IBM Processor for Network Resources

Revision 2.5

Databook

Preliminary

IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction,
NY 12533-6351

The IBM home page can be found at
http://www.ibm.com

The IBM Microelectronics Division home page
can be found at http://www.chips.ibm.com

pnr25.01
August 14, 2000

Contents

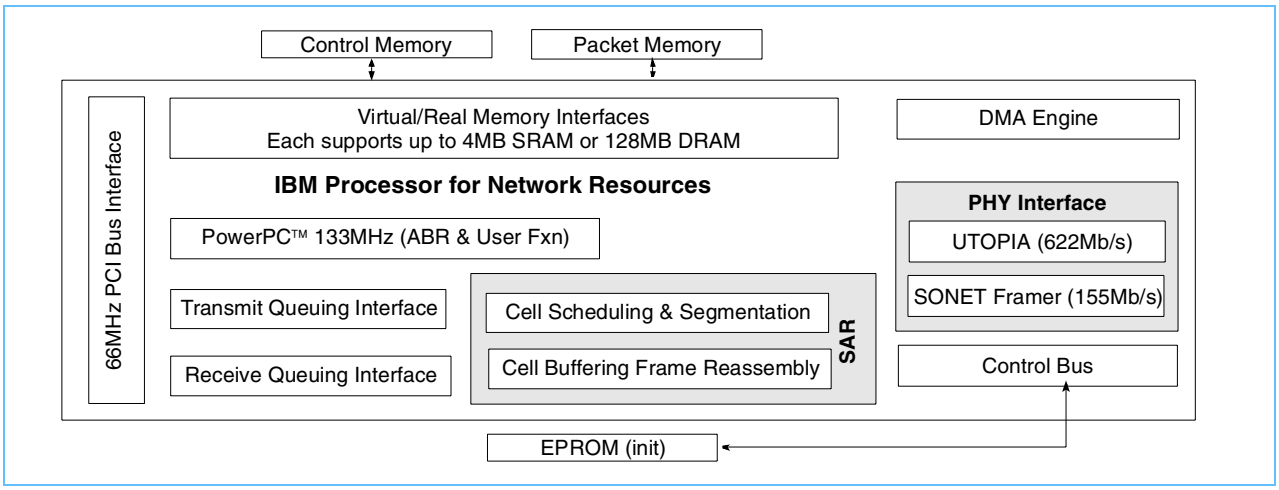
Features

- Supports multiple protocols, including ATM, POS, Frame Relay, and 10/100/Gigabit Ethernet
- Has a customizable on-chip 133 MHz PowerPC™ processor core
- Manages up to 65535 simultaneous logical channels, individually or in groups
- Integrated 155 Mb/s SONET (Synchronous Optical Network) Framer for simpler, low bandwidth designs
- Flexible ATM Forum-compliant UTOPIA II interface with up to four PHYs
- Switch Interface Extensions
- PCI 32/64-bit interface up to 66MHz.
- Configurable for sustained performance through the subsystem:
  - 155Mb/s full duplex internal SONET framer
  - 622Mb/s full duplex using an external SONET framer
  - 622Mb/s across up to four full duplex 155Mb/s links using an external quad framer
- JTAG Test Interface
- Package: 624 lead, 32 mm x 32 mm CBGA
- Power Supply: 2.6 V ±2%; 3.3 V ±5%.

Description

The IBM Processor for Network Resources (IBM3206K0424) is an Asynchronous Transfer Mode (ATM) support device. It is an interface and translator between a Peripheral Component Interconnect (PCI) bus and an ATM Utopia or similar interface to an ATM PHY. The IBM3206K0424 has an integrated Packet/Frame Memory (DRAM controller) and performs Segmentation and Reassembly (SAR) functions for several of the ATM Adaptation Layers (AALs). The IBM3206K0424 functions are illustrated in the *Block Diagram* on page 21. A Network Interface Card example is shown in *System Context of an ATM Subsystem* on page 34.

Block Diagram (See page 29 for descriptions of subsystems)



Ordering Information

Part Number	Description
IBM3206K0424	Network Resource Manager

Conventions

The bit notation is non-IBM, meaning that bit zero is the least significant bit and bit 31 is the most significant bit for a four-byte word.

The internal addressing view of the IBM3206K0424 registers and memory is big endian. In most cases, a system will wire its PCI bus interface to make the register view transparent, that is, the most significant bit in this specification will be the most significant bit in the register. If registers are read and written 32 bits at a time (which is the only way to access many of the registers), the endian-ness should not be a programming issue with respect to the registers.

The IBM3206K0424 DMA controller can transfer data in either big endian or little endian mode. See *General Purpose DMA (GPDMA)* on page 175 for details.

Numeric notation is as follows:

- Hexadecimal values are usually preceeded by x or X. For example: X’0B00’. For individual registers, Address values are hexadecimal without any special markings. For example, XXXX 1C3C.
- Binary values in text are either spelled out (zero and one) or appear in quotation marks. For example: ‘10101’.
- Binary values in the Default and Description columns of the register sections are often isolated from text as in this example:

  0: No action on read access

  1: Auto-reset interrupt request register upon read access

Standards Compliance

The IBM Processor for Network Resources, part number IBM3206K0424, has been designed with a number of standards in mind. These standards are listed below, grouped according to the area of IBM3206K0424 functionality they address.

- Network (defined by ITU-TS (formerly CCITT), ANSI and ATM Forum)
  - ITU Recommendation I-361 - B-ISDN ATM layer specification
  - ITU Recommendation I.362 - B-ISDN ATM Adaptation Layer (AAL) functional description
  - ITU Recommendation I.363 - B-ISDN ATM Adaptation Layer (AAL) specification
  - ITU Recommendation I.413 - B-ISDN user-network interface
  - ITU Recommendation I-432 - B-ISDN user-network interface - Physical Layer specification
  - ITU Recommendation I-610 - OAM principles of B-ISDN access
  - ANSI T1.ATM-199x Draft, Broadband ISDN - ATM Layer Functionality and Specification
  - ANSI T1.CBR-199x Draft, Broadband ISDN - ATM Adaptation Layer for Constant Bit Rate Service Functionality and Specification
  - ATM Forum 93-620R2 - ATM User-Network Interface Specification - Version 2.3 (July 27, 1993)
  - Bellcore TA-NWT-001248 Generic Requirements for Operations of Broadband Switching Systems (October 1993)
- System Interface
  - PCI Local Bus Specification, Production Version, Revision 2.1, June 1, 1995. Interface Technical Reference, 11/89, Part number 15F2160
- PHY Interface
  - SATURN User Network Interface, PMC-Sierra, Inc., February 1995
  - ATM Forum 93-727 An ATM PHY Data path interface, Version 2.01, March 24, 1994
  - Am7968/Am7969 TAXIchip(tm) Handbook, Transparent Asynchronous Transmitter/Receiver Interface, published by Advanced Micro Devices, 1994

Environmental Ratings

Absolute Maximum Ratings

Parameter	Rating	Unit	Note
Supply Voltage, $V_{DD}1$	2.3 to 2.7	V	1
Supply Voltage, $V_{DD}2$	3.0 to 3.6	V	1
Storage Temperature	-65 to 150	°C	1
Ambient Temperature with Power Applied	-40 to 100	°C	1

1. These are the maximum ratings that can be applied to the device without damage. The device function and specifications are valid only within the Recommended Operating Conditions.

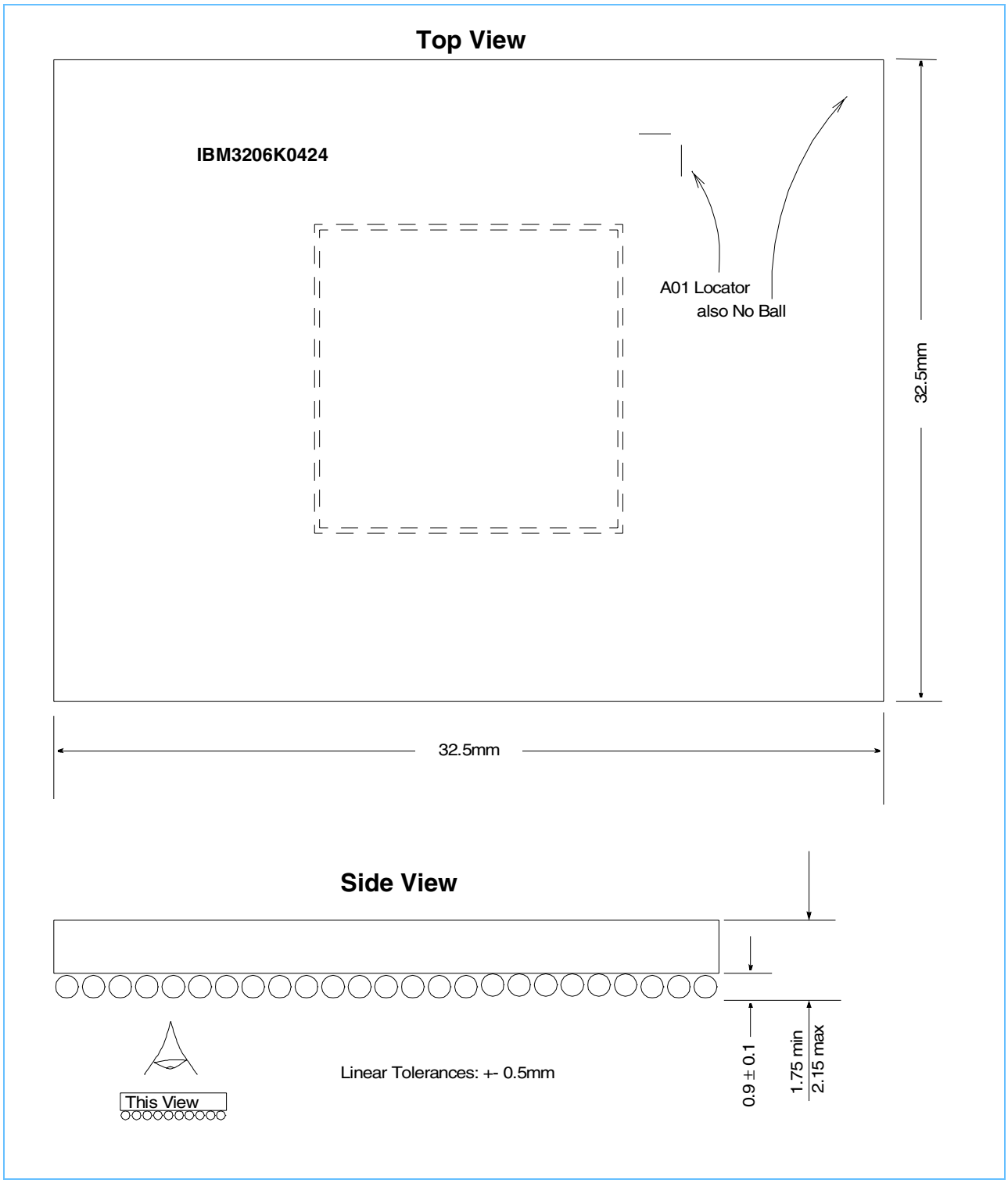
Recommended Operating Conditions

Parameter	Rating	Unit
Junction Temperature	0 to 85	°C
Supply Voltage, $V_{DD}1$, with respect to Ground	$2.6 \pm 2\%$	V
Supply Voltage, $V_{DD}2$, with respect to Ground	$3.3 \pm 5\%$	V

Power Dissipation

Parameter	Rating	Unit
$V_{DD}1$ (nominal)	6	W
$V_{DD}2$ (nominal)	2	W

Package Diagram



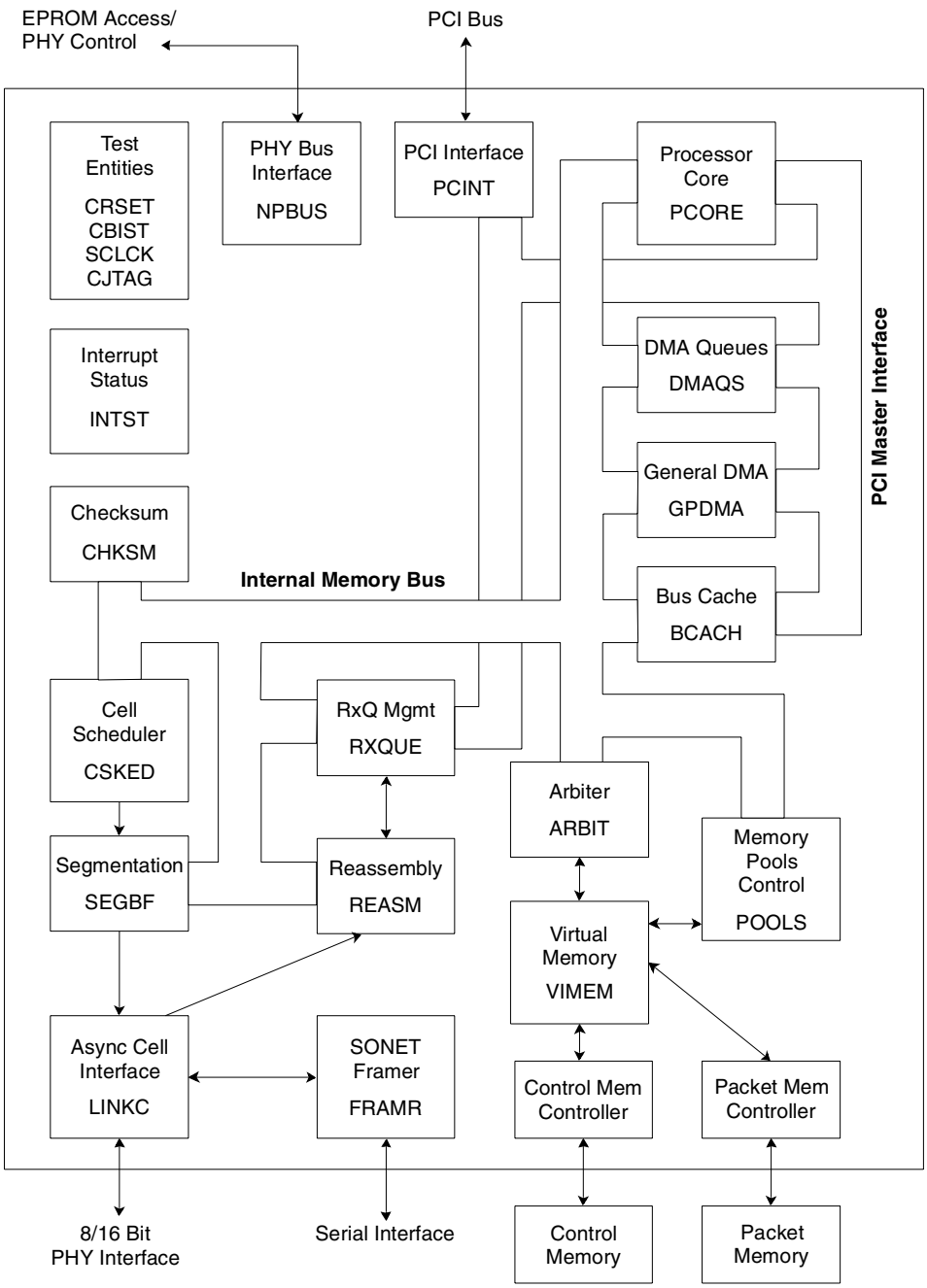
Pinout Viewed from Above

AE	AD	AC	AB	AA	Y	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A			
S	S	T	S	T	S	T	S	S	S	DCT	DCT	DCT	DCT	DCT	S	S	S	T	S	T	S	T	S	o	1		
S	G	S	VDD4	S	G	S	VDD	S	G	S	VDD4	DCT	VDD	S	G	S	VDD	S	G	S	VDD4	S	G	S	2		
T	S	T	S	T	S	T	S	S	S	S	DCT	DCT	DCT	S	S	S	S	T	S	T	S	T	S	T	3		
S	VDD	S	G	S	VDD	S	G	S	VDD4	S	G	DCT	G	S	VDD4	S	G	S	VDD	S	G	S	VDD	S	4	SIGNAL	467
T	S	T	S	T	S	S	S	S	S	S	T	DCT	T	S	S	S	S	S	S	T	S	T	S	T	5	VDD	44
S	G	S	VDD3	S	G	S	VDD4	S	G	S	VDD	DCT	VDD	S	G	S	VDD4	S	G	S	VDD5	S	G	S	6	VDD2	9
T	S	T	S	S	S	S	S	S	S	S	T	DCT	T	S	S	S	S	S	S	S	S	T	S	T	7	VDD3	9
S	VDD3	S	G	S	VDD	S	G	S	VDD4	S	G	DCT	G	S	VDD4	S	G	S	VDD	S	G	S	VDD5	S	8	VDD4	9
S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	9	VDD5	9
S	G	S	VDD	S	G	S	VDD3	S	G	S	VDD	S	VDD	S	G	S	VDD5	S	G	S	VDD	S	G	S	10	GND	77
DCT	S	S	S	S	S	S	S	S	S	VDD	S	G	S	VDD	S	S	S	S	S	S	S	S	S	DCT	11	ORIENT	1
DCT	VDD3	DCT	G	T	VDD3	T	G	S	VDD	S	G	VDD	G	S	VDD	S	G	T	VDD5	T	G	DCT	VDD5	DCT	12	TOTAL	625
DCT	DCT	DCT	DCT	DCT	DCT	DCT	DCT	S	S	G	VDD	G	VDD	G	S	S	DCT	DCT	DCT	DCT	DCT	DCT	DCT	DCT	13		
DCT	VDD	DCT	G	T	VDD3	T	G	S	VDD	S	G	VDD	G	S	VDD	S	G	T	VDD5	T	G	DCT	VDD	DCT	14		
DCT	S	S	S	S	S	S	S	S	S	VDD	S	G	S	VDD	S	S	S	S	S	S	S	S	S	DCT	15		
S	G	S	VDD	S	G	S	VDD3	S	G	S	VDD	S	VDD	S	G	S	VDD5	S	G	S	VDD	S	G	S	16	V1=VDD	
S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	17	V2=VDD2	
S	VDD3	S	G	S	VDD	S	G	S	VDD2	S	G	DCT	G	S	VDD2	S	G	S	VDD	S	G	S	VDD5	S	18	=VDD3	
T	S	T	S	S	S	S	S	S	S	S	T	DCT	T	S	S	S	S	S	S	S	S	T	S	T	19	=VDD4	
S	G	S	VDD3	S	G	S	VDD2	S	G	S	VDD	DCT	VDD	S	G	S	VDD2	S	G	S	VDD5	S	G	S	20	=VDD5	
T	S	T	S	T	S	S	S	S	S	S	T	DCT	T	S	S	S	S	S	S	T	S	T	S	T	21		
S	VDD	S	G	S	VDD	S	G	S	VDD2	S	G	DCT	G	S	VDD2	S	G	S	VDD	S	G	S	VDD	S	22	DCT=DC Test	
T	S	T	S	T	S	T	S	S	S	S	DCT	DCT	DCT	S	S	S	S	T	S	T	S	T	S	T	23	T=TST=AC Test	
S	G	S	VDD2	S	G	S	VDD	S	G	S	VDD2	DCT	VDD	S	G	S	VDD	S	G	S	VDD2	S	G	S	24		
S	S	T	S	T	S	T	S	S	S	DCT	DCT	DCT	DCT	DCT	S	S	S	T	S	T	S	T	S	S	25	View	
																										|/	
																										======	MLC
																										oooooo	BSM Ball

BSM - 32X32 MM CBGA @ 1.27MM PITCH

IBM ASIC 09/27/98

Dataflow


EPROM Access/
PHY Control
PCI Bus
Test Entities
CRSET
CBIST
SCLCK
CJTAG
PHY Bus Interface
NPBUS
PCI Interface
PCINT
Processor Core
PCORE
Interrupt Status
INTST
DMA Queues
DMAQS
General DMA
GPDMA
PCI Master Interface
Checksum
CHKSM
Internal Memory Bus
Bus Cache
BCACH
Cell Scheduler
CSKED
RxQ Mgmt
RXQUE
Arbiter
ARBIT
Memory Pools Control
POOLS
Segmentation
SEGBF
Reassembly
REASM
Virtual Memory
VIMEM
Async Cell Interface
LINKC
SONET Framer
FRAMR
Control Mem Controller
Packet Mem Controller
8/16 Bit
PHY Interface
Serial Interface
Control Memory
Packet Memory

Functional Description

The IBM3206K0424 has been designed by breaking the implementation of the various functions and data-flows into separate entities (major functional units).

This processor acts as a conversion unit from a bus memory interface (which is Work Queue oriented) to a PHY level ATM. To accomplish this, the IBM3206K0424 contains the major functional units listed below and shown in the *Dataflow* on page 27.

Control Processor Bus Interface

PCINT	PCI Interface entity
INTST	Interrupt Status entity
GPDMA	General Purpose DMA entity
DMAQS	Queue control for DMA activity

Memory Control

MEMRY	DRAM controlling entity containing the COMET (Control) and PAKIT (Packet) memory controllers
VIMEM	Virtual Memory controller
ARBIT	Memory subsystem requestor arbitration
BCACH	Bus Cache entity
POOLS	Memory Pool manager

Transmit Data Path

CSKED	Cell Scheduler
SEGBF	Cell Segmentation entity

Receive Data Path

REASM	Cell Re-assembly entity
RAALL	AAL processor
RXQUE	Receive Queue manager

PHY Level Interfaces

LINKC	Asynchronous Physical Layer interface
NPBUS	Nodal Processor Bus interface
FRAMR	Full SONET framing support logic

Hardware Protocol Assist

CHKSM	TCP/IP Checksum Logic
PCORE	Embedded 401 Processor Core

Base Device Functions

SCLCK	System Clock Generation and Repowering entity
CRSET	Hardware and Software Reset Controlling entity
CBIST	Built-In Self Test logic entity
CJTAG	JTAG Test Interface Logic entity

Subsystem Blocks

The IBM Processor for Network Resources provides the host bus interfacing, memory management for buffers and control, cell segmentation and reassembly, and PHY hardware control for an ATM adapter.

External Memory consists of a number of SRAM modules, or two SDRAM arrays used for the storage of packet data and the control structures used by the IBM3206K0424. Both the Packet and Control Memory arrays consist of two 32-bit wide banks.

When running at 102Mb/s or slower (full duplex aggregate throughput), a single array of memory can be used. Both control and data store are contained in this single array of memory. For a detailed description of the external memory organization refer to *The DRAM Controllers (COMET/PAKIT)* on page 184.

The PHY (Physical) Layer interface connects to several available hardware support devices. This layer of hardware converts a parallel data stream into a serial data stream to be shipped to and from the PMD layer.

The PHY and PMD end of a card design can be implemented as one of several encoding schemes and speeds, supporting both copper and fibre optic serial links. The interface will support the ATM Forum “Utopia spec,” the PMC chip, and a 25Mb/s serial interface to the IBM UTP solution. (See *Standards Compliance* on page 23 for documents which describe these interfaces.)

The PMD (Physical Media Dependent) Layer interface connects to the line drivers and receivers. This could be either a copper or a fibre optic transceiver.

External Architecture

The IBM Processor for Network Resources has four major interfaces:

A System Bus which acts as an actively cached memory slave and as a master for the PCI 32-bit bus.

The Physical (PHY) Interface which supports several physical layer hardware devices that perform parallel to serial data conversion and the rest of the transmission convergence.

An External DRAM Interface that controls one or two arrays of two-bank interleaved DRAM with 60 ns access time for Packet and Control Memory. The interface is direct drive to the DRAM.

The Control and Configuration Interface which covers a number of functions. It gives access from the system bus to the PHYs and to EPROM. The EPROM can also be used to hold initial device configuration, up to and including PVC configurations.

Note: IBM3206K0424 has built-in, self-test logic.

The four major interfaces allow the IBM Processor for Network Resources to be used in both "deep" and "shallow" adaptors with minimal external logic. (See *Block Diagrams of Possible Systems* on page 33 for examples.)

Internal Architecture

Logical Channel Support

The Logical Channel is the unit of resource allocation in ATM. At one level, the End Station negotiates with the Network Interface to determine the characteristics of each End Station-to-End Station connection. The resources that may be reserved in the network are defined in the ATM UNI (User Network Interface) Specification (see references in *Standards Compliance* on page 23). These resources include (but are not limited to) the peak and average bandwidth to be used by the logical channel, the maximum burst length that may be transmitted at the burst rate, the latency and variance of the connection, and the loss probability.

The term Logical Channel rather than virtual circuit or VPI/VCI is used in this databook to provide a level of abstraction from these specific instances.

A Switched Virtual Circuit (SVC) can be negotiated with specific characteristics specifically for it.

A virtual path can be negotiated with the network. Several virtual circuits within that path can then be multiplexed, using the VCI on that single VPI, without having to renegotiate for each additional VCI. The Logical channel, with respect to the network, would be the Virtual Path. There would be multiple logical channels internal to the End Station based on the Virtual Circuits used within the path.

Using ATM Adaptation Layers 3 and 4, a Multiplexing IDentifier (MID) can be used to provide multiple Logical Channels across a single VPI/VCI.

All of these Logical Channels are dealt with uniformly in IBM3206K0424. A hierarchy of Logical Channel Descriptors can be built up, and frames or buffers can be queued to each of the LCDs. See *Transmit Buffer (CSKED)* on page 273 for details.

Virtual Memory Support

The Packet Memory space appears on the bus as a group of *up to* 128K buffers (configurable size). A level of indirection has been added to the addressing of Packet memory to provide these large frame buffers without requiring memory behind all of them at the same time. This has been done for a number of reasons:

- The frames on the network can be up to 64KB long.
- The receiver does not know how long a frame will be until it is completely received.
- Software generally has a much easier time dealing with contiguous memory.

The memory does not page or swap. There are two major efficiencies used internally:

- The first N bytes of memory in a buffer are directly referenced.
- The blocks that make up the buffers are of multiple sizes.

Queues

The IBM Processor for Network Resources makes extensive use of cached single memory operation atomic queues:

Transmit queues	The interface to the scheduling entity. Blocks and Frames can be queued to Logical Channels.
Receive queues	Based on the settings in the Logical Channel Descriptor (receive side). Cells arriving can be queued individually, collected into frames, or stored in FIFO buffers.
Event queues	When a frame is transmitted, its memory can be “garbage collected” or a reference to the frame can be placed on an event queue for software to handle. If either a FIFO buffer scheme or frame buffer scheme is used to source or sink data on a logical channel, it is possible to set thresholds on the buffering that will cause events to be queued. When a threshold is crossed (for instance if a transmitting LC is about to run out of data to transmit), an event will be queued. Software can read these events either by polling or by being interrupted and can schedule tasks to provide more data. Events can be scheduled on the reception of the first N bytes of a frame so that header processing can begin even before the complete frame is received. This will allow “cut-through” routing to be supported.

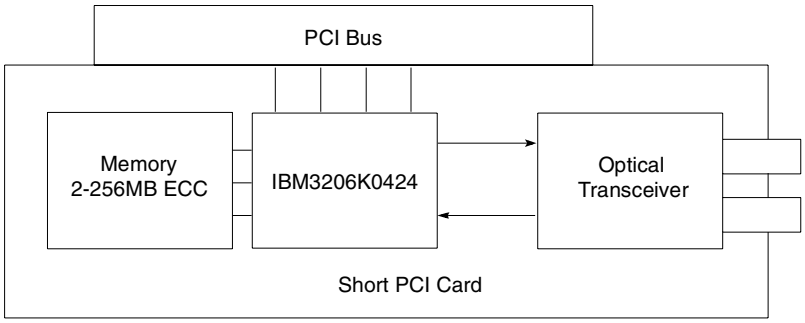
Note: In order to maintain the atomicity of 64-bit atomic transfers, the user must ensure that 64-bit transfers are bus atomic within the particular bus system in which the IBM3206K0424 is being used.

Scheduling

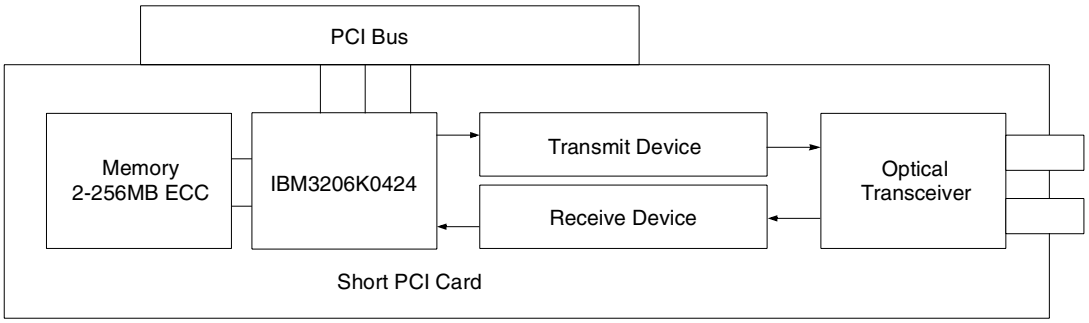
There is extensive support for transmit scheduling. Please see *Transmit Path* on page 37 and *Transmit Scheduling Capabilities* on page 38 for details.

Block Diagrams of Possible Systems

Shallow Adapter for 155Mb/s



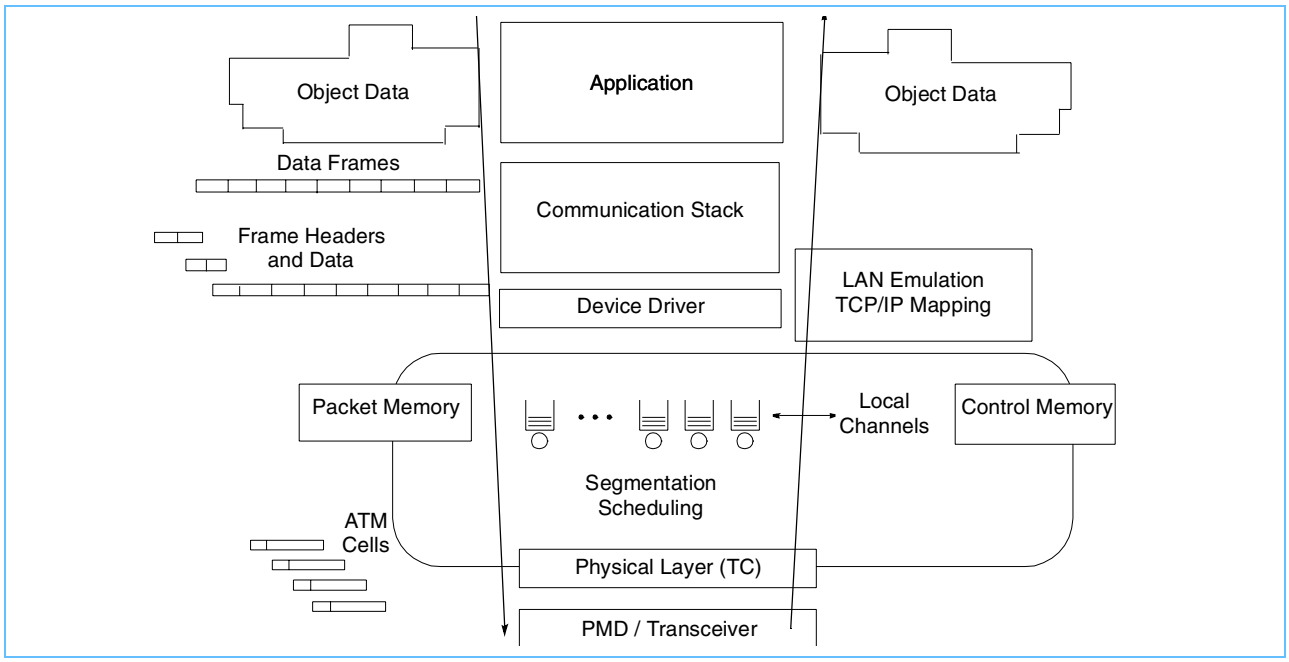
100 Mb/sec Card Using a TAXI Chipset



System Environment

The dataflow context of an ATM subsystem is shown in the diagram below. The purpose of the communications subsystem of any digital device is to allow the application to share data and to adjudicate the flow of control with other devices.

System Context of an ATM Subsystem



As shown in the figure above, data, in the form of application objects or control structures, are divided into communication frames at the communication stack interface. The stack may further partition the frames to fit reliability, efficiency, latency, and protocol requirements.

In most cases, the communication stack encapsulates the data frame with protocol headers and/or trailers. These header blocks are often located in memory in areas apart from the data frames. A device driver is often given the task of moving this scattered memory to the actual transmission device. Scatter DMA is often used to make this operation efficient.

In the case of the IBM Processor for Network Resources, the data can be DMAed into virtually contiguous buffers connected to and controlled by the IBM3206K0424. It is also possible to write the frame headers directly from the processor to the IBM3206K0424 memory. The fully assembled frame is enqueued for transmission over a particular logical channel. (See more on the richness of logical channels in ATM and the IBM3206K0424 in *Data Structures* on page 61).

The logical channels with pending work are serviced by the ATM Segmentation Layer which breaks the enqueued data into 48-byte chunks (depending on the ATM Adaptation Layer (AAL)) and prefixes it with a five-byte header (yielding the prime number 53) in preparation for transmission.

A Transmission Convergence (TC) sublayer appropriate for the Physical Layer (PHY) and Physical Media Dependent (PMD) connection is then exercised, making ATM cells suitable for transmission.

The receiving process is the reverse of the transmission process, except that the scheduling performed during transmission is replaced by an identification-demultiplexing step during the reception of cells.

Note: Not all of these separate parts or steps described in this section are necessary for a dedicated function system. IBM3206K0424 can easily be used in dedicated systems due to the goal of minimal processor intervention for steady state operations.

Data Flows

This section describes the data and control flow to and through the IBM3206K0424. In order for cell traffic to flow through an ATM interface, the cells require that Logical Channels be allocated. For information on Logical Channels, please see *Data Structures* on page 61.

Feature summary

- Virtual memory
- Memory pools
- Register read/write interface for memory allocation
- Transmit path scheduling
- Receive path demultiplexing
- Event queues

Operation summary

- Basic Assurance Tests (BATs)
- Initialize and configure
- Test path to switch
- Permanent Virtual Circuit setup
- Identify LAN servers
- Initialize SVCs
- Run, initializing circuits (Q.93B) and transmitting data

Transmit Path

A typical transmit operation begins with the software requesting a buffer from POOLS and filling it with data via slave DMA, master DMA, or processor writes. If virtual buffers are being used, the data write operation can fail due to lack of physical buffers. In the event of a failure, the header of the packet is updated to indicate the failure. The software can audit the header after the buffer has been completely transferred, and either take action to recover the data immediately or allow CSKED to generate an event later in the transmit cycle for any buffers that have had a data write failure.

Before the data can be transmitted, the buffer header must be updated to contain information required for correct transmission. Information such as data length, starting offset, and Logical Channel (LC) address are just a few of the fields that must be correctly reflected in the buffer header. For a complete list of the fields in the buffer header refer to *Packet Header* on page 61.

In addition to the fields in the buffer header, the scheduling and segmentation sections of the Logical Channel Descriptor (LCD), such as peak rate, average rate, and AAL type, must also be set up correctly prior to transmission. For a complete list of the fields in the LCD, refer to *Transmit Logical Channel Descriptor Data Structures* on page 66.

After the data have been transferred into packet storage and both the buffer header and the LCD structure have been correctly initialized, the buffer address is queued to CSKED. When it receives a buffer, CSKED checks the buffer header (Packet Memory) to make sure that the data transfer operation that filled the buffer completed without error. If it finds an error, CSKED posts an event to software and does nothing further with this buffer. If no error is indicated in the buffer header, CSKED fetches several fields from the LCD (Control Memory) indicated in the buffer header to determine the current state of that LCD. If the LCD is busy sending another buffer, the new buffer is queued to this LCD and will be processed when all previously enqueued buffers have been transmitted. If the LCD is not busy, CSKED updates the LCD based on several fields in the buffer header and queues the LCD to the next time slot on the time wheel (Control Memory).

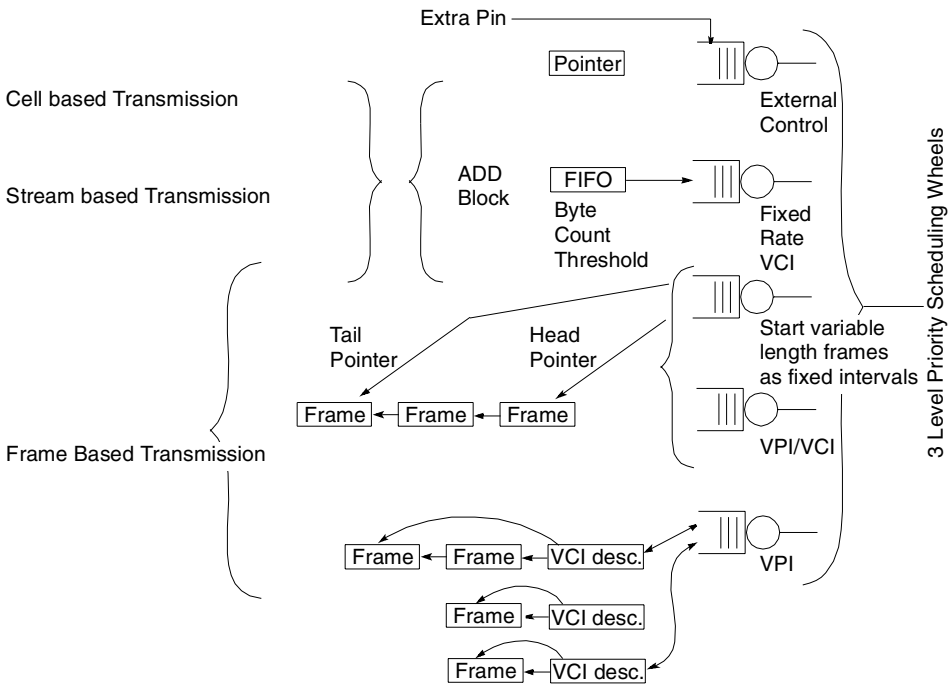
When CSKED detects a previously enqueued LCD on the time wheel, several fields are retrieved from the LCD. Among other things, these fields are used by CSKED to determine where on the time wheel to reschedule this LCD. The LCD address is then provided to SEGBF for processing.

When CSKED provides an LCD address to SEGBF, the segmentation portion of the LCD is retrieved from Control Memory to determine both the current address at which to continue buffer segmentation and the type of cell to construct. Depending on the AAL type bits in the segmentation portion of the LCD, the cell is constructed in an internal array using data from the LCD as well as data fetched from Packet Memory. When the cell construction is complete, status is raised to LINKC indicating that a new cell is available for transmission.

Transmit opportunities are repeatedly provided to SEGBF by CSKED at the desired rate until all the data in the buffer has been passed to LINKC via the cell buffer array. When SEGBF detects that no more data exists for a buffer, the LCD address is passed back to CSKED, indicating buffer completion. At this point, CSKED removes the LCD from the time wheel if no more buffers are queued in it. If more buffers are queued, the LCD is updated and the segmentation process continues until all buffers on the LCD queue are serviced. A bit in the buffer header generates a transmit complete event when no buffers remain in the queue.

Transmit Scheduling Capabilities

Transmit Queues and Logical Channel Traffic Shaping



Receive Path

As cells arrive, they pass from LINKC to REASM. REASM uses a portion of the ATM header to look up the LCD address for this cell. The LCD address is then passed to RAALL. RAALL reads the receive portion of the LCD, and then processes the cell based on the LCD information. For example, the LCD specifies what AAL to use and maintains the current reassembly state. Using the current reassembly state, the cell data is written to Packet Memory. While the data is written to Packet Memory, other functions such as CRC generation and verification are performed in parallel. If a packet is complete, all trailer verification is performed. If the packet is good, an event is placed on a receive queue in the RXQUE entity. For error scenarios, see *Receive Queues (RXQUE)* on page 366. At this point, software can dequeue the packet event from RXQUE using the dequeue operation. It can then examine headers, DMA the data into user space, and perform TCP checksums. When these actions are complete, the buffer is returned to the IBM3206K0424 by performing a POOLS-free buffer operation.

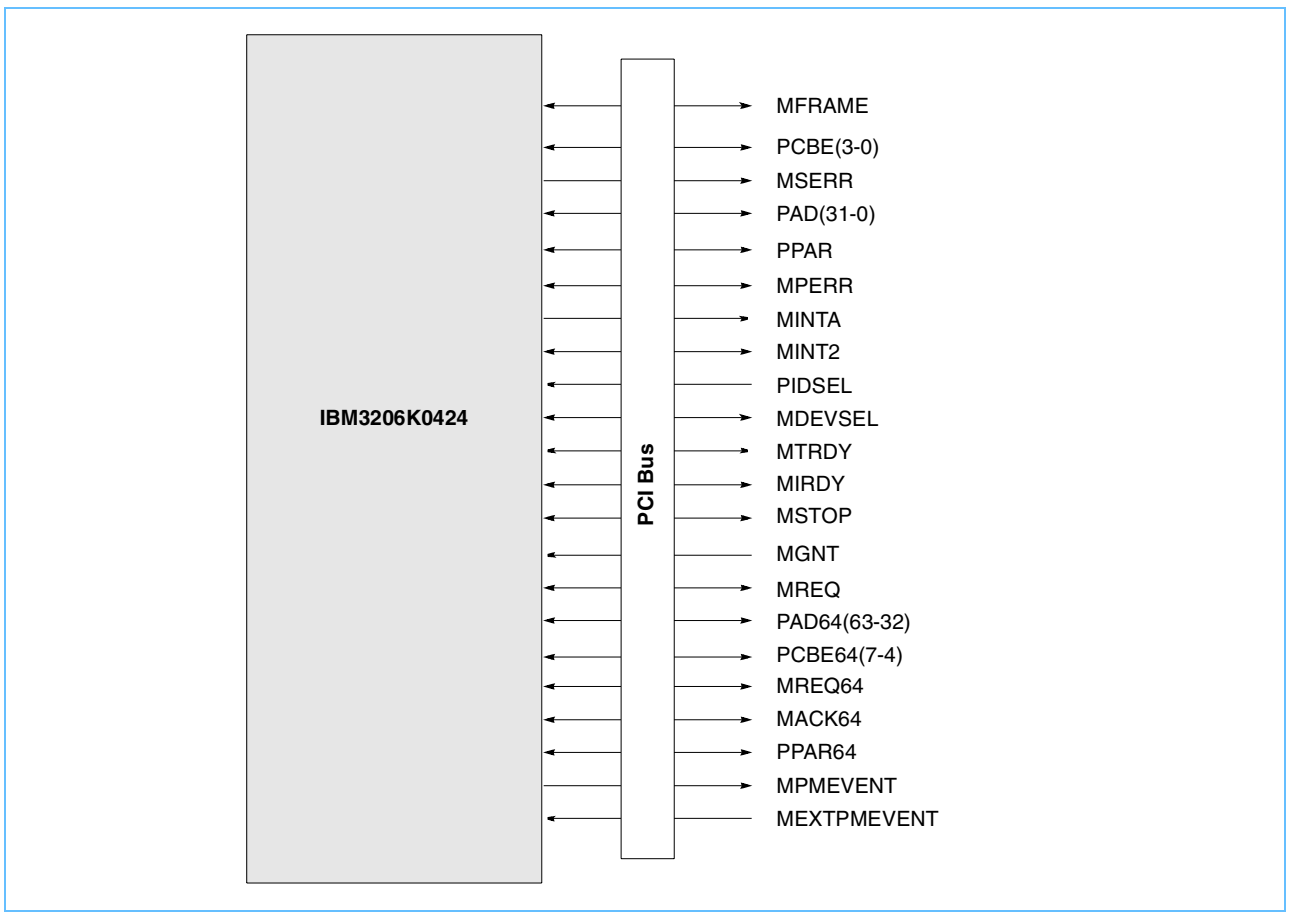
Input/Output Definitions

The several interfaces to IBM3206K0424 are described in the following sections. There are 443 active I/O pins, assigned as follows:

- 90 for the PCI bus
- 86 for the NPBUS
- 76 for the PHY bus
- 156 for the Packet Memory interface
- 35 strictly for configuration and testing

DRAM Memory Bus Interface

PCI Bus Connections



PCI Bus Interface Pin Descriptions

Quantity	Pin Name	Input/Output	Pin Description
1	MFRAME	S/T/S[1]	Cycle Frame is driven by the current master to indicate the beginning and duration of an access.
4	PCBE(3-0)	T/S	Bus Command and Byte Enables are multiplexed on the same PCI pins. During address phase they define the bus command; during data phase they define the byte enables.
1	MSERR	O/D	System Error reports address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic.
32	PAD(31-0)	S/T/S[1]	Address and Data are multiplexed on the same pins. A bus transaction consists of one address phase and one or more data phases.
1	PPAR	T/S	Parity is even parity across ad(31-0) and C/BE(3-0). Parity generation is required by all PCI agents.
1	MPERR	S/T/S[1]	Parity Error is for reporting data parity errors during all PCI bus transactions except Special Cycle.
1	MINTA	O/D	Interrupt A is used to request an interrupt.
1	MINT2	O/D or S/T/S[1]	This is an interrupt line that will go active low when sources within the IBM3206K0424 go active. It can be optionally connected to PCI interrupt B. See Entity 2: on page 135 for more details.
1	PIDSEL	IN	Initialization Device Select is a chip select during configuration transactions.
1	MDEVSEL	S/T/S[1]	Device Select indicates the driving device has decoded its address as the target of the current transaction.
1	MTRDY	S/T/S[1]	Target Ready signals the target agent's ability to complete the current data phase of the transaction.
1	MIRDY	S/T/S[1]	Initiator Ready indicates the bus master's ability to complete the current data phase.
1	MSTOP	S/T/S[1]	Stop indicates the current target is requesting the master to stop the current transaction.
1	MGNT	IN	Receives the Bus Grant line after a request has been made.
1	MREQ	S/T/S[1]	Requests the bus for an initiator transfer.
32	PAD64(63-32)	S/T/S[1]	Address and Data are multiplexed on the same pins and provide 32 additional bits. Also, these pins are multiplexed with the ENSTATE outputs, which allow debug of various internal state machines and signals.
4	PCBE64(7-4)	T/S	Bus Command and Byte Enables are multiplexed on the same PCI pins for 64-bit transfer support.
1	MREQ64	S/T/S[1]	Request 64-bit transfer. Has the same timing as MFRAME.
1	MACK64	S/T/S[1]	Acknowledge 64-bit transfer. Has the same timing as MDEVSEL.
1	PPAR64	S/T/S[1]	Parity Upper DWORD is the even parity bit that protects MAD64(63-32) and PCBE(7-4). When not on a PCI bus supporting 64 bits, this will drive ENSTATE outputs.

1. S/T/S = a sustained tri-state pin owned and driven by one and only one agent at a time. The agent that drives the S/T/S pin low must drive it high for at least one clock before letting it float. A new agent cannot start driving a S/T/S signal any sooner that one clock after the previous owner tri-states it. A pullup is required to sustain the inactive state until another agent drives it, and must be provided by the central resource.

PCI Bus Interface Pin Descriptions (Continued)

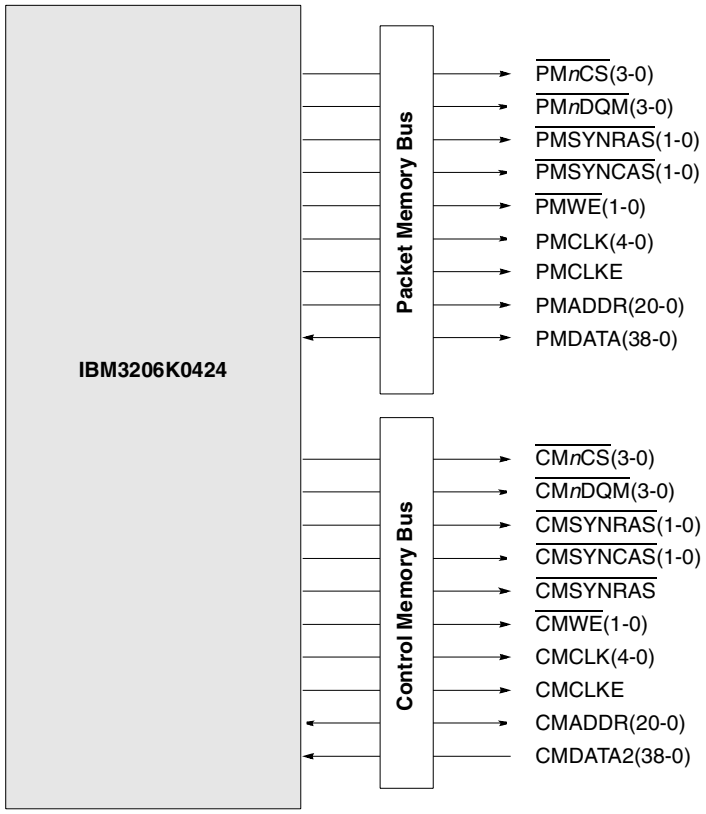
Quantity	Pin Name	Input/Output	Pin Description
1	MPMEVENT	O/D	As a PME source, this signal is active low and indicates a power management event signalled from the IBM3206K0424. The output need to be conditioned with a card-level FET circuit so that the resulting signal (PME# on the PCI bus) can be driven with the proper driver characteristics. This signal can also function as the PME_enable function for an external source when programmable in this mode in PCINT.
1	MEXTPMEVENT	IN	Active low by default but programmable, this input indicates a power management event signalled from some other card component to the IBM3206K0424.

1. S/T/S = a sustained tri-state pin owned and driven by one and only one agent at a time. The agent that drives the S/T/S pin low must drive it high for at least one clock before letting it float. A new agent cannot start driving a S/T/S signal any sooner that one clock after the previous owner tri-states it. A pullup is required to sustain the inactive state until another agent drives it, and must be provided by the central resource.

DRAM Memory Bus Interface

One Control Memory and one Packet Memory bus provide the attachment to the external DRAM. Up to two arrays of 32 data bits plus potential error detection bits may be connected to each bus.

DRAM Memory Bus Connections



DRAM Memory Bus Interface Pin Descriptions

Quantity	Pin Name	Input/Output	Pin Function	Pin Description
4	PM*n*CS(3:0)	Output	Packet Memory SRAM chip selects	PM*n*CS(3:2) are bank address lines 1 and 0 and PM*n*CS(1:0) are the chip selects for the two arrays when using SDRAM for Packet Memory. When using SRAM, they are either the four chip selects or are eight-encoded chip selects and a valid signal.
4	CM*n*CS(3:0)	Output	Control Memory SRAM chip selects	CM*n*CS(3:2) are bank address lines 1 and 0 and CM*n*CS(1:0) are the chip selects for the two arrays when using SDRAM for Control Memory. When using SRAM, they are either the four chip selects or are eight-encoded chip selects and a valid signal.
2	PM*n*DQM(3:0)	Output	Packet memory DQM lines	PMDQM(3:0) are the DQM lines when using SDRAM for Packet Memory. They are identical copies of output enable when using SRAM. PMDQM(3:2) is just another copy of PMDQM(1:0) to reduce loading on the nets.
2	CM*n*DQM(3:0)	Output	Control memory DQM lines	CM0DQM(3:0) are the DQM lines when using SDRAM for Control Memory. They are identical copies of output enable when using SRAM. CMDQM(3:2) is just another copy of CMDQM(1:0) to reduce loading on the nets.
2	PMSYNRAS(1:0)	Output	RAS signal for packet synchronous DRAM	PMSYNRAS(1:0) are identical copies of the RAS signal for Packet Memory when using SDRAM. They are byte enables (3:2) when using SRAM.
2	CMSYNRAS(1:0)	Output	RAS signal for control synchronous DRAM	CMSYNRAS(1:0) are identical copies of the RAS signal for Control Memory when using SDRAM. They are byte enables (3:2) when using SRAM.
2	PMSYNCAS(1:0)	Output	CAS signal for packet synchronous DRAM	PMSYNCAS(1:0) are identical copies of the CAS signal for Packet Memory when using SDRAM. They are byte enables (1:0) when using SRAM.
2	CMSYNCAS(1:0)	Output	CAS signal for control synchronous DRAM	CMSYNCAS(1:0) are identical copies of the CAS signal for Control Memory when using SDRAM. They are byte enables (1:0) when using SRAM.
2	PMWE(1:0)	Output	Packet Memory write enable	Packet memory write enable.
2	CMWE(1:0)	Output	Control Memory write enable	Control memory write enable.
5	PMCLK(4:0)	Output	Packet Memory clock	There are five copies to minimize loading.
1	PMCLKE	Input/Output	Packet Memory clock enable	Clock enable for Packet Memory when using SDRAM.
5	CMCLK(4:0)	Output	Control Memory clock	There are five copies to minimize loading.
1	CMCLKE	Input/Output	Control Memory clock enable	Clock enable output for Control Memory when using SDRAM.
21	PMADDR(20:0)	Output	Address signals to Packet Memory	
21	CMADDR(20:0)	Output	Address signals to Control Memory	
39	PMDATA(38:0)	Input/Output	Data signals to and from the Packet Memory	
39	CMDATA(38:0)	Input/Output	Data signals to and from the Control Memory.	

Memory I/O Cross Reference By Device Type

IBM3206K0424 I/O	Sync DRAM 2-Bank Device	Sync DRAM 4-Bank Device	SRAM
xxADDR(20:0)	Address(20:0)	Address(20:0)	Address(20:0)
$\overline{\text{xxCS}}$(3)	N/A	Bank Address(1)	Chip Select(3)
$\overline{\text{xxCS}}$(2)	Bank Address(0)	Bank Address(0)	Chip Select(2)
$\overline{\text{xxCS}}$(1:0)	Chip Select(1:0)	Chip Select(1:0)	Chip Select(1:0)
$\overline{\text{xxDQM}}$(3:2)	DQM(1:0)	DQM(1:0)	OE(1:0)[1]
$\overline{\text{xxDQM}}$(1:0)	DQM(1:0)	DQM(1:0)	OE(1:0)[1]
xxCLKE	CKE	CKE	
$\overline{\text{xxWE}}$(1:0)	WE(1:0)[1]	WE(1:0)[1]	WE(1:0)[1]
$\overline{\text{xxSYNRAS}}$(1:0)	RAS(1:0)[1]	RAS(1:0)[1]	Byte Enable(3:2)
$\overline{\text{xxSYNCAS}}$(1:0)	CAS(1:0)[1]	CAS(1:0)[1]	Byte Enable(1:0)
xxDATA(31:0)	Data(31:0)	Data(31:0)	Data(31:0)
xxDATA(35:32)	ECC(3:0)	ECC(3:0)	Parity(3:0)
xxDATA(38:36)	ECC(6:4)	ECC(6:4)	N/A

1. All signal groups marked by an asterisk are active at the same time.
2. xx = CM for Control Memory or PM for Packet Memory.
3. For SDRAMs, the DQM signals are active independently for shared ECC configurations.
4. For SDRAMs with split ECC, the DQMs are usually active unless doing burst length two and the DQM is needed to terminate a burst.

Possible Memory Configurations Using SDRAM With Shared ECC

Module Size	One Array plus ECC		Two Arrays plus ECC	
	Storage Size	Number of Devices	Storage Size	Number of Devices
1Mx16	4MB	3	8MB	5
2Mx8	8MB	5		
4Mx16	16MB	3	32MB	5
8Mx8	32MB	5		
16Mx16	64MB	3	128MB	5
32Mx8	128MB	5		

Note: While it is possible to connect more than five SDRAM modules to each controller on the IBM3206K0424, it is likely the capacitive loading will not allow the interface to work at 7.5ns. The memory interface would need to be slowed down to allow the interface to work.

Possible Memory Configurations Using SRAM

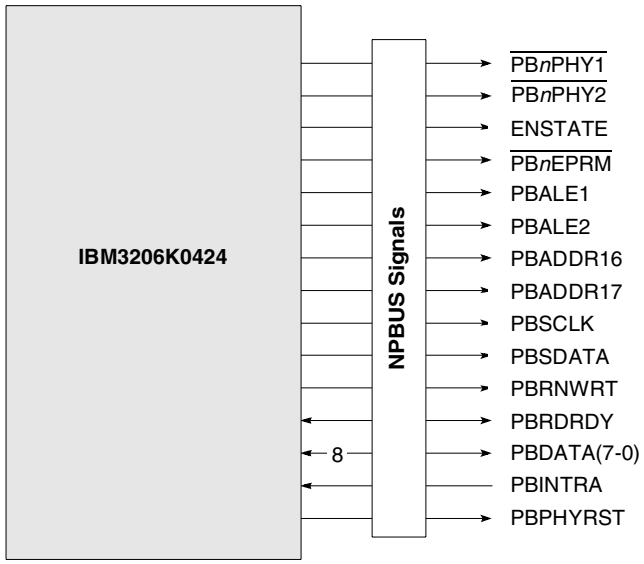
Module Size	Memory Size for One Module	Memory Size for Two Modules	Memory Size for Four Modules
2Mx18	N/A	8MB	16MB
1Mx18	N/A	4MB	8MB
512Kx18	N/A	2MB	4MB
256Kx18	N/A	1MB	2MB
1Mx36	4MB	8MB	16MB
512Kx36	2MB	4MB	8MB
256Kx36	1MB	2MB	4MB
128Kx36	N/A	1MB	2MB

1. For x18 SRAM modules, half the data bus goes to one module, and the other half goes to a second module. The chip select to the two modules is common. Therefore, two x18 modules can be connected to a single chip select while only one x32 module can. Therefore, given a constant number of chip selects, using pairs of x18 "x" Mb modules results in a memory that is twice as deep as what is possible with x32 modules. Using x18 modules also lowers the overall capacitance on the memory data nets.
2. While it is possible with the number of chip selects available (multiplexed or not) to connect more than four SRAM modules to each controller on the IBM3206K0424, it is likely the capacitive loading will not allow the interface to work at 7.5ns. The memory interface would need to be slowed down to allow the interface to work.

NPBUS

The NPBUS supports access to either an EPROM or PHY level hardware microprocessor interface. The NPBUS can operate with an eight-bit multiplexed addr/data bus or an eight-bit data bus with 18 separate address pins. Generic transfer control signals work with the PHY level hardware microprocessor interface or EPROM to accomplish data transfers.

NPBUS Connections



NPBUS Pin Descriptions

Quantity	Pin Name	Input/Output	Pin Function	Pin Description
1	$\overline{\text{PB}n\text{PHY1}}$	Output	Select PHY 1	When low, indicates that the IBM3206K0424 has selected PHY 1 to write to control registers inside PHY 1 or to read either the control or status registers.
1	$\overline{\text{PB}n\text{PHY2}}$	Output	Select PHY 2	When low, indicates that the IBM3206K0424 has selected PHY 2 to write to control registers inside PHY 2 or to read either the control or status registers. See *NPBUS Control Register* on page 428 for more details. If configured, this pin can also be odd parity across the eight-bit wide bidirectional data bus. It can also be configured as MPMDSEL - this control pin, under register bit control, can drive a logical value out. The intention is to select between the different PMD types on the 155 Mb/s copper card (UTP verses STP). If it is in cascade mode, this bit functions as PIDSELO (+idsel out), which the primary IBM3206K0424 will drive to the secondary IBM3206K0424 when trying to update configuration space via configuration cycles. This multiplexed pin also carries the PBDATAP signal.
63	ENSTATE (63-0)	Output		When programmed, drives out the real-time state-of-entity state machines, counters, etc. for debug purposes. The (47-32) bits of this bus are also PBADDR(15 - 0), which are the address lines for the external parallel EPROM or PHY. Additionally, bits 47 - 40 can be used as bi-directional data bus bits to extend the PBDATA bus by providing bits 15 - 8 of this bus. This allows operation with PHY parts that have a fixed 16-bit data but limits the addressing to this PHY to only eight address bits. (*LSSD test function - scanout(13 to 0) -SO -BDY*)
1	$\overline{\text{PB}n\text{EPRM}}$	Output	EPROM Select	When low, indicates that the IBM3206K0424 has selected the external EPROM to read from. After reset, the IBM3206K0424 will start accessing the optional on-card ROM/EPROM and do the chip initialization function if it does not find a serial EPROM attached.
1	PBALE1	Output	Address Latch Enable 1	When high, indicates that the IBM3206K0424 has generated an address on the PBDATA bus and should be latched by either a PHY that supports this muxing or an external octal latch TTL part. For an external EPROM, it will also latch bits 7-0 of the address for an external EPROM access.
1	PBALE2	Output	Address Latch Enable 2	When high, indicates that the IBM3206K0424 has generated an address on the PBDATA bus and should be latched by an external octal latch TTL part that holds bits 15-8 of the address for an external EPROM or PHY access.
1	PBADDR16	Output	Address Send 16	Supplies address 16 to an external EPROM. The pin will also function as PBALE3, an address latch enable, that indicates that the IBM3206K0424 has generated an address on the PBDATA bus and should be latched by an external octal latch TTL part that holds bits 23-16 of the address for an external EPROM access. The mechanism used to set this mode is to put a pull-down resistor on this pin. At reset time, it will be detected and set this bit in PBALE3 mode. Otherwise it will be in PBADDR16 mode.

1. S/T/S = a sustained tri-state pin owned and driven by one and only one agent at a time. The agent that drives the S/T/S pin low must drive it high for at least one clock before letting it float. A new agent cannot start driving a S/T/S signal any sooner that one clock after the previous owner tri-states it. A pullup is required to sustain the inactive state until another agent drives it and must be provided by the central resource.

NPBUS Pin Descriptions (Continued)

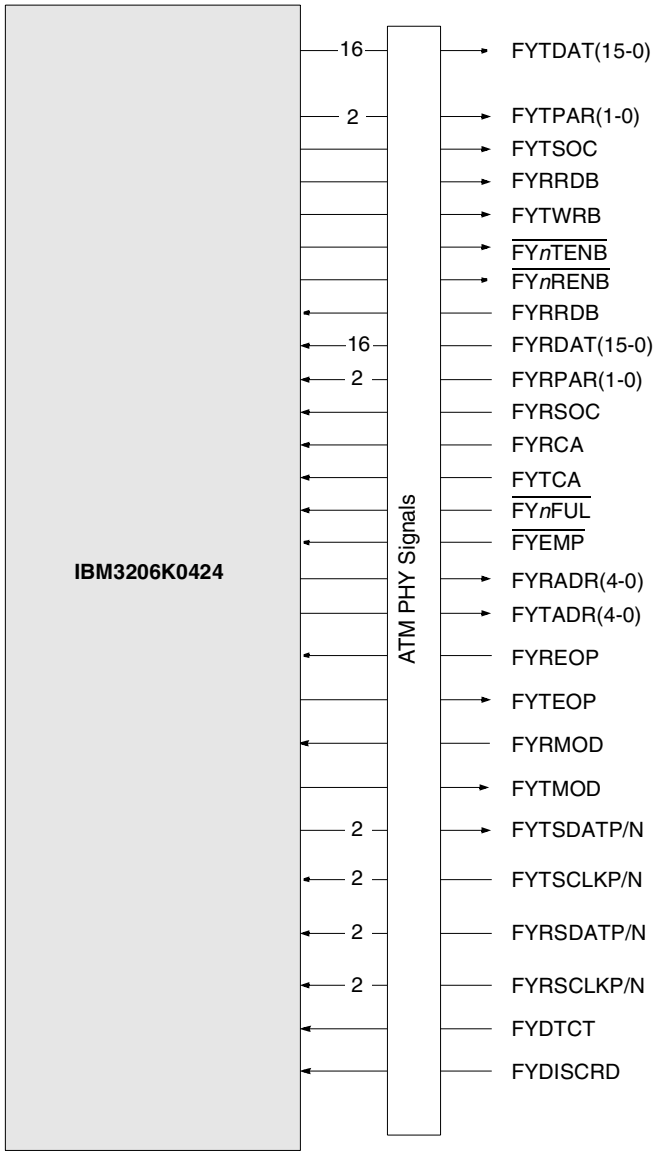
Quantity	Pin Name	Input/Output	Pin Function	Pin Description
1	PBADDR17	Output	Address Send 17	Supplies address 17 to an external EPROM.
1	PBSCLK	Output	Clock for the I^2C serial EPROM accesses	
1	PBSDATA	Output or Data		This is the data bit that connects to the external serial EPROM to read from or write to. It must have a pullup resistor attached and supports the I^2C protocol. The range of supported serial EPROM is from 256 to 2K bytes. After reset, the IBM3206K0424 will start accessing the optional on-card serial EPROM and do the chip initialization function. If this chip is pulled down (or no pullup), the IBM3206K0424 will assume that no serial EPROM is attached and will go try to fetch from a parallel EPROM.
1	PBRNWRT	Output	Read or Write	This pin allows the IBM3206K0424 to read from or write to internal registers of the PHY parts. This signal acts as the write strobe when talking to PMC-Sierra chips such as the Suni-Lite.
1	PBRDRDY	S/T/S[1]		This pin allows the IBM3206K0424 to read from or write into internal registers of the PHYs by acting as a data acknowledge signal from the memory slaves. This signal acts as the read strobe when talking to PMC-Sierra chips such as the Suni-Lite.
8	PBDATA(7-0)	Input or Output		The PB-Bus is an eight-bit wide bidirectional data bus used to interface the PHYs to the IBM3206K0424. When a data transfer is not happening, the lower four bits act as MLED(3-0) - four control pins that, under register bit control, can drive general status to LED devices.
1	PBINTRA	Input		This input from PHY A is an attention line that, when low, indicates that one or more unmasked flags are set in the status registers of PHY 1. If additional PHY parts are added, they should also dot their interrupt line onto this input.
1	PBPHYRST	Output	Implements the network safety features of the device	This signal implements the network safety features of the IBM3206K0424. It is the ORed value of RESET and all of the status bits cause the IBM3206K0424 to stop transferring data. It is asserted for a pulse, and then removed. This signal is asserted low.

1. S/T/S = a sustained tri-state pin owned and driven by one and only one agent at a time. The agent that drives the S/T/S pin low must drive it high for at least one clock before letting it float. A new agent cannot start driving a S/T/S signal any sooner that one clock after the previous owner tri-states it. A pullup is required to sustain the inactive state until another agent drives it and must be provided by the central resource.

ATM PHY Bus Interface

The PHY Bus consists of a transmit data path, receive data paths, and control signals.

PHY Bus Interface Connections



PHY Bus Pin Descriptions (Page 1 of 3)

Quantity	Pin Name	Input/Output	Pin Function	Pin Description
16	FYTDAT (15 - 0)	Output	PHY Transmit Data	When using an external PHY, this 16-pin bus carries the ATM CELL octets that are loaded in the PHY Transmit FIFO. When using the internal framer, bits 15, 14, and 13 are used for the RX HDLC interface signals OFPrxR1Data, OFPrxR1DS, and OFPrxRclk, respectively.
2	FYTPAR (1 - 0)	Output	Transmit Data Parity	When using an external PHY, these are byte parity signals for FYTDAT. When using the internal framer, bit 1 provides the RX Out-Of-Frame indication, OOF, and bit 0 provides the optical/ electrical module transmit shutdown control signal, OFPtxS-Down. When using a POS-PHY, bit zero provides the TERR signal.
1	FYTSOC	Output	Transmit start of Cell	When using an external PHY, this indicates the start of cell on FYTDAT. When using the internal framer, this provides the TX HDLC interface signal, OFPtxT1Dclk. When using a POS-PHY, this indicates TSOP.
1	FYTWRB	Output	Transmit write strobe	When using an external PHY, this signal is used to write ATM cells to the transmit FIFO. When using the internal framer, this signal provides the 19.44MHz TX clock, RefClkT. When using a Utopia Cell or POS-PHY interface, this signal provides the write clock based on the clock received on the TXCLK pin.
1	$\overline{\text{FY}n\text{TENB}}$	Output	Transmit write enable	When using an external PHY, this indicates that transmit data to the PHY is valid. When using the internal framer, this provides the TX HDLC interface signal, OFPtxT1DS.
1	$\overline{\text{FY}n\text{RENB}}$	Output	Receive write enable	When using an external PHY, this indicates to the PHY that the IBM3206K0424 is ready to accept data. When using the internal framer, this provides the clock recovery reset signal, RSTCRec1.
1	FYRRDB	Output	Receive ready strobe	When using an external PHY, this is used to read ATM cells from the PHY receive FIFO. When using the internal framer, this signal provides the 19.44MHz RX clock, RxByClk. When using a Utopia Cell or POS-PHY interface, this signal provides the write clock based on the clock received on the RXCLK pin.
16	FYRDAT(15 - 0)	Input	PHY Receive Data	When using an external PHY, this 16-pin bus carries the ATM CELL octets that are read from the PHY Receive FIFO.
2	FYRPAR(1 - 0)	Input	PHY Receive Data Parity	When using an external PHY, these are byte parity signals for FYRDAT. When using the internal framer, bit 1 provides the optical/electrical module low power indication signal, OFPtx-LPow, and bit 0 is not used. When using a POS-PHY, bit 0 should be connected to RERR.

Note: Because some of the PHY transmit I/Os are used for receive framer functions and vice versa, there are some restrictions on how the interfaces can be used.

1. If the transmit path is using an external PHY and the receive path is using the internal framer, FYTPAR(1) will assume the OOF function and not be available as a parity output. This is only a concern if the PHY uses a 16-bit data interface and parity is being used.
2. If the receive path is using an external PHY and the transmit path is using the internal framer, FYRPAR(1) will assume the OFPtx-LPow function and not be available as a parity input. This is only a concern if the PHY uses a 16-bit data interface.
3. If the transmit path is using an external PHY and the receive path is using the internal framer, and the external PHY has a 16-bit data interface, then the receive HDLC interface cannot be used. The three I/O for the RX HDLC interface will instead take on the function of FYTDAT(15-13).

PHY Bus Pin Descriptions (Page 2 of 3)

Quantity	Pin Name	Input/Output	Pin Function	Pin Description
1	FYRSOC	Input	Receive start of Cell	When using an external PHY, this signal indicates the start of cell on the FYRDAT bus.
1	FYRCA	Input	Receive Cell Available	When using an external PHY, this indicates that a cell is available in the receive FIFO. When using an internal framer, this signal is not used. When using a POS-PHY, this signal must be connected to PRPA.
1	FYTCA	Input	Transmit Cell Available	When using an external PHY, this indicates that a cell is available in the PHY transmit FIFO. When using the internal framer, this provides the TX HDLC interface signal, OFPtxT1Data. When interfacing to POS-PHY, this signal should be connected to PTPA.
1	$\overline{\text{FY}n\text{FUL}}$	Input	PHY Transmit Full	When using an external PHY, this is asserted low by the PHY when it can accept no more than four more data transfers before it is full. This pin should be pulled up to the inactive state when using a PHY that does not drive it. When using the internal framer, this provides the TX HDLC interface signal, OFPtxT1DFrm. When using a POS-PHY interface, this signal must be connected to STPA
1	$\overline{\text{FYEMP}}$	Input	PHY Receive Empty	When using an external PHY, this is asserted low by the PHY to indicate that in the current cycle there is no valid data for delivery to the IBM3206K0424. When the PHY does not drive FYEMP, this input should be tied to the inactive state. When using the internal framer, this signal is not used. When using a POS-PHY interface, this signal is connected to RVAL.
5	FYRADR(4-0)	Output	PHY Receive Address	When using an external PHY (Cell or POS-PHY based), this address is used to select and poll up to 31 PHYs on the receive side. Bits (1-0) are used to select which of the four PHYs and bit two is used to indicate the null address. When bit two is B'1', bits 1-0 are also '1'. When bit 2 is '0', bits 1-0 are "00", "01", "10", or "11".
5	FYTADR(4-0)	Output	PHY Transmit Address	When using an external PHY (Cell or POS-PHY based), this address is used to select and poll up to 31 PHYs on the transmit side. Bits (1-0) are used to select which of the four PHYs and bit two is used to indicate the null address. When bit two is '1', bits 1-0 are also '1'. When bit 2 is '0', bits 1-0 are "00", "01", "10", or "11".
1	FYREOP	Input	PHY Receive EOP	When using an external POS-PHY, this signal indicates if the FYRDATA (15-0) contains the last data of a packet. If the external PHY is not a POS-PHY, this signal should be tied to GND.

Note: Because some of the PHY transmit I/Os are used for receive framer functions and vice versa, there are some restrictions on how the interfaces can be used.

1. If the transmit path is using an external PHY and the receive path is using the internal framer, FYTPAR(1) will assume the OOF function and not be available as a parity output. This is only a concern if the PHY uses a 16-bit data interface and parity is being used.
2. If the receive path is using an external PHY and the transmit path is using the internal framer, FYRPAR(1) will assume the OFPtxLPow function and not be available as a parity input. This is only a concern if the PHY uses a 16-bit data interface.
3. If the transmit path is using an external PHY and the receive path is using the internal framer, and the external PHY has a 16-bit data interface, then the receive HDLC interface cannot be used. The three I/O for the RX HDLC interface will instead take on the function of FYTDAT(15-13).

PHY Bus Pin Descriptions (Page 3 of 3)

Quantity	Pin Name	Input/Output	Pin Function	Pin Description
1	FYTEOP	Output	PHY Transmit EOP	When using an external POS-PHY, this signal indicates if the FYTDATA (15-0) contains the last data of a packet. If the external PHY is not a POS-PHY, this signal should ignored.
1	FYRMOD	Input	PHY Receive MOD	When using an external POS-PHY, this signal indicates if the FYRDATA (7-0) contains valid data. If FYRMOD is '1', FYRDATA(7-0) is ignored. FYRMOD is only relevant when FYREOP is '1'. A value of '1' any other time will be ignored. If a POS-PHY is not connected, this signal should be tied to GND.
1	FYTMOD	Output	PHY Transmit MOD	When using an external POS-PHY, this signal indicates if the FYTDATA (7-0) contains valid data. If FYTMOD is '1', FYRDATA(7-0) will be ignored. FYTMOD is only driven to a '1' when FYTEOP is '1'.
2	FYTSDATP/N	Output	SERDES Transmit Data (Differential)	When using the internal framer and the internal SERDES, these signals provide the serial transmit data stream. When the transmit side of LINKC is connected to an Echo interface, this differential driver will provide the clock that goes with the data FYRDAT(7-0), parity FYTPAR(7-0), and SOC FYTSOC.
2	FYTSCLKP/N	Input	SERDES Transmit Clock (Differential)	When using the internal framer and the internal SERDES, the reference 155.52MHz clock is supplied on these signals. When not in use, these should be tied to (TBD).
2	FYRSDATP/N	Input	SERDES Receive Data (Differential)	When using the internal framer and the internal SERDES, the recovered receive data is supplied on these signals. When not in use, these should be tied to (TBD).
2	FYRSCLKP/N	Input	SERDES Receive Clock (Differential)	When using the internal framer and the internal SERDES, the recovered 155.52MHz clock is supplied on these signals. When not in use, these should be tied to (TBD).
1	FYDTCT	Input	PHY Carrier Detect	When using an external PHY, the PHY uses this signal to indicate carrier detect. When using the internal framer, this signal provides the deserializer lock detect signal, ELockDet, from the deserializer.
1	FYDISCRD	Input	PHY Cell Discard	When using an external PHY, this signal causes the current cell being received to be discarded. In this case it should only be asserted for the duration of one of the 53 bytes of the ATM cell. When using the internal framer, this signal provides the optical/electrical module Loss-Of-Signal indication, LossSig.

Note: Because some of the PHY transmit I/Os are used for receive framer functions and vice versa, there are some restrictions on how the interfaces can be used.

1. If the transmit path is using an external PHY and the receive path is using the internal framer, FYTPAR(1) will assume the OOF function and not be available as a parity output. This is only a concern if the PHY uses a 16-bit data interface and parity is being used.
2. If the receive path is using an external PHY and the transmit path is using the internal framer, FYRPAR(1) will assume the OFPtxLPow function and not be available as a parity input. This is only a concern if the PHY uses a 16-bit data interface.
3. If the transmit path is using an external PHY and the receive path is using the internal framer, and the external PHY has a 16-bit data interface, then the receive HDLC interface cannot be used. The three I/O for the RX HDLC interface will instead take on the function of FYTDAT(15-13).

Transmit PHY I/O Cross Reference

IBM3206K0424 I/O	Suni-Lite	Utopia	POS-PHY	Internal Framer
FYTWRB	TFCLK	TxCLK	TFCLK	RefClkT
FYTCA	TCA	TxClav	PTPA	OFPtxT1Data
$\overline{\text{FY}n\text{FUL}}$	N/A	$\overline{\text{TxFull}}$	STPA	OFPtxT1DFrm
$\overline{\text{FY}n\text{TENB}}$	$\overline{\text{TWRENB}}$	$\overline{\text{TxEnb}}$	$\overline{\text{TEnb}}$	OFPtxT1DS
FYTSOC	TSOC	TxSOC	TSOP	OFPtxT1Dclk
FYTDAT(15)	N/A	TxData(15)	TData(15)	OFPrxR1Data
FYTDAT(14)	N/A	TxData(14)	TData(14)	OFPrxR1DS
FYTDAT(13)	N/A	TxData(13)	TData(13)	OFPrxR1Dclk
FYTDAT(12-8)	N/A	TxData(12-8)	TData(12-8)	N/A
FYTDAT(7-0)	TDAT(7-0)	TxData(7-0)	TData(7-0)	TxExtDat(7-0)
FYTPAR(1)	N/A	TxPrty(1)	TPRTY	OOF
FYTPAR(0)	N/A	TxPrty(0)	TERR	OFPtxSDown
FYTADR(4-0)	N/A	TxADDR(4-0)	TADR(4-0)	N/A
FYTMOD	N/A	N/A	TMOD	N/A
FYTEOP	N/A	N/A	TEOP	N/A
FYTDATP	N/A	N/A	N/A	TSDATP
FYTDATN	N/A	N/A	N/A	TSDATN
FYTCLKP	N/A	N/A	N/A	TSCLKP
FYTCLKP	N/A	N/A	N/A	TSCLKN

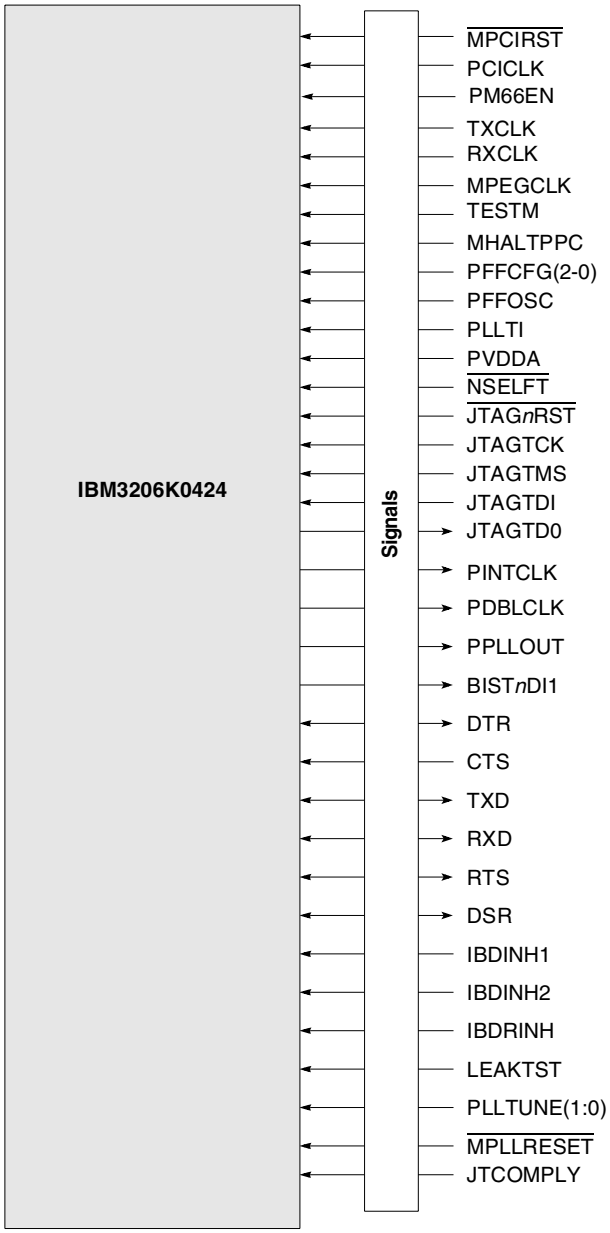
Note: Signals marked with an overbar are active low. Inputs listed as N/A should be tied to their inactive Utopia state.

Receive PHY I/O Cross Reference

IBM3206K0424 I/O	Suni-Lite	Utopia	POS-PHY	Internal Framer
FYRRDB	RFCLK	RxCLK	RFClk	RxByClk
FYRCA	RCA	RxClav	PRPA	N/A
$\overline{\text{FYEMP}}$	N/A	$\overline{\text{RxEmpty}}$	RVAL	N/A
$\overline{\text{FYRENB}}$	$\overline{\text{RRDENB}}$	$\overline{\text{RxEnb}}$	$\overline{\text{RENB}}$	RSTCRec1
FYRSOC	RSOC	RxSOC	RSOP	N/A
FYRDAT(15-8)	N/A	RxData(15-8)	RDat(15-8)	N/A
FYRDAT(7-0)	RDAT(7-0)	RxData(7-0)	RDat(7-0)	N/A
FYRPAR(1)	N/A	RxPrty(1)	RPRTY	OFPtxLPow
FYRPAR(0)	N/A	RxPrty(0)	RERR	N/A
FRYADR(4-0)	N/A	RXADDR(4-0)	RADR(4-0)	N/A
FYRMOD	N/A	N/A	RMOD	N/A
FYREOP	N/A	N/A	REOP	N/A
FYRSDATP	N/A	N/A	N/A	RSDATP
FYRSDATN	N/A	N/A	N/A	RSDATN
FYRSCLKP	N/A	N/A	N/A	RSCLKP
FYRSCLKN	N/A	N/A	N/A	RSCLKN
FYDTCT	N/A	N/A	N/A	DTCT
FYDISCRD	N/A	N/A	N/A	DISCRD
FYSETCLP	N/A	N/A	N/A	SETCLP

Note: Signals marked with an overbar are active low. Inputs listed as N/A should be tied to their inactive Utopia state.

Clock, Configuration, and LSSD Connections



Clock, Configuration, and LSSD Pin Descriptions

Quantity	Pin Name	Input/Output	Pin Description
1	$\overline{\text{MPCIRST}}$	Input	This signal causes a hardware reset when asserted low. See Entity 20: on page 511 for more details on resets.
1	PCICLK	Input	The PCICLK is a 0-33, 66MHz clock.
1	PM66EN	Input	This pin is high when on a PCI bus that runs at 33MHz - 66MHz. It is used to tell the on-chip PLL how to generate clocks.
1	TXCLK	Input	This is the LinkC asynchronous transmit clock.
1	RXCLK	Input	This is the LinkC asynchronous receive clock. An oscillator should be connected to RXCLK even if it is not functionally used. Without the RXCLK input oscillating, the chip may not reset properly.
1	MPEGCLK	Input	This is the MPEG asynchronous clock.
1	TESTM	Input	When the test mode pin is not asserted, this chip runs as specified. When the test mode pin is asserted, the chip is in LSSD test mode. Transparent latches become clocked latches and I/Os change to primary test inputs and test outputs. The precise connections are specified in the VHDL. This signal is asserted high when in test mode.
1	MHALTPPC	Input	Used by RISCWatch to halt the Power PC core for debug purposes. This does not need to be in a TEST/NOSCAN I/O location.
3	PFFCFG (2 - 0)	Input	These bits control the "find frequency" function which sets the range bits of the PLL. Below is the encoded meaning of these bits. 000 = Enable internal register of these bits in CRSET 001 = Disable auto range function: set range to < 25.0MHz operation 010 = Disable auto range function: set range to 25.0MHz - 33.3MHz-011 = Disable auto range function: set range to 50.0MHz operation 100 = Enable auto range function for 19.44MHz 101 = Reserved 110 = Enable auto range function for 25.0MHz 111 = Enable auto range function for 32.0MHz
1	PFFOSC	Input	This input is the auto range known frequency input that is used to time the PCI clock input. This should be connected to some oscillator on the card, for example, the PHY oscillator. This is also used as the BIST clock. Without this input oscillator, the chip will not run BIST nor will it properly reset; it is required for proper operation.
1	PLLTI	Input	When tied to '1', this input will cause the PLL to do a parametric testing at the wafer and module level. Normal mode for this pin is a '0'.
1	PVDDA	Input	Filtered Vdd source to the PLL logic. See technology application notes for filter circuit.
1	$\overline{\text{NSELFT}}$	Input	Minus active SELFTEST input. Normal mode is a '1'.
1	$\overline{\text{JTAG}n\text{RST}}$	Input	JTAG Test Reset provides an asynchronous initialization of the TAP controller.
1	JTAGTCK	Input	JTAG Test Clock is used to clock state information and test data into and out of the device during operation of the TAP.
1	JTAGTMS	Input	JTAG Test Mode Select is used to control the state of the TAP controller in the device. (*LSSD test function - RARRYTCLKC - SC*)
1	JTAGTDI	Input	JTAG Test Data Input is used to serially shift test data and test instructions into the device during TAP operation. (*LSSD test function - CLKDIVTCLKC-SC*)

Clock, Configuration, and LSSD Pin Descriptions (Continued)

Quantity	Pin Name	Input/Output	Pin Description
1	JTAGTDO	Output	Test Data Output is used to serially shift test data and test instructions out of the device during TAP operation. (*LSSD test function - PRSRAMABDONE and PLLLOCK output*)
1	PINTCLK	Output	This is the external test point to measure the jitter effects of the phase-lock loop circuit. PINTCLK does not serve any LSSD or MFG test function. It does not need to be on a TEST/NOSCAN location.
1	PDBLCLK	Output	This is the external test point that is double the frequency of the PINTCLK. It is used to clock ENSTATE state signals at this frequency. PDBLCLK does not serve any LSSD or MFG test function. It does not need to be on a TEST/NOSCAN location.
1	PPLLOUT	Output	This is an observation output only. This makes the output of the PLL observable. This is also the DTR signal when the SELRS232 is active.
1	$\overline{\text{BIST}n\text{DI1}}$	Output	Drives the DI input during BIST.
1	DTR	Input or Output	RS232 DTR for the core debugger. (*LSSD test function - TCLKA-AC)*
1	CTS	Input	RS232 CTS for the core debugger. (*LSSD test function - LPRA bypass-TI)*
1	TXD	Input or Output	RS232 TXD for the core debugger. (*LSSD test function - CLKDIVTCLKB-BC)*
1	RXD	Input or Output	RS232 RXD for the core debugger. (*LSSD test function - BSCANTCLKB-BC)*
1	RTS	Input or Output	RS232 RTS for the core debugger. (*LSSD test function - BSCANTCLKC-SC)*
1	DSR	Input or Output	RS232 DSR for the core debugger. (*LSSD test function - pll testout)*
1	IBDINH1	Input	This is the Boundary Scan input for BSINH1.
1	IBDINH2	Input	This is the Boundary Scan input for BSINH2(*).
1	IBDRINH	Input	This is the Boundary Scan input for rinh.
1	LEAKTST	Input	This is the STI driver/receiver leak test input.
2	PLLTUNE(1:0)	Input	These inputs help tune the PLL operation. (*LSSD test function - SCANOUT(15,14))*
1	$\overline{\text{MPLLRESET}}$	Input	This input is active low and resets the PLL at power up to avoid VCO runaway. This requires a reset circuit that delays a low-to-high level after power-on-reset by 150 µs. (*LSSD test function - this pin functions as the TESTCT [Test Clock Tree] input. When not asserted, this chip runs as specified. When asserted, the clock tree uses this input to control the clokc tree outputs - TI)*
1	JTCOMPLY	Input	This input is high for JTAG compliance and low for RISCWatch/BIST-friendly use. When this pin is high, JTAG boundary scan operations may be used to test chip I/O operation and card wiring without supplying clocks to the rest of the chip. Also, when the TAP controller enters the TEST LOGIC RESET state, the JTAG instruction is IDCODE. When this pin is low, the JTAG boundary scan logic works only if the other chip clocks are running in a normal functional manner. When the TAP controller enters the TEST LOGIC RESET state, the JTAG instruction is BYPASS in order to make this more compatible with RISCWatch. (*LSSD test function - SRAM BIST result output)*

Data Structures

These structures reside in Control Memory for each of the logical channels that are set up for transmission or reception.

Packet Header

Each packet buffer consists of two parts. The first part is the control information used by the IBM3206K0424. The second portion of the packet buffer is used to hold the actual packet data. The following figures show the structure of the transmit and receive packet headers:

Transmit Packet Header Structure

```
struct tx_min_packhead {
      bit32 next_buffer;
      bit8  AAL5_user_byte1;
      bit8  buffer_offset;
      bit16 buffer_length;
      bit25 lc_address;
      bit1  EFCI_status;
      bit1  reserved;
      bit1  dma_on_xmit;
      bit1  generate_CRC10;
      bit1  free_on_xmit;
      bit1  queue_on_xmit;
      bit1  cell_loss_priority;
      };
```

```
struct tx_packhead {
bit32 next_buffer;
bit8  AAL5_user_byte;
bit8  buffer_offset;
bit16 buffer_length;
bit25 lc_address;
bit1  EFCI_status;
bit1  reserved;
bit1  dma_on_xmit;
bit1  generate_CRC10;
bit1  free_on_xmit;
bit1  queue_on_xmit;
bit1  cell_loss_priority;
bit32 dma_desc_addr;

bit16  AAL5_user_byte1_2;
bit16  reserved
};
```

The minimum transmit packet header size (and transmit offset) is 0xC bytes.

Receive Packet Header Structure

```
struct rx_packhead {
      bit32 rx_label_flags;

      bit8  AAL5_user_byte1;
      bit8  buffer_offset;
      bit16 buffer_length;

      bit25 lc_address;
      bit1  EFCI_status;
      bit5  reserved;
      bit1  cell_loss_priority;

      bit32 optional_words[7];

      };
```

See *RXAAL Packet Header Configuration* on page 352 for available word choices and definitions.

Receive Packet Definitions

```
// aal 5 definition
      struct rx_label flags {
      bit16 rx_label;
      bit2  ip_chksm_flags;
      bit2  proto_chksm_flags;
      bit1  toobig_status;
      bit1  memchk_status;
      bit1  fabort_status;
      bit1  badlen_status;
      bit1  badcpi_status;
      bit1  badcrc_status;
      bit1  reserved;
      bit1  reserved;
      bit1  reserved;
      bit1  route_status;
      bit1  error_status;
      bit1  done_status;
        };
```

```
// raw definition
      struct rx_label flags {
      bit16 rx_label;
      bit2  reserved;
      bit2  reserved;
      bit1  toobig_status;
      bit1  memchk_status;
      bit1  reserved;
      bit1  reserved;
      bit1  reserved;
      bit1  badcrc_status;
      bit1  reserved;
      bit1  reserved;
      bit1  reserved;
      bit1  route_status;
      bit1  error_status;
      bit1  done_status;
        };
```

```
// packet mode definition
      struct rx_label flags {
      bit16 rx_label;
      bit2  ip_chksm_flags;
      bit2  proto_chksm_flags;
      bit1  toobig_status;
      bit1  memchk_status;
      bit1  reserved;
      bit1  reserved;
      bit1  reserved;
      bit1  reserved;
      bit1  reserved;
      bit1  reserved;
      bit1  reserved;
      bit1  route_status;
      bit1  error_status;
      bit1  done_status;
        };
```

The minimum receive packet offset is 0xC bytes. When the optional fields are enabled, the receive packet offset increases and should be set appropriately in the receive LCD. See *RXAAL Packet Header Configuration* on page 352 for available word choices and definitions.

Transmit and Receive Packet Header Field Descriptions (Page 1 of 3)

Field Name	Field Description
next_buffer	This field is used by the hardware to chain buffers together on queues. It contains the address of the next buffer if one exists. For transmit buffers allocated in virtual memory, this field is written by the hardware with a distinctive pattern ('zzzzzBAD'x) where zzzzz is the offset of the failure when a write operation was not able to complete due to a shortage of the real buffers needed to map into the virtual address space. This field can be checked after all buffer write operations and the appropriate recovery actions are taken immediately, or when a buffer that has had a write failure is enqueued to CSKED (an event will be generated and the buffer will not be processed by CSKED). A status bit also exists in the BCACH status register indicating that a write to virtual memory has failed. With cache performance in mind, this status bit could be checked first; if it is not set, there is no need to access the header of the packet. **Note:** This automatic error recovery mechanism results in the restriction that this first four bytes of a transmit packet must never be written via programmed IO or DMA during preparation for transmission. If this field is written by a software or DMA operation, the automatic error detection will not work properly and undesirable results are likely.
AAL5_user_byte1	On transmit, this field contains the value to be sent in the user byte in the last cell of an AAL5 packet, if INTST is configured for one user byte. On receive, this field contains the user byte from the AAL5 trailer, if INTST is configured for one user byte. When not configured for AAL5, this field is redefined. The two most significant bits contain the drop number. When in packet mode, the low five bits of this byte contain the number of bytes dropped due to the dropN-Bytes field in the LCD.
dma_on_xmit	If this bit is set, a DMA descriptor address placed in the packet header (offset 0xC) will be queued for execution.
generate_CRC10	If this bit is set, CRC10 will be generated over the cell(s) in this packet.
free_on_xmit	If this bit is set, the buffer will be freed after the transmission completes.
queue_on_xmit	If this bit is set, the buffer will be queued on the transmit complete queue after the transmission completes.

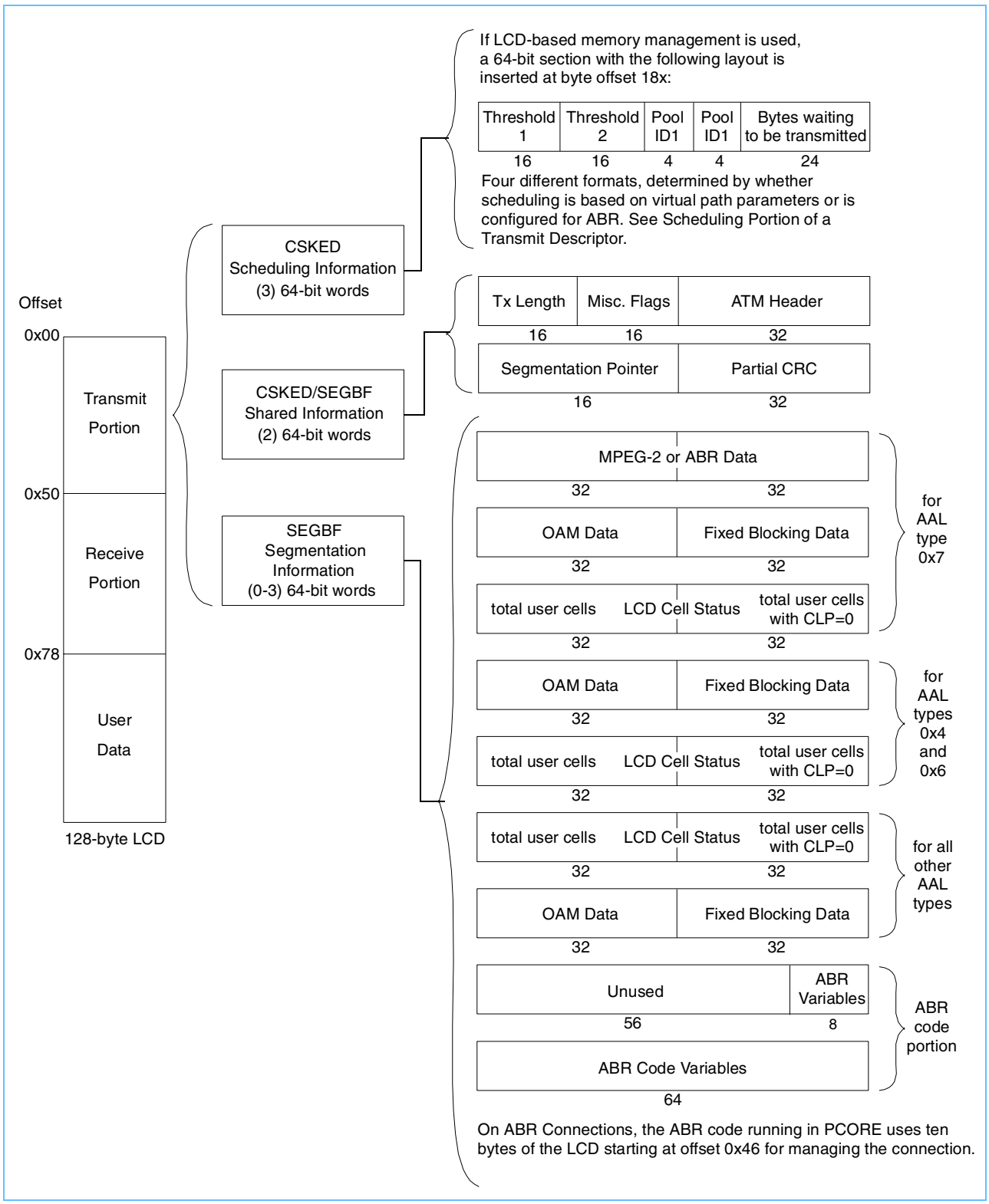
Transmit and Receive Packet Header Field Descriptions (Page 2 of 3)

Field Name	Field Description
EFCT_status	This bit is used on both transmit and receive: Transmit - If this bit is set, the EFCI bit in the ATM cell header will be set for each cell in this packet. Receive - This bit contains the ORed EFCI bits across all the cells that comprised this packet if this LCD is using AAL5.
cell_loss_priority	This bit is used on both transmit and receive: Transmit -If this bit is set, the cell loss priority bit in the ATM cell header will be set for each cell in this packet Receive - This bit contains the ORed cell loss priority bits across all the cells that comprised this packet if this LCD is using AAL5.
buffer_offset	This field contains the offset into the buffer where the data starts.
buffer_length	This field contains the length of the packet.
lc_address	This is the address of the logical connection descriptor on which this packet was received.
rx_atm_header	On reception, the four-byte ATM header (no HEC) is copied from the first and last cell into this area. This is an optional header word.
AAL5_user_byte2 (tx)	This field contains the value to be sent in the user bytes, one and two, in the last cell of an AAL5 packet if INTST is configured for two-user byte. The normal one-byte AAL5 user byte field is not used.
AAL5_user_byte2 (rx)	This field contains the AAL5 user bytes, one and two, in the last cell of an AAL5 packet if INTST is configured for two-user byte. The normal one-byte AAL5 user byte field is not filled in. This is an optional packet header word.
bit16 rx_label	This field is written with "RA" in ASCII (0x5241) to signal that this buffer was used by RAALL.
bit2 ip_chksm_flags	This field contains the IP checksum status if enabled. The possible values are: 00 Not checked 01 Good 10 Bad 11 Unknown
bit2 proto_chksm_flags	This field contains the protocol checksum status if enabled. The possible values are: 00 Not checked 01 Good 10 Bad 11 Unknown
bit1 toobig_status	Indicates the current packet exceeded the maximum packet size.
bit1 memchk_status	Indicates the current packet had a memory check (real size exceeded or virtual error).
bit1 fabort_status	Indicates the current packet was aborted (AAL5 forward abort).
bit1 badlen_status	Indicates the current packet had a bad AAL5 length in the trailer.
bit1 badcpi_status	Indicates the current packet had a bad AAL5 CPI field (not zero).
bit1 badcrc_status	Indicates the current packet had a bad AAL5 CRC.
bit1 route_status	This bit is written when the packet is completed if it is internally routed.
bit1 error_status	This bit is written when the packet is completed if an error condition occurs.
bit1 done_status	This bit is written when the packet is completed. It can be used when thresholding.
host_data	This is an optional word that can be copied from the LCD to the packet header.

Transmit and Receive Packet Header Field Descriptions (Page 3 of 3)

Field Name	Field Description
VBA	This is the IBM3206K0424 virtual buffer address of the current receive buffer. This is an optional packet header word.
TxQueueLenH/M/L	These words provide the current transmit queue length. See *RXAAL Transmit Queue Length Compression Configuration* on page 351 for more information on the format of these words. These are optional packet header words.
start_timestamp	A timestamp can be inserted in the packet header when the packet is started. The timestamp is sourced from the RXQUE timestamp register. The location of this timestamp is important so it is not overwritten when the packet is completed. For this reason, it should be the last word in the packet header. This is an optional header word.
end_timestamp	A timestamp can be inserted in the packet header when the packet is started. The timestamp is sourced from the RXQUE timestamp register. This is an optional header word.
dma_desc_addr	If the dma_on_xmit bit is set in the packet header, this field contains the address of the DMA descriptor that will be queued when transmission is complete.

Logical Channel Data Structure



General LCD Layout

```
struct lcd_struct {
  tx_lcd_struct  tx_lcd ;/* transmit portion of the lc descriptor */
  bit8  fill[N] ;/* the fill area depends on the type of tx lcd */
  rx_lcd_struct  rx_lcd ;/* receive portion of the lc descriptor */
} ;
```

Transmit Logical Channel Descriptor Data Structures

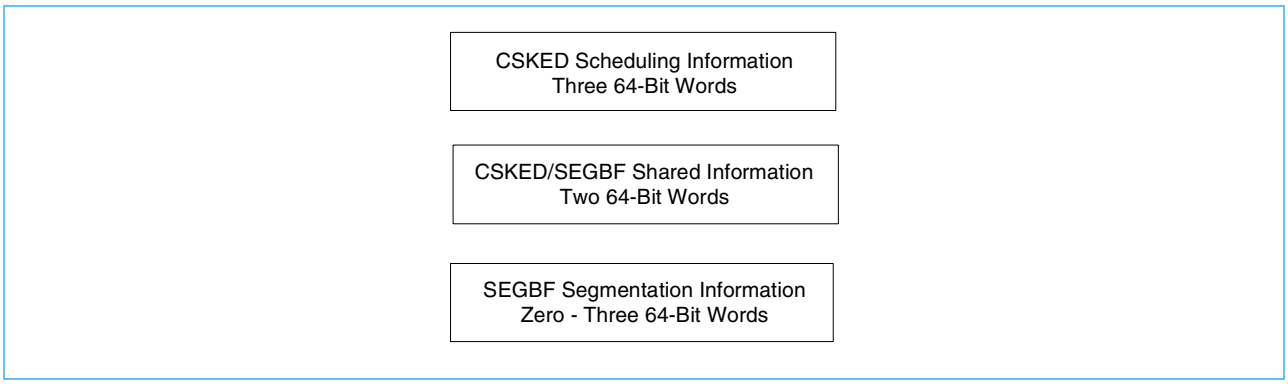
Logical Channel Data Structure on page 65 and *Scheduling Portion of a Transmit Descriptor* on page 67 show the layout of the transmit portion of a Logical Channel Descriptor. When initializing an LCD, any locations that are not written to a specific value should be initialized to zeros. Fields that typically need to be initialized to a non-zero value are flagged with a # in the structure below.

Note: This is only one possible layout of the transmit portion of the LCD. Some field locations vary and are further defined later in this section.

Care must be taken when updating fields in the LCD and then immediately causing the updated fields to be accessed by other IBM3206K0424 entities. For example, it is possible, although not likely, under the right conditions, for a normal LCD update followed by a SEGBF cell enqueue operation to actually execute in reverse order. This is due to IBM3206K0424 internal priority levels and could result in SEGBF fetching the LCD data before it has been updated to the new value. For this reason, it is highly advisable to use the LCD update mechanism in *Cell/Packet Re-assembly (REASM)*.

The transmit portion of the LCD can be subdivided into three distinct parts based on which chip functions or entities access that particular part of the LCD. The first three 64-bit words are scheduling related and are accessed only by CSKED. The next two 64-bit words (four if using switch fabric header) are related to both scheduling and segmentation and are accessed and shared by both CSKED and SEGBF. The words following these shared locations are related only to segmentation and are accessed only by SEGBF. The number of 64-bit words in this portion of the LCD can vary from one to three. The actual number being used in an LCD is determined by the segmentation processor code entry point field of that LCD. The following figure is the layout of the entire transmit portion of the LCD.

Overall Transmit LCD Layout

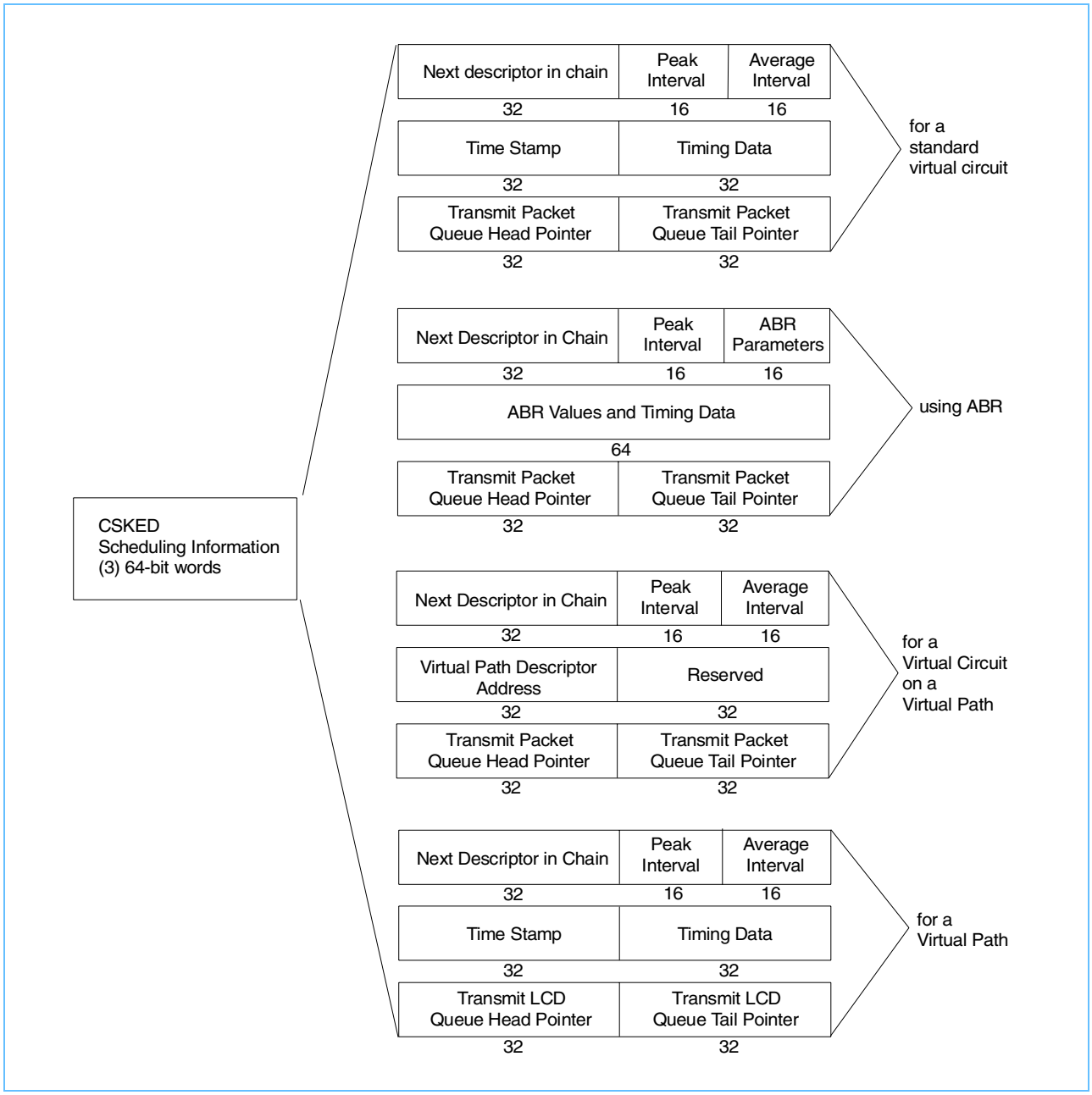


The three 64-bit words containing CSKED scheduling information can have four different formats depending upon whether scheduling is based on virtual path parameters or is configured for ABR.

If LCD-based memory management is used, a 64-bit section is inserted at byte offset ‘18’X.

On ABR connections, the ABR code running in PCORE will use ten bytes of the LCD for managing the connection, starting at offset 0x46 in the LCD.

Scheduling Portion of a Transmit Descriptor



Transmit Logical Channel Descriptor Structure

```
  .
typedef struct {
    bit32 next_lcd;
    bit16 #peak_interval;
    bit16 #average_interval;

    bit32 #timestamp;
    bit11 Reserved;
    bit1  remove_lcd;
    bit1  lc_on_timewheel;
    bit3  #alter_sched;
    bit2  #transmit_priority;
    bit1  #max_resolution;
    bit3  #max_burst_mult;
    bit10 #max_burst_value;

    bit26 head_packet_pointer;
    bit1  free_on_xmit;
    bit1  queue_on_xmit;
    bit1  conn_suss;
    bit1  #drop;
    bit26 tail_packet_pointer;
    bit6  reserved;

    bit16 transmit_length;
    bit8  buffer_offset;

    bit2  xmt_cmp_evt_mod;
    bit2  reserved;
    bit4  seg_prc_Entry_point;

    bit32 #ATM_header;

    bit32 segmentation_pointer;
    bit32 current_CRC;

    bit32 xmit_stat1;
    bit32 xmit_stat2;

  } tx_lcd_struct, *tx_lcd_struct_ptr;
```

Transmit Logical Path Descriptor Structure

```
  .
typedef struct {

    bit32 next_lcd;
    bit16 #peak_interval;
    bit16 #average_interval;

    bit32 #timestamp;
    bit11 Reserved;
    bit1  remove_lcd;
    bit1  lc_on_timewheel;
    bit3  #alter_sched;
    bit2  #transmit_priority;
    bit1  #max_resolution;
    bit3  #max_burst_mult;
    bit10 #max_burst_value;

    bit26 forward_LCD_pointer;
    bit6  reserved;

    bit26 backwared_LCD_pointer;
    bit6  reserved;


} tx_lpd_struct, *tx_lpd_struct_ptr;
```

Redefinition of Transmit Logical Channel Descriptor for Connections Sharing (Logical Path Bandwidth)

```
typedef struct {
    bit32 next_lcd;
    bit32 reserved;

    bit32 lcd_pointer;
    bit11 reserved;
    bit1  remove_lcd;
    bit1  lc_on_timewheel;
    bit3  #alter_sched;
    bit2  #transmit_priority;
    bit14 reserved;

    bit26 head_packet_pointer;
    bit1  free_on_transmit;
    bit1  queue_on_transmit;
    bit1  dma_on_transmit;
    bit1  conn_suss;
    bit2  #drop;
    bit26 tail_packet_pointer;
    bit6  reserved;

    bit16 transmit_length;
    bit8  buffer_offset;
    bit2  xmt_cmp_evt_mod;
    bit2  reserved;
    bit4  seg_prc_Entry_point;
    bit32 #ATM_header;

    bit32 segmentation_pointer;
     bit32 current_CRC;

    bit32 xmit_stat1;
    bit32 xmit_stat2;


} tx_lcd_struct, *tx_lcd_struct_ptr;
```

Redefinition of Shared and Segmentation Portion of Transmit LCD for ABR

```
typedef struct {
    bit16 transmit_length;
    bit8  buffer_offset;

    bit2  xmt_cmp_evt_mod;
    bit2  reserved;
    bit4  seg_prc_Entry_point;

    bit32 #ATM_header;

    bit32 segmentation_pointer;
    bit32 current_CRC;

    bit32 xmit_stat1;
    bit32 xmit_stat2;


    bit16 reserved;
    bit16 explicit_rate;
    bit16 current_rate;
    bit16 minimum_rate;

    bit32 backward_ptr;

    bit32 reserved;

} tx_lcd_struct, *tx_lcd_struct_ptr;
```

Owing to the use of certain LCD fields, a connection running ABR cannot be set up for segmentation AAL types 0x6 (fixed-sized blocking) or 0x7 (MPEG-2 assist).

Redefinition of Segmentation Portion of Transmit LCD for Fixed Size AAL5 Blocking (segmentation type 0x6)

```
typedef struct {

    bit16 transmit_length;
    bit8  buffer_offset;

    bit2  xmt_cmp_evt_mod;
    bit2  reserved;
    bit4  seg_prc_Entry_point;

    bit32 #ATM_header;

    bit32 segmentation_pointer;
    bit32 current_CRC;

    bit32 xmit_stat1;
    bit32 xmit_stat2;

    bit8  #Packets_per_AAL5_frame
    bit8  #Blocking_size (4 bytes x'2F' for MPEG-2)
    bit8  Current_transport_stream_packet
    bit8  Current_Blocking_Count (4 bytes)
    bit32 reserved;

} tx_lcd_struct, *tx_lcd_struct_ptr;
```

Redefinition of Segmentation Portion of Transmit LCD for MPEG2 (segmentation type 0x7)

```
typedef struct {

    bit16 transmit_length;
    bit8  buffer_offset;

    bit2  xmt_cmp_evt_mod;
    bit2  reserved;
    bit4  seg_prc_Entry_point;

    bit32 #ATM_header;

    bit32 segmentation_pointer;
    bit32 current_CRC;

    bit32 xmit_stat1
    bit32 xmit_stat2

    bit8  #Packets_per_AAL5_frame
    bit8  #Blocking_size (4 bytes x'2F' for MPEG-2)
    bit8  Current_transport_stream_packet
    bit8  Current_Blocking_Count (4 bytes)
    bit32 reserved;

    bit32 reserved;
    bit32 reserved;

} tx_lcd_struct, *tx_lcd_struct_ptr;
```

Redefinition of Scheduling Portion of Transmit LCD for ABR

```
typedef struct {
    bit32 next_lcd;
    bit16 #peak_interval;
    bit3  #Nrm;
    bit3  #Trm;
    bit10 #Tadtf;

    bit8  #Nc;
    bit8  #Ncrm;
    bit16 Reserved;
    bit11 Tlrm1;
    bit1  remove_lcd;
    bit1  lc_on_timewheel;
    bit3  #alter_sched;
    bit2  #transmit_priority;
    bit1  #max_resolution;
    bit13 Tlrm2;

    bit26 head_packet_pointer;
    bit1  free_on_xmit;
    bit1  queue_on_xmit;
    bit4  reserved;
    bit26 tail_packet_pointer;
    bit6  reserved;

} tx_lcd_struct, *tx_lcd_struct_ptr;
```

Redefinition of Scheduling Portion of Transmit LCD for Timers

```
typedef struct {
    bit32 next_lcd;
    bit32 #timer_period;

    bit32 #timestamp;
    bit12 reserved;
    bit1  lc_on_timewheel;
    bit1  reserved;
    bit2  #timer_type;
    bit2  #transmit_priority;
    bit1  #max_resolution;
    bit13 reserved;

    bit32 #dma_desc_addr;
    bit32 reserved;

} tx_lcd_struct, *tx_lcd_struct_ptr;
```

Definition of LCD-Based Memory Management of Transmit LCD

```
typedef struct {
    bit16 #threshold_1;
    bit16 #threshold_2;
    bit4  #pool_id1;
    bit4  #pool_id2;
    bit24 #bytes_queued;
} tx_lcd_struct, *tx_lcd_struct_ptr;
```

Definition of ABR Code Variables

```
typedef struct {
    bit8  #CRM;
    bit8  #iCDF;
    bit16 #iMCR;
    bit16 #PCR;
    bit8  #iRDF;
    bit8  #iRIF;
    bit16 #ICR;
} tx_lcd_struct, *tx_lcd_struct_ptr;
```

Field Definitions

The following is a detailed description of the fields listed above. This data structure should be initialized at connection setup but not modified while transmission is occurring on the connection. Only those fields marked with a # typically need to be initialized to something other than zero.

ABR Code Variables Definitions (Page 1 of 4)

Field Name	Field Description
next_lcd	This field is used by the hardware to chain LCDs together on queues. It contains the address of the next LCD if one exists.
peak_interval #	This field contains the minimum spacing allowed between consecutive cells on this connection. This spacing is expressed in cell times. A connection that can transmit every cell time would have a value of ‘1’ for this field.
average_interval #	This field contains the minimum average spacing allowed between cells transmitted on this connection. It is the inverse of the Sustainable Cell Rate. The value for this field is expressed in cell times.
Nrm #	This field specifies the maximum number of cells a source may send for each forward RM cell. Number of cells = (2**Nrm)+1.
Trm #	This field provides an upper bound on the time between forward RM cells for an active source. Time = 100•(2**-Trm) msec.
Tadtf #	The ACR Decrease Time Factor is the time permitted between sending RM cells before the rate is decreased to ICR. Time = Tadtf• 0.01 sec.
Nc #	This field is used as a counter to determine when IBM3206K0424 cells have been sent. It should be initialized at connection setup time to ‘0’.
Ncrm #	This field is used as a counter to determine when CRM RM cells have been sent. It should be initialized at connection setup time to CRM.
timestamp #	This field contains a timestamp used by the hardware to determine if transmit opportunity credits exist and if the Burst Tolerance has been exceeded. It should be initialized at connection setup time to the value in the current timeslot counter.
Tlrm1 & 2	These fields are used by the hardware to determine when the last RM cell was sent. They should be initialized to ‘0’.
lc_on_timewheel #	This field indicates if the LCD is currently queued to the timewheel. It should be initialized to ‘0’.
remove_lcd #	If this bit is set, the LCD will be removed from the time wheel at next transmission opportunity. It should be initialized to ‘0’.

ABR Code Variables Definitions (Page 2 of 4)

Field Name	Field Description
alter_sched #	These encoding bits alter the scheduling of cells on a Virtual Circuit (VC) 000 = Normal Scheduling — Scheduling is not altered. 001 = VC on VP — This VC is contained on a virtual path and will share the VP bandwidth after one packet is sent. The scheduling parameters are contained in the descriptor for the virtual path that is pointed to by the Virtual Path Descriptor Address field in this LCD. 010 = MPEG-2 Scheduling — The cells being sent out on this connection are monitored for a Peak Cell Rate (PCR). If a PCR is found, the AAL5 packet is terminated at the end of the MPEG-2 frame and the last cell is scheduled to go out at the time specified in the PCR. 011 = Packet-based scheduling — Packets will be scheduled at the average interval and cells within the packet will be scheduled at the peak interval. This is useful for sending information where variably-sized packets need to be sent at regular intervals. 100 = ABR scheduling — This VC will send Resource Management cells and adjust its transmission rate according to the behaviors specified in the ATM Forum Traffic Management Specification, Version 4.0. 101 = Fair VC on VP — This VC is contained on a virtual path and will share the VP bandwidth after one cell is sent. The scheduling parameters are contained in the descriptor for the virtual path which is pointed to by the Virtual Path Descriptor Address field in this LCD. 110 = Reserved — For MPEG-2 scheduling. 111 = Reserved
transmit_priority #	This field specifies the priority of transmission on this connection: 0=high, 1=medium, 2=low.
max_resolution #	If this bit is set, the lower eight bits of the average interval and peak interval parameters contain a fractional component. This allows a finer resolution for scheduling. For example, for a peak interval of 1.5 time units, the value written to the peak_interval field should be hex 0180. If this bit is set, the initial value of timestamp should contain the current timeslot counter shifted 16 bits to the left.
max_burst_mult #	The values in this field and the next field are used to limit the number of cells that can be transferred at the peak rate. The max_burst_value will be multiplied by four to the power of the value in this field to yield the maximum credit time. This time is expressed in cell times and represents the time it would take to acquire the maximum number of cell credits. This maximum credit time should equal the maximum number of cells that can be transferred at the peak rate (MBS) times the difference between the average and intervals. Maximum credit time = MBS * (AI-PI) where MBS = maximum burst size, AI = average interval, and PI = peak interval. MBS must be at least one to transmit at peak rate. If MBS is not at least one, the peak interval should be set to the average interval.
max_burst_value #	The value in this field will be multiplied by four to the power of the value in the max_burst_mult field to yield the maximum credit time.
head_packet_pointer	This field is used to chain buffers to LCDs.
tail_packet_pointer	This field is used to chain buffers to LCDs.
transmit_length	This field contains the length of the currently transmitted packet.
free_on_xmit	This bit is set if the header of the currently transmitted packet has specified that the packet is to be freed after transmission.
queue_on_xmit	This bit is set if the header of the currently transmitted packet has specified that the packet is to be queued after transmission.
dma_on_xmit	This bit is set if the header of the currently transmitted packet has specified that a DMA descriptor is to be queued after transmission.

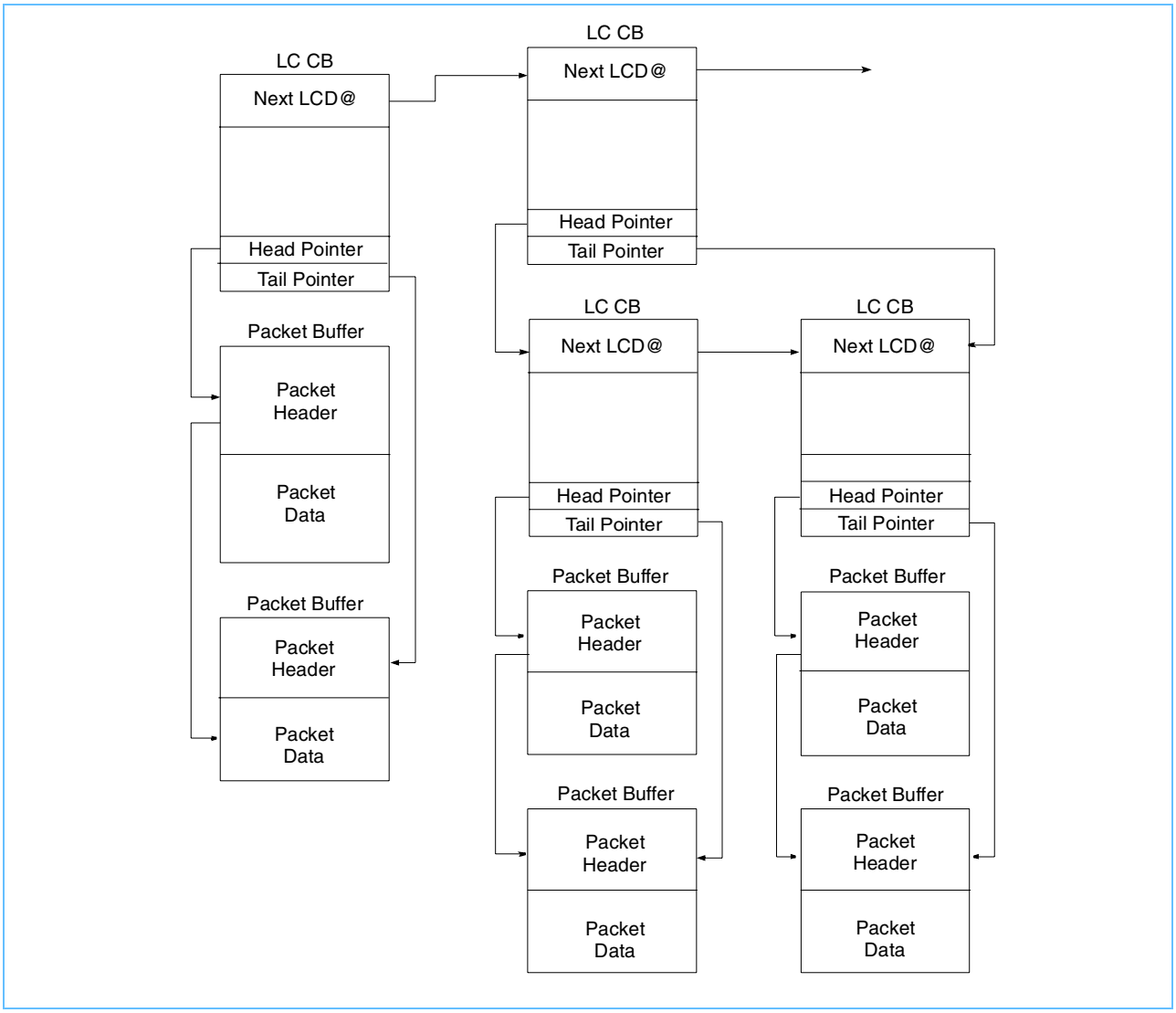
ABR Code Variables Definitions (Page 3 of 4)

Field Name	Field Description
conn_suss	This bit is used on ABR connections to suspend transmission. It should be initialized to '0'.
drop	These two bits are used to indicate which physical drop will be used for this connection. Traffic can be scheduled on up to four drops.
buffer_offset	This field contains the offset into the buffer that the transmit data starts.
xmt_cmp_evt_mod	This two-bit field can optionally (based on a bit in the SEGBF control register) be logically ORed with bits 8-7 of a transmit complete buffer event when it is generated by the segmentation logic. This field will only be ORed when buffer address events are being generated. It will have no affect when LCD addresses are being enqueued when a transmit complete event occurs.
seg_prc_Entry_point #	This four-bit field is loaded into the instruction pointer for both the LCD update processor and the cell generation processor when a cell opportunity occurs for the LCD. The value loaded into this field defines what type of cell will be generated by the segmentation logic. Possible types include raw 48- and 52-byte cells, AAL5 cells, switch bound cells (48-byte payload), extended switch bound cells (54-byte payload), and frame-based cells. The actual values that are associated with each type of cell will be defined at a later time.
ATM_header #	This field contains the first four bytes of the ATM header.
segmentation_pointer	This field contains a pointer to the next data to be transmitted. In normal operation, this field is initialized by the cell scheduler when a new frame is queued for segmentation.
current_CRC	This field contains the CRC as it is being built.
Current_Blocking_Count	This eight-bit field contains the current count of four-byte values that have been assembled into cells and sent out on this LCD for all fix block or MPEG AAL types. Other than initialization, this field should only be accessed by the hardware.
Fixed_Blocking_size #	This eight-bit field should be initialized by the software to contain the number of four-byte values that constitute a packet. For MPEG2, this register should be set to x'2F' (4•x'2F' = 188 byte transport stream packet).
Current_transport_stream_packet	This eight-bit field contains the number of the current transport stream packet that is being segmented. Other than initialization, this field should only be accessed by the hardware.
Packets_per_AAL5_frame #	This eight-bit field should be initialized by the software to indicate how many packets should be concatenated into an AAL5 frame.
explicit_rate	This 16-bit field contains the explicit cell rate as defined for ABR traffic on this LCD.
current_rate	This 16-bit field contains the current cell rate as defined for ABR traffic on this LCD.
minimum_rate	This 16-bit field contains the minimum cell rate as defined for ABR traffic on this LCD.
backward_ptr	When software needs to send a backward RM cell, this 32-bit field should be updated with the address of a buffer that contains the desired backward RM cell. After the segmentation logic transmits the cell, this field is cleared by the hardware.
xmit_stat1	This 32-bit field contains a count of one of three things: the total number of user cells that have been sent on this LCD, the total number of bytes that have been sent on this LCD, or the total number of frames that have been sent on this LCD. This field should zeroed when the connection is initialized. An event will be generated when this count wraps.
xmit_stat2	This 32-bit field contains a count of one of three things: the total number of user cells that have been sent on this LCD, the total number of bytes that have been sent on this LCD, or the total number of frames that have been sent on this LCD. This field should be zeroed when the connection is initialized. An event will be generated when this count wraps.
threshold_1&2	These fields are compared to the upper 24 bits of the bytes_queued field to determine when a threshold is crossed and the POOL ID for the received LCD should be changed.

ABR Code Variables Definitions (Page 4 of 4)

Field Name	Field Description
pool_id1&2	These fields are used to change the POOL ID when a threshold is crossed.
bytes_queued	This field is used to keep track of the number of bytes queued for transmission on this LCD.
timer_type #	These encoded bits determine the time of timer: 00 = Relative non-periodic timer — The expiration time will be in one timer period. The timer will not be scheduled again automatically. 01 = Relative periodic timer — The expiration time will be in one timer period. The timer will be automatically scheduled again. 10 = Absolute non-periodic timer — The timer will expire at the time specified by the timestamp field. The timer will not be automatically scheduled again. 11 = Absolute periodic timer — The timer will expire at the time specified by the timestamp field. The timer will be scheduled again, automatically, using the time specified in the timer_period field.
timer_period #	This field specifies the number of timeslots before the timer expires.
dma_desc_addr #	The DMA descriptor pointed to by this field will be queued for execution when the timer expires.
CRM	Missing RM cell count. CRM limits the number of forward RM cells that may be sent in the absence of received backward RM cells. CDF is written to the NCRM field whenever a backward RM cell is detected.
iCDF	Cutoff Decrease Factor (CDF) controls the decrease in ACR associated with CRM. CDF is zero or a power of 2 value in the range of 1/64 to 1. iCDF represents the power of 2 that is in the denominator of CDF. CDF = 1/(2**iCDF) so iCDF = log (base 2) of 1/CDF. Range = 1 to 6. A zero value for CDF should be represented as 0xFF for iCDF.
iMCR	This is the reciprocal of the Minimum Cell Rate (MCR). MCR is represented in the ABR Rate format. iMCR needs to be an integer representing the interval between cells, in units of timeslots per cell. The following formula illustrates the conversion from MCR to iMCR. iMCR = 1/(MCR) * 53 * 8 * (TSP/CI*(2**23)). The Timeslot Prescalar (TSP) is defined by the CSKED Timeslot Prescalar Register. The clock interval (CI) is determined by the CRSET Clock Control Register. With the TSP set to one timeslot per cell transmission time, iMCR =1 for a full bandwidth, 2 for a half bandwidth, etc.
PCR	The Peak Cell Rate (PCR) is the cell rate that the source may never exceed. PCR should be in the ABR Rate format.
iRDF	Rate Decrease Factor (RDF) controls the decrease in the cell transmission rate. RDF is a power of 2 value in the range of 1/32,768 to 1. iRDF represents the power of 2 that is in the denominator of RDF. RDF = 1/(2**iRDF) so iRDF = log (base 2) of 1/RDF. Range = 1 to 15.
iRIF	Rate Increase Factor (RIF) controls the increase in the cell transmission rate. RIF is a power of 2 value in the range of 1/32,768 to 1. iRIF represents the power of 2 that is in the denominator of RIF. RIF = 1/(2**iRIF) so iRIF = log (base 2) of 1/RIF. Range = 1 to 15.
ICR	The Initial Cell Rate is the rate at which a source should send initially and after an idle period. ICR should be in the ABR Rate format.

Transmit Data Structure Linkage



Receive LCD Data Structure and Modes

The format of the receive LCD structure depends on which AAL is being configured and which options are used. It is also dependent on if TCP/IP checksum verification has been enabled. When TCP/IP checksum verification is enabled, 16 additional bytes are added to the LCD format. TCP/IP checksum is enabled in the REASM Mode Register. The following are the basic layouts of the receive LCD:

Basic Receive LCD Layout

```
struct BasicRxLcd {
  bit32 packedInfo;
  bit32 crcState;

  bit32 misc;
  bit32 buffPtr;

  bit32 stat0;
  bit32 stat1;

  bit32 hostData;

  bit32 misc2;


  bit32 reserved;
  bit32 reserved;
  bit32 reserved;

  bit32 reserved;

};

struct ipWrd0  {;
  bit32  ipWrd0;
    bit4  frameType;
    bit6  skipCount;
    bit6  reserved;
    bit16 reserved;
  };
```

```
struct BasicRxLcdIP {
  bit32 packedInfo;
  bit32 crcState;

  bit32 ipWrd0;
  bit32 ipWrd1;
  bit32 ipWrd2;
  bit32 ipWrd3;

  bit32 misc;
  bit32 buffPtr;



  bit32 stat0;
  bit32 stat1;

  bit32 hostData;

  bit32 misc2;

};
```

The basic layout is the same for all LCD types. Only the packed Info and misc fields vary between the different LCD types. The following sections detail the receive LCD and the differences from the basic layout for each major option.

Raw LCD

Raw LCD Packed and Miscellaneous Field Layouts

```
struct Packed {
  bit4  aalType;                  // 0000 - raw
  bit2  ppMode;                   // 00   - normal
  bit2  state;                    // 00->down 01->idle/enabled

  bit1  reserved;                 // set to zero
  bit1  reserved;                 // set to zero
  bit1  reserved;                 // set to zero
  bit1  size;                     // 1->52 byte cell 0->48 byte cell
  bit1  storeCrc10;
  bit3  reserved;

  bit4  rxqNum;
  bit4  rxPoolId;
  bit8  rxOffset;
};

struct Misc {
  bit6  reserved;
  bit2  packHeadSel;
  bit8  reserved;
  bit16 oamMask;
};

struct Misc2 {
  bit32 reserved;
};
```

A raw LCD allows raw ATM cells to be received with no reassembly. The user can select to receive 52- or 48-byte cells. The packet header may or may not contain the ATM header. The cell data is then placed after the packet header at the configured receive offset. The 52-byte mode stores the entire cell minus the HEC, and the 48-byte mode stores only the ATM cell payload. Optional CRC-10 checking is available in raw modes.

Raw Routed LCD

Raw Routed LCD Packed and Miscellaneous Field Layouts

```
struct Packed {
  bit4  aalType;                    // 0000 - raw
  bit2  ppMode;                     // 01   - routed
  bit2  state;                      // 00->down 01->idle/enabled

  bit1  reserved;                   // set to zero
  bit1  reserved;                   // set to zero
  bit1  reserved;                   // set to zero
  bit1  size;                       // 1->52 byte cell 0->48 byte cell
  bit1  reserved;
  bit3  reserved;

  bit4  rxqNum;
  bit4  rxPoolId;
  bit8  rxOffset;
};

struct Misc {
  bit6  reserved;
  bit2  packHeadSel;
  bit8  reserved;
  bit16 oamMask;
};

struct Misc2 {
  bit32 routedLcd;
};
```

A raw routed LCD receives data in the same way that a raw LCD does. Once received, the cell buffer is routed internally to the scheduler and rescheduled for transmission. Normally, when a cell is received, the receive LCD address is written into the packet header and the buffer is surfaced to the user. When a cell is routed, the routedLcd field is used to fill in the LCD address in the packet header. This allows cells to be routed out the transmit interface with the same or different VPI/VCI.

The low order bits in the routedLcd field should be set correctly to free the buffer on transmission. These bits correspond to the flag bits in the packet header. Raw routing is also called forwarded or fast forward mode.

Raw Routed Early Drop LCD

Raw Routed Early Drop LCD Packed and Miscellaneous Field Layouts

```
struct Packed {
  bit4  aalType;              // 0001 - raw early drop
  bit2  ppMode;               // 01   - routed
  bit2  state;                // 00->down 01->idle/enabled 11->error

  bit1  reserved;             // set to zero
  bit1  reserved;             // set to zero
  bit1  reserved;             // set to zero
  bit1  size;                 // 1->52 byte cell 0->48 byte cell
  bit4  finalPoolId;          // pool id

  bit4  rxqNum;
  bit4  rxPoolId;
  bit8  rxOffset;
};

struct Misc {
  bit6  reserved;
  bit2  packHeadSel;
  bit8  reserved;
  bit16 oamMask;
};

struct Misc2 {
  bit32 routedLcd;
};
```

A raw routed early drop LCD receives data in the same way that a raw LCD does. Once received, the cell buffer is then routed internally to the scheduler and rescheduled for transmission. Normally when a cell is received, the receive LCD address is written into the packet header and the buffer is surfaced to the user. When a cell is routed, the routedLcd field is used to fill in the LCD address in the packet header. This allows cells to be routed out the transmit interface with the same or different VP/VC.

This mode should only be used when the routed cell stream is actually an AAL5 packet stream. In this mode, a cell being dropped due to resource causes the LCD to go into error mode until the cell that contains the user indicate (UIND) bit is received. All cells received in error mode are dropped, except the final cell which is forwarded. This conserves bandwidth while maintaining the AAL5 integrity. The finalPoolId provides a second poolid for the final cells to use to be sure that these final cells are always forwarded even when resources are low. The finalPoolId is only used if no buffers are available in the normal pool.

The low order bits in the routedLcd field should be set correctly to free the buffer on transmit. These bits correspond to the flag bits in the packet header.

This is also called forwarded or fast forward mode.

Raw Scatter/Cut-Through LCD

Raw Scatter/Cut-Through LCD Packed and Miscellaneous Field Layouts

```
struct Packed {
  bit4  aalType;            // 0000 - raw
  bit2  ppMode;             // 10   - scatter/cut through
  bit2  state;              // 00->down 01->idle/enabled

  bit1  reserved;           // set to zero
  bit1  reserved;           // set to zero
  bit1  reserved;           // set to zero
  bit1  size;               // 1->52 byte cell 0->48 byte cell
  bit1  storeCrc10;
  bit1  reserved;
  bit2  cutThruSel;

  bit4  rxqNum;
  bit4  rxPoolId;
  bit8  rxOffset;
};

struct Misc {
  bit6  reserved;
  bit2  packHeadSel;
  bit8  reserved;
  bit16 oamMask;
};

struct Misc2 {
  bit32 reserved;
};
```

A raw scatter/cut-through LCD receives data in the same way that a raw LCD does. Once received, the cutThruSel field is used to select one of four configurations. Each configuration specifies a receive queue and a DMA queue. The cut-through selector is used to select a cut-through/scatter configuration. The DMA descriptor is then built using the cell buffer address and the data length and the flags specified in the cut-through configuration. After being built, it is enqueued to the DMA queue specified. If there is no DMA descriptor available, then a no descriptor event is enqueued.

AAL5 LCD

AAL5 LCD Packed and Miscellaneous Field Layouts

```
struct Packed {
  bit4  aalType;              // 0101 - aal5
  bit2  ppMode;               // 00   - normal
  bit2  state;                // 00->down 01->idle/enabled 10->reasm 11->error

  bit1  reserved;             // set to zero
  bit1  rtoTest;              // set to zero
  bit1  rtoEnable;
  bit1  tmpCLP;
  bit1  tmpCongestion;
  bit3  reserved;

  bit4  rxqNum;
  bit4  rxPoolId;
  bit8  rxOffset;
};

struct Misc {
  bit6  reserved;
  bit2  packHeadSel;
  bit8  reserved;
  bit16 oamMask;
};

struct Misc2 {
  bit32 reserved;
};
```

An AAL5 LCD allows AAL5 packets to be received with no special processing.

AAL5 Routed LCD

AAL5 Routed LCD Layout

```
struct Packed {
  bit4  aalType;            // 0101 - aal5
  bit2  ppMode;             // 01   - routed
  bit2  state;              // 00->down 01->idle/enabled 10->reasm 11->error

  bit1  reserved;           // set to zero
  bit1  rtoTest;            // set to zero
  bit1  rtoEnable;
  bit1  tmpCLP;
  bit1  tmpCongestion;
  bit3  reserved;

  bit4  rxqNum;
  bit4  rxPoolId;
  bit8  rxOffset;
};

struct Misc {
  bit6  reserved;
  bit2  packHeadSel;
  bit8  reserved;
  bit16 oamMask;
};

struct Misc2 {
  bit32 routedLcd;
};
```

An AAL5 Routed LCD allows AAL5 packets to be received. Once received, the packet buffer is then routed internally to the scheduler and rescheduled for transmission. Normally when a packet is received, the receive LCD address is written into the packet header and the buffer is surfaced to the user. When a packet is routed, the routedLcd field is used to fill in the LCD address in the packet header. This allows packets to be routed out the transmit interface with the same or different VP/VC.

The low order bits in the routedLcd field should be set correctly to free the buffer on transmission. These bits correspond to the flag bits in the packet header.

AAL5 routing is also called forwarded or fast forward mode.

Note: Non-user data cells are terminated.

AAL5 Cut-Through/Scatter Mode LCD

AAL5 Cut-Through/Scatter Mode LCD Packed and Miscellaneous Field Layouts

```
struct Packed {
    bit4   aalType;     // 0101 - aal 5
    bit2   ppMode;      // 10   - scatter
    bit2   state;       // 00->down 01->idle/enabled 10->reasm 11->error


    bit1   reserved;    // set to zero
    bit1   rtoTest;     // set to zero
    bit1   rtoEnable;
    bit1   tmpCLP;
    bit1   tmpCongestion;
    bit1   reserved;
    bit2   cutThruSel;


    bit4   rxqNum;
    bit4   rxPoolId;
    bit8   rxOffset;
  };


  struct Misc {
    bit6  numDesc;
    bit2  packHeadSel;
    bit8  reserved;
    bit16 oamMask;
  };

  struct Misc2 {
    bit5  reserved;
    bit1  useCrcNumHead;
    bit10 numHeadBytes;
    bit16 reserved;
  };
```

Packet LCD

Packet LCD Packed and Miscellaneous Field Layouts

```
struct Packed {
  bit4  aalType;            // 0111 - packet
  bit2  ppMode;             // 00   - normal
  bit2  state;              // 00->down 01->idle/enabled 10->reasm 11->error

  bit1  reserved;           // set to zero
  bit1  rtoTest;            // set to zero
  bit1  rtoEnable;
  bit5  reserved;

  bit4  rxqNum;
  bit4  rxPoolId;
  bit8  rxOffset;
};

struct Misc {
  bit6  reserved;
  bit2  packHeadSel;
  bit8  reserved;
  bit16 reserved;
};

struct Misc2 {
  bit5  dropNBytes;
  bit11 reserved;
  bit16 reserved;
};
```

A packet LCD allows packets from the POS-PHY to be received with no special processing. The header-Thresh can be used to allow packet header thresholding events to be surfaced.

Packet Routed LCD

Packet Routed LCD Packed and Miscellaneous Field Layouts

```
struct Packed {
  bit4  aalType;              // 0111 - packet
  bit2  ppMode;               // 01   - routed
  bit2  state;                // 00->down 01->idle/enabled 10->reasm 11->error

  bit1  reserved;             // set to zero
  bit1  rtoTest;              // set to zero
  bit1  rtoEnable;
  bit5  reserved;

  bit4  rxqNum;
  bit4  rxPoolId;
  bit8  rxOffset;
};

struct Misc {
  bit6  reserved;
  bit2  packHeadSel;
  bit8  reserved;
  bit16 reserved;
};

struct Misc2 {
  bit32 routedLcd;
};
```

A packet routed LCD allows packets to be received from the POS-PHY. Once received, the packet buffer is then routed internally to the scheduler and rescheduled for transmission. Normally, when a packet is received, the receive LCD address is written into the packet header and the buffer is surfaced to the user. When a packet is routed, the routedLcd field is used to fill in the LCD address in the packet header. This allows packets to be routed out the transmit interface with the same or different LCD.

The low order bits in the routedLcd field should be set correctly to free the buffer on transmit. These bits correspond to the flag bits in the packet header.

This is also called forwarded or fast forward mode.

Packet Cut-Through Scatter Mode LCD

Packet Scatter Mode LCD Packed and Miscellaneous Field Layouts

```
struct Packed {
  bit4  aalType;            // 0111 - aal5
  bit2  ppMode;             // 10   - scatter
  bit2  state;              // 00->down 01->idle/enabled 10->reasm 11->error

  bit1  reserved;           // set to zero
  bit1  rtoTest;            // set to zero
  bit1  rtoEnable;
  bit3  reserved;
  bit2  cutThruSel;

  bit4  rxqNum;
  bit4  rxPoolId;
  bit8  rxOffset;
};

struct Misc {
  bit6  numDesc;
  bit2  packHeadSel;
  bit8  reserved;
  bit16 reserved;
};

struct Misc2 {
  bit5  dropNBytes;
  bit1  useCrcNumHead;
  bit10 numHeadBytes;
  bit16 reserved;
};
```

Field Definitions

The following are the definitions of the LCD fields, grouped by major function. All reserved fields should be set to zero.

Common Field Definitions

Field Name	Field Description	Note
aalType	Specifies the AAL for this LCD. The following are the valid values: 0000 = Raw Mode 0001 = Raw Mode - Early Drop 0101 = AAL5 0110 = AAL5 - 54-Byte Mode 0111 = Packet	1
ppMode	Specifies the post processing mode for this LCD. The following are the valid values: 00 = Normal 01 = Routed 10 = Scatter/Cut-Through 11 = Reserved	1
state	This specifies the reassembly state for this LCD. This field is used by the IBM3206K0424, but in order to receive cells, an LCD must be initialized to idle state. The following are the valid values: 00 = Down State 01 = Idle State 10 = Reassembling State 11 = Error State	1
cutThruSel	Specifies a cut-through configuration. The cut-through configuration specifies a receive queue to get a descriptor from, a DMA queue to enqueue the descriptor to, and a set of cut-through flags.	1
rxqNum	Specifies to which receive queue normal events should be posted. Note: some events may be routed to the error queue based on your RXQUE setup.	1
rxPoolId	Specifies which POOL ID should be used when getting buffers for received packets.	1
rxOffset	Specifies the offset into the IBM3206K0424 buffer where the received packet should be placed. A value of ‘0’ is equivalent to 256 bytes.	1
stat0	LC statistic word zero. Default counts the total users cells with CLP=0 received on this LC. For accurate counts, this should be initialized to zero.	1
stat1	LC statistic word one. Default counts the total users cells received on this LC. For accurate counts, this should be initialized to ‘0’.	1
hostData	If enabled, the contents of this field are placed in packet of each received packet for this LCD. One use of this is to place a correlator to a host-specific data structure for this LCD.	1
ipWrd0-3	When TCP/IP Verification is enabled, these words are used by the IBM3206K0424 to store state between cells for the TCP/IP verification process. The frameType field specifies an offset for the checksum nano-program. This field is used to jump to a specific algorithm to find the IP header. The skipCount field allows the user to specify a fixed amount that is always skipped before looking for the IP header using the specified algorithm. The remainder of the words should be initialized to ‘0’.	1

1. Software should set up this field.

Raw Mode Field Definitions

Field Name	Field Description	Note
size	This field specifies how many bytes are stored when cell is received: 01 = 52-Byte cell (no HEC) 00 = 48-Byte cell	1
storeCrc10	When set, the CRC-10 state bit is written into the packet header. A ‘1’ is written in the error status bit in word 0 if a bad CRC-10 is detected.	1
routedLcd	When routing cells, this field is used to fill in the LCD field of the packet header. This allows the user to dynamically route cells back out the interface using a different LCD. The user should be sure to set the free on transmit bit in this field as if it were in a packet header.	1

1. Software should set up this field.

Packet/AAL5 Field Definitions

Field Name	Field Description	Note
rtoTest	This is the reassembly timeout processing test-and-set bit. It is used by the IBM3206K0424 but should be initialized to '0'.	1
rtoEnable	If set, reassembly processing is enabled for this LC, if the LC is running AAL5.	1
tmpCLP	Used by the IBM3206K0424 to track the current state of the ORed CLP bit for the current AAL5 packet. This field should be set to '0' at initialization. After initialization, the IBM3206K0424 maintains this field.	1
tmpCongestion	Used by the IBM3206K0424 to track the current state of the ORed congestion bit for the current AAL5 packet. This field should be set to '0' at initialization time.	1
headerThresh	Specifies how much data should be received before poping a packet start event. If it is set to '0', only complete packet events will be popped.	1
buffPtr	This field is used by the IBM3206K0424, but should be initialized to zero by software. This field is used to track the current packet under reassembly.	1
crcState	This field is used by the IBM3206K0424 to maintain the CRC residue as the current packet is reassembled. It should be initialized to '0'.	
routedLcd	When routing cells, this field is used to fill in the LCD field of the packet header. This allows the user to dynamically route cells back out the interface using a different LCD. The user should be sure to set the free on transmit bit in this field as if it were in a packet header.	1
cutThruThresh	Used to determine how much cut-through data is DMAed.	1
dmaedHeader	This is used by the IBM3206K0424 for cut-through Mode 7 processing. This should be initialized to '0'.	1
numDesc	This is used by theIBM3206K0424 for scatter processing, and should be initialized to '0'.	1
packHeadSel	Specifies which packet header should be used for this connection. See *RXAAL Packet Header Configuration* on page 352 for more information.	1
oamMask	Specifies how OAM traffic should be filtered. See *ATM OAM Cell Processing* on page 322 for more information.	
useCrcNumHead	When set, specifies that receive CRC will determine how many bytes to use for numHeadBytes. This is useful when a connection is carrying IP traffic, and the TCP/IP header lengths can be determined.	1
numHeadBytes	Specifies how many bytes of data should be kept with the packet header when DMAing the final DMA list for a completed scatter packet. Must be less than the page size.	1
dropNBytes	Allows 0 - 31 bytes of packet data to be dropped from the beginning of the packet on POS-PHY networks.	1

1. Software should set up this field.

Internal Organization: Entity Descriptions

This part contains detailed descriptions of the entities which, working together, make up the IBM3206K0424. The data flows through the chip have already been described; now the details of the registers and algorithms will be revealed. The entity descriptions are numbered for easy reference.

Note on Set/Clear Type Registers

There are many registers in IBM3206K0424 that operate as a set/clear type. These registers have two addresses. The base address is for clearing bits in the register, and base address +4 bytes is for setting bits in the register. The setting or clearing operations occur only for those bits that have the value of'1' on the write of the register. Either of the addresses can be used for reading the register.

Control Processor Bus Interface Entities

Entity 1: The IOP Bus Specific Interface Controller (PCINT)

This entity provides PCI specific interfacing between the external connection and the internal entities. It will support the following functions:

- PCI memory target
- PCI master
- Address and data latching
- Provide parity error detection and generation
- Provide configuration space registers
- 64-bit data path for master and slave operation
- 64-bit addressing support for master and slave operation
- Auto 64-bit slot detection supported
- 66MHz PCI bus clock operation supported

PCI Options Taken

- Medium address decode design point
- Locking as a memory target supported
- Interrupt A will be supported, with interrupt 2 as a the sideband signal
- Registers will not burst, but cause retries when a burst is attempted
- BIST defaults set at the PCI 2 second maximum

PCI Target Response

- A Target Retry is issued if a burst crosses the end of the IBM3206K0424's memory space.
- A Target Abort will be issued if AD and command bus have bad parity (address phase parity error). Optionally, if $\overline{\text{SERR}}$ is enabled, it will also be returned.
- If enabled, the $\overline{\text{PERR}}$ signal will be driven on bad parity during data write cycles (data phase parity error) when the IBM3206K0424 is the target of the command.
- A Target Retry will be issued by the IBM3206K0424 if internal contention will cause a large bus access delay.

PCI Master Response

- A Master Abort will be issued if $\overline{\text{DEVSEL}}$ is not asserted after five clocks.
- If enabled, the $\overline{\text{PERR}}$ signal will be driven on bad parity during data read cycles (data phase parity error) when the IBM3206K0424 is the initiator of the command.

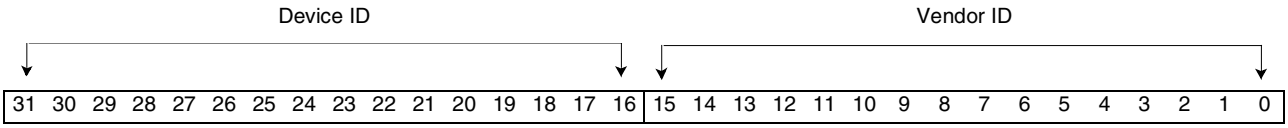
PCI Master Retry

- IBM3206K0424 will retry when requested by the slave.

1.1: PCINT Config Word 0

Identifies this device and vendor type, allocated by PCI SIG.

Length	32 bits
Type	Read Only
Address	XXXX 0000
Restrictions	Can be read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset value (Big Endian)	X'00A11014', but alterable at power-up/reset time with Crisco code. See Entity 15: on page 428 for details.
Power on Reset value (Little Endian)	X'1410A100'

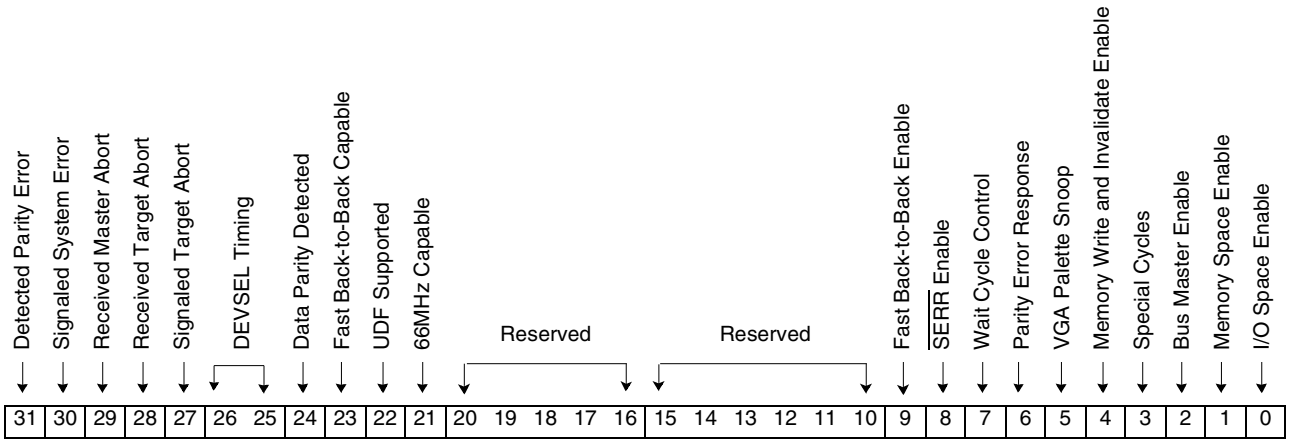


Bit(s)	PCI Spec	Name	Description
31-16	15-0	Device ID	This is a unique two-byte device ID assigned to this adapter.
15-0	15-0	Vendor ID	This is a unique two-byte vendor ID.

1.2: PCINT Config Word 1

The Status register is used to record status information for the PCI bus related events. Writing ’1’ to a bit in this register will reset that bit. The Command register provides coarse control over a device’s ability to generate and respond to PCI cycles. Access type of the Command register is read/write. See bit definitions.

Length	32 bits
Type	Read/Write and Read/Reset
Address	XXXX 0004
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset value (Big Endian)	X’02B00000’
Power on Reset value (Little Endian)	X’0000B002’



Bit(s)	PCI Spec	Name	Description
31	15	Detected Parity Error	This bit is set by the device whenever it detects a parity error, even if parity error handling is disabled (as controlled by bit 6 of PCINT Configuration Word 1).
30	14	Signaled System Error	This bit is set whenever the device asserts $\overline{\text{SERR}}$.
29	13	Received Master Abort	This bit is set by a master device whenever its transaction is terminated with master-abort, except for Special Cycle.
28	12	Received Target Abort	This bit is set by a master device whenever its transaction is terminated with target-abort.
27	11	Signaled Target Abort	This bit is set by a target device whenever its transaction is terminated with target-abort.
26-25	10-9	DEVSEL Timing	These bits are hard-wired to ’01’, assuming medium address decode.
24	8	Data Parity Detected	This bit is implemented by this bus master. It is set when this agent asserts $\overline{\text{PERR}}$ or observeS $\overline{\text{PERR}}$ asserted, AND this agent setting the bit acted as the bus master for the operation in which the error occurred, AND bit 6 of PCINT Configuration Word 1 is set.

Bit(s)	PCI Spec	Name	Description
23	7	Fast Back-to-Back Capable	Defaults to '1' unless by Crisco code. See Entity 15: on page 428 for details.
22	6	Reserved	Defaults to '0' unless set by Crisco. See Entity 15: on page 428 for details.
21	5	66MHz Capable	Defaults to '1' unless set by Crisco. See Entity 15: on page 428 for details.
20	4	Capabilities List	This bit on indicates that this device implements the pointer for a New Capabilities linked list at the offset 34th. See the PCI spec revision 2.2 for more details on New Capabilities. Defaults to '1' unless set by Crisco code. See Entity 15: on page 428 for details for more on Crisco.
19-16	3-0	Reserved	Reserved
15-10	15-10	Reserved	Reserved
9	9	Fast Back-to-Back Enable	This bit can be set to a value, but is ignored by this DMA master since it never drives these types of cycles. This slave, as indicated by bit 23, however, can handle fast back-to- back addresses to it. Initialization software will set this bit if all targets are fast back-to-back capable.
8	8	$\overline{\text{SERR}}$ Enable	If this bit is '1', the $\overline{\text{SERR}}$ driver is enabled.
7	7	Wait Cycle Control	This bit is hard-wired to '0' because stepping is not supported by this master.
6	6	Parity Error Response	When this bit is '1', normal action is taken when a parity error is detected. When it is '0', any parity errors detected are ignored and normal operation is continued.
5	5	VGA Palette Snoop	This bit is not implemented.
4	4	Memory Write and Invalidate Enable	This bit is not implemented.
3	3	Special Cycles	This bit is set to '0', and will not monitor Special Cycle operations.
2	2	Bus Master Enable	If this bit is '1', this device will be allowed to act as a bus master.
1	1	Memory Space Enable	If this bit is '1', this device will respond to memory space accesses.
0	0	I/O Space Enable	If this bit is '1', this device will respond to I/O space accesses.

1.3: PCINT Config Word 2

The Class Code is used to identify the generic function for this device. The Revision ID is used to identify the level of function for this device. See bit definitions.

Length 32 bits

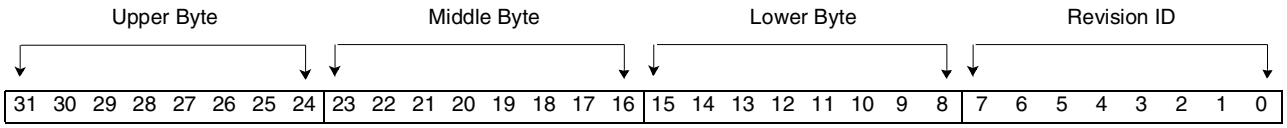
Type Read Only

Address XXXX 0008

Restrictions Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.

Power on Reset value (Big Endian) X'02030025'

Power on Reset value (Little Endian) X'25000302'



Bit(s)	PCI Spec	Name	Description
31-24	23-16	Upper Byte	The upper byte of the Class Code is a base code that broadly classifies the type of function this device performs. Code chosen is: X'02' - Network controller
23-16	15-8	Middle Byte	The middle byte of the Class Code is a sub-class code that identifies more specifically the function of this device. Code chosen is: X'03' - ATM controller
15-8	7-0	Lower Byte	The lower byte of the Class Code identifies a specific register-level programming interface so that device independent software can interact with this device. There are two secret functions here. By writing bit 15 to '1', the class code of '03' (ATM) changes to '00 (Ethernet). The read value of bit 15 remains '0'. By writing bit 14 to '1', bit 23 of Config Word 3 (also known as bit 7 of the Header Type byte) will reflect a value of '1', indicating that it is a multi-function device. The read value of bit 14 remains '0'. Bits 13 down to eight are still available to be set to indicate a register-level programming interface.
7-0	7-0	Revision ID	This is the revision level of this chip.

1.4: PCINT Config Word 3

This word specifies the system cache size in units of 32-bit words, the value of the Latency Timer for this PCI bus master, the Header Type which identifies the layout of bytes in configuration space, and the register for the control and status of BIST (Built-in self-test). See bit definitions.

Length 32 bits

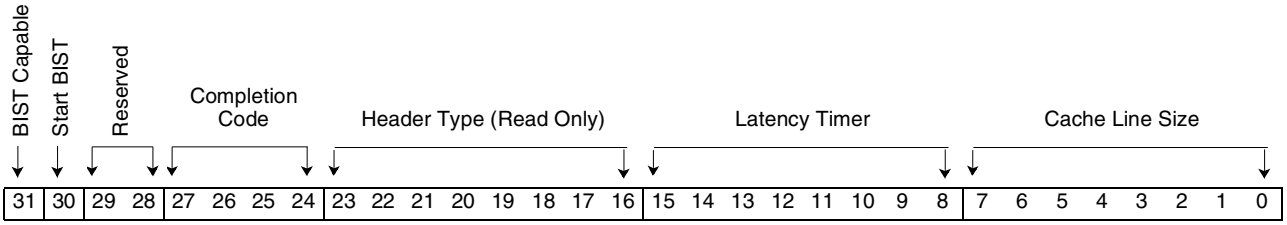
Type Read/Write

Address XXXX 000C

Restrictions Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.

Power on Reset value (Big Endian) X'80000000'

Power on Reset value (Little Endian) X'00000080'



Bit(s)	PCI Spec	Name	Description
31	7	BIST Capable	This bit is a '1' because this device supports BIST.
30	6	Start BIST	Writing this bit '1' invokes BIST. This bit is reset to '0' after BIST is complete. This bit has two seconds to reset after a start BIST action.
29-28	5-4	Reserved	Reserved
27-24	3-0	Completion Code	A value of '0' means this device has passed BIST. If bit 27 is on, the PRPG value failed. If bit 26 is on, the MISR value failed. Bits 25 and 24 are always '0'.
23-16	7-0	Header Type (Read Only)	The encoding chosen is X'00'.
15-8	7-0	Latency Timer	This register specifies a value of latency in units of PCI bus clocks.
7-0	7-0	Cache Line Size	This register is used to best determine what read command should be used by this master. Any cache line size is supported.

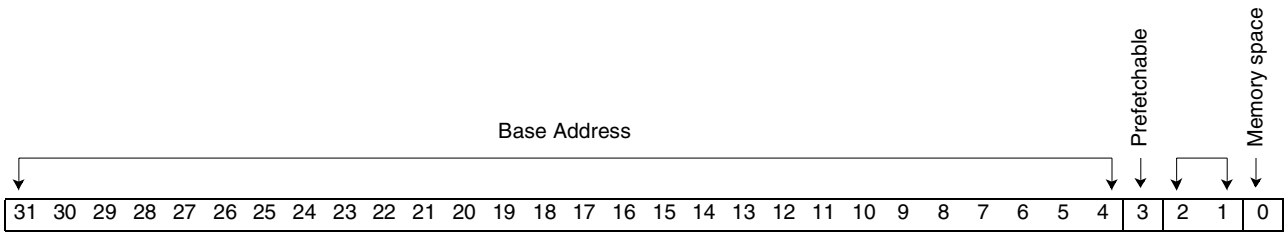
1.5: PCINT Base Address 1 (I/O for Register)

This register specifies the base address of where in PCI I/O or memory space the IBM3206K0424 registers will be mapped. When written with '1's and read back, the least significant bits read back as '0' will indicate the amount of I/O space required for this device to operate. For example, when a value of 'FFFFFFFF' is written, a value read of 'FFFFFF00' indicates that 256 bytes of address space is required. See bit definitions.

The programming of this bit depends on whether the IBM3206K0424 is in 64-bit addressing mode or not. When in 64-bit addressing mode, bit 4 of the PCINT 64-bit Controller Register is set to '1', and this register specifies a memory address. When the IBM3206K0424 is not in 64-bit addressing mode because bit 4 of the PCINT 64-bit Control Register is set to '0', this register specifies an I/O address. See bit definitions and *PCINT 64-bit Control Register* on page 116.

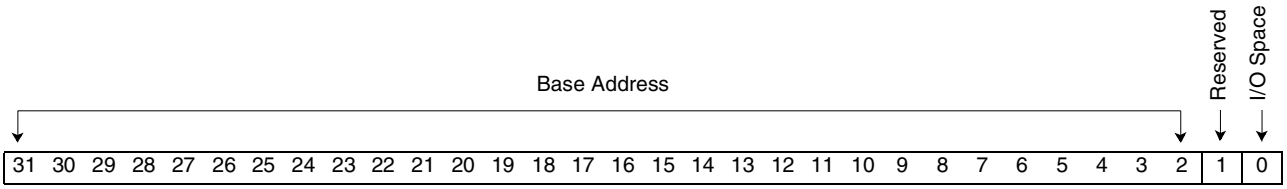
Length	32 bits
Type	Read/Write
Address	XXXX 0010
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register. Bit 17 in the PCINT Base Address Control Register must be set to allow the IBM3206K0424 to decode addresses for this range.
Power on Reset value (Big Endian)	X'00000001'
Power on Reset value (Little Endian)	X'01000000'

When in 64-bit Addressing Mode (that is, bit 64 of PCINT 64-bit Control Register is set to '1'):



Bit(s)	PCI Spec	Name	Description
31-4	31-4	Base Address	This register is used to hold the address where the target device will decode for memory accesses. The size is 32K of addressing, naturally aligned. This means that only bits 31-15 are writable.
3	3	Prefetchable	Reserved and set to '0'.
2-1	2-1		This base address can be mapped anywhere in 32-bit address space. The value of these bits is 00b.
0	0	Memory space	This is memory space, so the bit is set to '0'.

When not in 64-bit Addressing Mode (that is, bit 64 of PCINT 64-bit Control Register is set to ’0’):



Bit(s)	PCI Spec	Name	Description
31-2	31-2	Base Address.	This register is used to hold the address where the target device will decode for I/O accesses. The size is 16K of addressing, naturally aligned. This means that only bits 31-14 are writable. The PCI specification only allows 256 bytes of I/O Base Address, so this address is only for special applications. Using the feature of non-postable writes for I/O cycles must accompany enough I/O space in the system memory map.
1	1	Reserved.	Reserved and set to ‘0’.
0	0	I/O Space.	This is I/O space, so this bit is set to a ‘1’.

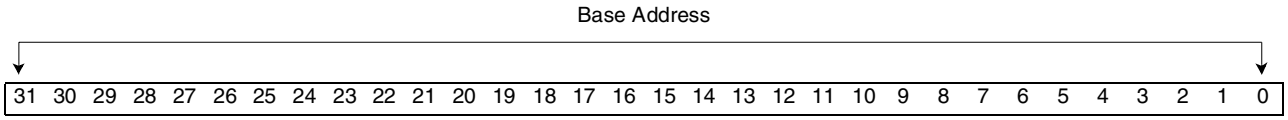
1.6: PCINT Base Address 2 (Mem for Register)

This register specifies the base address of where in PCI memory space the IBM3206K0424 registers will be mapped. When written with ’1’s and read back, the least significant bits read back as ’0’ will indicate the amount of memory space required for this device to operate. For example, when a value of ’FFFFFFFF’ is written, a value read of ’FFFFFF00’ indicates that 256 bytes of address space this required. See bit definitions.

The programming of this bit depends on whether the IBM3206K0424 is in 64-bit addressing mode or not. When in 64-bit addressing mode, bit 4 of the PCINT 64 bit Controller Register is set to ’1’, and this register specifies a memory address. When the the IBM3206K0424 is not in 64-bit addressing mode because bit 4 of the PCINT 64-bit Control Register is set to ‘0’, this register specifies an I/O address. See bit definitions and *PCINT 64-bit Control Register* on page 116.

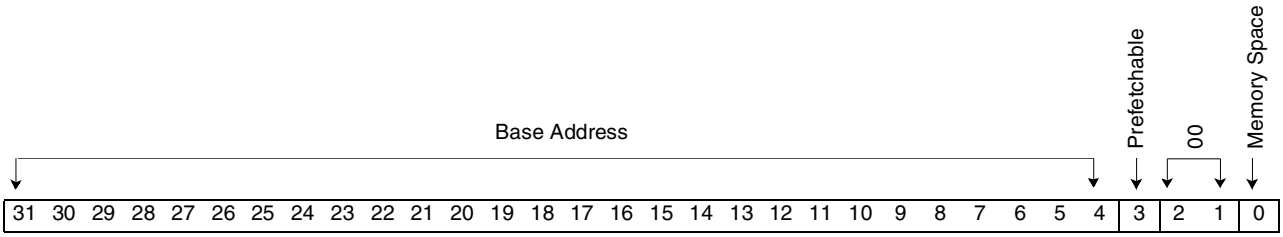
Length	32 bits
Type	Read/Write
Address	XXXX 0014
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register. Bit 16 in the PCINT Base Address Control Register must be set to allow the IBM3206K0424 to decode addresses for this range.
Power on Reset value (Big Endian)	X’00000000’
Power on Reset value (Little Endian)	X’00000000’

When in 64-bit Addressing Mode (that is, bit 64 of PCINT 64-bit Control Register is set to ’1’):



Bit(s)	PCI Spec	Name	Description
31-0	31-0	Upper part of Base Address	This register is used to hold the upper 32 bits of address during a 64 bit addressing dual cycle access.

When not in 64-bit Addressing Mode (that is, bit 64 of PCINT 64-bit Control Register is set to ’0’):



Bit(s)	PCI Spec	Name	Description
31-4	31-4	Base Address	This register is used to hold the address where the target device will decode for memory accesses. The size is 32K of addressing, naturally aligned. This means that only bits 31-15 are writable.
3	3	Prefetchable	This memory space is non-prefetchable, so this bit is set to ‘0’. This means that there are side effects on reads.
2-1	2-1		This base address can be mapped anywhere in 32 bit address space. The value of these bits is ‘00’.
0	0	Memory Space	This is memory space, so this bit is set to ’0’.

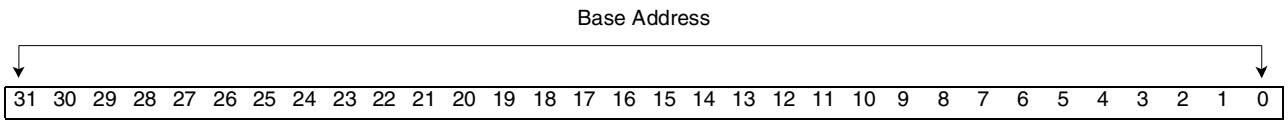
1.7: PCINT Base Addresses 3-6 (Memory)

This register specifies the base address of where in PCI memory space the IBM3206K0424 memory will be mapped. When written with ’1’s and read back, the least significant bits read back as ’0’ will indicate the amount of memory space required for this device to operate. For example, when a value of ’FFFFFFFF’ is written, a value read of ’FFFFFF00’ indicates that 256 bytes of address space this required. See bit definitions.

The mapping for the base address of registers into IBM3206K0424 memory is one-to-one, assuming a memory windowing option is not set in the PCINT Base Addr Control Register for that base address register (BAR). Multiple BARs are only used to use a given system memory map more efficiently. As required by the BAR, the addresses are size-aligned. For example, that means a 16MB size could be represented with one BAR as one 16MB size aligned on a 16MB boundary. However, four-4MB BARs could represent the same 16MB size but be aligned on any 4MB boundary. The value in any of the BARs does not map directly to any particular IBM3206K0424 memory structure, such as Control Memory. The addresses are mapped using the Virtual, Packet, and Control base address registers in VIMEN.

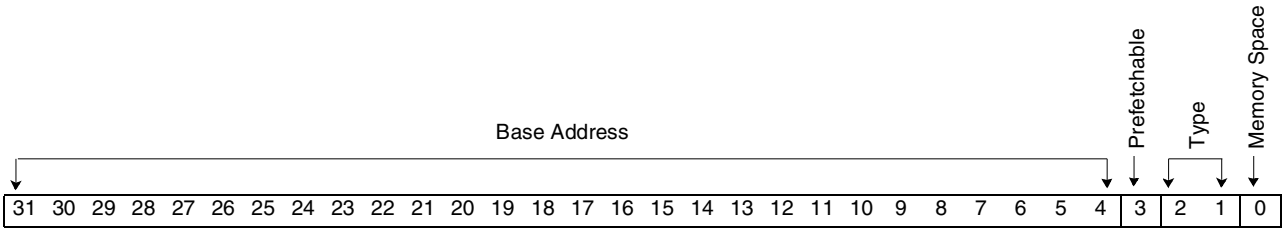
Length	32 bits	
Type	Read/Write	
Address	Reg 3	XXXX 0018
	Reg 4	XXXX 001C
	Reg 5	XXXX 0020
	Reg 6	XXXX 0024
Power on Reset value (Big Endian)	X’00000008’	
Power on Reset value (Little Endian)	X’08000000’	
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register. If one of these registers is not enabled (see PCINT Base Address Control Register), then a read of that register will return all ’0’s. The power on value stated below assumes that the register is enabled. Normally, configuration code will just read these registers to find out what is there. To enable more that the default of registers 3 and 4, the use of Crisco code could be used. See Entity 15: on page 428 for details.	

When in 64-bit Addressing Mode (that is, bit 64 of PCINT 64-bit Control Register is set to ’1’):



Bit(s)	PCI Spec	Name	Description
31-0	31-0	Upper part of Base Address	This register is used to hold the upper 32 bits of address during a 64 bit addressing dual cycle access.

When not in 64-bit Addressing Mode (that is, bit 64 of PCINT 64-bit Control Register is set to ’0’):



Bit(s)	PCI Spec	Name	Description
31-4	31-4	Base Address	This register is used to hold the address where the target device will decode for memory accesses. The size of addressing is naturally aligned and determined by what is set in the PCINT Base Address Control Register.
3	3	Prefetchable	This memory space is prefetchable, so this bit is set to a ‘1’. This means that there are no side effects on reads, all bytes are returned on reads regardless of byte enables, and host bridges can merge processor writes into this range without causing errors.
2-1	2-1	Type	This base address can be mapped anywhere in 32-bit address space. The value of these bits is 00b.
0	0	Memory Space	This is memory space, so this bit is set to a ‘0’.

Note: These registers power up to X’08000000’ if accessed little endian.

1.8: PCINT CardBus CIS Pointer

This register contains the offset to where the Card Information Structure (CIS) is located. See bit definitions.

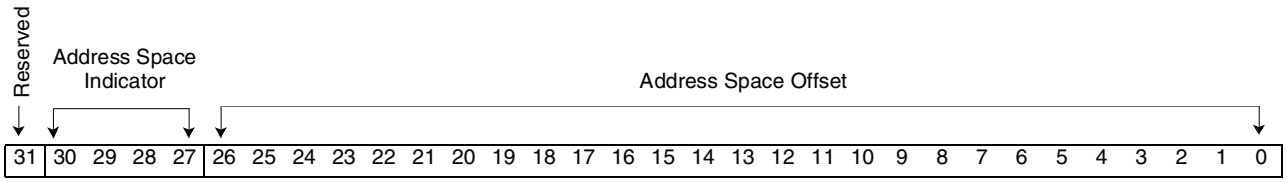
Length 32 bits

Type Read/Write

Address XXXX 0028

Restrictions Cannot be written unless by Crisco, or the PCI configuration space override write bit is on. See Entity 15: on page 428 for details.

Power on Reset value X'00000000'



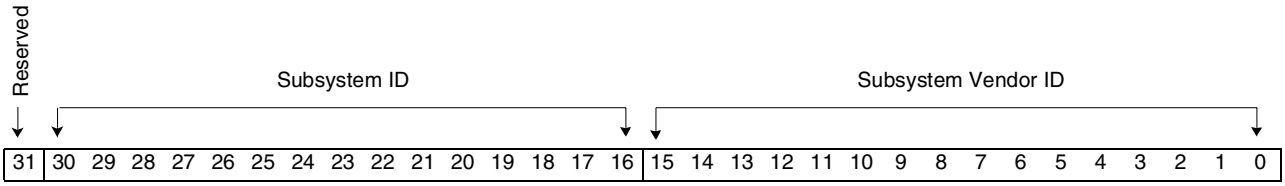
Bit(s)	Name	Description
31	Reserved	Reserved
30-27	Address Space Indicator	Can be set by Crisco code, likely to be in expansion ROM space. See Entity 15: on page 428 for details.
26-0	Address Space Offset	This field has the offset into expansion ROM that is the location of the CIS. See the PCMCIA v2.10 specification for details of the CIS.

1.9: PCINT Subsystem ID/Vendor ID

This register contains the Subsystem ID and Subsystem Vendor ID. See bit definitions.

Other possible codes that could be returned for the Subsystem ID are listed below. The correctness of their value is superseded by higher (IOA card) levels of documentation.

Length	32 bits
Type	Read/Write
Address	XXXX 002C
Restrictions	Cannot be written unless by Crisco, or the PCI configuration space override write bit is on.
Power on Reset value (Big Endian)	X'xxxx1014'
Power on Reset Value (Little Endian)	X'1410xxxx



Bit(s)	Name	Description
31	Reserved	Reserved
31-16	Subsystem ID	Generally will be set by crisco code. If not set by CRISCO, this value defaults to zero. See Entity 15: on page 428 for details.
15-0	Subsystem Vendor ID	Default value is the IBM vendor ID.

1.10: PCINT ROM Base Address

This register specifies the base address of where in PCI memory space the IBM3206K0424 ROM will be mapped. When written with ones and read back, the least significant bits read back as '0' will indicate the amount of memory space required for this device to operate. For example, when a value of 'FFFFFFFF' is written, a value read of 'FFFFFF00' indicates that 256 bytes of address space is required. See bit definitions.

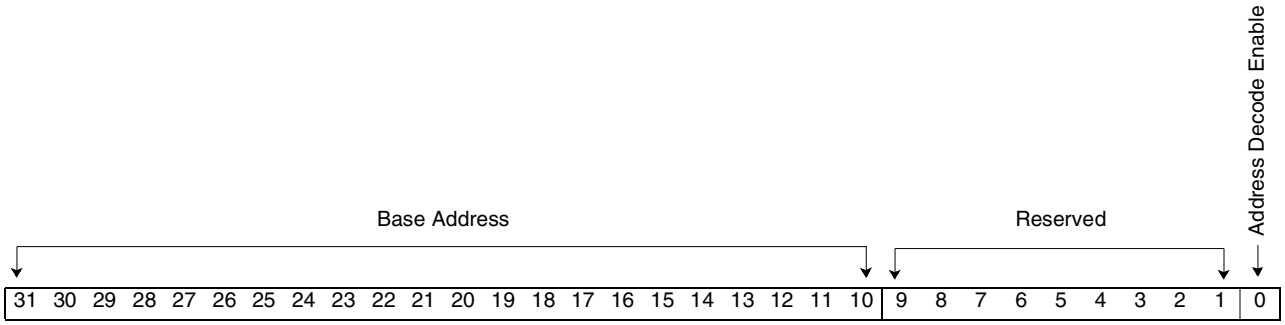
Length 32 bits

Type Read/Write

Address XXXX 0030

Restrictions Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as Big Endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.

Power on Reset value X'00000000'

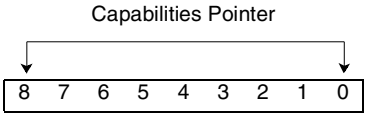


Bit(s)	PCI Spec	Name	Description
31-10	31-11	Base Address	This register is used to hold the address where the target device will decode for expansion ROM. The size is fixed at 2K of addressing, naturally aligned.
9-1	10-1	Reserved	Reserved and set to '0'.
0	0	Address Decode Enable	This bit set to '1' will enable accesses to expansion ROM only if Memory Space Enable bit (bit 1 in PCINT Configuration Word 1) is also set.

1.11: Capabilities Pointer

This register contains the Capabilities Pointer. See bit definitions.

Length	8 bits
Type	Read only
Address	XXXX 0034
Restrictions	Cannot be written by Crisco or when the PCI configuration space override write bit is on.
Power on Reset value (Big Endian)	X'000000C0'
Power on Reset value (Little Endian)	X'C0000000'

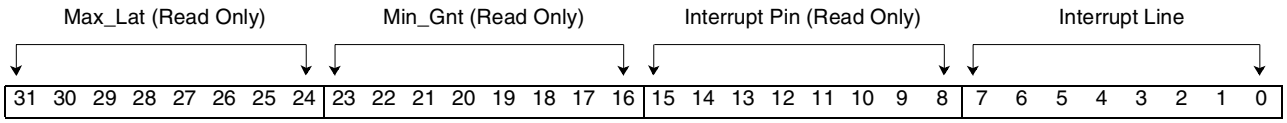


Bit(s)	Name	Description
7-0	Capabilities Pointer	Used to point to a linked list of new capabilities implemented by this device. The register is valid only if bit 4 of PCINT Config Word 1 is set. Bits 0 and 1 are always '0'.

1.12: PCINT Config Word 15

This register is used to communicate interrupt line routing information, tell which interrupt pin this device uses, and specify the desired setting for Latency Timer values. See bit definitions.

Length	32 bits
Type	Read/Write
Address	XXXX 003C
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as Big Endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset value (Big Endian)	X'00010100'
Power on Reset value (Little Endian)	X'00010100'



Bit(s)	PCI Spec	Name	Description
31-24	7-0	Max_Lat (Read Only).	This value specifies a period of time in units of 1/4 microsecond. Max_Lat is used for specifying how often this device needs to gain access to the PCI bus.
23-16	7-0	Min_Gnt (Read Only).	This value specifies a period of time in units of 1/4 microsecond. Min_Gnt is used for specifying how long a burst period this device needs, assuming a 33MHz clock rate.
15-8	7-0	Interrupt Pin (Read Only).	This device used $\overline{\text{INTA}}$ for its PCI bus interrupt. Value of this field is '01'.
7-0	7-0	Interrupt Line.	Software will write the routing information into this register as it initializes and configures the system.

1.13: PCINT Endian Control Register

This register allows control and status to the big/little endian address selection. It controls the byte order across the PCI bus. See bit definitions.

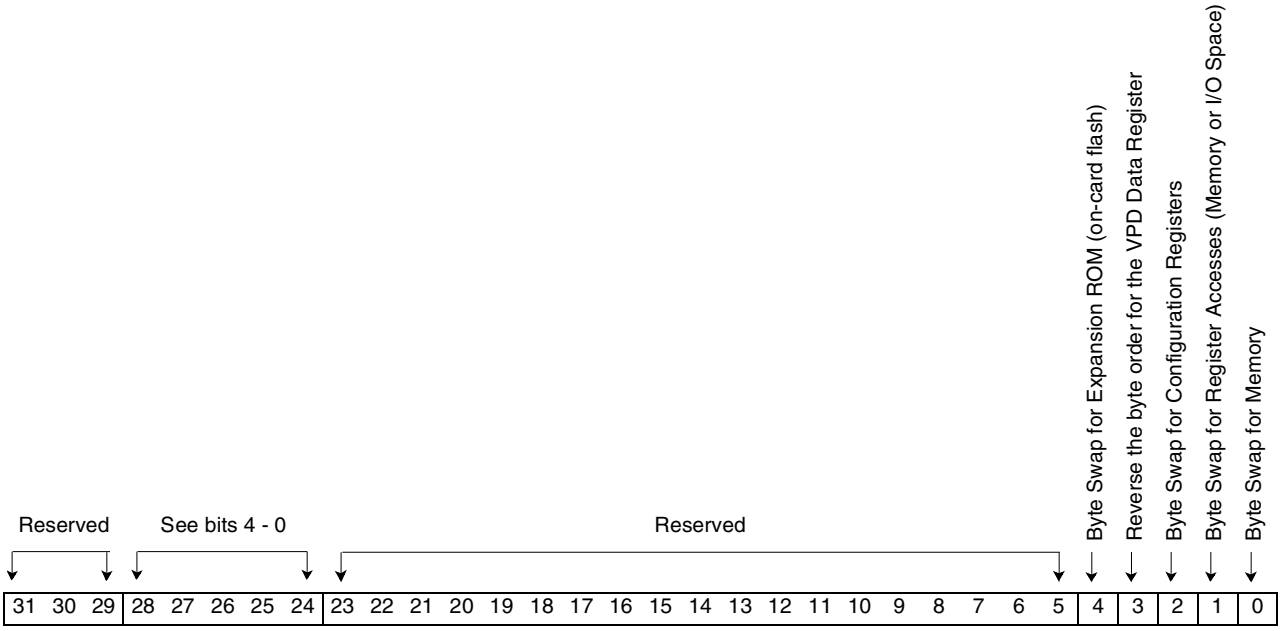
Length 32 bits

Type Read/Write

Address XXXX 0058

Restrictions Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle.

Power on Reset value X'00000000'

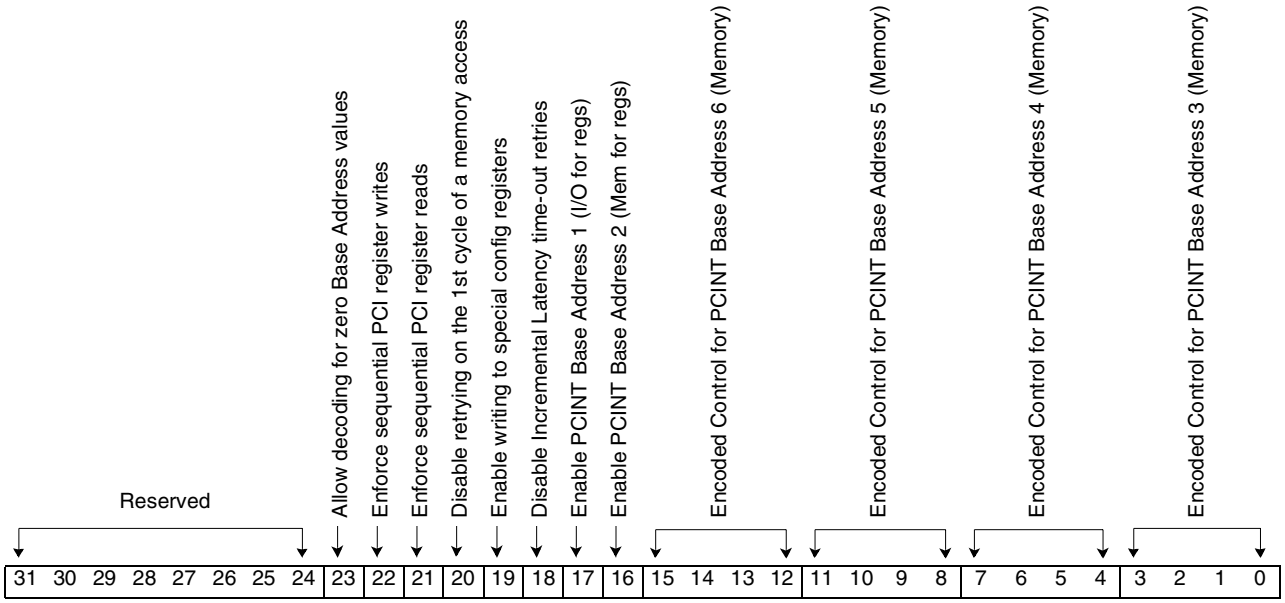


Bit(s)	Name	Description
31-29	Reserved	Reserved.
28-24	Same as the definitions for bits 4-0	
23-5	Reserved	Reserved.
4	Byte Swap For Expansion ROM (on-card flash)	When this bit is set to '1', the bytes of an internal Expansion ROM access (big endian view) will be swapped to and from the PCI interface.
3	Reverse the byte order for the VPD Data Register	When this bit is set to '1', the bytes of the Vital Product Data Interface - Word 2 register access will be swapped in reverse order to which bits 2 or 1 are set.
2	Byte Swap for Configuration Registers	When this bit is set to '1', the bytes of an internal Configuration register access (big endian view) will be swapped to and from the PCI interface.
1	Byte Swap for Register Access (Memory or I/O space)	When this bit is set to '1', the bytes of an internal register access (big endian view) will be swapped to and from the PCI interface.
0	Byte Sway for Memory	When this bit is set to '1', the bytes of an internal Packet Memory access (big endian view) will be swapped to which bits 2 or 1 are set.

1.14: PCINT Base Address Control Register

This register controls all the base address registers that map to memory. See bit definitions.

Length	32 bits
Type	Read/Write
Address	XXXX 005C
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset value (Big Endian)	X'0001000F'
Power on Reset value (Little Endian)	X'0F001100'



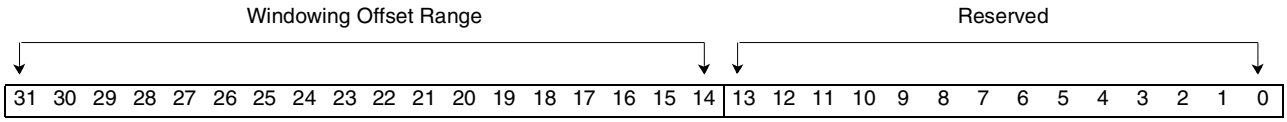
Bit(s)	Function	Description
31-24	Reserved	Reserved
23	Allow decoding for zero Base Address values	Setting this bit to '1' will enables decoding of a BAR address that is set to '0'. Normally, the PCI specification does not allow for a zero address to be a valid decode.
22	Enforce sequential PCI register writes	Setting this bit to '1' ensures that PCI register writes will occur in sequential order of prior memory accesses or register reads. The cost for doing this is possible extra retry cycles for accesses not dependent on other posted accesses to complete.
21	Enforce sequential PCI register reads	Setting this bit to '1' ensures that PCI register reads will occur in sequential order of prior memory accesses or register writes. The cost for doing this is possible extra retry cycles for accesses not dependent on other posted accesses to complete.

Bit(s)	Function	Description
20	Disable retrying on the 1st cycle of a memory access.	Setting this bit to ‘1’ disables the retrying of a memory access to IBM3206K0424. This causes a PCI spec violation, but not a data integrity problem. It solves the rare problem in which two masters are accessing Control Memory at the same time and retries happen to both endlessly.
19	Enable writing to special config registers	Setting this bit to ’1’ enables writing to certain registers that are normally read-only. An example of this is the vendor and function ID register (PCINT Configuration Word 0).
18	Disable Incremental Latency time-out retries	Setting this bit to ’1’ disables PCI retries due to cycles taking more than eight cycles on burst accesses after the first access.
17	Enable PCINT Base Address 1 (I/O for regs)	Setting this bit to ’1’ enables PCINT Base Address 1 (I/O for registers). This does the same function as bit 0 in the PCINT Configuration Word 1 register, but also makes the PCINT Base Address 1 (I/O for regs) read back ’0’s even when written to with values. It guards against anything that BIOS code may do to PCINT Configuration Word 1 register bit 0 if I/O accesses are not desired.
16	Enable PCINT Base Address 2 (Mem for regs)	Setting this bit to ’1’ enables PCINT Base Address 2 (Mem for regs) so IBM3206K0424 registers can be accessed by PCI memory cycles.
15-12	Encoded Control for PCINT Base Address 6 (Memory)	Same as bits 3-0.
11-8	Encoded Control for PCINT Base Address 5 (Memory)	
7-4	Encoded Control for PCINT Base Address 4 (Memory)	
3-0	Encoded Control for PCINT Base Address 3 (Memory).	Encoding of bits: X'0': Disable this Base Address. X'1': Configured to respond to a 2 GB address size. X'2': Configured to respond to a 1 GB address size. X'3': Configured to respond to a 512 MB address size. X'4': Configured to respond to a 256 MB address size. X'5': Configured to respond to a 128 MB address size. X'6': Configured to respond to a 64 MB address size. X'7': Configured to respond to a 32 MB address size. X'8': Configured to respond to a 16 MB address size. X'9': Configured to respond to a 8 MB address size. X'A': Configured to respond to a 4 MB address size. X'B': Configured to respond to a 2 MB address size. X'C': Configured to respond to a 1 MB address size. X'D': Configured to respond to a 64K address size, and enables internal windowing of memory. X'E': Configured to respond to a 32K address size, and enables internal windowing of memory. X'F': Configured to respond to a 16K address size, and enables internal window

1.15: PCINT Window Offsets for Base Addresses 3-6

These registers specify the amount of memory space required for this device to operate. See bit definitions.

Length	32 bits	
Type	Read/Write	
Address	Reg 3	XXXX 0060
	Reg 4	XXXX 0064
	Reg 5	XXXX 0068
	Reg 6	XXXX 006C
Power on Value	X'00000000'	
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.	

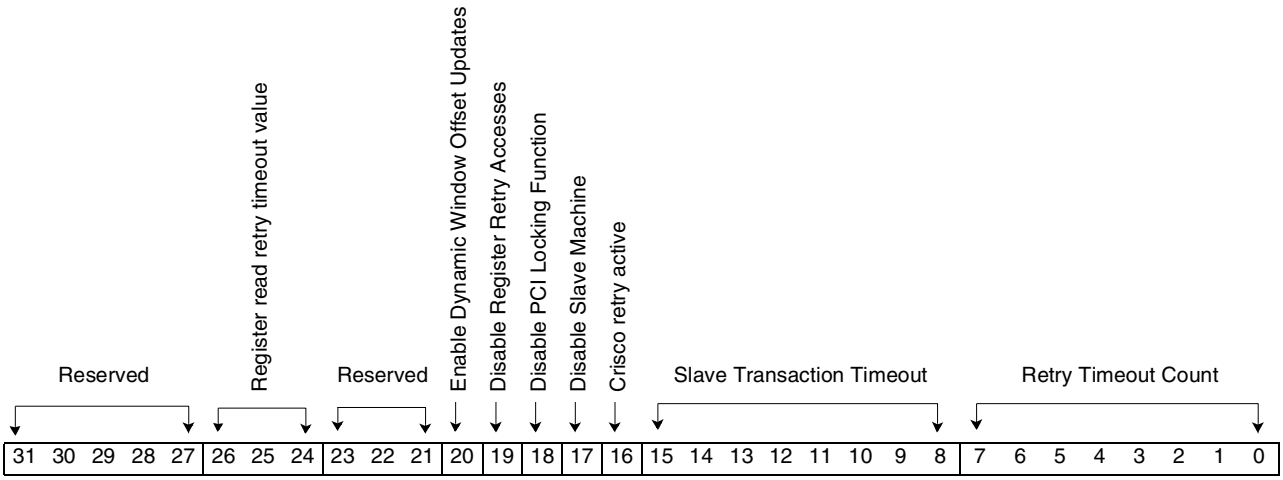


Bit(s)	Function	Description
31-14	Windowing Offset Range.	This register is used to hold the address offset, which is added to the PCI address (when windowing is enabled) to form the internal memory address. Bits 15 and 14 may or may not be used, depending on how bits are set in the PCINT Base Address Control Register. When bit 20 of PCINT Count Timeout Register is set, Window Offset register three can be updated with the address returned from a good get buffer from POOLS. This will save a write from code to this register. When bit 20 of PCINT Count Timeout Register is set, Window Offset register four can be updated with the address returned from a dequeue from the receive queue. This will save a write from code to this register.
13-0	Reserved.	Reserved

1.16: PCINT Count Timeout Register

This register holds the count limit of PCI slave retry cycles. See bit definitions.

Length	32 bits
Type	Read/Write
Address	XXXX 0070
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as Big Endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset Value (Big Endian)	X'0200FFFF'
Power on Reset Value (Little Endian)	X'FFFF0003'



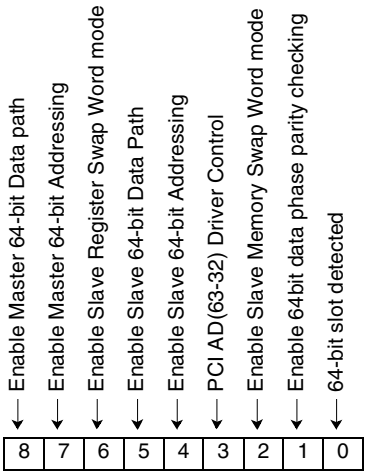
Bit(s)	Function	Description
31-27	Reserved	Reserved
26-24	Register read retry timeout value	The bits can be set to determine how many PCI cycles a register access will wait for an internal cycle to complete for a read access. It can be programmed to wait for up to seven cycles. A value of '0' will not timeout this access with a retry.
23-21	Reserved	Reserved
20	Enable Dynamic Window Offset Updates	Setting this bit to '1' enables the values of PCINT Window Offsets for Base Addresses 3-6 so that it updated with a good get primitive or certain receive queue dequeues.
19	Disable Register Retry Accesses	Setting this bit to '1' disables PCI retry signaling during a register or primitive access.
18	Disable PCI Locking Function	Setting this bit to '1' disables this PCI locking function when set to '1'
17	Disable Slave Machine	This bit is for Crisco code use. When set to '1', it disables all responses to the PCI bus in slave mode. In general, never turn this bit on. Bit 19 of the PCINT Base Address Control Register must be set before this bit can be changed.
16	Reserved	Reserved

Bit(s)	Function	Description
15-8	Slave Transaction Timeout	These bits hold a value that is used to count the number of PCI clocks times 256 when a PCI slave cycle is in progress. If the count is reached, due to some internal chip hang condition, a target abort is issued. A value of ’0’ disables target aborts from this function.
7-0	Retry Timeout Count	These bits hold a value that is used to count the number of PCI retries. The maximum count is 256 times 16 retries. If the count is reached, a target abort is issued. A value of ‘0’ will disable target aborts from this function.

1.17: PCINT 64-bit Control Register

This register contains miscellaneous control bits.

Length	9 bits
Type	Read/Write
Address	XXXX 0078
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset Value (Big Endian)	X'00000XXX', where the 'X' values depend on whether bit 0 is set and values of the enable bits in PCINT 64-bit Enable Register.
Power on Reset Value (Little Endian)	X'XX0X0000', where the 'X' values depend on whether bit 0 is set and values of the enable bits in PCINT 64-bit Enable Register.



Bit(s)	Name	Description
8	Enable Master 64-bit Data path	This bit set to '1' will enable master 64-bit data path for dma transfers.
7	Enable Master 64-bit Addressing	This bit set to '1' will enable master 64-bit addressing.
6	Enable Slave Register Swap Word mode	This bit set to '1' will enable word swapping of the each of the four groups of data bytes in an eight-byte register transfer. 2
5	Enable Slave 64-bit Data Path	This bit set to '1' will enable the slave 64-bit data path for registers and Packet Memory.
4	Enable Slave 64-bit Addressing	This bit set to '1' will enable slave 64-bit addressing, making base addresses 1 and 2 available for register accesses (memory cycles only) and base addresses 3 and 4 available for Packet Memory.
3	PCI AD(63-32) Driver Control	This bit set to '1' will cause the AD(63-32) PCI drivers to force to tri-state unless a 64-bit access is occurring. Otherwise, when set to '0', the drivers will always drive active.
2	Enable Slave Memory Swap Word mode	This bit set to '1' will enable word swapping of the each of the four groups of data bytes in an eight-byte slave memory transfer through BCACH. 2

Bit(s)	Name	Description
1	Enable 64bit data phase parity checking	This bit set to '1' will enable the data phase parity checking on bits 32 to 63 of the AD PCI bus.
0	64-bit slot detected	This bit will set when the REQ64# I/O pin was low bus when RST# went inactive. This bit is a read-only status bit. This bit on, combined with the status of the corresponding bit in the PCINT 64-bit Enable Register will determine the value of other bits in this register.

1.18: PCINT 64-bit Enable Register

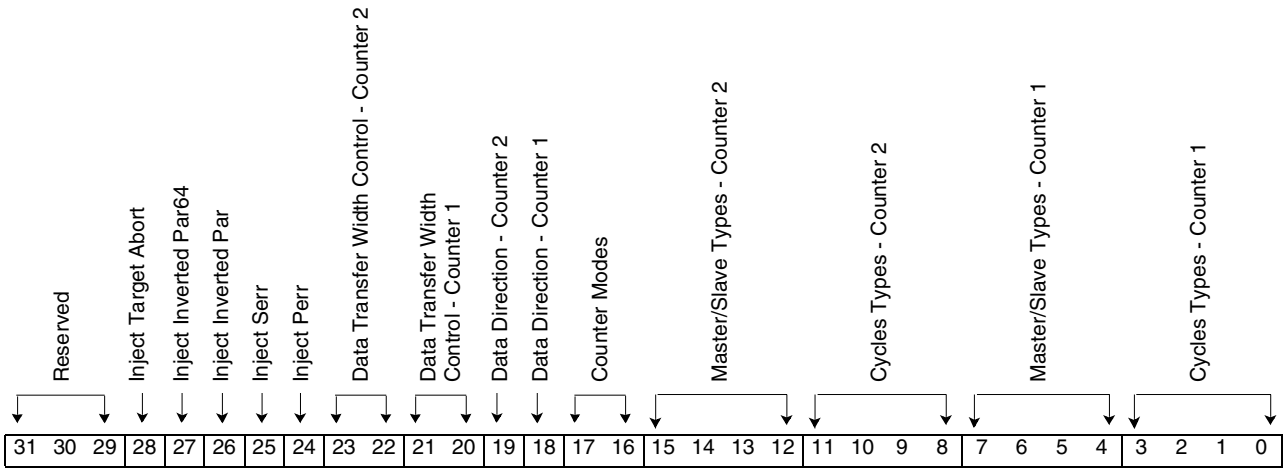
See the *PCINT 64-bit Control Register* on page 116 for the bitwise description that the corresponding bit in this register will enable (a value of '1' means enabled). Any bit in this register ANDed with bit 0 of PCINT 64-bit Control Register will determine if the other bits in PCINT 64-bit Control Register are set.

Length	9 bits
Type	Read/Write
Address	XXXX 0088
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see PCINT Base Address Control Register), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset Value (Big Endian)	X'0000012A'
Power on Reset Value (Little Endian)	X'2A010000

1.19: PCINT Perf Counters Control Register

This register contains control bits for the PCINT performance Counter 1 and PCINT Performance Counter 2.

Length	32 bits
Type	Read/Write
Address	XXXX 007C
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see the *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset Value (Big Endian)	X'00000000'
Power on Reset Value (Little Endian)	X'00000000'



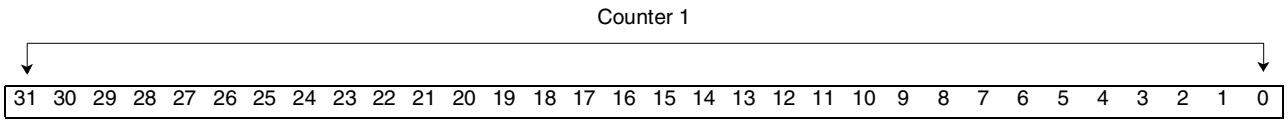
Bit(s)	Function	Description
28	Inject Target Abort	This bit on will make the target respond with a target abort sequence, provided all the other conditions are set correctly in this register.
27	Inject Inverted Par64	This bit on will invert the value of PCI PAR64, provided all the other conditions are set correctly in this register.
26	Inject Inverted Par	This bit on will invert the value of PCI PAR, provided all the other conditions are set correctly in this register.
25	Inject Serr	This bit on will flow a PCI Serr#, provided all the other conditions are set correctly in this register. Bit 8 of the PCINT Config Word 1 does not need to be set to cause this condition.
24	Inject Perr	This bit on will flow a PCI Perr#, provided all the other conditions are set correctly in this register. Bit 6 of the PCINT Config Word 1 does not need to be set to cause this condition.
23-22	Data Transfer Width Control - Counter 2	These bits will determine which kind of cycle to count based on the transfer size for counter 2. The defines are the same as bits 21-20.

Bit(s)	Function	Description
21-20	Data Transfer Width Control - Counter 1	These bits will determine which kind of cycle to count based on the transfer size for counter 1. X'0': All transfers X'1': 32 bit transfers only X'2': 64 bit transfers only X'3': Enable Data Direction (bits 18 or 19)
19	Data Direction - Counter 2	These bits will determine which kind of cycle to count based on the data direction - in or out of the IBM3206K0424 for counter 2.
18	Data Direction - Counter 1	These bits will determine which kind of cycle to count based on the data direction - in or out of the IBM3206K0424 for counter 1. X'0': Data In X'1': Data Out
17-16	Counter Modes	These bits will determine which kind of mode both counters will operate in. X'0' Stop on overflow X'1' Interrupt on wrap X'2' Event on wrap X'3' Inject active errors on overflow
15-12	Master/Slave Types - Counter 2	These bits determine which kind of PCI cycle owners to be counted for counter 2. The definitions are the same as bits 7-4.
11-8	Cycles Types - Counter 2	These bits determine what kind of PCI events are to be counted for counter 2. The definitions are the same as bits 3-0.
7-4	Master/Slave Types - Counter 1	These bits determine which kind of PCI cycle owners to be counted for counter 1. X'0' All Devices on the PCI bus X'1' All Devices but IBM3206K0424 X'2' IBM3206K0424 only (master or slave) X'3' IBM3206K0424 master X'4' IBM3206K0424 slave (all types) X'5' IBM3206K0424 slave register accesses X'6' IBM3206K0424 slave memory accesses
3-0	Cycles Types - Counter 1	These bits will determine what kind of PCI events are to be counted for counter 1. X'0' Off X'1' All PCI clock cycles X'2' Active PCI bus cycles (frame + irdy + trdy) X'3' PCI Data Xfer Opportunities ((irdy + trdy) & devsel) X'4' PCI Data Xfers (irdy & trdy) X'5' PCI Retries (irdy & no trdy & devsel & stop) X'6' PCI Address Phase (frame & not frame delayed) X'7' PCI Disconnects (irdy & trdy & devsel & stop)

1.20: PCINT Perf Counter 1

This register contains PCI performance counter 1.

Length	32 bits
Type	Read/Write
Address	XXXX 0080
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset Value (Big Endian)	X'00000000'
Power on Reset Value (Little Endian)	X'00000000'

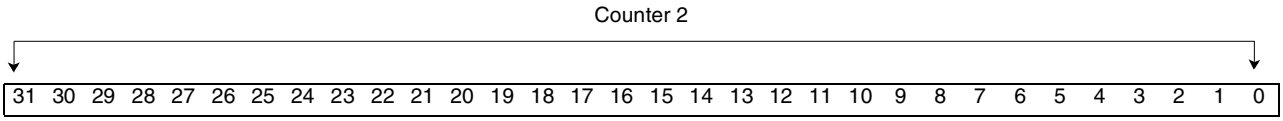


Bit(s)	Name	Description
31-0	Counter 1	See *PCINT Perf Counters Control Register* on page 119 for information on how this counter will increment.

1.21: PCINT Perf Counter 2

This register contains PCI performance counter 2.

Length	32 bits
Type	Read/Write
Address	XXXX 0084
Restrictions	Can be written or read during configuration cycle, memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset Value (Big Endian)	X'00000000'
Power on Reset Value (Little Endian)	X'00000000'

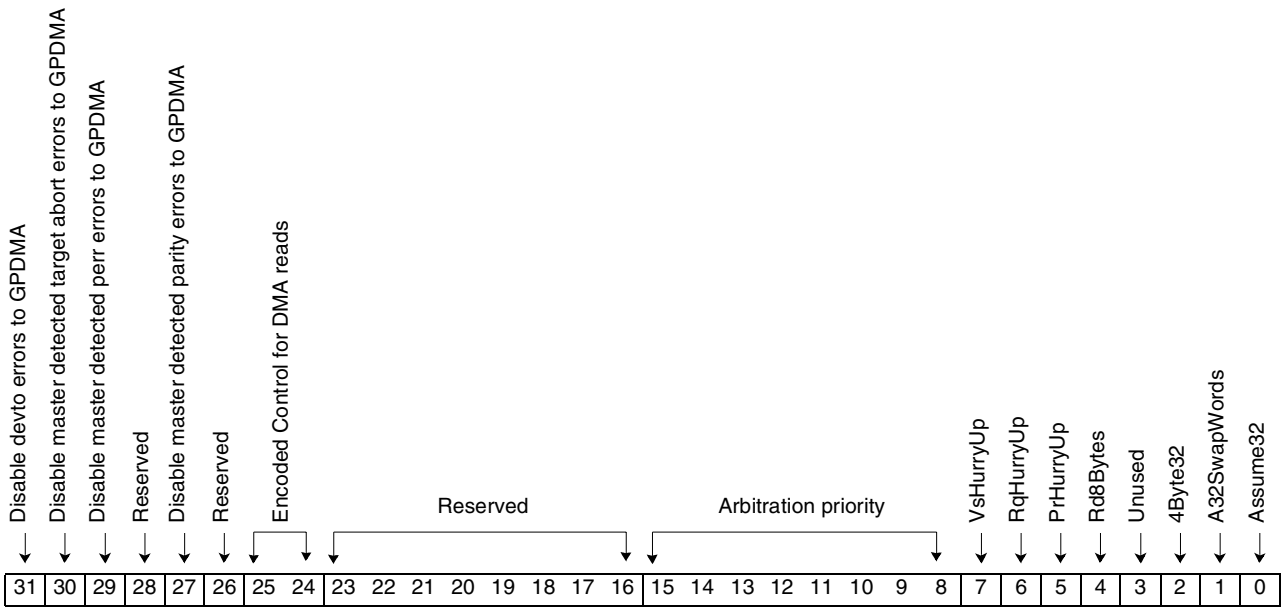


Bit(s)	Name	Description
31-0	Counter 2	See *PCINT Perf Counters Control Register* on page 119 for information on how this counter will increment.

1.22: PCI Master Options Control

This register contains the control register when the IBM3206K0424 is the PCI master.

Length	32 bits
Type	Read/Write
Address	XXXX 008C
Restrictions	None
Power on Reset Value (Big Endian)	X'00000000'
Power on Reset Value (Little Endian)	X'00000000'



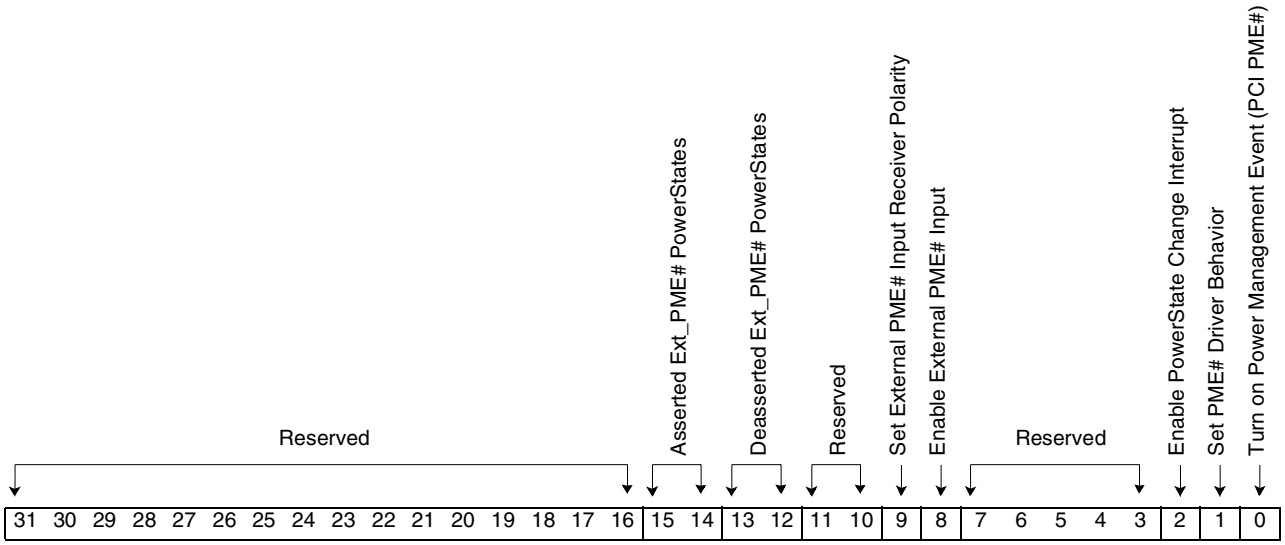
Bit(s)	Name	Description
31	Disable devto errors to GPDMA	Setting this bit to '1' will disable device timeout errors from stopping a GPDMA transfer.
30	Disable master detected target abort errors to GPDMA	Setting this bit to '1' will disable master detected target abort errors from stopping a GPDMA transfer.
29	Disable master detected perr errors to GPDMA	Setting this bit to '1' will disable master detected perr errors from stopping a GPDMA transfer.
28	Reserved	Reserved
27	Disable master detected parity errors to GPDMA	Setting this bit to '1' will disable master detected parity errors from stopping a GPDMA transfer
26	Reserved	Reserved

Bit(s)	Name	Description
25-24	Encoded Control for DMA reads	Encoding of bits: X'0': Let the IBM3206K0424 pick the best memory read command based on the cacheline size bits and the DMA count. X'1': Fix the read DMA command to Memory Read Multiple. X'2': Fix the read DMA command to Memory Read Line. X'3': Fix the read DMA command to Memory Read.
23-16	Reserved	Reserved
15-8	Arbitration priority	PCI master will cease using a default round-robin scheme for internal requestor arbitration if these bits are not all '0'. Bits 15-14 are the priority level for GPDMA. Bits 13-12 are the priority level for PCORE. Bits 11-10 are the priority level for RXQUE. Bits 9 - 8 are the priority level for INTST. Valid levels are 3,2,1, and 0. Only 4 levels must be used.
7	VsHurryUp	PCI master will inform GPDMA to HurryUp if VSTAT is waiting.
6	RqHurryUp	PCI master will inform GPDMA to HurryUp if RXQUE is waiting.
5	PrHurryUp	PCI master will inform GPDMA to HurryUp if PCORE is waiting.
4	Rd8Bytes	PCI master will force all byte enables active for reads.
3	Unused	PCI master does not use this bit.
2	4Byte32	PCI master will transfer a four-byte DMA as a 32-bit transfer.
1	A32SwapWords	PCI master will always swap words for any Assume32 transfer.
0	Assume32	PCI master will not request a 64-bit transfer.

1.23: Power Management Program Control

This register contains the control register for power management signalling.

Length	32 bits
Type	Read/Write
Address	XXXX 0090
Restrictions	Can be written or read during configuration cycle or memory cycle when enabled (see *PCINT Base Address Control Register on page 111*), or as an I/O cycle. This register is documented as big endian, but how data is presented on the PCI bus depends on how the controls are set in the PCINT Endian Control Register.
Power on Reset Value (Big Endian)	X'00000000'
Power on Reset Value (Little Endian)	X'00000000'



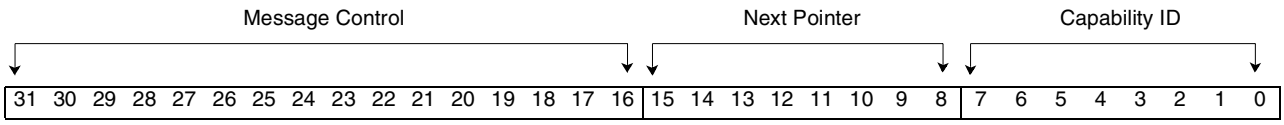
Bit(s)	Name	Description
31-16	Reserved	Reserved
15-14	Asserted Ext_PME# Power-States	These bits reflect what power state the Ext_PME# pin will indicate that it is in when the Ext_PME# is asserted.
13-12	Deasserted Ext_PME# Power-States	These bits reflect what power state the Ext_PME# pin will indicate that it is in when the Ext_PME# is de-asserted. Also, these are the default bits read back for the pmi2 power-states bits 1-0 when Ext_PME# is not enabled.
11-10	Reserved	Reserved
9	Set External PME# Input Receiver Polarity	Setting this bit to '1' will make the chip input called Ext_PME# to be used as a positive active signal. Otherwise it is negative active.
8	Enable External PME# Input	Setting this bit to '1' will enable the chip input called Ext_PME# to be used as a source to driver the PM# state.
7-3	Reserved	Reserved
2	Enable PowerState Change Interrupt	Setting this bit to '1' will enable a change of power states to cause an interrupt bit to turn on in INTST.

Bit(s)	Name	Description
1	Set PME# Driver Behavior	Setting this bit to ’1’ will make the PME# driver behave like a push-pull driver. Setting this bit to ’0’ will make the PME# driver behave like an open-drain.
0	Turn on Power Management Event (PCI PME#)	Setting this bit will assert the PCI bus signal PME#.

1.24: Message Signaled Interrupts-Word 1

This register contains the part of the Message Signaled Interrupts structure. See bit definitions.

Length	32 bits
Type	Read Only/Read/Write
Address	XXXX 00C0
Restrictions	Cannot be written unless by Crisco, or the PCI config space override write bit is on.
Power on Reset Value (Big Endian)	X'0082D005'
Power on Reset Value (Little Endian)	X'05D08200'

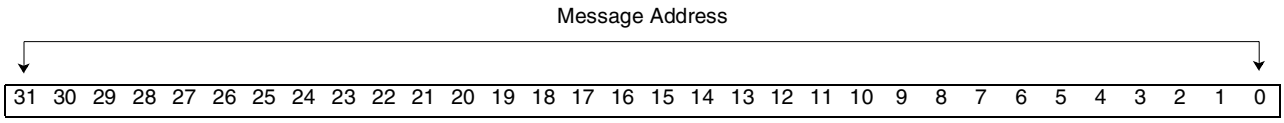


Bit(s)	Name	Description
31-16	Message Control	See PCI spec revision 2.2 for more details. Bits 31-24 are 0, and bit 23 is 1. Bits 22-20 are the Multiple Message Enable field, and bits 19-17 are the Multiple Message Capable field. Bit 16 is MSI Enable. Only bits 16, 20, 21, and 22 are writable.
15-8	Next Pointer	Pointer to the next item in the capabilities list.
7-0	Capability ID	Set to 05h to identify this function as Message Signaled Interrupt capable.

1.25: Message Signaled Interrupts-Word 2

This register contains the part of the Message Signaled Interrupts structure. See bit definitions.

Length	32 bits
Type	Read/Write
Address	XXXX 00C4
Restrictions	
Power on Reset Value (Big Endian)	X'00000000'
Power on Reset Value (Little Endian)	X'00000000'

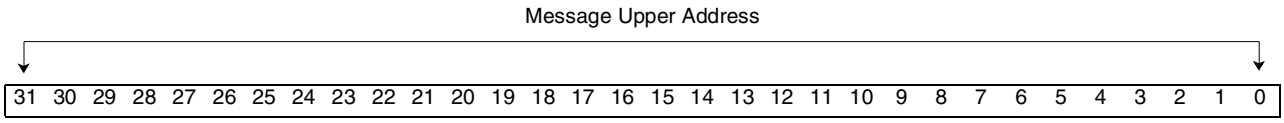


Bit(s)	Name	Description
31-0	Message Address	See PCI spec revision 2.2 for more details. Bits 31-24 are '0', and bit 23 is '1'. Bits 22-20 are the Multiple Message Enable field, and bits 19-17 are the Multiple Message Capable field. Bit 16 is MSI Enable. Only bits 16, 20, 21, and 22 are writable.

1.26: Message Signaled Interrupts-Word 3

This register contains the part of the Message Signaled Interrupts structure. See bit definitions.

Length	32 bits
Type	Read/Write
Address	XXXX 00C3
Restrictions	
Power on Reset Value (Big Endian)	X'00000000'
Power on Reset Value (Little Endian)	X'00000000'



Bit(s)	Name	Description
31-0	Message Upper Address	See PCI spec revision 2.2 for more details. Bits 31-0 hold the upper 32 bits of system address for the MSI memory write DMA transaction.

1.27: Message Signaled Interrupts-Word 4

This register contains the part of the Message Signaled Interrupts structure. See bit definitions.

Length 16 bits

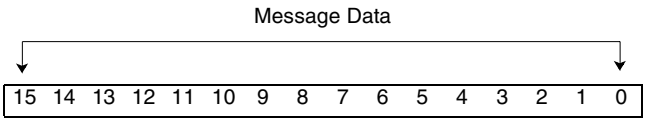
Type Read/Write

Address XXXX 00CC

Restrictions

Power on Reset Value (Big Endian) X'00000000'

Power on Reset Value (Little Endian) X'00000000'

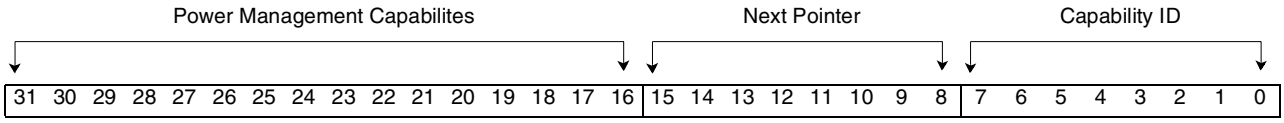


Bit(s)	Name	Description
15-0	Message Data	See PCI spec revision 2.2 for more details. Bits 15-0 hold the data for the MSI memory write DMA transaction.

1.28: Power Management Interface-Word 1

This register contains the part of the Power Management Interface structure. See bit definitions.

Length	32 bits
Type	Read Only
Address	XXXX 00D0
Restrictions	Cannot be written unless by Crisco, or the PCI config space override write bit is on.
Power on Reset Value (Big Endian)	X'00000000'
Power on Reset Value (Little Endian)	X'00000000'

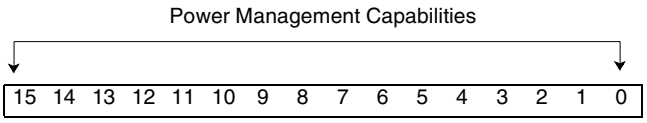


Bit(s)	Name	Description
31-16	Power Management Capabilities	See PCI Bus Power Management Interface Spec, Version 1.0 for more details. Bits 31-27 are for PME_Support. Bit 26 is for D2_Support. Bit 25 is for D1_Support. Bits 24-22 are reserved. Bit 21 is the Device Specific Initialization bit. Bit 20 is reserved. Bit 19 is for PME Clock. Bits 18-16 are version compliance level.
15-8	Next Pointer	Pointer to the next item in the capabilities list.
7-0	Capability ID	Set to 01h to identify this function as PCI Bus Power Management Interface.

1.29: Power Management Interface-Word 2

This register contains the part of the Power Management Interface structure. See bit definitions.

Length	16 bits
Type	Read Only/Read Clear/Read Write
Address	XXXX 00D4
Restrictions	
Power on Reset Value (Big Endian)	X'00000000'
Power on Reset Value (Little Endian)	X'00000000'

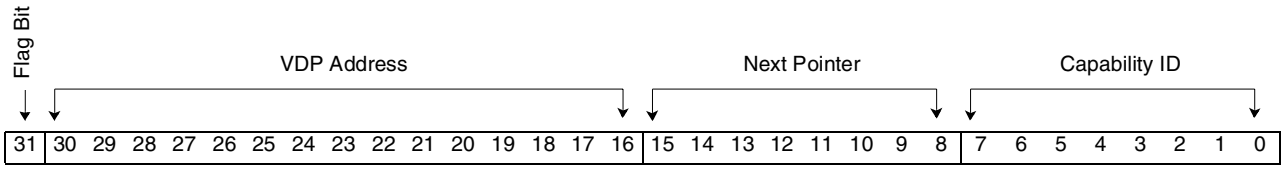


Bit(s)	Name	Description
15-0	Power Management Capabilities	See PCI Bus Power Management Interface Spec Version 1.0 for more details. Bit 15 is for PME_Status. Bit 14-13 are not used. Bits 12-9 are not used. Bit 8 is PME_En. Bits 7-2 are reserved. Bit 1-0 are the PowerState.

1.30: Vital Product Data Interface-Word 1

This register contains the part of the Vital Product Data Interface structure. See bit definitions.

Length	32 bits
Type	Read Only/Read Write
Address	XXXX 00D8
Restrictions	Cannot be written unless by Crisco, or if the PCI config space override write bit is on.
Power on Reset Value (Big Endian)	X'00000003'
Power on Reset Value (Little Endian)	X'00000000'



Bit(s)	Name	Description
31	Flag Bit	See PCI Spec Revision 2.2, Appendix 1 for more details. Read/Write Flag bit. For VPD reads, this bit is set to '0' and the VPD Address is set. When the hardware sets the bit to '1', valid VPD data can be read. For VPD writes, the VPD Data is written first. Then this bit is set to '1', along with the VPD Address. The write is active until the hardware resets this bit.
30-16	VDP Address	See PCI Spec Revision 2.2, Appendix 1 for more details. VPD Address.
15-8	Next Pointer	Pointer to the next item in the capabilities list.
7-0	Capability ID	Set to 03h to identify this function as Vital Product Data Interface.

1.31: Vital Product Data Interface-Word 2

This register contains the part of the Vital Product Data Interface structure. See bit definitions.

Length 32 bits

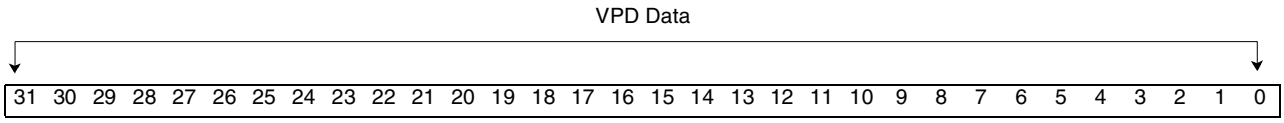
Type Read/Write

Address XXXX 00DC

Restrictions

Power on Reset Value (Big Endian) X'00000000'

Power on Reset Value (Little Endian) X'00000000'



Bit(s)	Name	Description
31-0	VPD Data	See PCI Spec Revision 2.2, Appendix 1 for more details. Four bytes of data are always read or written through this data field.

Entity 2: Interrupt and Status/Control (INTST)

This entity contains the masking registers that choose which interrupt/status source will be gated onto one of the two available interrupt I/O pins. A new delayed interrupt function has been added. This function allows IBM3206K0424 status registers to be read and placed in system memory before the interrupt signal is raised. For details, see *DMA QUEUES (DMAQS)* on page 154.

A bus timer function is provided in this entity that times a single bus access to make sure that the cycle is terminated before the system timer times out. This allows the user code an opportunity to recover from the error as opposed to the subsystem common code.

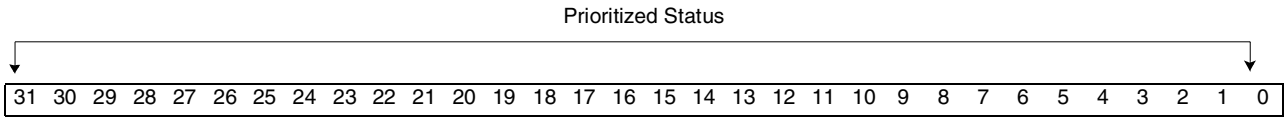
Below is a summary of this entity's functions:

- Interrupt Prioritized Status Registers
- Interrupt Source Register
- Interrupt Enable Registers
- Bus timer function
- Control Processor error register with enable register

2.1: INTST Interrupt 1 Prioritized Status

Used to help quickly parse which interrupting entity of the IBM3206K0424 is active.

Length	32 bits
Type	Read Only
Address	XXXX 0400
Restrictions	None
Power on Reset value	X'00000000'



Bit(s)	Function	Description
31-0	Prioritized Status	Reading this register will give a prioritized value of the bits in the INTST Interrupt Source and INTST Enable for Interrupt 1 (MINTA) registers ANDed together, returning a value that will be a hex number equal to bit number n + 1. For example, if bit 31 is on, X'20' will be read back.

2.2: INTST Interrupt 2 Prioritized Status

Used to help quickly parse which interrupting entity of the IBM3206K0424 is active.

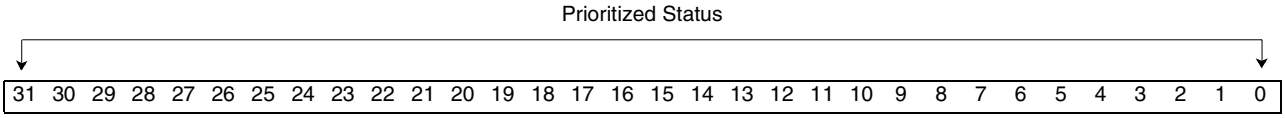
Length 32 bits

Type Read Only

Address XXXX 0404

Restrictions None

Power on Reset value X'00000000'



Bit(s)	Function	Description
31-0	Prioritized Status	Reading this register will give a prioritized value of the bits in the INTST Interrupt Source and INTST Enable for Interrupt 1 (MINT2) registers ANDed together, returning a value that will be a hex number equal to bit number n + 1. For example, if bit 31 is on, X'20' will be read back.

2.3: INTST Control Register

This register is used to control various IBM3206K0424 functions. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

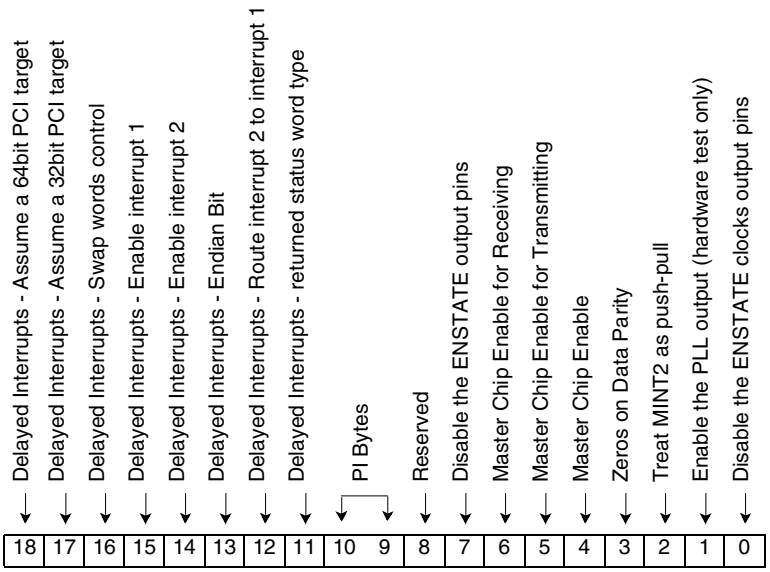
Length 18 bits

Type Clear/Set

Address XXXX 0408 and 0C

Restrictions None

Power on Reset value X'0010200'



Bit(s)	Name	Description
18	Delayed Interrupts - Assume a 64bit PCI target	This bit set will help the mastering logic determine how to best move data to a 64-bit PCI target. This bit is set when software has system knowledge of its targets.
17	Delayed Interrupts - Assume a 32bit PCI target	This bit set will help the mastering logic determine how to best move data to a 32-bit PCI target. This bit is set when software has system knowledge of its targets.
16	Delayed Interrupts - Swap words control	This bit determines the word order of the status word dma transfer for delayed ints. The default value of '1' is to swap the words. A value of '0' will not swap them.
15	Delayed Interrupts - Enable interrupt 1	When set, the delayed int mechanism for int 1 is enabled.
14	Delayed Interrupts - Enable interrupt 2	When set, the delayed int mechanism for int 2 is enabled.
13	Delayed Interrupts - Endian Bit	This bit determines the endian of the status word DMA transfer for delayed ints. When this bit is set, the endian is little. The default of '0' is big endian.
12	Delayed Interrupts - Route interrupt 2 to interrupt 1	When set, the int 2 signal is routed and raised as int 1. This bit allows both sets of int masks in intst to be used, while still using only a single hardware int. When set, both delayed int's should be enabled if they are being used.
11	Delayed Interrupts - returned status word type	When this bit is set, the INTST Interrupt Source word will be anded with the corresponding enable register. Otherwise, the INTST Interrupt Source register alone will be returned.

Bit(s)	Name	Description
10-9	PI Bytes	These bits are encoded to tell how many bytes long the AAL 5 CPI field is. The following are the encodings: ’00’ CPI field is zero bytes long. In this case, the two bytes containing the CPI field and the AAL5 user-to-user byte are copied into the packet header. See the definition of the packet header for the locations. ’01’ CPI field is one byte long and is always ’0’. In this case, the one byte AAL5 user-to-user byte is copied into the packet header. '10' CPI field is two bytes long and is always ’0’. '11' Treated the same as B'00'
8	Reserved	Reserved
7	Disable the ENSTATE output pins	When this bit is set to '0', the chip i/o ENSTATES will be driven with the output of the internally muxed debug states. When set to '1', these outputs will be quiet.
6	Master Chip Enable for Receiving	When this bit is set to '1', various state machines in the receive part of the chip will be enabled.
5	Master Chip Enable for Transmitting	When this bit is set to '1', various state machines in the transmit part of the chip will be enabled.
4	Master Chip Enable	When this bit is set to '1', various state machines in the chip will be enabled. This must be set to '1' transmit or receive anything.
3	Zeros on Data Parity	When this bit is set to '1', zeros will be forced on the data bus parity line(s) during a slave read data phase or a master address phase or a master write data phase.
2	Treat MINT2 as push-pull	When this bit is set to '1', the chip I/O MINT2 will be driven active high as well as low, like a push-pull driver. This is for use as a specific sideband application, not as a general shared open-drain interrupt line.
1	Enable the PLL output (hardware test only)	When this bit is set to '1', the chip i/o PPLLOUT will be driven with the output of the internal PLL. When set to '0', this output will be quiet.
0	Disable the ENSTATE clocks output pins	When this bit is set to '0', the chip i/o PINTCLK and PDBLCLK will be driven with the output of the internal clock tree. When set to '1', these outputs will be quiet.

2.4: INTST Interrupt Source

This register indicates the source(s) of the interrupt(s) pending. It can also be used as a status register when the bits are enabled. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. Note that bits in this register always reflect the state of the source register bit: Writing a value will have no effect. Reserved bits will not take on the written value. The delay of running through a latch has been removed. For the delayed interrupts feature, writing this register at the end of an interrupt handling routine will guarantee that interrupt1 and interrupt2 (if enabled) will pulse off, allowing the logic to get ready for the next interrupt DMA.

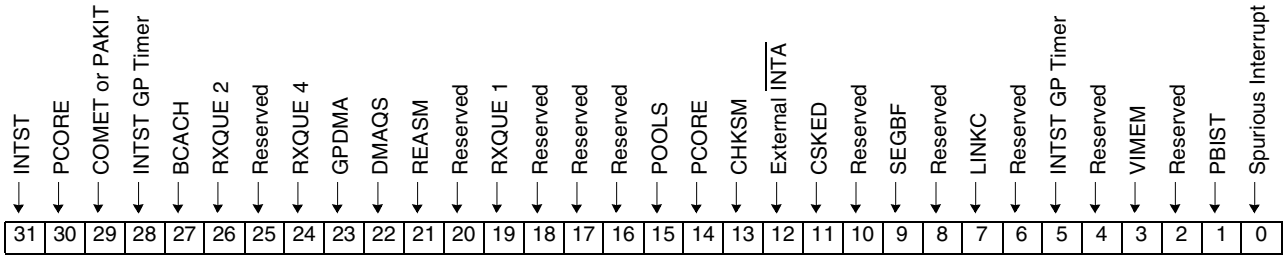
Length 32 bits

Type Read Only

Address XXXX 0410 and 14

Restrictions None

Power on Reset value X'00000000'



Bit(s)	Name	Description
31	INTST	A Control Processor related condition has occurred. A read of the INTST CPB Status and INTST CPB Status Enable must be done for more information. See *INTST CPB Status on page 142* and *INTST CPB Status Enable on page 144*.
30	PCORE	The PCORE entity has hardware interrupts that need handling.
29	COMET or PAKIT	The COMET or PAKIT entities have interrupts that need handling.
28	INTST GP Timer	The INTST General Purpose Timer Counter has reach the INTST General Purpose Timer Compare value and caused an interrupt.
27	BCACH	The BCACH entity has interrupts that need handling.
26	RXQUE 2	The RXQUE entity has interrupts that need handling.
25	Reserved	Reserved
24	RXQUE 4	The RXQUE entity has interrupts that need handling.
23	GPDMA	The GPDMA entity has interrupts that need handling.
22	DMAQS	The DMAQS entity has interrupts that need handling.
21	REASM	The REASM entity has interrupts that need handling.
20	Reserved	Reserved
19	RXQUE 1	The RXQUE entity has interrupts that need handling.
18	Reserved	Reserved
17	Reserved	Reserved

Bit(s)	Name	Description
16	Reserved.	Reserved.
15	POOLS	The POOLS entity has interrupts that need handling.
14	PCORE	The PCORE entity has User Defined interrupts that need handling.
13	CHKSM	The CHKSM entity has interrupts that need handling.
12	External $\overline{\text{INTA}}$	This bit will be set when the IBM3206K0424 detects that MINTA is low and, conditionally, when the same bit in INTST Enable for PCORE Normal Interrupt or INTST Enable for PCORE Critical Interrupt is set. This bit is for use by the PCORE entity, and it is recommended that interrupts directed out which drive output (MINTA) be disabled.
11	CSKED	The CSKED entity has interrupts that need handling.
10	Reserved	Reserved
9	SEGBF	The SEGBF entity has interrupts that need handling.
8	Reserved	Reserved
7	LINKC	The LINKC entity has interrupts that need handling.
6	Reserved	Reserved
5	INTST GP Timer	The INTST General Purpose Timer Counter has reached the INTST General Purpose Timer Compare value and caused an interrupt.
4	Reserved	Reserved
3	VIMEM	The VIMEM entity has interrupts that need handling.
2	Reserved	Reserved
1	PBIST	This bit is set when the PBIST entity did not indicate that it was done. It is also not clearable.
0	Spurious Interrupt	Under normal conditions, this bit should never be set. However, if one of the other bits in this register turns on, then off, a spurious interrupt condition will occur. The manual vector passed to the processor will point to this bit being on.

2.5: INTST Enable for Interrupt 1 (MINTA)

This register serves as an enable for interrupt 1. See the *INTST Interrupt Source* register on page 139 for the bitwise description that the corresponding bit in this register will enable. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length 32 bits

Type Clear/Set

Address XXXX 0418 and 1C

Restrictions None

Power on Reset value X'00000000'

2.6: INTST Enable for Interrupt 2 (MINT2)

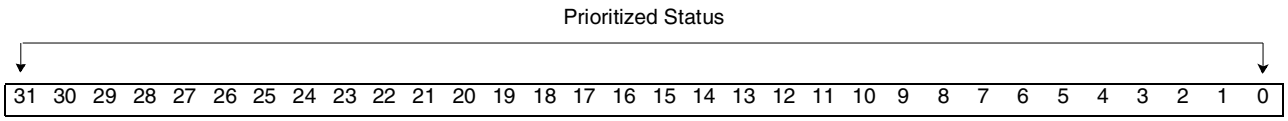
This register serves as a enable for interrupt 2. See the *INTST Interrupt Source* Register on page 139 for the bitwise description that the corresponding bit in this register will enable. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	32 bits
Type	Clear/Set
Address	XXXX 0420 and 24
Restrictions	None
Power on Reset value	X'00000000'

2.7: INTST Interrupt Source without Enables

This register is used to help quickly parse which interrupting bit of INTST Interrupt Source is active. It does not matter what state the Enable registers are set to because the value returned does not depend on them.

Length	32 bits
Type:	Read Only
Address	XXXX 0428
Restrictions	None
Power on Reset value	X'00000000'



Bit(s)	Function	Description
31-0	Prioritized Status	Reading this register gives a prioritized value of the bits in the INTST Interrupt Source, returning a value that is a hex number equal to bit number n + 1. For example, if bit 31 is on, X'20' will be read back.

2.8: INTST CPB Status

This register holds the status bits for errors on the Control Processor bus. These bits, when disabled, will set a bit in the INTST Interrupt Source register. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

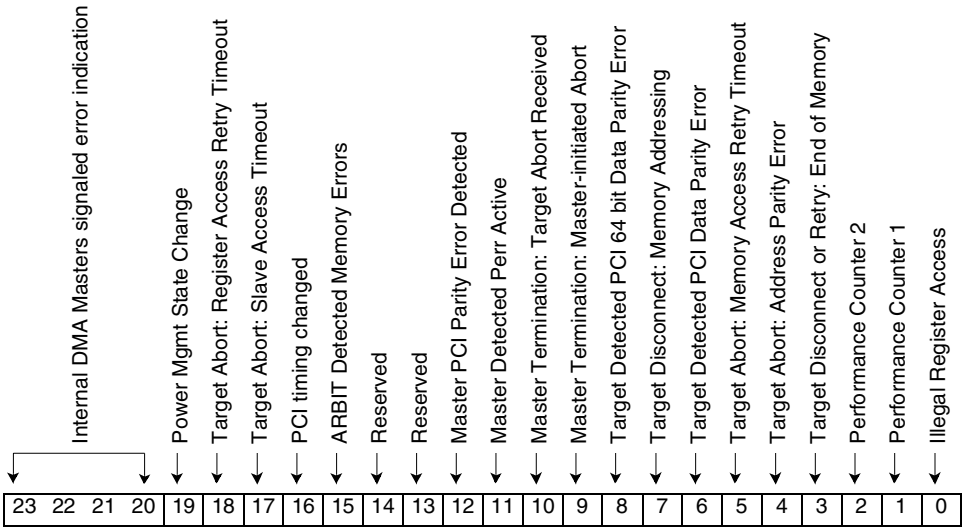
Length 24 bits

Type Clear/Set

Address XXXX 0430 and 34

Restrictions None

Power on Reset value X'00000'



Bit(s)	Function	Description
23-20	Internal DMA Masters signaled error indication	The PCINT entity will set these bits when it signals a DMA error indication to one of the internal requesting masters. Bit 23 is GPDMA, bit 22 is PCORE, bit 21 is RXQUE, and bit 20 is INTST.
19	Power Mgmt State Change	This bit is set when the conditions in PCINT are met to trigger a Power Management State change.
18	Target Abort: Register Access Retry Timeout	This bit is set when this slave does more retry cycles than the specified amount in the PCINT Count Timeout Register during a register access.
17	Target Abort: Slave Access Timeout	This bit is set when this slave does not access the IBM3206K0424 in the specified amount in the PCINT Count Timeout Register.
16	PCI timing changed	The PCI bus clock has changed range. The IBM3206K0424 should be reset and re-initialized.
15	ARBIT Detected Memory Errors	This bit is set when error conditions detected by ARBIT are enabled. Note: This bit is a reflection of the arbitrator status bits and does not need to be reset if the arbitrator condition has been reset.
14	Reserved	Reserved
13	Reserved	Reserved
12	Master PCI Parity Error Detected	This bit is set when a PCI bus data parity error is detected in master mode.

Bit(s)	Function	Description
11	Master Detected Perr Active	This bit is set when a target has driven $\overline{\text{PERR}}$.
10	Master Termination: Target Abort Received	This bit is set when in master mode and the transfer is aborted by the target.
9	Master Termination: Master-initiated Abort	This bit is set when in master mode and the transfer is aborted by this master.
8	Target Detected PCI 64 Bit Data Parity Error	This bit is set when a PCI data parity error is detected in 64 bit target mode (the upper DWORD has the data parity error).
7	Target Disconnect: Memory Addressing	This is set when a memory access is occurring and bits 0 and 1 of the address are not ’0’.
6	Target Detected PCI Data Parity Error	This bit is set when a PCI data parity error is detected in target mode.
5	Target Abort: Memory Access Retry Timeout	This bit is set when this slave does more retry cycles than the specified amount in the PCINT Count Timeout Register during a memory access.
4	Target Abort: Address Parity Error	This bit is set when an address parity error is detected.
3	Target Disconnect or Retry: Wrap of 2 GB internal address slave counter	This bit is set when the slave address counter is its maximum counter value will indicate a termination condition on the PCI bus. This is primarily a debug bit and can turn on during normal operation. It most likely will be useful when the IBM3206K0424 slave mode is configured in 64-bit addressing mode.
2	Performance Counter 2	The PCINT Performance Counter 2 has overflowed.
1	Performance Counter 1	The PCINT Performance Counter 1 has overflowed.
0	Illegal Register Access	This bit is set when an IBM3206K0424 register is being accessed by fewer than four bytes at a time. This is not true for configuration registers during a configuration cycle.

2.9: INTST CPB Status Enable

This register serves as an enable for the INTST CPB Status register. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See the *INTST CPB Status on page 142* for the bitwise description that corresponding bit in this register will enable. This enable will initialize to the disabled state.

Length	19 bits
Type	Clear/Set
Address	XXXX 0438 and 3C
Restrictions	None
Power on Reset value	X'00000'

2.10: INTST IBM3206K0424 Halt Enable

This register serves as an enable for the INTST CPB Status register and gates which errors will reset bit 4 (Master chip enable), bit 5 (Master chip enable for Transmitting), and bit 6 (Master chip enable for Receiving), all in the INTST Control Register register. This allows selected bits to disable the IBM3206K0424, especially in the case of severe hardware detected errors. See the *INTST CPB Status on page 142* for the bitwise description that corresponding bit in this register will enable. This enable will initialize to the disabled state. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	19 bits
Type	Clear/Set
Address	XXXX 0440 and 44
Restrictions	None
Power on Reset value	X'0009F71'

2.11: INTST CPB Capture Enable

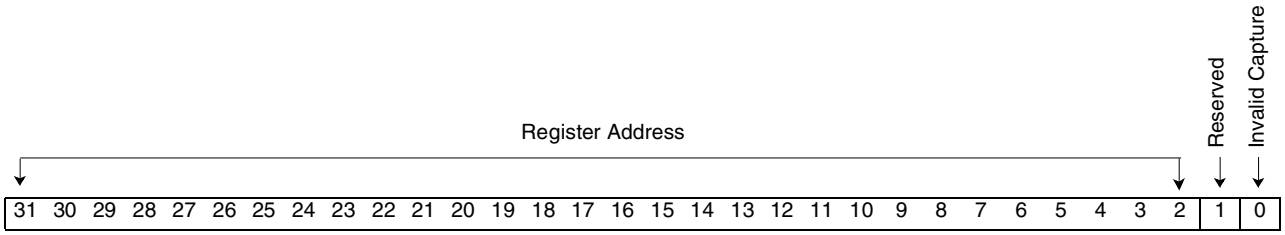
This register serves as an enable for the INTST CPB Status that will determine on which error type the INTST CPB Captured Address register will be updated. See the *INTST CPB Status on page 142* for the bitwise description that corresponding bit in this register will enable. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	19 bits
Type	Clear/Set
Address	XXXX 0450 and 54
Restrictions	None
Power on Reset value	X'00069F71'

2.12: INTST CPB Captured Address

This information can be used to attempt a retry in the exception handling microcode. This register holds the value of the IBM3206K0424 register address on the PCI during a bus error condition. This only latches values from sources that are enabled in the INTST CPB Capture Enable register.

Length 32 bits
Type Read/Write
Address XXXX 0458
Restrictions None
Power on Reset value X'00000001'



Bit(s)	Function	Description
31-2	Register Address	Captured IBM3206K0424 register address.
1	Reserved	Reserved.
0	Invalid Capture	When this bit is reset to '0', a valid capture has been made.

2.13: INTST General Purpose Timer Pre-scaler

This is the pre-scaler for the INTST General Purpose Timer Compare.This register will hold the value of the pre-scale count. The default value is 1 tick every 10.02uS, assuming a 33MHz or 66MHz PCI Bus clock, producing a 66MHz system clock (count is system clock). The pre-scale count value is n-1, where n is the desired increment count.

Owing to a physical design problem, the function of this register was lost. It should be set to a non-zero value, so that the INTST General Purpose Timer Counter can be used with a prescale of only the default clock (one tick every 30ns, assuming a 33-MHz system clock).

Length 32 bits
Type Read/Write
Address XXXX 0464
Restrictions None
Power on Reset value X'0000029B'

2.14: INTST General Purpose Timer Compare

This is the compare value for the general purpose timer. This register holds the value of the data that is compared to the count value in the INTST General Purpose Timer Counter, setting the INTST General Purpose Timer Status bits. See *INTST General Purpose Timer Mode Control* on page 148 for details on the operation of this register.

Length	32 bits
Type	Read/Write
Address	XXXX 0468
Restrictions	None
Power on Reset value	X'0800 0000'

2.15: INTST General Purpose Timer Counter

This is the general purpose timer counter.

This register holds the value of the counter. It always counts up. See *INTST General Purpose Timer Mode Control* on page 148 for details on operation of this register.

Length	32 bits
Type	Read/Write
Address	XXXX 046C
Restrictions	None
Power on Reset value	X'0000 0000'

2.16: INTST General Purpose Timer Status

This is the status of the general purpose timer counter. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

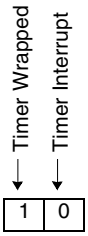
Length 2 bits

Type Clear/Set

Address XXXX 0470 and 74

Restrictions None

Power on Reset value X'0'



Bit(s)	Function	Description
1	Timer Wrapped	This bit is set when the INTST General Purpose Timer Counter wraps around to a '0' count value.
0	Timer Interrupt	See *INTST General Purpose Timer Mode Control* on page 148 for details on how this bit is set. For mode 0: This bit is set when the INTST General Purpose Timer Counter matches the value in the INTST General Purpose Timer Compare register. The comparing condition must be changed (write INTST General Purpose Timer Counter or INTST General Purpose Timer Compare) before resetting this bit, or the bit will set again.

2.17: INTST General Purpose Timer Mode Control

This register controls the operating modes of the general purpose timer counter. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

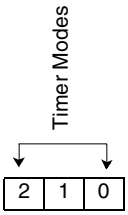
Length 3 bits

Type Clear/Set

Address XXXX 0478 and 7C

Restrictions None

Power on Reset value X'4'



Bit(s)	Description
2-0	These bits are encoded to provide eight different timer operation modes. Encoding is as follows: Mode 0 The INTST General Purpose Timer Counter is a free-running up-counter and sets bit 0 of INTST General Purpose Timer Status when equal to INTST General Purpose Timer Compare. Mode 1 The INTST General Purpose Timer Counter is a free-running up-counter and sets bit 0 of INTST General Purpose Timer Status when equal to INTST General Purpose Timer Compare. A write to INTST General Purpose Timer Compare will reset bit 0 of INTST General Purpose Timer Status. Mode 2 The INTST General Purpose Timer Counter is a free-running up-counter and sets bit 0 of INTST General Purpose Timer Status when equal to or greater than INTST General Purpose Timer Compare. A write to INTST General Purpose Timer Compare will reset bit 0 of INTST General Purpose Timer Status. Mode 3 The INTST General Purpose Timer Counter is a up-counter and sets bit 0 of INTST General Purpose Timer Status when equal to or greater than the INTST General Purpose Timer Compare. The INTST General Purpose Timer Counter is also reset when a comparison is made. A write to INTST General Purpose Timer Compare will reset bit 0 of INTST General Purpose Timer Status and INTST General Purpose Timer Counter. Mode 4 The INTST General Purpose Timer Counter is disabled and no status bits will be set. Modes 5-7 Reserved

2.18: INTST Enable for PCORE Normal Interrupt

This register serves as an enable for the PCORE normal interrupt input. See *INTST Interrupt Source on page 139* for the bitwise description that the corresponding bit in this register will enable. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	32 bits
Type	Clear/Set
Address	XXXX 0480 and 84
Restrictions	None
Power on Reset value	X'00000000'

2.19: INTST Enable for PCORE Critical Interrupt

This register serves as an enable for the PCORE critical interrupt input. See *INTST Interrupt Source on page 139* for the bitwise description that the corresponding bit in this register will enable. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	32 bits
Type	Clear/Set
Address	XXXX 0488 and 8C
Restrictions	None
Power on Reset value	X'00000000'

2.20: INTST Debug States Control

This register serves as the control for external debug states.

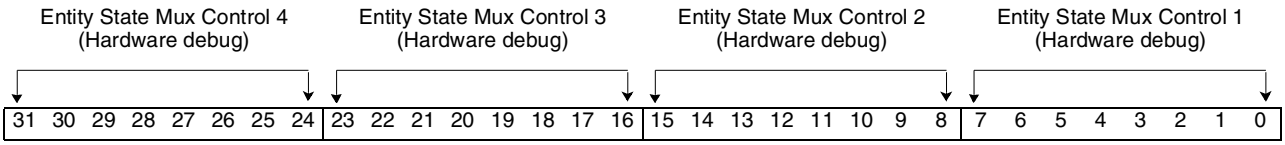
Length 32 bits

Type Read/Write

Address XXXX 0490

Restrictions None

Power on Reset value X'38030201'



Bit(s)	Function	Description
31-24	Entity State Mux Control 4 (Hardware debug)	Selection of these bits allows internal state machines, counters, etc. to show up on chip outputs enstate (63-48). Selection encoding is the same as multiplexer 1 control.
23-16	Entity State Mux Control 3 (Hardware debug)	Selection of these bits allows internal state machines, counters, etc. to show up on chip outputs enstate (7 -32). Selection encoding is the same as multiplexer 1 control.
15-8	Entity State Mux Control 2 (Hardware debug)	Selection of these bits allows internal state machines, counters, etc. to show up on chip outputs enstate (31-16).

Bit(s)	Function	Description
7-0	Entity State Mux Control 1 (Hardware debug)	Selection of these bits allows internal state machines, counters, etc. to show up on chip outputs enstate (15-0) which are multiplexed over ad64 (15-0). X'00' Disabled (no transition on outputs). X'01' Select CRSET 15-0 states. X'02' Select NPBUS 15-0 states. X'03' Select PCINT 15-0 states. X'04' Select PCINT 31-16 states. X'05' Select COMET 15-0 states. X'06' Select COMET 31-16 states. X'07' Select PAKIT 15-0 states. X'08' Select PAKIT 31-16 states. X'09' Select RXQUE 15-0 states. X'0A' Select RXQUE 31-16 states. X'0B' Select RAALL 15-0 states. X'0C' Select RAALL 31-16 states. X'0D' Select RAALL 47-32 states. X'0E' Select RAALL 63-48 states. X'0F' Select REASM 15-0 states. X'10' Select LINKC 15-0 states. X'11' Select SEGBF 15-0 states. X'12' Select SEGBF 31-16 states. X'13' Select SEGBF 47-32 states. X'14' Select SEGBF 63-48 states. X'15' Select CSKED 15-0 states. X'16' Select CHKSM 15-0 states. X'17' Select CHKSM 31-16 states. X'18' Select GPDMA 15-0 states. X'19' Select BCACH 15-0 states. X'1A' Select BCACH 31-16 states. X'1B' Select POOLS 15-0 states. X'1C' Select POOLS 31-16 states. X'1D' Select POOLS 47-32 states. X'1E' Select POOLS 63-48 states. X'1F' Select POOLS 79-64 states. X'20' Select POOLS 95-80 states. X'21' Select POOLS 111-96 states. X'22' Select POOLS 127-112 states. X'23' Select VIMEM 15-0 states. X'24' Select VIMEM 31-16 states. X'25' Select VIMEM 47-32 states. X'26' Select ARBIT 15-0 states. X'27' Select ARBIT 31-16 states. X'28' Select ARBIT 47-32 states. X'29' Select ARBIT 63-48 states. X'2A' Select PCORE 15-0 states. X'2B' Select PCORE 31-16 states. X'2C' Select PCORE 47-32 states. X'2D' Select PCORE 63-48 states. X'2E' Select PCORE 79-64 states. X'2F' Select PCORE 95-80 states. X'30' Select PCORE 111-96 states. X'31' Select PCORE 127-112 states. X'32' Select DMAQS 15-0 states. X'33' Select DMAQS 31-16 states. X'34' Select DMAQS 47-32 states. X'35' Select DMAQS 63-48 states. X'36' Select SCLCK 15-0 states. X'37' Select SCLCK 31-16 states. X'38' Select SCLCK 39-32 states. X'39'-X'FF'Reserved (do not toggle as well)

2.21: INTST Delayed Interrupts DMA System Address 1

This register serves as the Delayed Interrupts DMA System Address. This register holds the value of the system DMA address to which the delay interrupt status data will be moved.

Length	64 bits
Type	Read/Write
Address	XXXX 0498 and 9C
Restrictions	None
Power on Reset Value	X'00000000'

2.22: INTST Delayed Interrupts DMA System Address 2

This register serves as the Delayed Interrupts DMA System Address. This register holds the value of the system DMA address to which the delay interrupt status data will be moved.

Length	64 bits
Type	Read/Write
Address	XXXX 04a0 and a4
Restrictions	None
Power on Reset Value	X'00000000'

2.23: Current PCI Master Address Counter for Debug

This register holds the current PCI Master address counter value. This register holds the value of the PCI Master DMA address. The function of this register is primarily for debug when a severe error occurs that stops the DMA engine from running.

Length	64 bits
Type	Read Only
Address	XXXX 04a8 and ac
Restrictions	None
Power on Reset Value	X'00000000'

2.24: External Entity States Read

This register will get a snapshot value of the enstates pin I/O. This register will return whatever is on the output of the enstates mux output. It is strictly for debug and a convenient way to look at the current state of IBM3206K0424 internal logic. It is controlled by INTST Debug States Control.

Length	64 bits
Type	Read Only
Address	XXXX 04b0 and b4
Restrictions	None
Power on Reset Value	X'xxxxxxxx'

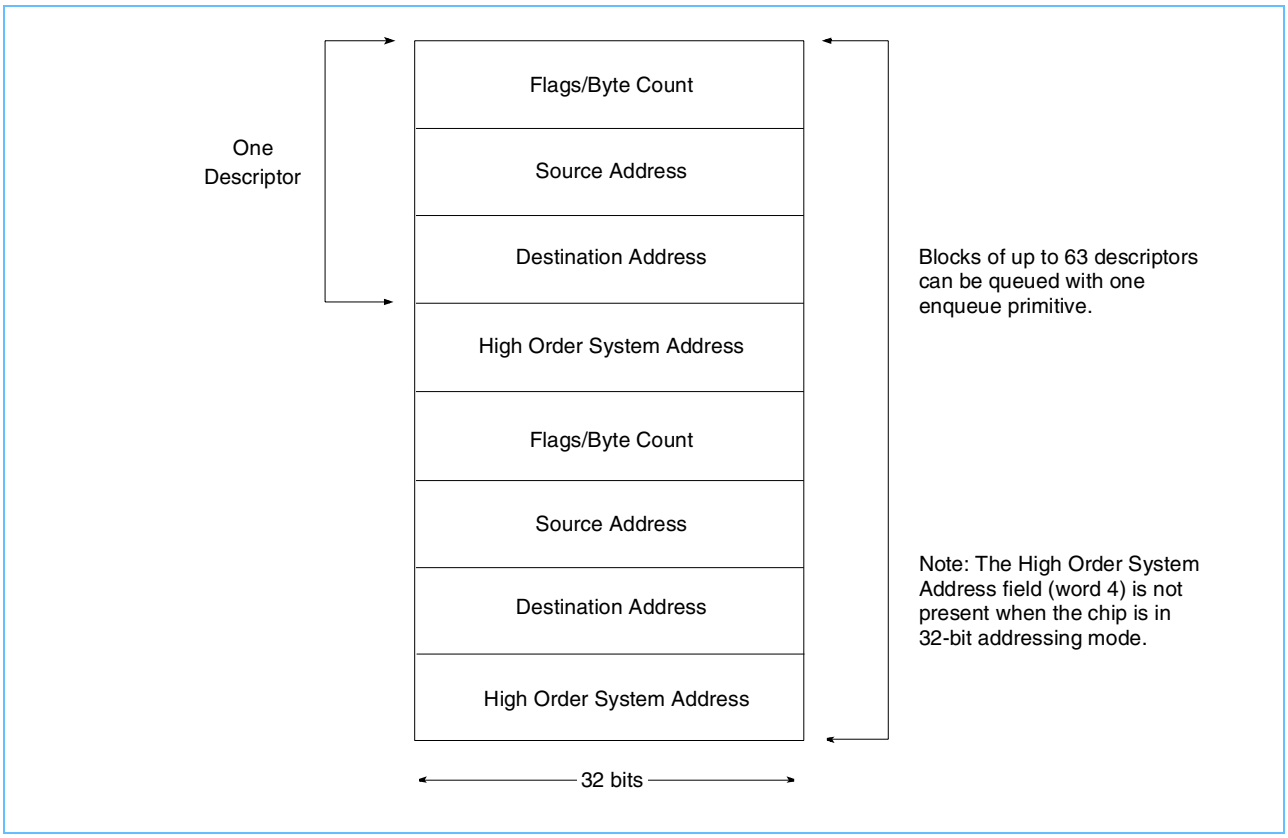
Entity 3: DMA QUEUES (DMAQS)

DMAQS provides the interface to the IBM3206K0424's DMA master capability (described in *General Purpose DMA (GPDMA)* on page 175). It provides three DMA queues that hold DMA descriptor chains that are executed in a multiplexed fashion. Together with GPDMA, a very powerful interface is provided to software to complete complex tasks including TCP/IP checksumming for transmit and receive packets. The following sections describe the features of DMAQS, how to set up DMAQS, and some troubleshooting tips.

DMA Descriptors

DMA descriptors can reside in either PCI/system memory space or the IBM3206K0424 memory space. Certain types of descriptors, called cut-through DMA descriptors, must be located in the IBM3206K0424 memory space. DMA descriptors that are located in the IBM3206K0424 memory space are more efficient to process because they do not need to be moved across the PCI bus. However, it is more costly for software to update across the bus. The best option is to mix descriptors in both locations. DMA descriptors that are infrequently changed should reside in the IBM3206K0424 memory, while dynamic descriptors should be placed in system memory. Descriptors located in the IBM3206K0424 memory space must fall in a definable address range. See *DMAQS Local Descriptor Range Registers* on page 173.

DMA Descriptor Layout



DMA Types and Options

The DMA descriptor is very versatile and can perform many actions. The following list shows some examples and possible flags to use. Other combinations are possible: see *GPDMA Transfer Count and Flag Register* on page 179.

DMA Types and Flags

Hex Flags	DMA Operation
3000	Clear the current TCP checksum and include this DMA in the TCP checksum.
1000	Include this DMA in the TCP checksum and use previous checksum as seed.
0800	This DMA transfer is done in Little Endian mode.
0400	Upon completion of this DMA descriptor, the destination address from this descriptor is used as a packet address to be enqueued to transmit.
0100	Queue a DMA complete event when DMA is complete.
0080	Status in the status register is inhibited for this descriptor. This can be useful if ints/polling are being used to track when a particular DMA is complete.
0001	Move system memory to the IBM3206K0424 memory.
0010	Move the IBM3206K0424 memory to system memory.
0012	Move a single IBM3206K0424 register to system memory.
0013	Move IBM3206K0424 memory to system memory and free buffer. Upon DMA completion, the source address is used to free the IBM3206K0424 buffer.
0017	Auto-increment source address and move IBM3206K0424 memory to system memory and free buffer. Upon DMA completion, the source address is used to free the IBM3206K0424 buffer.
0002	Move single IBM3206K0424 register to IBM3206K0424 memory.
0020	Move IBM3206K0424 Memory to single IBM3206K0424 register.
0021	Move system Memory to single IBM3206K0424 register.
0031	Move system memory to a new IBM3206K0424 buffer. A get buffer operation will be done to fill in the destination address using the low four bits of the destination address as a get pool ID.
0050	Move something to source address of next descriptor. Allows indirection.
0062	Move single IBM3206K0424 register to destination address of the next descriptor. Allows a get buffer operation in descriptor chain. See the get buff flag for a better option.
0008	Use source address as immediate data. Allows up to four bytes of immediate data in the DMA descriptor.
0004	Auto-increment the source address. The source address picks up where it left off from the previous DMA descriptor.
000C	Auto-increment the source address and use as immediate data. One use is to free a packet after DMAing data. See the free buff flags for better option.
0040	Auto-increment the destination address. The destination address picks up where it left off from the previous DMA descriptor. One use is transmit scatter into an IBM3206K0424 virtual buffer.
2200	Hold the destination address. Useful for freeing a scatter DMA list, or doing a repetitive write to an IBM3206K0424 register.
1200	Hold the source address. Useful for doing a repetitive read from an IBM3206K0424 register.

Note: These are not the only options. Some of the above can be ORed together also.

Using the above, you can efficiently do TCP checksumming, place user events in receive queues, do register reads/writes, free buffers, and get buffers.

Descriptor Based DMAs

This is the recommended approach to processing DMAs. A single descriptor or a descriptor chain is built that describes the actions to take. The descriptor is then enqueued to the proper DMA Queue. The number of the descriptor in the DMA chain is placed in the lower six bits of the descriptor address as it is enqueued.

Register Based DMAs

While register based DMAs can be enabled and used, they are not recommended because they are not as efficient and they do not leave a debug trail as the descriptors do in the DMA queue. These should not be used concurrently with descriptor-based DMAs for a particular queue, but register-based and descriptor-based DMAs can be used on different queues. One possible use for register-based DMAs is doing DMAs from the core.

Polling, Interrupts, or Events

There are several choices for handling DMA completion. First, the status register can be polled. While not very efficient, it is the easiest option. Second, you can use interrupts to tell when a DMA is done. Again, not very efficient. However, interrupts should be used to tell when a DMA error has occurred.

One way to deal with DMA completes is to use the RXQUE event mechanism. By generating events, the user can dump in DMA descriptor and clean up at a later time when it is convenient. The user can use the automatic DMA events using the queue on DMA complete flag, or the user can place a user event on an arbitrary queue by writing a DMA descriptor that does an explicit RXQUE enqueue with user data.

Error Detection and Recovery

Ideally, there should not be any errors. Errors are usually user errors in the DMA descriptor which need to be fixed and are not recoverable. Errors on the PCI bus (that is, parity) should not occur in a normal working system, and typically you do not want to recover them. However, if recovery is desired, the current DMA must be recovered in GPDMA. Upon successful completion of the recovered DMA, DMAQS will resume operation.

DMA/Queue Scheduling Options

There are three DMA queues. Queue 0 is higher priority than the other two. This high priority queue is always scheduled to go if the current descriptor is ready. The other two queues (Q1 and Q2) are of equal priority and are scheduled in a round robin fashion when the descriptor is ready. This is meant to provide a transmit DMA queue, receive DMA queue, and a high priority DMA queue. However, these queues can be used for any purpose by setting the routing registers properly.

The queues can be arbitrated after each DMA request length operation, after complete DMA descriptor chains complete, or after a single DMA descriptor in a chain completes. The queues can also be placed in true round robin mode, where all three queues have equal priority.

Address Size

DMAQS can be operated with either 32- or 64-bit System Addresses. See PCINT 64-bit Control Register. All DMAQS address registers are 64 bits wide. In 32-bit addressing mode, the high order portion of address registers are initialized at reset to ‘0’, and cannot be modified. In 32 bit addressing mode, word four of the DMA Buffer Descriptor is ignored.

Data Width

DMAQS recognizes 64 bit writes to 64 bit internal registers. DMAQS internal 64-bit registers may be written either as a 64-bit entity, or by two 32-bit writes. All DMAQS registers are memory mapped on a 64-bit boundary (address bits 2:0 = 0). When in 64-bit addressing mode, an address register is updated with 32 bit writes (atomicity of update cannot be guaranteed). The user should use semaphores to assure the integrity of the operation.

Initialization of DMAQS

DMAQS is very simple to set up by following these steps:

1. Set up each of the three DMA queues.

   To do this, you need to know the size of each queue (see *DMAQS Upper Bound Registers on page 159* for choices). Given this information, the DMA queue is set up with two register writes in diagnostic mode (see *DMAQS Control Register* on page 164).

   ```
   dmaqs->lowerBound[q] = baseAddress // should be aligned with size of queue
   dmaqs->upperBound[q] = encodedSize; // et encoded size of dma queue
   or dmags->bounds[q] = baseAddress + encodedSize; // if 64 bit data is enabled
   ```

   The data structure for the DMA queue is now set up.

2. Set up the queue thresholds if they are being used:

   ```
   dmaqs->threshold[q]=threshold //setthresholdsizetobeinterrupted on
   //may also need to set int mask
   ```

3. Set up the local DMA descriptor range if local descriptors are being used:

   ```
   dmaqs->localDescriptorLowerBound = localDescriptorBase // set base addr of local desc in
   IBM3206KO424 memory
   dmaqs->localDescriptorUpperBound = localDescriptorEnd; // set ending addr of local desc in
   IBM3206KO424 memory
   ```

4. Set up any options that are being used in the DMAQS Control Register:

   ```
   dmaqs->control[set] = ENABLE_DMA_QUEUES | CLR_CHECKSUM_TO_FOXES; // set options/modes
   ```

5. Finally, clear the diagnostic bit:

   ```
   dmaqs->control[clr] = DIAG_MODE // clear the diag mode bit
   ```

6. Need to set up memory bank selection if necessary, but normally Control Memory is used.

3.1: DMAQS Lower Bound Registers

These registers specify the lower bound of the corresponding DMA queue data structure. These registers specify the lower bound of the corresponding DMA queue data structure. The head, tail, and length of the DMA queue are initialized when this register is written. When the DMA queue wraps past the upper bound, it wraps back to the value in the lower bound register, thus implementing the DMA queue as a circular buffer.

When this register is written, the corresponding DMA queue is essentially reset. This is because the head, tail, and length of the queue are all reset.

Length	32 bits	
Type	Read/Write	
Address	Queue 0	604
	Queue 1	684
	Queue 2	704
Power on Value	X'00000000'	
Restrictions	During normal operations, these registers are read only. These registers can only be written when the diagnostic bit has been set in the DMAQS Control Register. The alignment should correspond to the size specified in the upper bound register. For example, it should be 4K aligned if the upper bound specifies 4K size. The low order nine bits are not writable and read back '0'.	

3.2: DMAQS Upper Bound Registers

These registers specify the encoded size/upper bound of the corresponding DMA queue data structure. The actual upper bound is calculated by adding the decoded queue size to the lower bound. When the DMA queue wraps past the upper bound, it wraps back to the lower bound register, thus implementing the DMA queue as a circular buffer.

Length	3 bits	
Type	Read/Write	
Address	Queue 0	XXXX 0604
	Queue 1	XXXX 0644
	Queue 2	XXXX 0684
Power on Value	X'00000000'	
Restrictions	During normal operations, these registers are read only. These registers can only be written when the diagnostic bit has been set in the DMAQS Control Register.	

Bit(s)	Description
2-0	Encoding is as follows: 000 59512 bytes of memory 001 611K bytes of memory 010 632K bytes of memory 011 654K bytes of memory 100 678K bytes of memory 101 6916K bytes of memory 110 7132K bytes of memory 111 7364K bytes of memory

3.3: DMAQS Head Pointer Registers

These registers point to the head element of the corresponding DMA queue. During normal operations, these registers do not need to be read or written; they are used by the IBM3206K0424 to implement the DMA queues. These registers are initialized when the DMAQS Lower Bound Registers for the corresponding DMA queue is written.

Length	32 bits	
Type	Read/Write	
Address	Queue 0	608
	Queue 1	688
	Queue 2	708
Power on Value	X'00000000'	
Restrictions	During normal operations, these registers are read only. They can only be written when the diagnostic bit has been set in the DMAQS Control Register. The head pointer registers are 4-byte aligned (low order two bits are always '0'). Bits 31-17 are calculated internally and are not writable.	

3.4: DMAQS Tail Pointer Registers

These registers point to the next free element of the corresponding DMA queue.

Length	32 bits	
Type	Read/Write	
Address	Queue 0	60C
	Queue 1	68C
	Queue 2	70C
Power on Value	X'00000000'	
Restrictions	During normal operations, these registers are read only. They can only be written when the diagnostic bit has been set in the DMAQS Control Register. The head pointer registers are four-byte aligned (low order two bits are always '0'). Bits 31-17 are calculated internally and are not writable.	

3.5: DMAQS Length Registers

These registers specify the length in bytes of the corresponding DMA queue. This register is cleared when the corresponding DMAQS Lower Bound Registers is written.

Length	17 bits	
Type	Read	
Address	Queue 0	614
	Queue 1	694
	Queue 2	714
Power on Value	X'00000000'	
Restrictions	The lengths are calculated and cannot be written.	

3.6: DMAQS Threshold Registers

These registers specify a queue length threshold at which the corresponding status bit is generated.

These registers should be set equal to the queue length that should cause status to be generated. For example, if the value was set to five, then no interrupt would be generated until the queue was length five or more for the corresponding DMA queue. The threshold is level sensitive, so as long as the length is greater than or equal to the threshold, the corresponding status bit is set. When this register is set to '0', no thresholding occurs.

Length	17 bits	
Type	Read/Write	
Address	Queue 0	61C
	Queue 1	69C
	Queue 2	71C
Power on Value	X'0000'	
Restrictions	Must be a multiple of four.	

3.7: DMAQS Interrupt Status

This register indicates the source(s) of the interrupt(s) pending.

See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

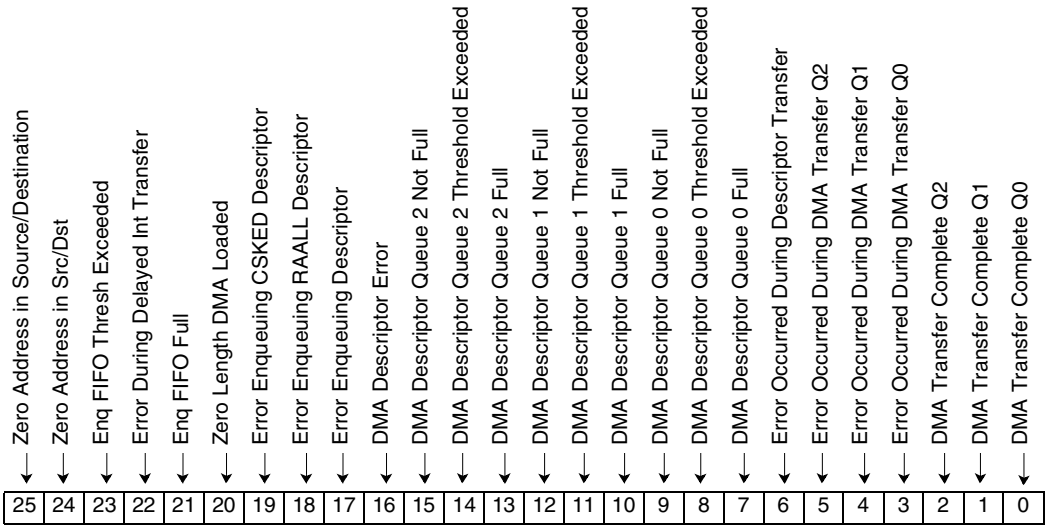
Length 26 bits

Type Read/Write

Address XXXX 6F0 and 6F4

Restrictions None

Power on Value X'00009200'



Bit(s)	Function	Description
25	Zero Address in Source/Destination	When set, a zero address was detected in the source or destination field of a DMA descriptor. The remainder of the descriptor chain was skipped and an event was enqueued to the DMA complete queue. This may or may not be an error condition. It is not an error if the get buffer mode is being used and no buffer was available. In this case, the descriptor can be retried or discarded by software.
24	Enq FIFO Thresh Exceeded	When set, the DMA enqueue FIFO length threshold has been exceeded.
23	Enq FIFO Full	When set, the DMA enqueue FIFO is full and further enqueues will be held off. This bit is hot and cannot be reset.
22	Enqueue Primitive Sequence Error	This bit is set when an improper sequence is detected for loading the DMAQS Enqueue DMA Descriptor Primitive in 64 bit addressing mode.
21	Zero Length DMA Loaded	A descriptor was loaded that had a DMA length equal to zero. This will not stop the DMA engine, but it is technically a user error.
20	Error Enqueuing PCORE Descriptor	A descriptor was enqueued from PCORE with a chain length of zero.
19	Error Enqueuing CSKED Descriptor	A descriptor was enqueued from CSKED with a chain length of zero.
18	Error Enqueuing REALL Descriptor	A descriptor was enqueued from REAALL with a chain length of zero.

Bit(s)	Function	Description
17	Error Enqueuing Descriptor	A descriptor was enqueued with a chain length of zero.
16	DMA Descriptor Error	An invalid transfer was described by the value loaded into the Transfer Count and Flag register.
15	DMA Descriptor Queue 2 Not Full	The DMA descriptor Queue 2 is not full. This bit always contains the status of the queue and is therefore is not writable.
14	DMA Descriptor Queue 2 Threshold Exceeded	The threshold for DMA descriptor Queue 2 has been exceeded.
13	DMA Descriptor Queue 2 Full	The DMA descriptor Queue 2 is full. This bit always contains the status of the queue and is therefore is not writable.
12	DMA Descriptor Queue 1 Not Full	The DMA descriptor Queue 1 is not full. This bit always contains the status of the queue and is therefore is not writable.
11	DMA Descriptor Queue 1 Threshold Exceeded	The threshold for DMA descriptor Queue 1 has been exceeded.
10	DMA Descriptor Queue 1 Full	The DMA descriptor Queue 1 is full. This bit always contains the status of the queue and is therefore is not writable.
9	DMA Descriptor Queue 0 Not Full	The DMA descriptor Queue 0 is not full. This bit always contains the status of the queue and is therefore is not writable.
8	DMA Descriptor Queue 0 Threshold Exceeded	The threshold for DMA descriptor Queue 0 has been exceeded.
7	DMA Descriptor Queue 0 Full	The DMA descriptor Queue 0 is full. This bit always contains the status of the queue and is therefore is not writable.
6	Error Occurred During Descriptor Transfer	Hardware errors occurred transferring the DMA descriptor. The transfer stopped after detecting the error. If the descriptor transfer is finished or is to be terminated, the byte count register must be written to clean up the failed descriptor transfer. Before this bit is reset, the DMA descriptor queue must contain the valid descriptor data or the &regdmtdqcn. must be written to the value it contained prior to the descriptor enqueue.
5	Error Occurred During DMA Transfer Q2	Hardware errors occurred during the last transfer on Queue 2. The transfer stopped after detecting the error. Inspect GPDMA registers for actual location of error.
4	Error Occurred During DMA Transfer Q1	Hardware errors occurred during the last transfer on Queue 1. The transfer stopped after detecting the error. Inspect GPDMA registers for actual location of error.
3	Error Occurred During DMA Transfer Q0	Hardware errors occurred during the last transfer on Queue 0. The transfer stopped after detecting the error. Inspect GPDMA registers for actual location of error.
2	DMA Transfer Complete Q2	The DMA transfer has completed for Queue 2.
1	DMA Transfer Complete Q1	The DMA transfer has completed for Queue 1.
0	DMA Transfer Complete Q0	The DMA transfer has completed for Queue 0.

3.8: DMAQS Interrupt Enable

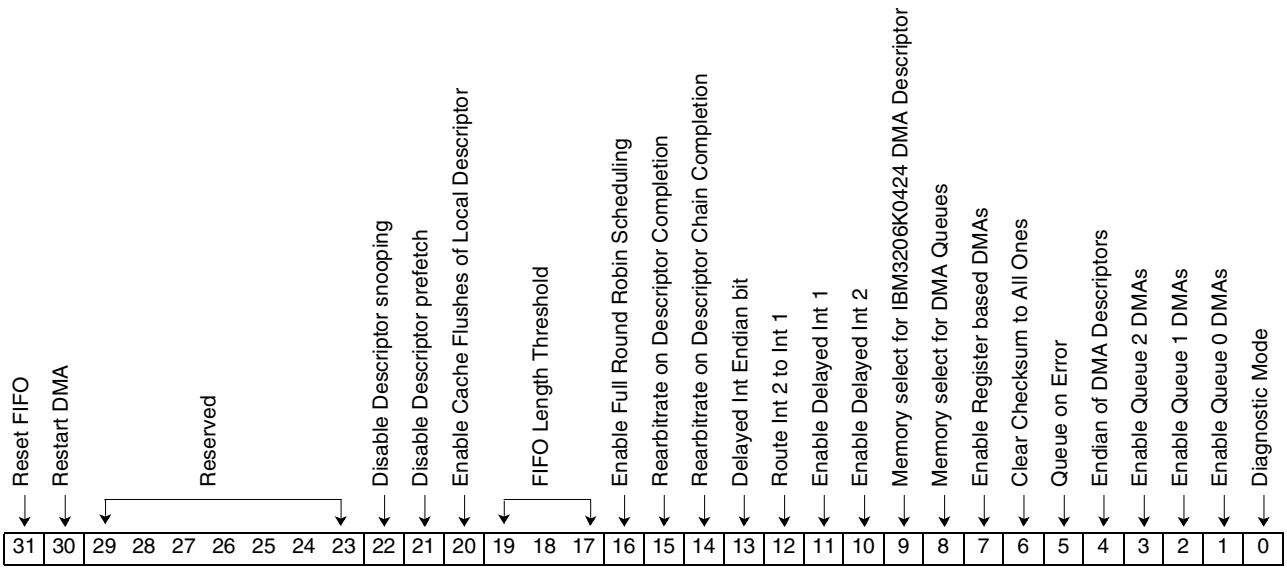
This register serves as a mask for DMAQS Interrupt Status. See *DMAQS Interrupt Status* on page 162 for the bitwise description that the corresponding bit in this register will mask. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	26 bits
Type	Read/Write
Address	XXXX 670 and 674
Restrictions	None
Power on Value	X'00260078'

3.9: DMAQS Control Register

Used to set options for DMAQS. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	32 bits
Type	Read/Write
Address	XXXX 0770 and 774
Restrictions	See bit descriptions.
Power on Value	X'000C0001'



Bit(s0	Function	Description
31	Reset FIFO	When this bit is set, the internal DMA enqueue FIFO is flushed, and this bit is reset. The result is this bit will always be read as a '0'. This bit can only be set in diagnostic mode.

Bit(s0	Function	Description
30	Restart DMA	When this bit is set, the internal DMA state machine restarts the current DMA that is stopped, and this bit is reset. The result is this bit will always be read as a '0'. This bit should only be used at the specific recommendation of an IBM3206K0424 developer.
29-23	Reserved	Reserved.
22	Disable Descriptor snooping	When set, this bit is the DMA descriptor snooping logic is disabled. When this bit is enabled, IBM3206K0424 performance may be enhanced.
21	Disable Descriptor prefetch	When set, this bit is the next descriptor prefetch logic is disabled. Otherwise Performance may be enhanced by enabling this function.
20	Enable Cache Flushes of Local Descriptor	When set, this bit is all local DMA descriptors are flushed out of BCACH before being used. This only needs to be used if local DMA descriptors are in Packet Memory and are updated via the slave interface. Cut-through descriptors do not fall in this category.
19-17	FIFO Length Threshold	This value is used to set the FIFO length threshold. When this threshold is exceeded, bit 23 of the Interrupt Status Register is set.
16	Enable Full Round Robin Scheduling	When set, this bit is all three DMA queues are of equal priority. When cleared, Queue 0 is higher priority than queues 1 and 2.
15	Rearbitrate on Descriptor Completion	When set, this bit is the DMA queues are rearbitrated after each individual DMA descriptor completes.
14	Rearbitrate on Descriptor Chain Completion	When set, this bit is the DMA queues are rearbitrated after full DMA descriptor chains complete. This bit takes precedence over bit 15. When both bits 14 and 15 are cleared, the queues are rearbitrated after each DMA request length operation.
13	True Queue Zero Preference Scheduling Mode	Bit 13 is provided to ensure compatibility with previous chips. In version 2.1 and before, Queue 0 would not be scheduled immediately after itself if queue 1 or queue 2 were ready when a queue 0 DMA completed. This was because it took at least one cycle to reload the queue registers. In IBM3206K0424, the queue registers are loaded while the arbitration for the next DMA is being done, if preloading or snooping is enabled. In this case, with bit 17 set, a Queue 0 DMA may be immediately followed by another Queue 0 DMA. With bit 17 reset, the scheduling (with all queues ready) is q0q1q0q2q0q1.... This mode is provided to give Queue 0 scheduling preference without permitting it to lock out the other two queues.
12	Queue 2 uses on chip SRAM	This bit directs queue 2 to fetch all DMA descriptors from the On-Chip SRAM. Bits 63-18 of the system descriptor address will be ignored.
11	Queue 1 uses on chip SRAM	This bit directs queue 1 to fetch all DMA descriptors from the On-Chip SRAM. Bits 63-18 of the system descriptor address will be ignored.
10	Queue 0 uses on chip SRAM	This bit directs queue 0 to fetch all DMA descriptors from the On-Chip SRAM. Bits 63-18 of the system descriptor address will be ignored.
9	Memory select for IBM3206K0424 DMA Descriptor	When this bit is set, the DMA descriptors that are located in the IBM3206K0424 are located in Packet Memory. Otherwise they are located in Control Memory.
8	Memory select for DMA Queues	When this bit is set, the DMA Queues are located in Packet Memory. Otherwise they are located in Control Memory.
7	Enable Register based DMAs	When set, this bit is source, destination, count, and system descriptor address (SDA) registers can be written to start a DMA.
6	Clear Checksum to All Ones	When this bit is set and the DMAQS Checksum Register is cleared, the DMAQS Checksum Register is set to 0xffff. When this bit is cleared and the DMAQS Checksum Register is cleared, the DMAQS Checksum Register is set to '0'. This option should be used if the TCP/IP checksum should never be set to '0' (0xffff is '0' also).
5	Queue on Error	When set, this bit causes any DMA error to log an error event.
4	Endian of DMA Descriptors	When set, this bit indicates that DMA descriptors in system memory are in little endian format. The default is big endian.
3	Enable Queue 2 DMAs	This bit enables DMA Queue 2.
2	Enable Queue 1 DMAs	This bit enables DMA Queue 1.
1	Enable Queue 0 DMAs	This bit enables DMA Queue 0.
0	Diagnostic Mode	When this bit is set, DMAQS is in diagnostic mode.

3.10: DMAQS Enqueue DMA Descriptor Primitive

This register enqueues a DMA descriptor chain to the corresponding DMA queue. The write data is the address of the descriptor chain that describes the DMA transfers. The low six bits contain a count of the number of DMA descriptors in this chain. After the DMA descriptors are enqueued by writing to this register, the chain of descriptors is fetched from system memory and the DMA transfers described by the chain of descriptors are performed. In 32 bit addressing mode, the low-order 32 bits of the register are written, and the high-order 32 bits are reset when the register is loaded.

Length	64 bits	
Type	Write	
Address	Queue 0	XXXX 0620
	Queue 1	XXXX 06A0
	Queue 2	XXXX 0720
Power on Value	Queue 0	X'00000000000000000'
	Queue 1	X'00000000'
	Queue 2	X'00000000'
Restrictions	None	

3.11: DMAQS Source Address Register

This register is used to set and keep track of the Source Address during a DMA transfer. This is the source for the current DMA transfer. A bit in the Transfer Count and Flag Register determines whether the source address is internal to the IBM3206K0424 or is a system address. In 32-bit addressing mode, the low-order 32 bits of the register are written, and the high-order 32 bits are reset when the register is loaded.

Length	64 bits	
Type	Read/Write	
Address	Queue 0	XXXX 0638
	Queue 1	XXXX 06B8
	Queue 2	XXXX 0738
Power on Value	X'0000000000000000'	
Restrictions	None	

3.12: DMAQS Destination Address Register

This register is used to set and keep track of the destination address during a DMA transfer. This is the Destination address for the current DMA transfer. In 32-bit addressing mode, the low-order 32 bits of the register are written, and the high-order 32 bits are reset when the register is loaded. A bit in the Transfer Count and Flag Register determines whether the destination address is internal to the IBM3206K0424 or is a system address.

Length	32 bits	
Type	Read/Write	
Address	Queue 0	XXXX 0628
	Queue 1	XXXX 06A8
	Queue 2	XXXX 0728
Power on Value	X'0000000000000000'	
Restrictions	None	

3.13: DMAQS Buffer Address Register

This register is used to set and keep track of the POOLS Buffer address during a DMA transfer. When the DMA Descriptor directs that a new buffer address be obtained from POOLS, this is Buffer Address for the current DMA transfer. A bit in the Transfer Count and Flag Register determines whether a buffer address has been obtained for this descriptor. This register can be written to an RXQUE queue. The low-order seven bits should be set to 0x2a, the event code for Assign Pool Buffer events.

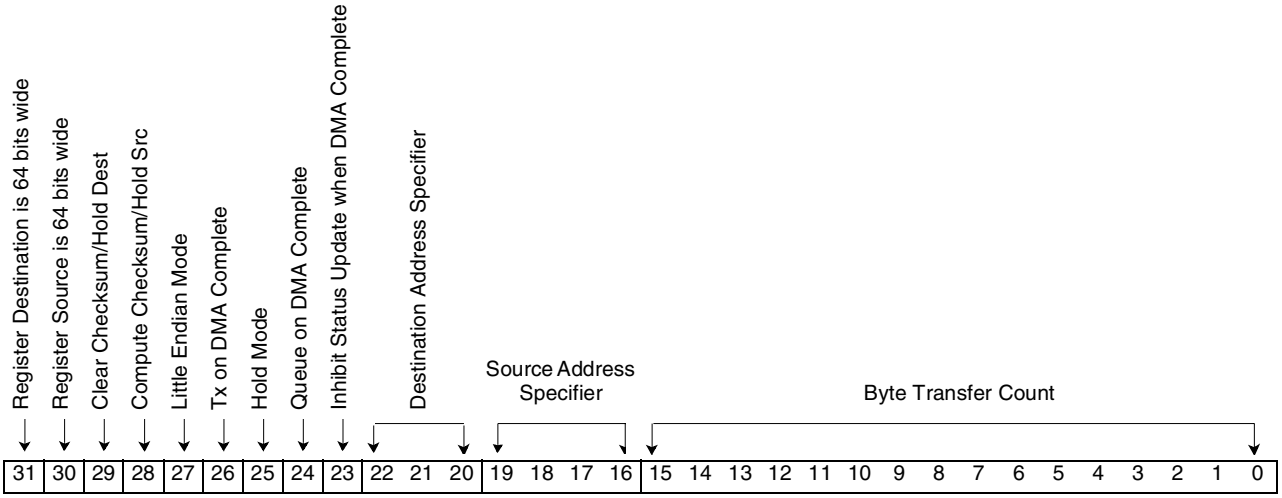
This is the Destination address for the current DMA transfer. In 32-bit addressing mode, the low-order 32 bits of the register are written, and the high-order 32 bits are reset when the register is loaded. A bit in the Transfer Count and Flag Register determines whether the destination address is internal to the IBM3206K0424 or is a system address.

Length	32 bits	
Type	Read/Write	
Address	Queue 0	XXXX 0630
	Queue 1	XXXX 06B0
	Queue 2	XXXX 0730
Power on Value	X'0000002a'	
Restrictions	These registers should not be written during normal system operation. The low-order 7 bits are set to 0x2a and should not be modified. This is the event code for BFA events in RXQUE.	

3.14: DMAQS Transfer Count and Flag Register

This register specifies the type and number of bytes transferred during a DMA transfer. The lower 16 bits are a counter of the number of bytes transferred during a DMA transfer. The upper 16 bits specify the type of transfer.

Length	32 bits	
Type	Read/Write	
Address	Queue 0	XXXX 644
	Queue 1	XXXX 06C4
	Queue 2	XXXX 0744
Power on Value	X'00000000'	
Restrictions	None	



Bit(s)	Function	Description
31	Register Destination is 64 bits wide	These bits must be used with bits 25 and 29 (see below).
30	Register Source is 64 bits wide	
29	Clear Checksum/Hold Dest	When this bit is set, the checksum and the alignment state are cleared.
28	Compute Checksum/Hold Src	When this bit is set, a checksum will be computed over this DMA segment.
27	Little Endian Mode	When this bit is written to '0', this DMA channel operates in big endian mode. When set to '1', the channels operate in little endian mode. In little endian mode, both the source and destination must be aligned on four-byte boundaries.
26	Tx on DMA Complete	When set, the destination address is used as the packet address that is to be enqueued to CSKED to be transmitted. The lower bits are set to '0' so the buffer base is used for the CSKED enqueue operation.

Bit(s)	Function	Description
25	Hold Mode	When set, bits 31-28 are redefined to allow the source or destination address to be held instead of incremented. Bit 29 becomes hold destination address and bit 28 becomes hold source address. This allows a single DMA descriptor to do an N-to-1 or 1-to-N transfer. For example, an entire scatter DMA list can be freed to a receive queue enqueue register. The address being held must be a register address. When holding, the maximum length is 252 bytes. When holding, the source or destination is incremented by four when the DMA completes (for auto-increment mode). Bit 31 becomes destination address 64 bits wide. Bit 30 becomes source address 64 bits wide. This destination is required to properly update 64 bit wide registers when hold mode is asserted.
24	Queue on DMA Complete	When this bit is set, the upper 26 bits of the DMAQS System Descriptor Address register will be queued to the DMA event queue when the DMA completes. If descriptors are not being used to set up the DMA, then before starting the DMA, the DMAQS System Descriptor Address register should be loaded before starting the DMA with a value to identify this transfer. If descriptors are being used, the DMAQS System Descriptor Address register will be loaded automatically with the system address of the descriptor block at the time it is processed.
23	Inhibit Status Update when DMA Complete	Normally a bit will be set in the status register when the DMA completes without error. If this bit is set, this update will not be done. This bit is useful when multiple DMAs are to be done and an interrupt is only desired on the last transfer. The DMA error status bits are not affected by this bit.
22-20	Destination Address Specifier	These bits specify how the destination address should be used for this DMA descriptor. The following are the valid patterns: 000 IBM3206K0424 memory address: The destination address specifies an IBM3206K0424 internal memory address. 001 PCI Bus Address: The destination address specifies a PCI bus address. 010 IBM3206K0424 Register Address: The destination address specifies an IBM3206K0424 register address. Only the low 16 bits must be specified. 011 Get IBM3206K0424 Buffer: The low four bits of the destination address specify a pool ID from which to get a buffer. If a buffer is not available, a zero destination address event or appropriate status is raised. Otherwise the buffer address is used as an IBM3206K0424 memory address. 100 Auto Increment Destination Address: The destination address is sourced from the previous DMA instead of the destination address specified in the descriptor. 101 Next Source Address: The destination address is the address of the source address field of the next descriptor in the current DMA chain. Using this feature allows indirection. 110 Next Destination Address: The destination address is the address of the destination address field of the next descriptor in the current DMA chain. Using this feature allows operations like doing a get buffer in the DMA descriptor chain. 111 Offset Destination Address: The Destination Address is a positive offset from the DMAQS Buffer Address Register. Using this feature allows, for example, storing the checksum value in the header of the packet. Others Reserved: Reserved and flagged as errors.

Bit(s)	Function	Description
19-16	Source Address Specifier	These bits specify how the source address should be used for this DMA descriptor. The following are the valid patterns: 0000 IBM3206K0424 Memory Address: The source address specifies an IBM3206K0424 internal memory address. 0001 PCI bus address: The source address specifies a PCI bus address. 0010 IBM3206K0424 Register Address: The source address specifies an IBM3206K0424 register address. Only the low 16 bits must be specified. 0011 IBM3206K0424 Memory Address and Free Buffer when DMA Complete: The source address specifies an IBM3206K0424 internal memory address, and this address will be freed to POOLS when the DMA is complete. -100 Auto Increment Source Address: The source address is sourced from the previous DMA instead of the source address specified in the descriptor. -111 Auto Increment Source Address and Free Buffer when DMA Complete: The source address is sourced from the previous DMA instead of the source address specified in the descriptor. The source address specifies an IBM3206K0424 internal memory address, and this address will be freed to POOLS when the DMA is complete. 1000 Immediate Data: Use the Source Address Field as immediate data. The data is 4 bytes in 32-bit addressing mode or eight bytes when in 64 bit addressing mode. If the count is greater than the data size, the data is repeated. 1-11 Immediate Data and Free Buffer when DMA Complete: Use the Source Address Field as immediate data. The data is 4 bytes in 32 bit addressing mode or eight bytes when in 64-bit addressing mode. If the count is greater than the data size, the data is repeated. The source address will be freed to POOLS when the DMA is complete. Others Reserved: Reserved and flagged as errors.
15-0	Byte Transfer Count	These bits indicate the number of bytes to transfer. A non-zero value in this field will start the DMA transfer.

3.15: DMAQS System Descriptor Address

The upper 57 bits contain the address of the current descriptor block and the lower seven bits contain the number of descriptors in the chain that remain to be processed. When doing register-based DMAs, the low six bits are set to "000001" when the DMAQS Transfer Count and Flag Register is written. If DMA descriptors are used for DMA transfers, this register will contain the system address of the current descriptor block and the number of descriptors that remain to be processed. This address may be queued on DMA completion to correlate DMA transfers with system control blocks. In 32-bit addressing mode, the low-order 32 bits of the register are written, and the high-order 32 bits are reset when the register is loaded.

Length	64 bits	
Type	Read/Write	
Address	Queue 0	XXXX 0648
	Queue 1	XXXX 06C8
	Queue 2	XXXX 0748
Power on Value	X'0000000000000000'	
Restrictions	This register should not be written if descriptors are going to be used to set up DMA transfers. If it is used, it must be written to 0 before descriptors are enqueued.	

3.16: DMAQS Checksum Register

This register contains the accumulated checksum. This register contains the accumulated checksum value. It can also be used to initialize the checksum with a seed value. The most significant bit contains the alignment state (1 = odd, 0 = even alignment). The alignment state is significant between subsequent checksummed DMAs.

This register can be read at four different addresses. The base address returns the unmodified accumulated checksum. The base address +4 returns the inverted accumulated checksum. The base address + 8 returns the byte-swapped accumulated checksum. The base address + 12 returns the inverted byte-swapped accumulated checksum.

Length	17 bits	
Type	Read/Write	
Address	Q0 Sum	XXXX 0654
	Q0 Inv Sum	XXXX 065C
	Q0 Swapped Sum	XXXX 0664
	Q0 Inv Swapped	XXXX 066C
	Q1 Sum	XXXX 06D4
	Q1 Inv Sum	XXXX 06DC
	Q1 Swapped Sum	XXXX 06E4
	Q1 Inv Swapped	XXXX 06EC
	Q2 Sum	XXXX 0754
	Q2 Inv Sum	XXXX 075C

	Q2 Swapped Sum	XXXX 0764
	Q2 Inv Swapped	XXXX 076C
Power on Value	Q0 Sum	X'00000000'
	Q0 Inv Sum	X'0000ffff'
	Q0 Swapped Sum	X'00000000'
	Q0 Inv Swapped	X'ffff0000'
	Q1 Sum	X'00000000
	Q1 Inv Sum	X'0000ffff'
	Q1 Swapped Sum	X'00000000'
	Q1 Inv Swapped	X'ffff0000'
	Q2 Sum	X'00000000'
	Q2 Inv Sum	X'0000ffff'
	Q2 Swapped Sum	X'00000000'
	Q2 Inv Swapped	X'ffff0000'

Restrictions Only the base address accepts write data. All four addresses return read data.

3.17: DMAQS Local Descriptor Range Registers

These registers specify the lower and upper bounds respectively of the memory range for local DMA descriptors.

These registers contain the address of the lower and upper bound of the memory range of descriptors that are in the IBM3206K0424. If a descriptor block is enqueued, it is compared to these registers. If it falls within this range, only the descriptor address is placed on the queue. When the descriptor is to be loaded into the DMA registers, and it falls within this range, it will not be taken from the queue but loaded directly from the descriptor address. These registers are 4K aligned. The upper bound register contains the address of the last 4K block in the local descriptor address range.

Length	Lower Bound	64 bits
	Upper Bound	32 bits
Type	Read/Write	
Address	Lower Bound	XXXX 0798
	Upper Bound	XXXX 07A4
Power on Value	Lower Bound	X'0000000000000000'
	Upper Bound	X'00000000'
Restrictions	Can be written in diagnostic mode only. Upper bound is 32 bits. The upper 32 bits are internally generated, and are not different from the upper 32 bits of the lower bound register. The last four addresses in this range are reserved, and should not be used to hold descriptors.	

3.18: DMAQS Event Queue Number Register

This register specifies which DMAQS queue should be used when DMA descriptors are enqueued from CSKED (DMA on transmit comp). This register also indicates the RXQUE Event Queue to which events should be enqueued for each DMAQS queue register.

Length	20 bits
Type	Read/Write
Address	XXXX 07CC
Power on Value	X'00002210'
Restrictions	Can be written in diagnostic mode only. Invalid values (that is, 3), force queue number 2.

3.19: DMAQS DMA Request Size Register

This register specifies the maximum request size for DMA descriptor scheduling.

This is the amount of data that DMAQS will request GPDMA to move in a single request. For example, if a descriptor wants to move 2K of data and the request size is set to 512 bytes, then DMAQS will request 512 bytes to be moved and then rearbitrate the DMA queues. A value of '0' is the same as 0xffff.

Length	16 bits
Type	Read/Write
Address	XXXX 07C0
Power on Value	X'00000000'
Restrictions	None

3.20: DMAQS Enq FIFO Register

This register is for diagnostic use only. It holds DMA descriptor waiting to be placed on a DMA queue. Reading this register is destructive. The oldest entry is read on each read. If it is desired to re-dispatch the dequeued entries, they will have to be re-enqueued.

DMA Descriptors which reside in System Storage are not immediately copied into the queue, but space is reserved in the queue for the descriptors, and a descriptor is built which will copy the descriptor into the queue when needed.

Length	64 bits
Type	Read Only
Address	XXXX 07F8
Power on Value	X'0000000000000000'
Restrictions	Can not be written.

Entity 4: General Purpose DMA (GPDMA)

This entity provides DMA control between System Memory and IBM3206K0424 Packet Memory.

DMA transfers must be enabled in the GPDMA control registers for transmit and/or receive. There are two ways to initiate DMA transfers. The first is by directly writing the Source Address, Destination Address, and Transfer Count and Flag Registers. The second is by using DMA descriptors and enqueueing them using DMAQS. These two methods should not be used simultaneously. If using descriptors, refer to the DMAQS section beginning on *DMA QUEUES (DMAQS)* on page 154 for more information.

DMA transfers to system I/O space are not allowed.

4.1: GPDMA Interrupt Status

This register indicates the source(s) of the interrupt(s) pending, or is used as a status register when the bits are enabled. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

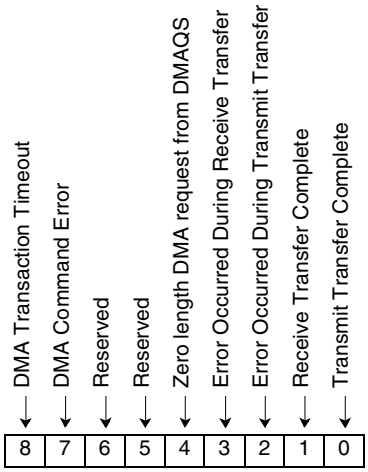
Length 9 bits

Type Read/Write

Address XXXX 0108 and 0C

Power on Reset value X'000'

Restrictions None



Bit(s)	Function	Description
8	DMA Transaction Timeout	The DMA Transaction Timeout specified in the GPDMA Interrupt Enable timed out.
7	DMA Command Error	An invalid transfer was described by the value loaded into the Transfer Count and Flag Register.
6	Reserved	Reserved
5	Reserved	Reserved
4	Zero length DMA request from DMAQS	DMAQS has requested a DMA with a length of zero. This bit is for information use only. This bit is not an error that will prevent GPDMA from processing additional DMA requests.

Bit(s)	Function	Description
3	Error Occurred During Receive Transfer	Hardware errors occurred during the last transfer. The transfer stopped after detecting the error.
2	Error Occurred During Transmit Transfer	Hardware errors occurred during the last transfer. The transfer stopped after detecting the error.
1	Receive Transfer Complete	The receive transfer is complete.
0	Transmit Transfer Complete	The transmit transfer is complete.

4.2: GPDMA Interrupt Enable

This register serves as a mask for GPDMA Interrupt Status. See *GPDMA Interrupt Status* on page 175 for the bitwise description that the corresponding bit in this register will mask. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length 9 bits

Type Read/Write

Address XXXX 0110 and 14

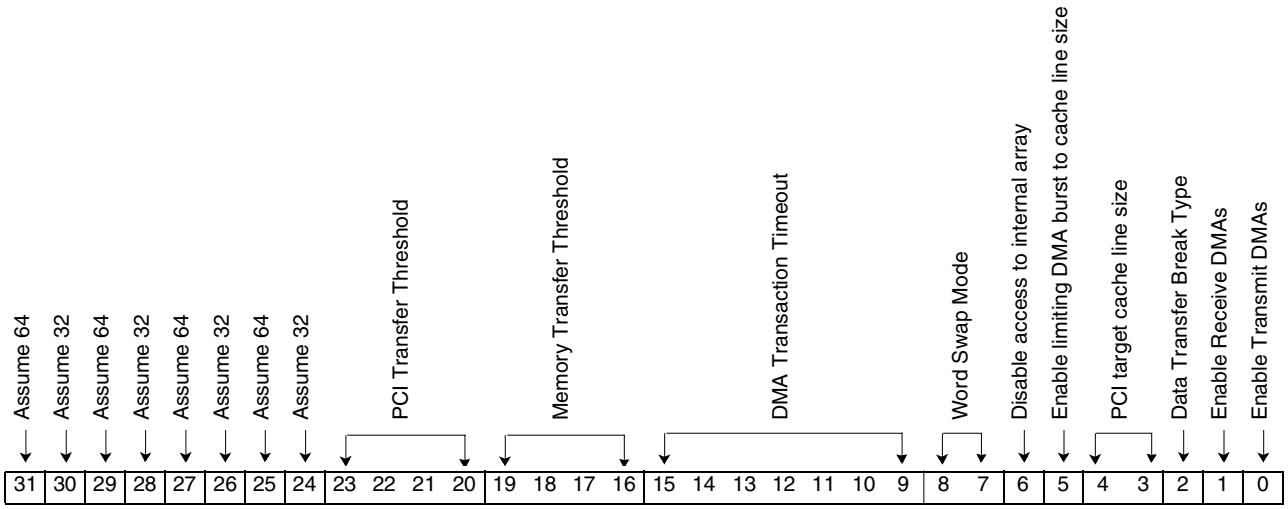
Power on Reset value X'9C'

Restrictions None

4.3: GPDMA Control Register

Used to set options for DMA operations. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	32 bits
Type	Read/Write
Address	XXXX 0118 and 1C
Power on Value	X'008800c7'
Restrictions	None



Bit(s)	Function	Description
31	Assume 64	Assume 64-bit PCI Interface for Descriptor Transfers
30	Assume 32	Assume 32-bit PCI Interface for Descriptor Transfers
29	Assume 64	Assume 64-bit PCI Interface for Queue 2 Transfers
28	Assume 32	Assume 32-bit PCI Interface for Queue 2 Transfers
27	Assume 64	Assume 64-bit PCI Interface for Queue 1 Transfers
26	Assume 32	Assume 32-bit PCI Interface for Queue 1 Transfers
25	Assume 64	Assume 64-bit PCI Interface for Queue 0 Transfers
24	Assume 32	Assume 32-bit PCI Interface for Queue 0 Transfers 2
23-20	PCI Transfer Threshold	The value of these bits multiplied by eight determines the number of bytes that must be ready to transfer before a DMA transfer is initiated on the PCI bus. This can be used to tune the performance of the PCI bus. If the number of bytes left to transfer is less than the threshold, the transfer will start when all remaining bytes are ready to be transferred.
19-16	Memory Transfer Threshold	The value of these bits multiplied by eight determine the number of bytes that must be ready to transfer before a transfer is initiated on the internal memory bus. This can be used to tune the performance of the memory subsystem.
15-9	DMA Transaction Timeout	These bits hold a value that is used to count the number of cycles that an unacknowledged DMA cycle is in progress. If the count is reached, due to an internal chip hang condition, the DMA is terminated. A value of '0' disables this function.

Bit(s)	Function	Description
8-7	Word Swap Mode	This field controls word swapping for data being transferred to PCINT. The word swapping is done as part of endian alignment. The bits are defined as follows: 00 No Swap: The 32-bit words being transferred will not be swapped by PCINT 01 Swap: The 32-bit words being transferred will always be swapped by PCINT 10 Endian Swap: The words will be swapped if byte swapping is being done. 11 Anti-endian Swap: The words will be swapped if byte swapping is not being done.
6	Disable access to internal array	When this bit is set, the internal array cannot be read or written. This can be used to ensure that the array is not inadvertently read or written while DMAs are in progress, causing unpredictable results.
5	Enable limiting DMA burst to cache line size	This bit on causes a DMA burst to terminate upon crossing a cache line boundary of the PCI target.
4-3	PCI target cache line size	This field indicates the cache line size if aligning DMAs to the cache line size of the PCI target (see bit 5). 00 32 bytes 01 64 bytes 10 128 bytes 11 256 bytes
2	Data Transfer Break Type	This bit on causes a re-arbitrate request from PCI to "immediately" stop moving data and resurface a new request to move the remainder of the data. When reset, GPDMA stops data movement at the next Cache line boundary.
1	Enable Receive DMAs	This bit on enables DMA transfers out of the IBM3206K0424.
0	Enable Transmit DMAs	This bit on enables DMA transfers into the IBM3206K0424.

4.4: GPDMA Source Address Register

Used to set and keep track of the Source Address during a DMA transfer. This is the system address that increments during a DMA transfer. A bit in the Transfer Count and Flag Register determines if the source address is internal to the IBM3206K0424 or is a system address.

Length 64 bits

Type Read/Write

Address XXXX 0128

Power on Value X'0000000000000000'

Restrictions None

4.5: GPDMA Destination Address Register

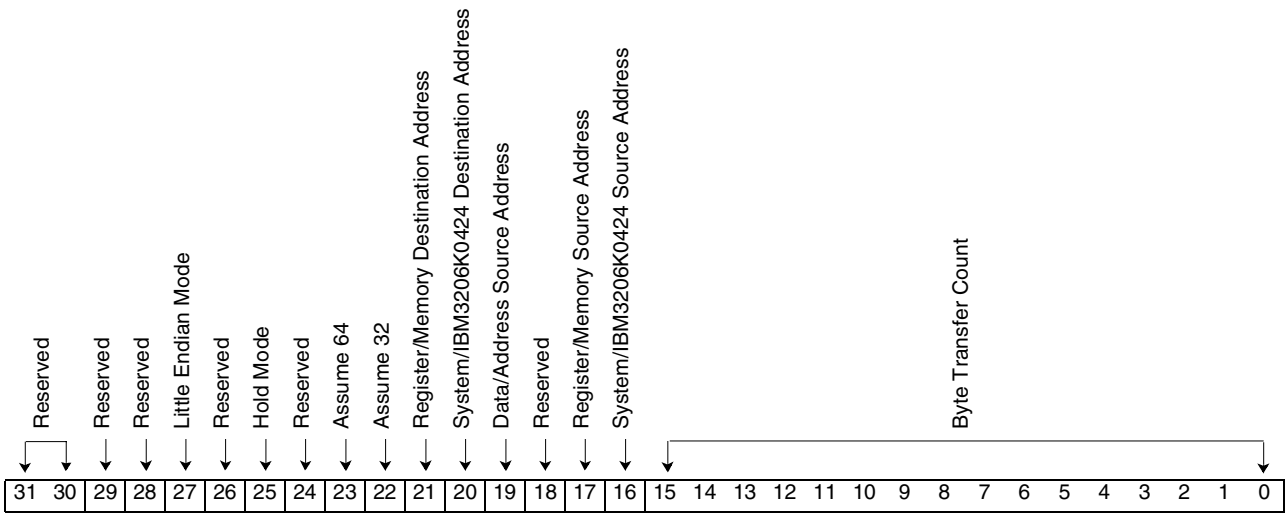
Used to set and keep track of the Destination address during a DMA transfer. This is the Destination address that increments during a DMA transfer. A bit in the Transfer Count and Flag Register determines if the destination address is internal to the IBM3206K0424 or is a system address.

Length	64 bits
Type	Read/Write
Address	XXXX 0130
Power on Value	X'0000000000000000'
Restrictions	None

4.6: GPDMA Transfer Count and Flag Register

Specifies the type and number of bytes transferred during a DMA transfer. The lower 16 bits are a counter of the number of bytes transferred during a DMA transfer. It is a count down counter; when zero is reached, the transfer ends. Writing a non-zero value to the lower 16 bits starts the DMA transfer. The upper 16 bits specify the type of transfer as follows.

Length	32 bits
Type	Read/Write
Address	XXXX 0138
Power on Value	X'000000'
Restrictions	None



Bit(s)	Function	Description
31-30	Reserved	These bits must be '0'.
29	Reserved	
28	Reserved	

Bit(s)	Function	Description
27	Little Endian Mode	When this bit is written to '0', this DMA channel operates in big endian mode. When '1', it operates in little endian mode. When in little endian mode, both the source and destination must be aligned on four-byte boundaries.
26	Reserved	Reserved.
25	Hold Mode	When set, bits 28-29 are redefined to allow the source or destination address to be held instead of incremented. Bit 29 becomes the hold destination address and bit 28 becomes the hold source address. The address being held must be a register address. When holding, the maximum length is 240 bytes.
24	Reserved	Reserved.
23	Assume 64	Assume the PCI Interface is 64-bits wide.
22	Assume 32	Assume the PCI Interface is 32-bits wide.
21	Register/Memory Destination Address	If this bit is set, the destination address is a register address. If this bit is not set, the destination address is a memory address. If the destination address is a system address, this bit should cleared. I/O DMA cycles on the PCI bus are not implemented.
20	System/IBM3206K0424 Destination Address	If this bit is set, the destination address is a PCI bus address. If this bit is not set, the destination address is internal to the chip.
19	Data/Address Source Address	If this bit is set, the Source Address Register contains the source data. If this bit is not set, the Source Address Register contains the source address.
18	Reserved	Reserved.
17	Register/Memory Source Address	If this bit is set, the source address is a register address. If this bit is not set, the source address is a memory address. If the source address is a system address, this bit should cleared. I/O DMA cycles on the PCI bus are not implemented.
16	System/IBM3206K0424 Source Address	If this bit is set, the source address is a PCI bus address. If this bit is not set, the source address is internal to the chip.
15-0	Byte Transfer Count	These bits indicate the number of bytes to transfer. A non-zero value in this field starts the DMA transfer.

4.7: GPDMA DMA Max Burst Time

Used to limit the number of cycles a master can burst on the PCI bus. When a DMA burst is started, a counter is loaded with the value in this register. When the counter expires and the current access completes, the PCI bus is released for use by another bus master. Writing a non-zero value to this register enables this function.

Length	24 bits
Type	Read/Write
Address	XXXX 0158
Power on Value	X'000'
Restrictions	None

4.8: GPDMA Maximum Memory Transfer Count

Used to limit the size of data requests to the Control/Packet Memories. This register defines the maximum number of bytes to be transferred in a single storage request to IBM3206K0424 Storage.

Length	7 bits
Type	Read/Write
Address	XXXX 0150
Power on Value	X'40'
Restrictions	None

4.9: GPDMA Checksum Register

This register contains the accumulated checksum value. It can also be used to initialize the checksum with a seed value. The most significant bit contains the alignment state (1 = odd, 0 = even alignment). This register can be read at four different addresses. The base address returns the unmodified accumulated checksum. The base address +4 returns the inverted accumulated checksum. The base address + 8 returns the byte-swapped accumulated checksum. The base address + 12 returns the inverted byte-swapped accumulated checksum.

Length	17 bits
Type	Read/Write
Address	XXXX 0160
Power on Value	X'00000'
Restrictions	None

4.10: GPDMA Read DMA Byte Count

This register counts the bytes transferred into the IBM3206K0424 by the DMA controller. Descriptor bytes can optionally be included. (See the *GPDMA Control Register* on page 177 for details.)

Length	32 bits
Type	Read/Write
Address	XXXX 0178
Power on Value	X'00000000'
Restrictions	None

4.11: GPDMA Write DMA Byte Count

This register counts the bytes transferred out of the IBM3206K0424 by the DMA controller.

Length	32 bits
Type	Read/Write
Address	XXXX 017C
Power on Value	X'00000000'
Restrictions	None

4.12: GPDMA Array Read Address

This register is used to read the GPDMA internal array. The internal array is used to hold data for the DMA. The array is organized as 64 32-bit words. The GPDMA Array Read Address is written with the address of the word that is to be read. The GPDMA Array Data Register is read to obtain the contents of the addressed word. The GPDMA Array Read Address is incremented each time the GPDMA Array Data Register is read, causing repeated reads of the GPDMA Array Data Register to obtain sequential words from the array.

Length	8 bits
Type	Read/Write
Address	XXXX 0140
Power on Value	X'00'
Restrictions	This address space is for diagnostic use only. It should not be read or written during normal operation. The low order two bits of GPDMA Array Read Address are place holders and are ignored. They should be set to '0'.

4.13: GPDMA Array Write Address

This register is used to write the GPDMA internal array. The internal array is used to hold data for the DMA. The array is organized as 64 32-bit words. The GPDMA Array Write Address is written with the address of the word that is to be written. The GPDMA Array Data Register is written to write the addressed word. The GPDMA Array Write Address is incremented each time the GPDMA Array Data Register is written, causing repeated writes of the GPDMA Array Data Register to write sequential words in the array.

Length	8 bits
Type	Read/Write
Address	XXXX 0144
Power on Value	X'00'
Restrictions	This address space is for diagnostic use only. It should not be read or written during normal operation. The low order two bits of GPDMA Array Read Address are place holders and are ignored. They should be set to '0'.

4.14: GPDMA Array

Reads the contents of the internal array. The internal array is used to hold data for the DMA.

Length	64 words x 32 bits
Type	Read/Write
Address	XXXX 0148
Power on Value	X'00000000'
Restrictions	This address space is for diagnostic use only. It should not be read or written during normal operation. The array is read/written at the location indicated by the GPDMA Array Read Address or GPDMA Array Write Address.

Memory Controlling Entities

Entity 5: The DRAM Controllers (COMET/PAKIT)

This section describes the function of the COMET/PAKIT entities. COMET is the memory controller for Control Memory, and PAKIT is the memory controller for Packet Memory.

Each controller can support the following types of memory:

- Synchronous DRAMs running at 133MHz (7.5 ns cycle time) with a CAS latency of two or three and a burst length of one or two. Memory sizes of 4MB, 8MB, 16MB, and 32MB are supported. Please note that the cycle time of the SDRAM clock is a constant on the IBM3206K0424. Any SDRAM part selected must be capable of running at 133MHz or faster at CAS latency 2 or 3.
- Synchronous SRAM running at 133MHz (7.5 ns cycle time) with a read latency of two and a write latency of zero or two. Memory sizes of 1MB, 2MB, 4MB, and 8MB are supported.

Note: For any memory configuration, modules must be selected such that the loading on any memory net (including card wiring) does not exceed 120pF.

The number of column address lines is programmable, allowing both DRAMs with symmetric address (same number of row and column address lines) and asymmetric address (typically having more row than column address lines).

If using SDRAM, the memory may be operated as having one or two arrays. The arrays are differentiated by their chip selects. If the memory is configured to have two arrays, the memory's address range is split equally between the two arrays.

Memory checking can be enabled/disabled, and the method of checking selected can be either ECC or parity. IF ECC is selected, seven data bits are used for ECC over the 32 data bits. If parity is selected, four data bits are used to provide parity over the 32 data bits.

COMET/PAKIT are designed so that memory contents are preserved over a reset. If the IBM3206K0424 is reset while a memory write cycle is in progress, the cycle is completed in an orderly fashion to ensure that valid ECC/parity is written. Memory timings are not violated when reset goes active. Refresh is maintained during the reset.

Memory Reset Sequence

After a reset, onboard ROM or external firmware must properly configure the control registers for COMET/PAKIT.

If using SRAM, the reset sequence is complete. If using SDRAM, bit 3 of the memory controller's SDRAM Command and Status Register must be written to a '1' to initiate forcing the SDRAMs out of the self refresh state and performing the POR sequence. When bits five and four of this register are '00', the SDRAMs are ready for use.

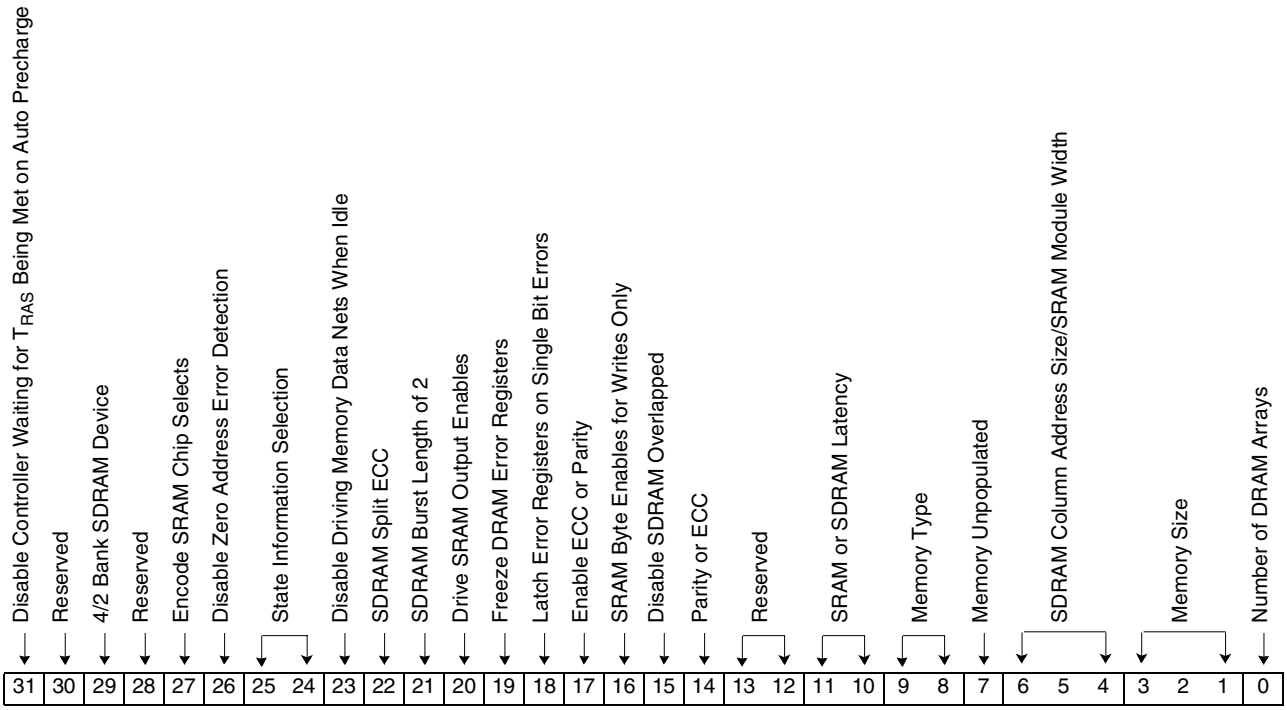
Note: Memory configuration errors occur if an attempt is made to use memory that is configured incorrectly or, if an attempt is made to use SDRAM before the POR sequence is completed.

Accesses to the first 0x20 bytes of memory (Control or Packet) are not allowed unless bit 26 of the corresponding memory control register is set. With this restriction in place, accesses with zero-valued pointers will cause the zero address error bit in the memory controller's status register to be set.

5.1: COMET/PAKIT Control Register

This register contains the information that controls the functions of the entity. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. Before this register can be altered, writing it must be enabled in *COMET/PAKIT Memory Controller Write Enable Register* (described on page 197).

Length	32 bits
Type	Read/Write
COMET Address	XXXX 0900 AND 04
PAKIT Address	XXXX 0980 AND 84
Power On Value	X'00000000'
Restrictions	None



Bit(s)	Function	Description
31	Disable Controller Waiting for T_{RAS} Being Met on Auto Pre-charge	Some SDRAM modules don't allow sending a read or write with auto precharge to them unless T_{RAS} has been met. If using ESDRAM (any SDRAM at burst length of two, or a part that allows sending an auto precharge before T_{RAS} has been met), this bit should be set. This bit has no meaning for SRAM.
30	Reserved	Reserved
29	4/2 Bank SDRAM Device	When this bit is '1', the SDRAMs attached are four-bank devices. When a '0', the SDRAMs are two-bank devices. A '0' setting works for either type device, but four-bank devices provide slightly better performance if this bit is set to '1'.
28	Reserved	Reserved
27	Encode SRAM Chip Selects	When using SRAM, this bit set to a '0' causes the chip select outputs to be direct chip selects. Set to '1', chip selects 2-0 carry an encoded value for one of eight chip selects. Chip select three indicates when chip selects 2-0 are valid.

Bit(s)	Function	Description
26	Disable Zero Address Error Detection.	When set to ‘1’, this bit disables the detection of zero address errors to memory.
25-24	State Information Selection.	These bits control what will be visible on the enstate outputs if COMET/PAKIT are selected for observation on the enstate pins.
23	Disable Driving Memory Data Nets When Idle	When set to ‘1’, this bit disables the memory controller from driving the memory data nets to ’0’ when the controller is idle.
22	SDRAM Split ECC	When set to ‘1’, this bit indicates that the ECC/parity for multiple arrays of memory are in separate modules and a slight increase in performance is possible. If this bit is ‘0’, the ECC/parity is in a shared module. If using neither ECC or parity, this bit should be set to ’1’ for a slight performance increase. This bit applies only when SDRAM is being used.
21	SDRAM Burst Length of 2	When set to ’1’, this bit indicates that the SDRAM should be driven assuming a burst length of two. This bit set to ’0’ indicates a burst length of one.
20	Drive SRAM Output Enables	When set to ’1’, this bit allows functional output enables to be driven for SRAMs. When set to ’0’, the output enables are driven active continuously.
19	Freeze Error Registers	When set to ’1’, this bit freezes the Memory Address Register and the Syndrome Register when a memory error occurs. When this bit is set to ’0’, the error registers are updated whenever an error is encountered. For this bit to have any meaning with single bit errors, bit 18 must also be a ’1’.
18	Latch Error Registers on Single Bit Errors	When set to ’1’, this bit allows error data to be latched into the Memory Error Address Register and the Syndrome Register when a single bit errors occurs. When this bit is set to ’0’, single bits errors do not latch data into the error registers.
17	Enable ECC or Parity	This bit set to ’1’ enables ECC detection/correction or parity error detection.
16	SRAM Byte Enables for Writes Only	When set to ’1’, this bit' causes byte enables to only be driven on writes to SRAM. The enables are driven inactive for reads. If the bit is set to ’0’, the byte enables are valid on both reads and writes.
15	Disable SDRAM Overlapped Bank Accesses/Shorten SRAM Write Duration	When the memory controller is configured for SDRAM, setting this bit to ’1’ disables the overlapping of bank accesses. When configured for SRAM, setting this bit to ’1’ shortens the time the IBM3206K0424 drives data on writes.
14	Parity or ECC	When set to ’1’, this bit causes parity to be generated. This bit set to ’0’ causes ECC to be generated. ECC is supported for DRAM only.
13-12	Reserved	Reserved
11-10	SRAM or SDRAM Latency	These bits indicate the delay between performing a read and the memory returning data. The bits are encoded as follows: 00 1 Cycle (SRAM only) 01 2 Cycles 10 3 Cycles (SDRAM only) 11 Reserved
9-8	Memory Type	These bits indicate the type of memory being used for memory. The bits are encoded as follows: 00 SRAM 01 ZBT SRAM 10 Synchronous DRAM (SDRAM) 11 Enhanced Synchronous DRAM (ESDRAM)
7	Memory Unpopulated.	If this bit is ’1’, there is no physical memory connected to this controller.

Bit(s)	Function	Description
6-4	SDRAM Column Address Size/SRAM Module Width	These bits indicate the number of column address lines. The bits are encoded for SDRAM as follows: '000' 8 column address lines (256 words/row) '001' 9 column address lines (512 words/row) '010' 10 column address lines (1K words/row) '011' Reserved '1XX' Reserved The bits are encoded for SRAM as follows: '000' 18-bit wide 4 Mb SRAM '001' 18-bit wide 8 Mb SRAM '010' 18-bit wide 16 Mb SRAM '011' 18-bit wide 32 Mb SRAM '100' 36-bit wide 4 Mb SRAM '101' 36-bit wide 8 Mb SRAM '110' 36-bit wide 16 Mb SRAM '111' 36-bit wide 32 Mb SRAM
3-1	Memory Size	These bits indicate the total amount of memory present, that is, two 64MB SDRAM arrays would result in a value of 128MB in these bits. The bits are encoded as follows: '000' 1 MB '001' 2 MB '010' 4 MB '011' 8 MB '100' 16 MB '101' 32 MB '110' 64 MB '111' 128 MB - SDRAM Only
0	Number of DRAM Arrays.	This two bit indicates the number of arrays of DRAM present. This bit set to '0' indicates one array; the bit set to '1' indicates two arrays.

5.2: COMET/PAKIT Status Register

This register contains status information for COMET/PAKIT. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length 6 bits

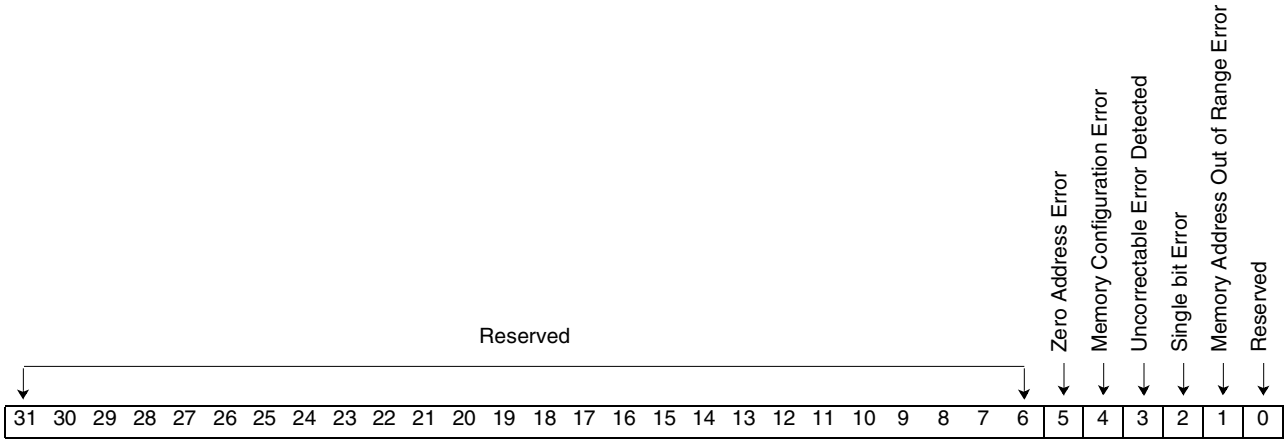
Type Read/Write

COMET Address XXXX 0908 AND 0C

PAKIT Address XXXX 0988 AND 8C

Power On Value X'00000000'

Restrictions



Bit(s)	Function	Description
31-6	Reserved	Reserved
5	Zero Address Error	This bit is set if COMET/PAKIT is presented an address of zero.
4	Memory Configuration Error	This bit is set if COMET/PAKIT is configured in an invalid combination.
3	Uncorrectable Error Detected	This bit is set if an uncorrectable error is detected.
2	Single bit Error	This bit is set if a single bit ECC error is detected.
1	Memory Address Out of Range Error	This bit is set if the address presented to the memory controller is out of the defined range.
0	Reserved	Reserved

5.3: COMET/PAKIT Interrupt Enable Register

This register contains bits corresponding to the bits in the COMET/PAKIT Status Register. If a bit in this register is set and the corresponding bit is set in the COMET/PAKIT Status Register, an interrupt is generated. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	6 bits
Type	Read/Write
COMET Address	XXXX 0910 AND 14
PAKIT Address	XXXX 0990 AND 94
Power On Value	X'0000003A'
Restrictions	

5.4: COMET/PAKIT Lock Enable Register

This register contains bits corresponding to the bits in the COMET/PAKIT Status Register. If a bit in this register is set and the corresponding bit is set in the COMET/PAKIT Status Register, a signal is sent to VIMEM indicating that memory should be locked. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	6 bits
Type	Read/Write
COMET Address	XXXX 0918 AND 1C
PAKIT Address	XXXX 0998 AND 9C
Power On Value	X'0000003A'
Restrictions	

5.5: COMET/PAKIT Memory Error Address Register

This register holds the address at which the last memory error occurred.

Length 32 bits

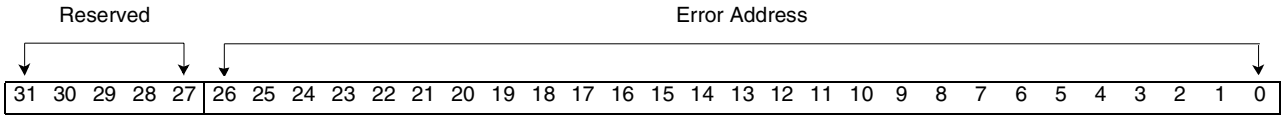
Type Read Only

COMET Address XXXX 0920

PAKIT Address XXXX 09A0

Power On Value X'00000000'

Restrictions



Bit(s)	Function	Description
31-27	Reserved	Reserved.
26-0	Error Address.	The read address of the last memory error.

5.6: COMET/PAKIT SDRAM Command and Status Register

This register is used to issue various commands to and control the timing operation of Synchronous DRAMS when they are attached to the IBM3206K0424. If the IBM3206K0424 is not configured for SDRAMs, any writes to bits 3-0 of this register are ignored. This register is also used to reflect the status of the Synchronous DRAMs. When a command bit in this register is set (bits 3-0 only), the command executes and resets the bit upon completion. Only one bit (3-0 only) may be set during any write. Software should poll this register to make sure the previous command has completed before issuing another write to this register. If more than one bit at a time is written to this register (3-0 only), the results may be unpredictable.

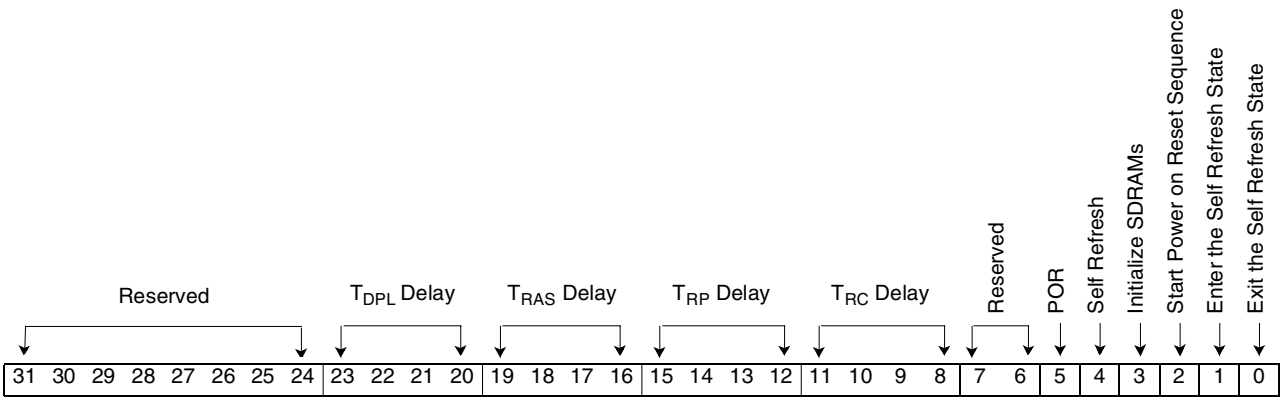
Length 32 bits

Type Read/Write

COMET Address XXXX 0924

PAKIT Address XXXX 09A4

Restrictions Power On Value: X'00003030'



Bit(s)	Function	Description
31-24	Reserved	Reserved.
23-20	T_{DPL} Delay	The value of these four bits determine the number of cycles after writing data before a precharge command may be issued. To determine the value needed in these bits, take the T_{DPL3} (for CAS latency 3) or T_{DPL2} (for CAS latency 2) parameter from the specification of the SDRAM of the part to be used, divide by 7.5 ns, and round up if necessary. Suggested values are: X'2' 6.8 ns SDRAM X'1' 6 ns ESDRAM X'1' 7.5 ns ESDRAM
19-16	T_{RAS} Delay	The value of these four bits determine the number of cycles a bank must be active. To determine the value needed in these bits, take the T_{RAS} parameter from the specification of the SDRAM of the part to be used, divide by 7.5 ns, and round up if necessary. Suggested values are: X'7' 6.8 ns SDRAM X'3' 6 ns ESDRAM X'3' 7.5 ns ESDRAM

Bit(s)	Function	Description
15-12	T_{RP} Delay	The value of these four bits determines the number of cycles after a bank precharge starts before the bank may be accessed again. To determine the value needed in these bits, take the T_{RP} parameter from the specification of the SDRAM of the part to be used, divide by 7.5 ns, and round up if necessary. If Tdpl is 2, Trp may need to be increased by one to insure correct operation. Suggested values are: X'4' 6.8 ns SDRAM X'2' 6 ns ESDRAM X'2' 7.5 ns ESDRAM
11-8	T_{RC} Delay	The value of these four bits determine the bank cycle time. To determine the value needed in these bits, take the T_{RC} parameter from the specification of the SDRAM of the part to be used, divide by 7.5 ns, and round up if necessary. Suggested values are: X'A' 6.8 ns SDRAM X'5' 6 ns ESDRAM X'5' 7.5 ns ESDRAM
7-6	Reserved	Reserved
5	POR	When set to '1', this bit indicates the POR sequence has not been performed on the SDRAMs. This bit automatically resets to '0' when the POR sequence has been performed.
4	Self Refresh	This bit reads '1' when the SDRAMs are in the self refresh state. This bit reads '0' when the SDRAMs are not in the self refresh state. This bit is '1' after a POR or reset. The exit self refresh operation must be performed before the POR sequence is initiated.
3	Initialize SDRAMs	This bit effectively encapsulates the functions provided by bits 2 and 0. Setting this bit to '1' causes the memory controller to take the SDRAMs out of self refresh and perform the POR sequence on them. This bit clears itself. When the initialization is complete, bits 4 and 5 should be a '0'.
2	Start Power on Reset Sequence	When set to '1', this bit causes the DRAM controller to initiate the SDRAM power on sequence. This includes an all-banks precharge, following by a command register write that sets the CAS latency to 3, the wrap type to sequential, and the burst length to 1, followed by two refresh cycles. After this sequence is initiated and completed, bit 5 resets and the SDRAMs are ready for normal use.
1	Enter the Self Refresh State	When set to '1', this bit causes the SDRAM controller to signal the SDRAMs to go into the self refresh state. All memory activity is suspended. Once the SDRAMs have entered the self refresh state, bit 4 will set. This bit will clear itself.
0	Exit the Self Refresh State	When set to '1', this bit causes the SDRAM controller to signal the SDRAMs to exit the self refresh state. Once the SDRAMs have exited the self refresh state, bit 4 will clear. This bit will clear itself.

5.7: COMET/PAKIT DRAM Refresh Rate Register

This register holds the value of a counter used to control the rate of refresh for the DRAM.

Length 32 bits

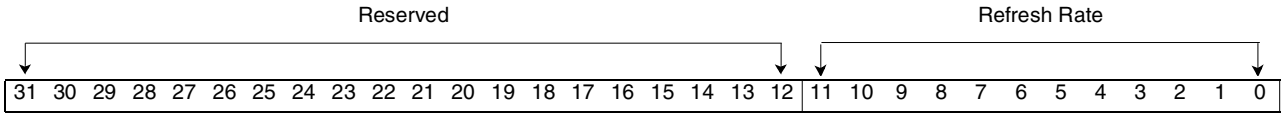
Type Read/Write

COMET Address XXXX 0928

PAKIT Address XXXX 09A8

Power On Value X'00000820'

Restrictions None

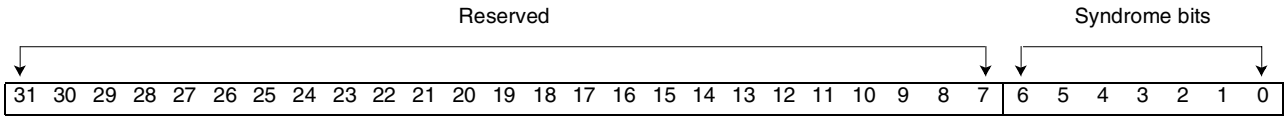


Bit(s)	Function	Description
31-12	Reserved	Reserved
11-0	Refresh Rate	The value of these bits multiplied by 7.5 ns gives the refresh rate. The POR value of X'820' yields a refresh rate of 15.6 ms.

5.8: COMET/PAKIT Syndrome Register

This register holds the syndrome bits that can be used to isolate the data or check bit in error when ECC is used. When parity is used, this register indicates which of the four bytes of the memory bus had a parity error.

Length	32 bits
Type	Read/Write
COMET Address	XXXX 092C
PAKIT Address	XXXX 09AC
Power On Value	X'00000000'
Restrictions	None



Bit(s)	Function	Description
31-7	Reserved	Reserved
6-0	Syndrome bits	When using ECC, a single bit error can be identified by matching the contents of this register to the corresponding bit in the table below. When using parity, only bits 3-0 are valid and are interpreted as follows: 0000 No parity error 0001 Parity error on bits 7-0 0010 Parity error on bits 15-8 0100 Parity error on bits 23-16 1000 Parity error on bits 31-24 Other Reserved

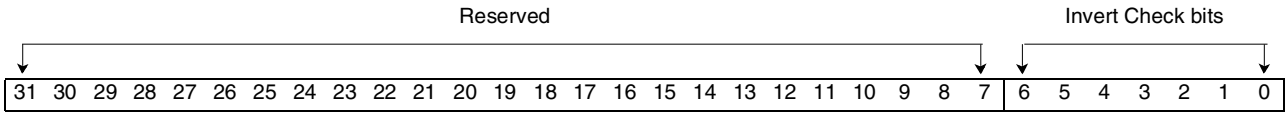
ECC Syndrome Bits

Bit in Error	Syndromes	Bit in Error	Syndromes
ECC(6)	’1000000’	ECC(5)	’0100000’
ECC(4)	’0010000’	ECC(3)	’0001000’
ECC(2)	’0000100’	ECC(1)	’0000010’
ECC(0)	’0000001’	N/A	N/A
DATA(31)	’0111000’	DATA(30)	’0110100’
DATA(29)	’0110010’	DATA(28)	’0101100’
DATA(27)	’1110000’	DATA(26)	’1101000’
DATA(25)	’1100100’	DATA(24)	’1100010’
DATA(23)	’0100101’	DATA(22)	’0010101’
DATA(21)	’0001101’	DATA(20)	’1100001’
DATA(19)	’0110001’	DATA(18)	’0101001’
DATA(17)	’0011001’	DATA(16)	’1000101’
DATA(15)	’1010001’	DATA(14)	’1001100’
DATA(13)	’1001010’	DATA(12)	’1000110’
DATA(11)	’1000011’	DATA(10)	’1011000’
DATA(09)	’1010100’	DATA(08)	’1010010’
DATA(07)	’0100011’	DATA(06)	’0010011’
DATA(05)	’0001011’	DATA(04)	’0000111’
DATA(03)	’0011010’	DATA(02)	’0100110’
DATA(01)	’0010110’	DATA(00)	’0001110’

5.9: COMET/PAKIT Checkbit Inversion Register

This register can be used for diagnostic purposes to invert the ECC/parity check bits that are written to memory.

Length 32 bits
Type Read/Write
COMET Address XXXX 0930
PAKIT Address XXXX 09B0
Power On Value X'00000000'
Restrictions None

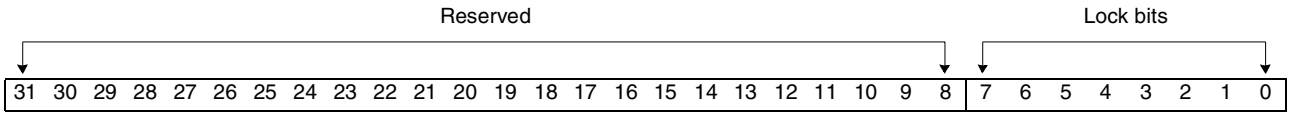


Bit(s)	Function	Description
31-7	Reserved	Reserved
6-0	Invert Check bits	Setting any of these bits inverts the corresponding check bit that is written to memory. Only bits 3-0 are valid when parity is used as a checking mechanism.

5.10: COMET/PAKIT Memory Controller Write Enable Register

This register must be written to a specific pattern before the Memory Control Register can be written.

Length 32 bits
Type Read/Write
COMET Address XXXX 0934
PAKIT Address XXXX 09B4
Power On Value X'000000B4'
Restrictions None



Bit(s)	Function	Description
31-8	Reserved	Reserved
7-0	Lock bits	This register must be written to a X'4' before the Memory Control Register can be written. This register will POR to X'4', but CRISCO code sets up the memory controller and clears this register back to X'0'.

5.11: COMET/PAKIT Memory Configuration Error Sense Register

This register can be read to help determine the source of a memory configuration error. The bits in this register reset automatically once the configuration error is resolved.

Length 32 bits

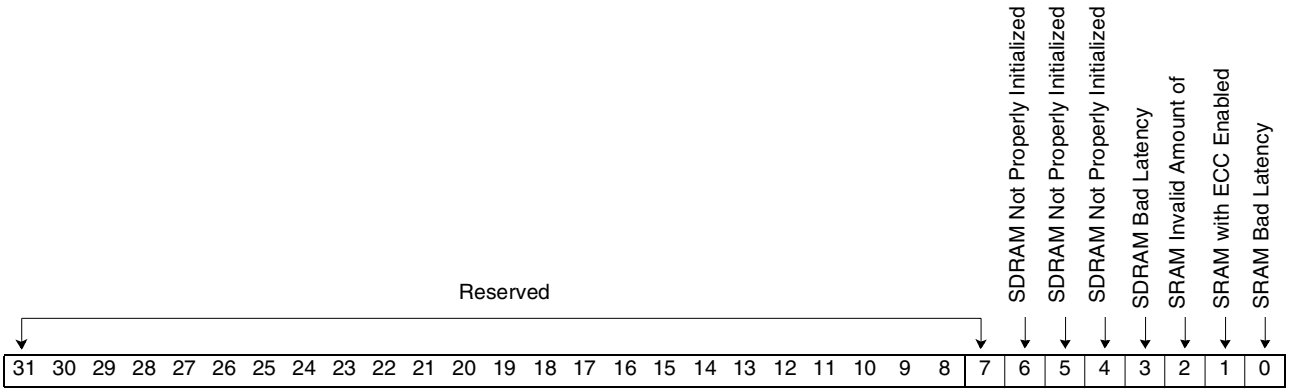
Type Read Only

COMET Address XXXX 0938

PAKIT Address XXXX 09B8

Power On Value X'00000000'

Restrictions None



Bit(s)	Function	Description
31-7	Reserved	Reserved
6	SDRAM Not Properly Initialized	An SDRAM access has been attempted without the SDRAMs being taken out of self refresh and/or the POR sequence being performed.
5	SDRAM Not Properly Initialized	Bits 6-4 of the control register have an invalid value.
4	SDRAM Not Properly Initialized	Bits 3-2 of the control register indicate less than 4M of memory.
3	SDRAM Bad Latency	Bits 11-10 of the control register indicate a latency of one or a reserved value.

Bit(s)	Function	Description
2	SRAM Invalid Amount of Memory	Bits 27, 13-12 and 3-1 of the control register indicate an invalid SRAM memory size. Acceptable sizes are: 4Mbit x18 unmuxed chip selects of sizes 1M, 2M, 4M. 8Mbit x18 unmuxed chip selects of sizes 2M, 4M, 8M. 16Mbit x18 unmuxed chip selects of sizes 4M, 8M, 16M. 32Mbit x18 unmuxed chip selects of sizes 8M, 16M, 32M. 4Mbit x18 muxed chip selects of sizes 2M, 4M, 8M. 8Mbit x18 muxed chip selects of sizes 4M, 8M, 16M. 16Mbit x18 muxed chip selects of sizes 4M, 16M, 32M. 32Mbit x18 muxed chip selects of sizes 8M, 32M, 64M. 4Mbit x36 unmuxed chip selects of sizes 1M, 2M. 8Mbit x36 unmuxed chip selects of sizes 1M, 2M, 4M. 16Mbit x36 unmuxed chip selects of sizes 2M, 4M, 8M. 32Mbit x36 unmuxed chip selects of sizes 4M, 8M, 16M. 4Mbit x36 muxed chip selects of sizes 1M, 2M, 4M. 8Mbit x36 muxed chip selects of sizes 2M, 4M, 8M. 4Mbit x36 muxed chip selects of sizes 4M, 8M, 16M. 8Mbit x36 muxed chip selects of sizes 8M, 16M, 32M.
1	SRAM with ECC Enabled	Bits 17, 14, and 9-8 of the control register indicate ECC is enabled with an SRAM configuration. This is invalid.
0	SRAM Bad Latency	Bits 11-10 of the control register indicate a latency of three or a reserved value.

Entity 6: ATM Virtual Memory Logic (VIMEM)

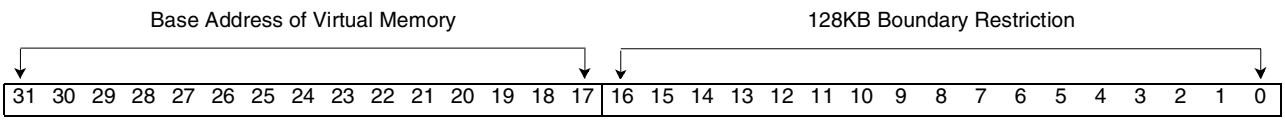
This entity is responsible for adjustment of all addresses provided to the memory control entities. All addresses can be categorized into three distinct types, based entirely upon the location of the requested address with respect to the three base registers defined in this entity. The three types of addresses are referred to as control, real packet, and virtual packet addresses.

All memory requests arriving on the Control Memory bus are handled as Control Memory accesses, and simply have the contents of the Control Memory Base Register subtracted from them before being passed on to the Control Memory Entity. When the processor accesses memory, the cache controller compares the requested address to the Real Packet Memory Base Register and if the address is less than the base register, the request is routed to the Control Memory bus; otherwise it is routed to the Packet Memory bus. All requests arriving on the Packet Memory bus are compared to the Virtual Memory Base Address Register. If the address of the request is less than the base register, the contents of the Real Packet Memory Base Register are subtracted from the address and this address is passed on to the Packet Memory Control Entity. If the requested address is greater than or equal to the base register, a more complex, but flexible scheme is used to determine the real address to provide to the Packet Memory Control Entity. For a detailed explanation of the virtual address generation scheme refer to *Virtual Memory Overview* on page 248 and the accompanying figures.

6.1: VIMEM Virtual Memory Base Address

This register defines the starting address of the virtual address space used to manage incoming and outgoing frames. Any time an access is made to Virtual Memory that falls within the defined bounds of Virtual Memory, the contents of this register are subtracted from the virtual address to derive the true offset into Virtual Memory. This true offset, along with the known length of all virtual buffers, allows the index of the specific virtual buffer to be derived by the Virtual Memory access hardware. This index can then be used to access the real buffer map associated with this virtual buffer.

Length	32 bits
Type	Read/Write
Address	XXXX 0D10
Power On Value	X'0040 0000'
Restrictions	The start of virtual address space must begin on a 128KB boundary. For this reason, the lowest 17 bits of this register are forced to '0' and are not implemented. Writes of any value to the low 17 bits of this register are ignored, and a read always returns '0' for the low 17 bits.



Bit(s)	Description
31-17	These bits contain the upper 15 bits of the base address of Virtual Memory.
16-0	These bits are forced to '0' because the Virtual Memory base address must start on a 128K byte boundary.

6.2: On-Chip Memory Base Address

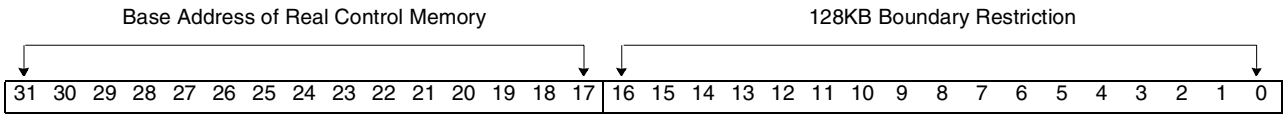
This register is used by various entities to generate the base address of the On-Chip Memory (OCM).

Length	17 bits
Type	Read/Write
Address	XXXX 0D9C
Power On Value	X'0010 0000'
Restrictions	The start of virtual address space must begin on a 128KB boundary. For this reason, the lowest 17 bits of this register are forced to '0' and are not implemented. Writes of any value to the low 17 bits of this register are ignored, and a read always returns '0' for the low 17 bits.

6.3: VIMEM Control Memory Base Address

This register defines the starting address of the Control Memory address space. Any time an access is made to Control Memory, the contents of this register are subtracted from the address before an access to memory occurs.

Length	32 bits
Type	Read/Write
Address	XXXX 0D14
Power On Value	X'0000 0000'
Restrictions	The start of real control address space must begin on a 128KB boundary. For this reason, the lowest 17 bits of this register are forced to '0' and are not implemented. Writes of any value to the low 17 bits of this register are ignored, and a read always returns '0' for the low 17 bits.

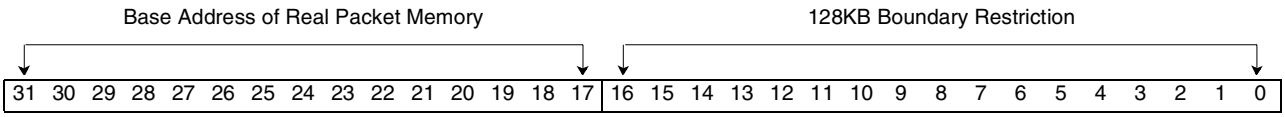


Bit(s)	Description
31-17	These bits contain the upper 15 bits of the base address of real Control Memory.
16-0	These bits are forced to '0' because the real Control Memory base address must start on a 128KB boundary.

6.4: VIMEM Packet Memory Base Address

This register defines the starting address of the Packet Memory address space. Any time an access is made to Packet Memory, the contents of this register are subtracted from the address before an access to memory occurs.

Length	32 bits
Type	Read/Write
Address	XXXX 0D18
Power On Value	X'0020 0000'
Restrictions	The start of real packet address space must begin on a 128KB boundary. For this reason, the lowest 17 bits of this register are forced to '0' and are not implemented. Writes of any value to the low 17 bits of this register are ignored, and a read always returns '0' for the low 17 bits. This register must also be set up before any of the Real Buffer Base Registers, or the Virtual Buffer Map Registers are written.



Bit(s0	Description
31-17	These bits contain the upper 15 bits of the base address of real Packet Memory.
16-0	These bits will be forced to '0' because the real Packet Memory base address must start on a 128KB boundary.

6.5: VIMEM Virtual Memory Total Bytes

This register defines the total number of bytes in the address space being allocated for Virtual Memory. The contents of this register, divided by the configured size of virtual buffers, yields the total number of virtual buffer indices that should be used to initialize POOLS. The value of the indices should range from this calculated value minus one, down to zero. If an address is determined to be above or equal to the Virtual Memory Base Register it is assumed to be a virtual access. If the virtual buffer index derived from the requested address indicates that the virtual buffer space being accessed is above the limit defined by this register, an error is generated.

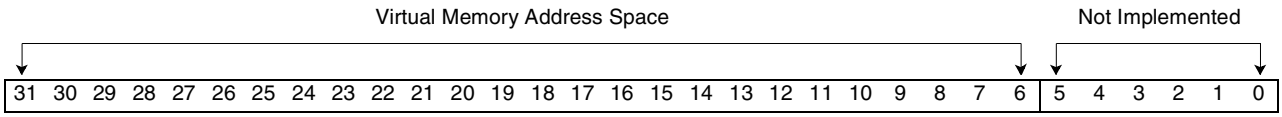
Length 32 bits

Type Read/Write

Address XXXX 0D0C

Power On Value X'0001 0000'

Restrictions The maximum value that should be set in this register is (65535 * virtual buffer size). For example, if 64-byte virtual buffers are configured, the maximum value that should be loaded into this register is X'3FFFC0'.



Bit(s)	Description
31-6	These bits contain the upper 26 bits of the total number of bytes of address space being reserved for Virtual Memory.
5-0	These bits are not implemented and are forced to '0' because the Virtual Memory block can only be allocated in increments of the current virtual buffer size (minimum size is 64 bytes).

6.6: VIMEM Virtual/Real Memory Buffer Size

This register defines the total number of bytes to be occupied by each of the virtual or real buffers as well as the spacing from one buffer to the next.

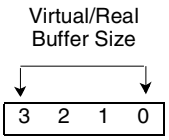
Length 4 bits

Type Read/Write

Address XXXX 0D04

Power On Value X'2'

Restrictions Care must be taken to set this register to a large enough value to contain the entire frame being sent as well as certain control information that the hardware stores in the buffer header. For example, if the maximum frame being sent or received is 1024 bytes long, then this register should be set to indicate 2048-byte frames to allow sufficient room for the buffer header information added by the hardware.



Bit(s)	Description
3-0	These bits contain the encoded four-bit value that defines the virtual/real buffer size. The encoding is as follows: 0000 64 bytes 0001 128 bytes 0010 256 bytes 0011 512 bytes 0100 1024 bytes 0101 2048 bytes 0110 4096 bytes 0111 8192 bytes 1000 16384 bytes 1001 32768 bytes 1010 65536 bytes 1011 131072 bytes 1100 -1111 Reserved

6.7: VIMEM Packet Memory Offset

This register contains the number that will be added by the VIMEM access logic to all accesses of real Packet Memory that occur. In a high performance configuration (separate control and packet store), this register should be written to all zeros to indicate that all accesses of real Packet Memory do not require any additional offset to be added. In a medium performance configuration (combined control and packet store), this register should be loaded with a value that indicates the logical partitioning between control and packet storage. If for instance, a single bank of 2 meg was configured and this register was loaded with X'00100000' (1 meg), then all accesses to real Packet Memory would be forced into the 1-meg to 2-meg range.

Length	32 bits
Type	Read/Write
Address	XXXX 0D3C
Power On Value:	X'0000 0000'
Restrictions	This register should only be loaded with a non-zero value if a medium performance configuration (combined control and packet store) exists. The value loaded must be between zero and the maximum of the total amount of memory in the single bank, and it must be on a 128KB boundary. Any time the value in this register is changed, the related base registers must be reloaded because the value loaded into them is affected by the contents of this register during the load operation. The related registers are the Virtual Buffer Map Base Address Register and all five real buffer base registers.

6.8: VIMEM Maximum Buffer Size

This register is used by the Virtual Memory logic to determine if an access to a virtual buffer falls into the region of the buffer that can be accessed. If a virtual buffer read or write accesses an offset in a virtual buffer that is greater than the contents of this register, the Virtual Memory logic can be configured to halt and generate an interrupt. The power up value of all ones causes this check to be disabled. This register is intended to provide the user with a means of providing additional protection to accesses of the virtual buffers. For example, if this register is loaded with X'FF8', all memory access up to and including the byte at address X'FFF' are allowed. Any access of offset X'1000' or above will cause an exception.

Length	17 bits
Type	Read/Write
Address	XXXX 0D34
Power On Value	X'1 FFF8'
Restrictions	All address logic based on this register only recognizes eight-byte words in memory. For this reason, the low three bits of this register are not implemented and are always forced to '0'.

6.9: VIMEM Access Control Register

The bits in this register control the configurable features of the Virtual Memory logic. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

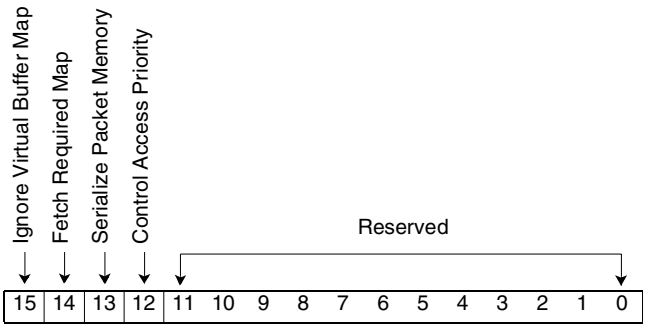
Length 16 bits

Type Read/Write

Address XXXX 0D80 and 84

Power On Value X'0'

Restrictions None



Bit(s)	Description
15	When set, this bit forces the Virtual Memory logic to ignore the virtual buffer map validity indication, and force all maps to appear valid.
14	When set, this bit forces the Virtual Memory logic to fetch the required map entry from storage on every new virtual access. If a Virtual Memory map is updated by the software for any reason, this bit should be toggled on and off after the map is updated and before any virtual access happens to ensure that the Virtual Memory logic is not using stale cached map segments. There is no hardware provided to make sure that the map entry required by the Virtual Memory logic is not contained in one of the BCACH lines. If is the responsibility of the software to ensure that all modified lines are flushed from the cache before the Virtual Memory logic needs them.
13	When set, this bit forces all accesses to Packet Memory to be serialized.
12	When set, this bit causes control accesses to always have priority over packet accesses in a single memory bank configuration. When reset, priority will toggle every time an access is initiated.
11-0	Reserved

6.10: VIMEM Access Status Register

This register contains information regarding the current status of the Virtual Memory logic mainly with respect to detected error access conditions. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

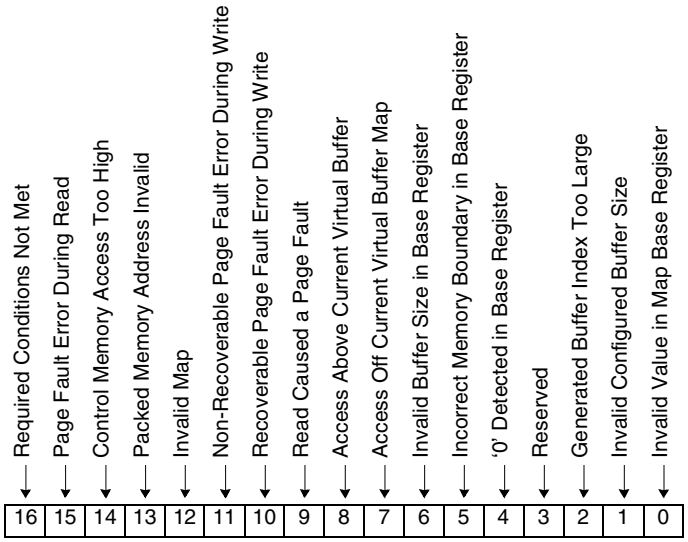
Length 17 bits

Type Read/Write

Address XXXX 0D60 and 64

Power On Value X'0000'

Restrictions None



Bit(s)	Description
16	When set, this bit indicates that the required conditions for the control, packet, and virtual base registers has not been satisfied. The required conditions are: control base < packet base < virtual base
15	When set, this bit indicates that the Virtual Memory logic has detected a page fault error when attempting to read memory. This indicates that no real buffer was available to map into the virtual address space when required. All virtual reads that fail during a page fault regardless of the requesting entity will cause this bit to be set. If the corresponding bit is reset in the lock register, the read operation will complete, but with invalid data.
14	When set, this bit indicates that a Control Memory access was detected that was above the value contained in the Packet Memory Offset Register for single bank configurations, or in a multiple bank configuration in which the high address bits 31 - 27 were not '0'.
13	When set, this bit indicates that a Packet Memory access of address zero was detected in single bank mode, or that a packet address was detected that contained an address out of range (high five bits non-zero).
12	When set, this bit indicates that the Virtual Memory logic has detected a Virtual Memory operation that attempted to access a map that was not marked as valid. A virtual buffer map is marked valid by the POOLs entity when the buffer is originally acquired, and is marked as invalid when the buffer is freed back to POOLs. Receiving this error indication typically means that the software is trying to use a buffer that has not been acquired through the normal means, or is trying to use a buffer that has already been freed, or that memory has been corrupted. The valid indication that is checked by the hardware is the value X'?656' in the first 16 bits of the eight-byte map entry being accessed. To determine the failing address, the memory control entity can be locked on this type of failure, and the information saved by the memory controller, along with the base registers in this entity can be used to determine which map was being accessed at the time of failure.

Bit(s)	Description
11	When set, this bit indicates that the Virtual Memory logic has detected a non-recoverable page fault error when attempting to write memory. This indicates that no real buffer was available to map into the virtual address space when required. All virtual writes that fail during a page fault, with the exception of BCACH and RAALL operations, cause this bit to be set.
10	When set, this bit indicates that the Virtual Memory logic has detected a recoverable page fault error when attempting to write memory. This indicates that no real buffer was available to map into the virtual address space when required. Operations from BCACH and RAALL cause this bit to be set instead of the non-recoverable bit because the software can recover from these failures. If a BCACH write to Virtual Memory fails in this manner, the packet header of the frame being updated is updated to indicate the failure. Software can check the field in the packet header to ensure that the DMA operation completed successfully. If such a packet is enqueued to CSKED, the packet header is checked and will prevent the frame from being passed on to the segmentation logic. When CSKED encounters a frame that has had this type of failure, there are several possible ways in which it can be configured (via the CSKED control register) to handle the situation. It can be configured to ignore the error and attempt to transmit the frame anyway (probably not a good way), or the buffer can be freed back to POOLs, or an event can be generated to allow the software to deal with the situation. If a RAALL write to Virtual Memory fails in this manner, the packet currently being received is dropped; it is up to the software to perform any recovery operations that are required.
9	When set, this bit indicates that the Virtual Memory logic has detected a read operation that caused a page fault. This is an invalid condition because the data required for a read operation should have been previously initialized by a write operation, so no page fault should ever occur on a read operation. If the corresponding bit in the lock register is reset, a page is mapped into the current virtual buffer segment and the data that previously was written in that page is returned. This bit can come on in several situations that are not really errors. In these cases, the associated interrupt and lock bits can be reset so that this error does not cause the adapter to halt normal operation. Several of these conditions are: When predictive fill is enabled, a read from the end of a buffer may cause a predictive read that crosses a virtual segment boundary and causes this bit to be set. If a small buffer (fits entirely in the cache) is copied from one IBM3206K0424 buffer to another IBM3206K0424 buffer, a subsequent read of the last bytes written causes this bit to be set if the cache hasn't been flushed between the write and the read, and the last write cycle did not write all four bytes, and the address that is being written/read is within the first 0x20 bytes of a virtual segment.
8	When set, this bit indicates that the Virtual Memory logic has detected an access of a virtual buffer that falls above the limit set by the buffer maximum size register.
7	When set, this bit indicates that the Virtual Memory logic has detected an access that does not fall in one of the currently mapped buffer segments based upon the currently-configured virtual buffer map size.
6	When set, this bit indicates that a virtual access has been detected that used a base register that had an invalid associated buffer size configured in the low order bits.
5	When set, this bit indicates that a virtual access has been detected that used a base register that was not on the correct memory boundary. For example, if a base register is set up to use 2K buffers, then the base register must be set up on a 2K boundary.
4	When set, this bit indicates that a virtual access has been detected that used a base register that contained a value of '0'.
3	Reserved.
2	When set, this bit indicates that the Virtual Memory logic has detected a memory access that resulted in the generation of a buffer index that was greater than the currently configured maximum derived from the VIMEM Virtual Memory total bytes register.
1	When set, this bit indicates that the currently configured size of buffers is invalid.
0	When set, this bit indicates that the map base register contains an invalid value. Two possible causes are that bits 5-2 are not '0' or bits 31-6 are '0'.

6.11: VIMEM Access Status Interrupt Enable Register

This register allows the user to enable interrupts for each of the conditions reported in the VIMEM Access Status Register. Each bit corresponds to the same bit in the status register and when set to '1' generates an interrupt to the processor if the condition is detected. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	17 bits
Type	Read/Write
Address	XXXX 0D68 and 6C
Power On Value	X'1FFFF'
Restrictions	None

6.12: VIMEM Memory Lock Enable Register

This register allows the user to selectively allow each of the conditions reported in the VIMEM Access Status Register to force a memory lock condition in the memory controller. Each bit corresponds to the same bit in the status register and when set to '1' causes a memory lock if the condition is detected. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	17 bits
Type	Read/Write
Address	XXXX 0D70 and 74
Power On Value	X'1FFFF'
Restrictions	None

6.13: VIMEM State Machine Current State

This register provides feedback to the user regarding the current status of the state machines in VIMEM. One use of this register is to make sure that the required initialization time has expired after loading the segment size register. This is accomplished by reading this register repeatedly until the initialization state machine is in the idle state.

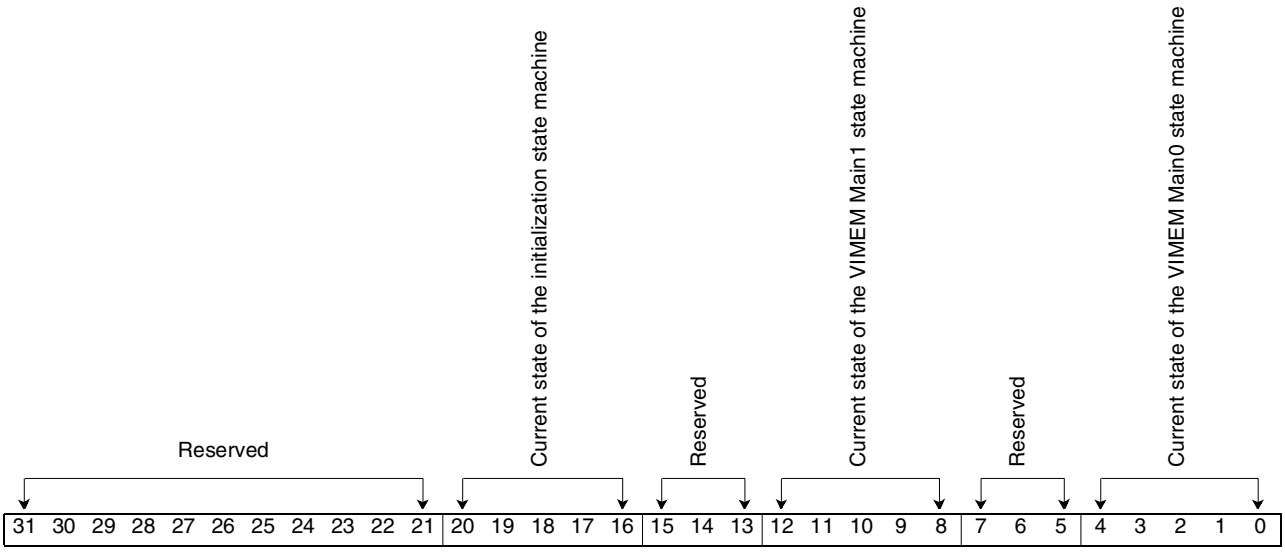
Length 32 bits

Type Read Only

Address XXXX 0D78

Power On Value X'1?0'

Restrictions None

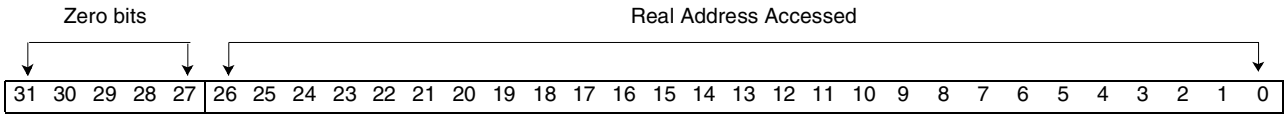


Bit(s)	Description
31-21	Reserved
20-16	These bits contain the current state of the initialization state machine. A value of "1----" indicates that the state machine is in the idle state.
15-13	Reserved
12-8	These bits contain the current state of the VIMEM Main1 state machine. A value of "00000" indicates that the state machine is in the idle state.
7-5	Reserved
4-0	These bits contain the current state of the VIMEM Main0 state machine. A value of "00000" indicates that the state machine is in the idle state.

6.14: VIMEM Last Processor Read Real Address

This register provides information to the user about the last read access of virtual Packet Memory by the processor. If a virtual address was accessed, this register contains the real address generated by the Virtual Memory logic that can be used to access the same location. This register is intended mainly as an aid in debugging to make virtual address translation easier. To perform the translation, the processor must read from the desired virtual address: after the read is complete, this register contains the real address that was accessed. The address contained in this register is an offset from the beginning of physical Packet Memory.

Length	32 bits
Type	Read Only
Address	XXXX 0D7C
Power On Value	X'0000 0000'
Restrictions	None

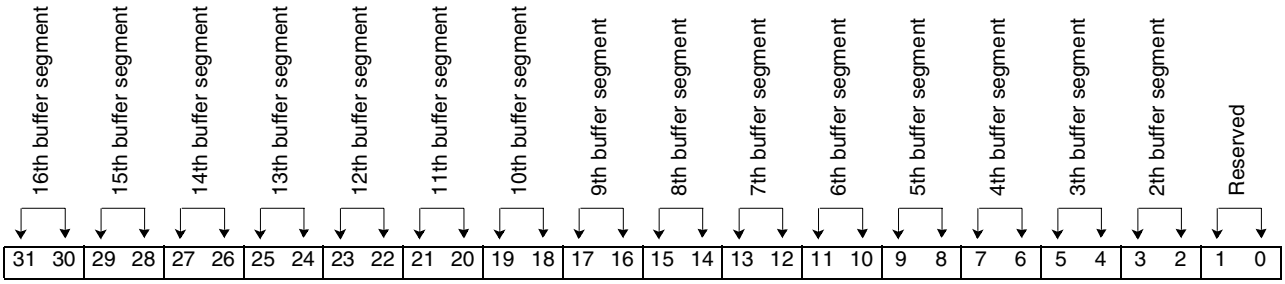


Bit(s)	Description
31-27	These bits always read as '0'.
26-0	After any read operation from the processor to Packet Memory, these bits will contain the real address that was accessed.

6.15: VIMEM Virtual Buffer Segment Size Register

This register, along with the lower four bits of the real buffer base registers, defines the size of the second through 16th real buffers that are concatenated to make up a virtual buffer. Two bits of this register are associated with each real buffer segment and indicate one out of four possible associations. The associative possibilities are shown in the bit table below. Every two bits defines the connection between a particular buffer segment and the real buffer base registers.

Length	32 bits
Type	Read/Write
Address	XXXX 0D00
Power On Value	X'0000 0000'
Restrictions	Care must be used when setting up this register to ensure that only values that correspond to real buffer sizes that POOLs has also been set up to provide are loaded. A write to this register causes the Virtual Memory logic to calculate the different real buffer boundaries within a virtual buffer. This calculation requires information from the real buffer base registers to determine the size of the different segments making up the virtual buffer. For this reason, it is required that this register be written after the real buffer base registers have been initialized. After writing this register, the software must wait at least 2 ms before accessing Virtual Memory.



Bit(s)	Description	Bit Association
31-30	Defines the 16th buffer segment's connection.	00 Associates this real buffer segment with Real Buffer Base Register 0 01 Associates this real buffer segment with Real Buffer Base Register 1 10 Associates this real buffer segment with Real Buffer Base Register 2 11 Associates this real buffer segment with Real Buffer Base Register 3
29-28	Defines the 15th buffer segment's connection.	
27-26	Defines the 14th buffer segment's connection.	
25-24	Defines the 13th buffer segment's connection.	
23-22	Defines the 12th buffer segment's connection.	
21-20	Defines the 11th buffer segment's connection.	
19-18	Defines the 10th buffer segment's connection.	
17-16	Defines the 9th buffer segment's connection.	
15-14	Defines the 8th buffer segment's connection.	
13-12	Defines the 7th buffer segment's connection.	
11-10	Defines the 6th buffer segment's connection.	
9-8	Defines the 5th buffer segment's connection.	
7-6	Defines the 4th buffer segment's connection.	
5-4	Defines the 3rd buffer segment's connection.	
3-2	Defines the 2nd buffer segment's connection.	
1-0	Reserved. The first real buffer is implicitly associated with the virtual buffer, these bits will always be read as '0'.	

6.16: VIMEM Buffer Map Base Address

This register contains the address in Packet Memory at which the buffer map table starts. The buffer map table consists of a variable number of eight-byte entries for each buffer that is allocated in the system. The first 16 bits of each eight-byte entry contains the POOL ID and various status flags associated with this buffer, thus this base register is used in both real and Virtual Memory modes.

In Virtual Memory mode, each of the three subsequent 16 bits contains an index which is associated with a buffer size base register using the Buffer Segment Limit Register. The index and buffer size base register are used to determine a real buffer address. If the map size is set to eight bytes, only one eight-byte entry is used for each buffer. If the map size is set to 16 bytes, two eight-byte entries are used for each buffer. If the map size is set to 32 bytes, four eight-byte entries are used for each buffer. If the map size is set to 64 bytes, five eight-byte entries are used for each buffer, and the remaining 24 bytes of the map are unused by the hardware.

Length 32 bits

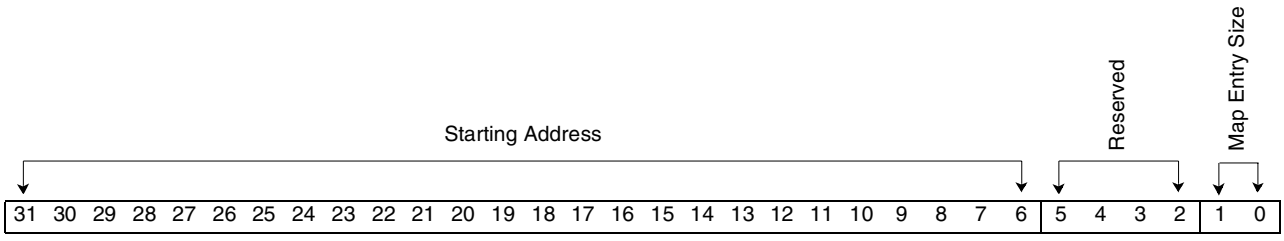
Type Read/Write

Address XXXX 0D08

Power On Value X'0020 0000' (This value is actually the power up contents of the Packet Memory Real Base Register added to the power up contents of this register (X'00000000') due to the automatic address adjustment explained below.)

Restrictions The base address for the buffer map must begin on a 64-byte boundary. When a base register is written, the hardware performs an automatic adjustment to the address using the contents of the Packet Memory Real Base Register, and the Packet Memory Offset Register. This results in the actual value being stored, not being the value that is written by the program. This is done to make the virtual accesses that use the base register execute more quickly.

The reverse adjustment is made when the read operation is performed, so that it appears to the program no different than a normal operation. Care must be taken, however, to ensure that both the Packet Memory Real Base Register and the Packet Memory Offset Register are set-up before any of the base registers are written. If the Packet Memory Base Register or the Packet Memory Offset Register are changed, Packet Memory should not be accessed until all the base registers have been written again.



Bit(s)	Description
31-6	Defines the starting address of the buffer map
5-2	Reserved, should be written with '0'

Bit(s)	Description
1-0	Defines the size of each map entry: 00 8 bytes 01 16 bytes 10 32 bytes 11 64 bytes

6.17: VIMEM Real Buffer Base Addresses

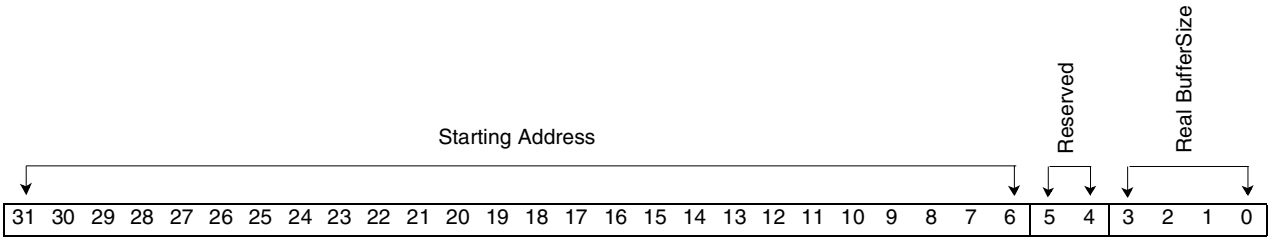
These registers contain the address in Packet Memory at which a block of memory begins that is used to provide a given size buffer. In general, the block allocated must be large enough to contain as many buffers as will be freed to POOLS on initialization. However, for Real Buffer Base 4, the size of the block reserved must be large enough so that one buffer is available for each of the virtual buffers freed to POOLS. These buffers must not be freed to POOLS because they are implicitly used as the first real buffer segment for each of the virtual buffers. If a given base register (and associated buffer size) is not used, the low four bits of the register should be set to X'F' to ensure that accesses of this buffer size are detected and flagged as an error. When using real memory mode (controlled in POOLS), all of these base registers are unused with the exception of base register zero, which contains the base address for all real memory buffers. In real mode, the low four bits of base register zero are of no significance. The size of the real buffers is controlled through the Buffer Size Register.

Buffer Size	0	1	2	3	4 (implicit)
Length	32 bits	32 bits	32 bits	32 bits	32 bits
Type	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Address	XXXX 0D20	XXXX 0D24	XXXX 0D28	XXXX 0D2C	XXXX 0D30
Power on Value	X'0020 000F'	X'0020 000F'	X'0020 000F'	X'0020 000F'	X'0020 000F'

Restrictions

The base address for any given buffer size must begin on a boundary that is equal to the buffer size. For example, the base address for 128-byte buffers must be on a 128-byte boundary, and the base address for 4096-byte buffers must be on a 4096-byte boundary.

When a base register is written, the hardware performs an automatic adjustment to the address using the contents of the Packet Memory real base register and the Packet Memory offset register. This results in the actual value being stored, not being the value that is written by the program. This is done to make the memory accesses that use the base register execute quicker. The reverse adjustment is made when the read operation is performed, so that it appears to the program no different than a normal operation. Care must be taken however to ensure that the Packet Memory Real Base Register and the Packet Memory Offset Register are set-up before any of the base registers are written. If the Packet Memory Base Register or the Packet Memory Offset Register is changed, Packet Memory should not be accessed until all the base registers have been written again. The power on value of these registers is actually the power on value of the Packet Memory Real Base Register added to the contents of the Packet Memory Offset Register added to the original contents of these registers (X'0000000F').



Bit(s)	Description
31-6	Defines the starting address in Packet Memory of the memory block used to provide real buffers of defined size
5-4	Reserved (User should write zeros and ignore read value)
3-0	Defines the size of the real buffers in this block of memory with the following encoding: 0000 64 bytes 0001 128 bytes 0010 256 bytes 0011 512 bytes 0100 1024 bytes 0101 2048 bytes 0110 4096 bytes 0111 8192 bytes 1000 16384 bytes 1001 32768 bytes 1010 65536 bytes 1011 131072 bytes 1100 -1111 Reserved

Entity 7: ATM Packet/Control Memory Arbitration Logic (ARBIT)

This section contains descriptions of the registers used by the arbiter logic.

7.1: ARBIT Control Priority Resolution Register High

The bits in this register define the priority of requesting entities to Control Memory.

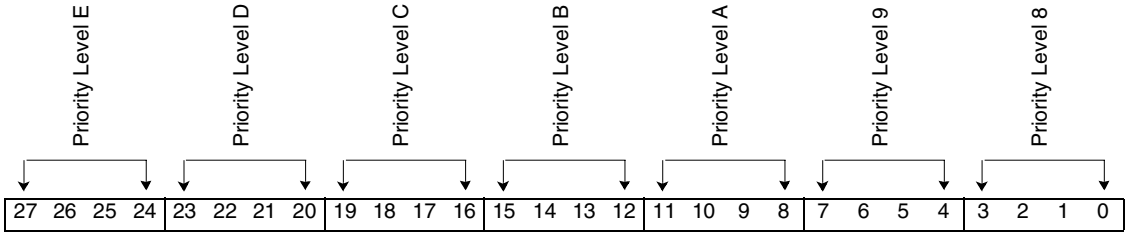
Length 28 bits

Type Read/Write

Address XXXX 0E00

Restrictions None

Power On Value X'EDC BA98'



Bit(s)	Description
27-24	The value loaded into these bits defines which entity will be requesting at priority level E (lowest priority). Value encoding is: F: Reserved 7: SEGBF E: CHKSM 6: TXLCD D: PCORE LO 5: RXLCD C: BCACH LO 4: GPDMA B: POOLS LO 3: DMAQS A: CSKED 2: PCORE HI 9: RXXLT 1: BCACH HI 8: RXQUE 0: POOLS HI
23-20	The value loaded into these bits defines which entity will request at priority level D. Value encoding is as listed in the description of bits 27-24.
19-16	The value loaded into these bits defines which entity will request at priority level C. Value encoding is as listed in the description of bits 27-24.
15-12	The value loaded into these bits defines which entity will request at priority level B.Value encoding is as listed in the description of bits 27-24.
11-8	The value loaded into these bits defines which entity will request at priority level A. Value encoding is as listed in the description of bits 27-24.
7-4	The value loaded into these bits defines which entity will request at priority level 9. Value encoding is as listed in the description of bits 27-24.
3-0	The value loaded into these bits defines which entity will request at priority level 8. Value encoding is as listed in the description of bits 27-24.

7.2: ARBIT Control Priority Resolution Register Low

The bits in this register define the priority of requesting entities to Control Memory.

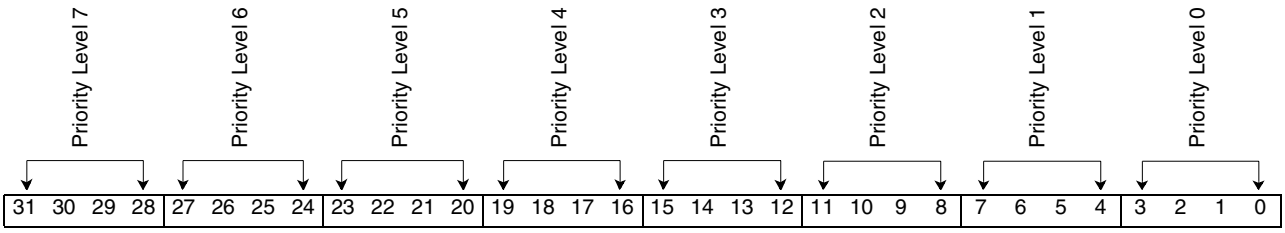
Length 32 bits

Type Read/Write

Address XXXX 0E04

Restrictions None

Power On Value X'7654 3210'



Bit(s)	Description
31-28	The value loaded into these bits defines which entity will request at priority level 7. Value encoding is: F: Reserved 7: SEGBF E: CHKSM 6: TXLCD D: PCORE LO 5: RXLCD C: BCACH LO 4: GPDMA B: POOLS LO 3: DMAQS A: CSKED 2: PCORE HI 9: RXXLT 1: BCACH HI 8: RXQUE 0: POOLS HI
27-24	The value loaded into these bits defines which entity will request at priority level 6. For value encoding, see the description of bits 31-28.
23-20	The value loaded into these bits defines which entity will request at priority level 5. For value encoding, see the description of bits 31-28.
19-16	The value loaded into these bits defines which entity will request at priority level 4. For value encoding, see the description of bits 31-28.
15-12	The value loaded into these bits defines which entity will request at priority level 3. For value encoding, see the description of bits 31-28.
11-8	The value loaded into these bits defines which entity will request at priority level 2. For value encoding, see the description of bits 31-28.
7-4	The value loaded into these bits defines which entity will request at priority level 1. For value encoding, see the description of bits 31-28.
3-0	The value loaded into these bits defines which entity will request at priority level 0 (highest priority). For value encoding, see the description of bits 31-28.

7.3: ARBIT Control Error Mask Register

The bits in this register control whether ARBIT detected error conditions on an entity's interface will lock the Control Memory subsystem. Bits in this register also control the locking of the Control Memory subsystem based on Control Memory, Packet Memory, Virtual Memory, and BCACH detected error conditions. Resetting the appropriate bit will force errors from that source to be ignored.

Length 20 bits

Type Clear/Set

Address XXXX 0E08 and 0C

Power On Value X'FFFFF'

Restrictions None

Bit(s)	Bit Name/Function
19	Reserved
18	ARBIT detected packet errors
17	PCORE errors
16	POOLS errors
15	BCACH errors
14	VIMEM errors
13	PAKTT errors
12	COMET errors
11	CHKSM
10	PCORE
9	BCACH
8	POOLS
7	CSKED
6	RXXLT
5	RXQUE
4	SEGBF
3	TXLCD
2	RXLCD
1	GPDMA
0	DMAQS

7.4: ARBIT Control Error Source Register

The bits in this register provide feedback to indicate the source of errors that have been detected by the memory subsystem.

Length 20 bits

Type Clear/Set

Address XXXX 0E18 and 1C

Power On Value X'00000'

Restrictions Bits 17 through 12 are driven from external entities and cannot be set or reset in this register. They must be set or reset in the entity of origin.

Bit(s)	Bit Name/Function
19	Reserved
18	ARBIT detected packet errors
17	PCORE errors
16	POOLS errors
15	BCACH errors
14	VIMEM errors
13	PAKTT errors
12	COMET errors
11	CHKSM
10	PCORE
9	BCACH
8	POOLS
7	CSKED
6	RXXLT
5	RXQUE
4	SEGBF
3	TXLCD
2	RXLCD
1	GPDMA
0	DMAQS

7.5: ARBIT Control Winner Register

The bits in this register indicate which entity currently owns Control Memory.

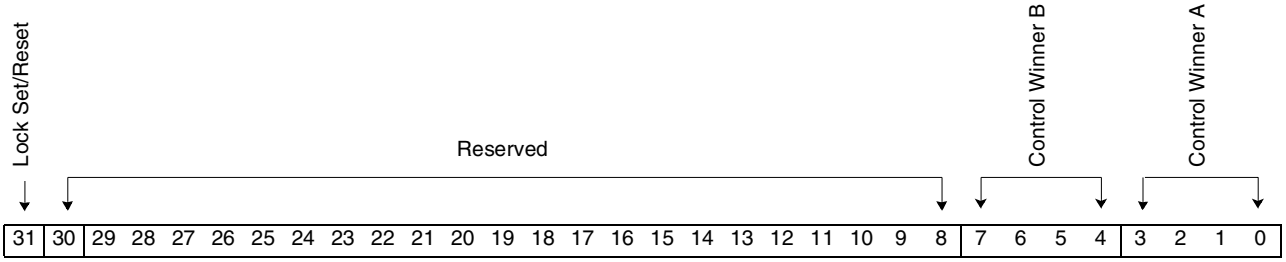
Length 32 bits

Type Read

Address XXXX 0E2C

Power On Value X'F'

Restrictions None

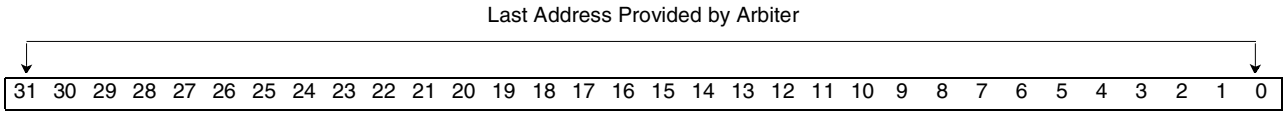


Bit(s)	Description
31	For performance reasons, two sets of operational latches (bank A and bank B) exist in the arbiter for Control Memory. When set, this bit indicates that the B latches are active, and when reset indicates that the A latches are active. When this bit is set and memory is locked, bits 7-4 of this register contain a value that indicates the entity that most recently was accessing memory. If this bit is reset and memory is locked, bits 3-0 of this register contain a value that indicates the entity that was accessing memory most recently.
30-8	Reserved. Will read '0'.
7-4	Control winner B.
3-0	Control winner A. F: Reserved 7: SEGBF E: CHKSM 6: TXLCD D: PCORE LO 5: RXLCD C: BCACH LO 4: GPDMA B: POOLS LO 3: DMAQS A: CSKED 2: PCORE HI 9: RXXLT 1: BCACH HI 8: RXQUE 0: POOLS HI

7.6: ARBIT Control Address Register A

If latch bank A is active, the bits in this register indicate the last address that was used to access Control Memory.

Length	32 bits
Type	Read
Address	XXXX 0E10
Power On Value	X'0000 0000'
Restrictions	None

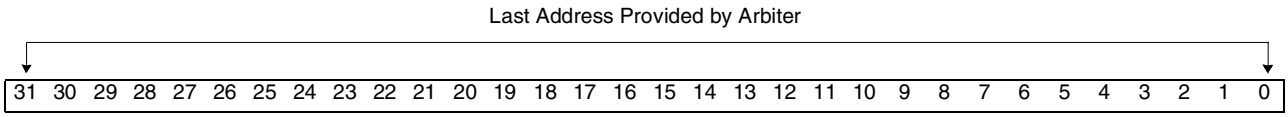


Bit(s)	Description
31-0	These bits contain the last address provided by the arbiter to the Control Memory controller.

7.7: ARBIT Control Address Register B

If latch bank B is active, the bits in this register indicate the last address that was used to access Control Memory.

Length	32 bits
Type	Read
Address	XXXX 0E20
Power On Value	X'0000 0000'
Restrictions	None

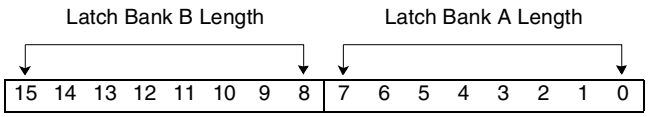


Bit(s)	Description
31-0	These bits contain the last address provided by the arbiter to the Control Memory controller.

7.8: ARBIT Control Length Register

The bits in this register indicate the last length that was used to access Control Memory.

Length 16 bits
Type Read
Address XXXX 0E14
Power On Value X'0000 0000'
Restrictions None



Bit(s)	Description
15-8	These bits contain the length used to access Control Memory through latch bank B.
7-0	These bits contain the length used to access Control Memory through latch bank A.

7.9: ARBIT Control Lock Entity Enable Register

The value programmed in this register controls what entity, if any, has access to Packet Memory immediately after memory has locked. This register powers up to a value that will not allow any entity to access memory after a lock condition until the lock condition has been properly cleared.

Length 4 bits

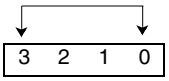
Type Read/Write

Address XXXX 0E28

Power On Value X'F'

Restrictions None

Bit Map Value



Bit(s)	Description
3-0	The value in these bits map to the following entities: F: Reserved 7: SEGBF E: CHKSM 6: TXLCD D: PCORE LO 5: RXLCD C: BCACH LO 4: GPDMA B: POOLS LO 3: DMAQS A: CSKED 2: PCORE HI 9: RXXLT 1: BCACH HI 8: RXQUE 0: POOLS HI

7.10: ARBIT Control Config Register

The bits in this register control the operation of the Control Memory arbiter.

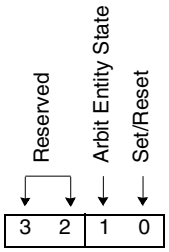
Length 4 bits

Type Clear/Set

Address XXXX 0E38 and 3C

Power On Value X'0'

Restrictions None



Bit(s)	Description
3-2	Reserved
1	This bit controls the arbit entity state debug mux. When set, the incoming entity requests and outgoing acknowledges are routed to the entity state pins. When reset, the internal state information is routed to the entity state pins.
0	When set, this bit forces all operations to Control Memory to be serialized. An operation from one entity must be entirely complete before an operation from another entity will be started. When reset, if the memory operation in process can be overlapped, a second operation will be started before the first operation is complete.

7.11: ARBIT Packet Priority Resolution Register High

The bits in this register define the priority of requesting entities to Packet Memory.

Length 28 bits

Type Read/Write

Address XXXX 0E80

Power On Value X'EDC BA98'

Restrictions None

Bit(s)	Description
27-24	The value loads into these bits defines which entity will request at priority level E (lowest priority). Value encoding is: F: Reserved 7: SEGBF E: CHKSM 6: Reserved D: PCORE LO 5: RXAAL C: BCACH LO 4: GPDMA B: POOLS LO 3: DMAQS A: CSKED 2: PCORE HI 9: Reserved 1: BCACH HI 8: RXQUE 0: POOLS HI
23-20	The value loaded into these bits defines which entity will request at priority level D. For value encoding, see the description of bits 27-24.
19-16	The value loaded into these bits defines which entity will request at priority level C. For value encoding, see the description of bits 27-24.
15-12	The value loaded into these bits defines which entity will request at priority level B. For value encoding, see the description of bits 27-24.
11-8	The value loaded into these bits defines which entity will request at priority level A. For value encoding, see the description of bits 27-24.
7-4	The value loaded into these bits defines which entity will request at priority level 9. For value encoding, see the description of bits 27-24.
3-0	The value loaded into these bits defines which entity will request at priority level 8. For value encoding, see the description of bits 27-24.

7.12: ARBIT Packet Priority Resolution Register Low

The bits in this register define the priority of requesting entities to Packet Memory.

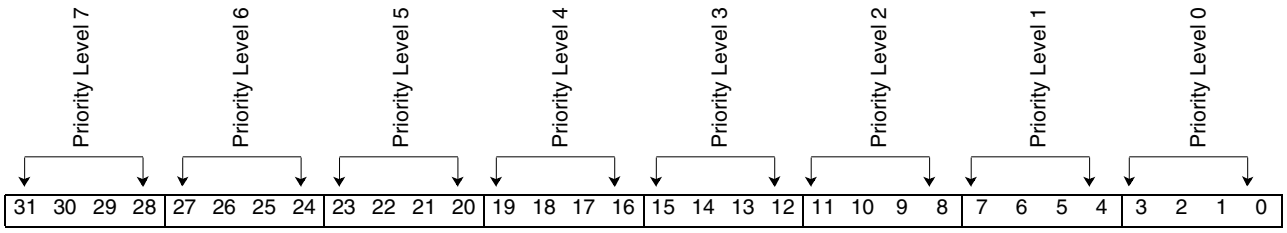
Length 32 bits

Type Read/Write

Address XXXX 0E84

Restrictions None

Power On Value X'7654 3210'



Bit(s)	Description
31-28	The value loaded into these bits define which entity will be requesting at priority level 7. Value encoding is: F: Reserved 7: SEGBF E: CHKSM 6: Reserved D: PCORE LO 5: RXAAL C: BCACH LO 4: GPDMA B: POOLS LO 3: DMAQS A: CSKED 2: PCORE HI 9: Reserved 1: BCACH HI 8: RXQUE 0: POOLS HI
27-24	The value loaded into these bits defines which entity will request at priority level 6. For value encoding, see the description of bits 31-28 above.
23-20	The value loaded into these bits defines which entity will request at priority level 5. For value encoding, see the description of bits 31-28 above.
19-16	The value loaded into these bits defines which entity will request at priority level 4. For value encoding, see the description of bits 31-28 above.
15-12	The value loaded into these bits defines which entity will request at priority level 3. For value encoding, see the description of bits 31-28 above.
11-8	The value loaded into these bits defines which entity will request at priority level 2. For value encoding, see the description of bits 31-28 above.
7-4	The value loaded into these bits defines which entity will request at priority level 1. For value encoding, see the description of bits 31-28 above.
3-0	The value loaded into these bits defines which entity will request at priority level 0 (highest priority). For value encoding, see the description of bits 31-28 above.

7.13: ARBIT Packet Entity Error Mask Register

The bits in this register control whether ARBIT detected error conditions on an entity's interface will lock the Packet Memory subsystem. Bits in this register also control the locking of the Packet Memory subsystem based on Control Memory, Packet Memory, Virtual Memory, and BCACHE detected error conditions. Resetting the appropriate bit will force errors from that source to be ignored.

Length 18 bits

Type Clear/Set

Address XXXX 0E88 and 8C

Power On Value X'FFFFF'

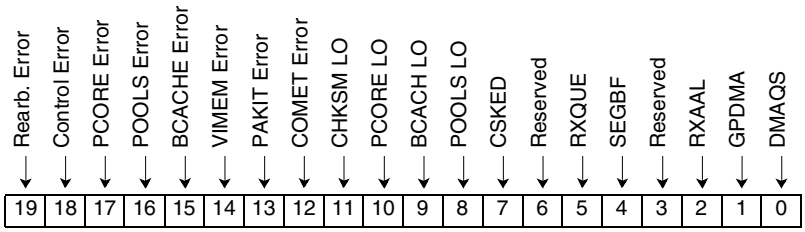
Restrictions None

Bit(s)	Bit Name/Function
19	Re-arbitration failure
18	ARBIT detected control errors
17	PCORE error
16	POOLS error
15	BCACH error
14	VIMEM error
13	PAKTT error
12	COMET error
11	CHKSM LO
10	PCORE LO
9	BCACH LO
8	POOLS LO
7	CSKED
6	Reserved
5	RXQUE
4	SEGBF
3	Reserved
2	RXAAL
1	GPDMA
0	DMAQS

7.14: ARBIT Packet Error Source Register

The bits in this register provide feedback to indicate the source of errors that have been detected by the memory subsystem.

Length	20 bits
Type	Clear/Set
Address	XXXX 0E98 and 9C
Power On Value	X'00000'
Restrictions	Bits 17, 16, and 11 through 14 are driven from external entities and can not be set/reset in this register. They must be set/reset in the entity of origin.



Bit(s)	Bit Name/Function
19	Rearbitration failure. Rearbitration detected while already handling rearbitration condition. This condition would indicate that the priorities programmed in the priority resolution logic were incorrectly programmed and POOLS HIGH was not given the highest priority.
18	ARBIT detected control errors
17	PCORE error
16	POOLS error
15	BCACH error
14	VIMEM error
13	PAKTT error
12	COMET error
11	CHKSM LO
10	PCORE LO
9	BCACH LO
8	POOLS LO
7	CSKED
6	Reserved
5	RXQUE
4	SEGBF
3	Reserved
2	RXAAL
1	GPDMA
0	DMAQS

7.15: ARBIT Packet Winner Register

The bits in this register indicate which entity currently owns Packet Memory.

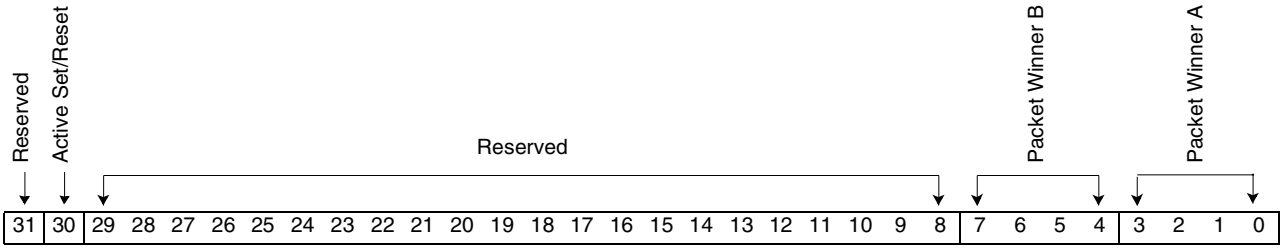
Length 32 bits

Type Read

Address XXXX 0EAC

Power On Value X'F'

Restrictions None

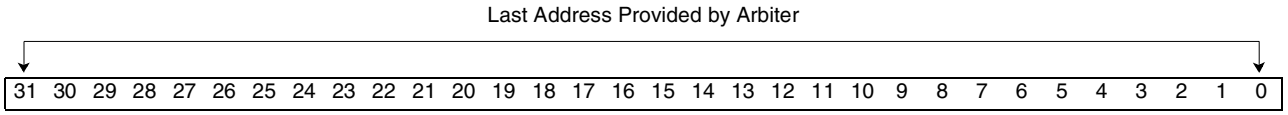


Bit(s)	Description
31	Reserved.
30	For performance reasons, two sets of operational latches (bank A and bank B) exist in the arbiter for Packet Memory. When set, this bit indicates that the B latches are active, and when reset it indicates that the A latches are active. When this bit is set and memory is locked, bits 7-4 of this register contain a value that indicates the entity that most recently was accessing memory. If this bit is reset and memory is locked, bits 3-0 of this register contain a value that indicates the entity that was accessing memory most recently.
29-8	Reserved. Will read '0'.
7-4	Packet winner B.
3-0	Packet winner A. Value encoding is: F: Reserved 7: SEGBF E: CHKSM 6: Reserved D: PCORE LO 5: RXAAL C: BCACH LO 4: GPDMA B: POOLS LO 3: DMAQS A: CSKED 2: PCORE HI 9: Reserved 1: BCACH HI 8: RXQUE 0: POOLS HI

7.16: ARBIT Packet Address Register A

If latch bank A is active, the bits in this register indicate the last address that was used to access Packet Memory.

Length 32 bits
Type Read
Address XXXX 0E90
Power On Value X'0000 0000'
Restrictions None

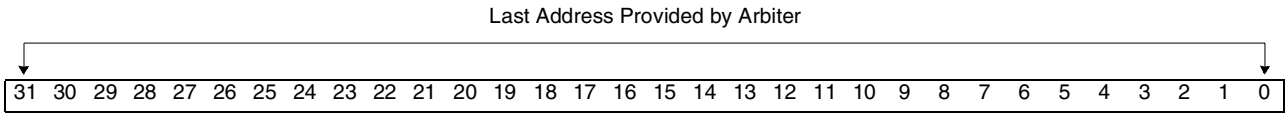


Bit(s)	Description
31-0	These bits contain the last address provided by the arbiter to the Packet Memory controller.

7.17: ARBIT Packet Address Register B

If latch bank B is active, the bits in this register indicate the last address that was used to access packet memory.

Length 32 bits
Type Read
Address XXXX 0EA0
Power On Value X'0000 0000'
Restrictions None

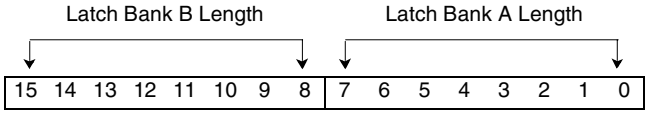


Bit(s)	Description
31-0	These bits contain the last address provided by the arbiter to the Packet Memory controller.

7.18: ARBIT Packet Length Register

The bits in this register indicate the last length that was used to access Packet Memory.

Length	16 bits
Type	Read
Address	XXXX 0E94
Power On Value	X'0000 0000'
Restrictions	None



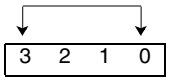
Bit(s)	Description
15-8	These bits contain the length used to access Packet Memory through Latch Bank B
7-0	These bits contain the length used to access Packet Memory through Latch Bank A

7.19: ARBIT Packet Lock Entity Enable Register

The value programmed in this register controls what entity, if any, has access to Packet Memory immediately after memory has locked. This register powers up to a value that will not allow any entity to access memory after a lock condition until the lock condition has been properly cleared.

Length	4 bits
Type	Read/Write
Address	XXXX 0EA8
Power On Value	X'F'
Restrictions	None

Bit Map Value

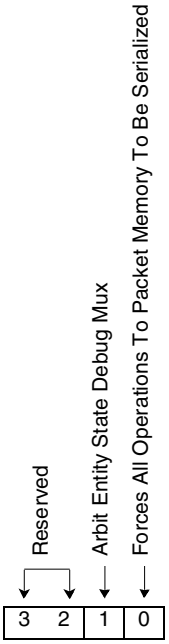


Bit(s)	Description
3-0	The value in these bits map to the following entities: Value encoding is: F: Reserved 7: SEGBF E: CHKSM 6: Reserved D: PCORE LO 5: RXAAL C: BCACH LO 4: GPDMA B: POOLS LO 3: DMAQS A: CSKED 2: PCORE HI 9: Reserved 1: BCACH HI 8: RXQUE 0: POOLS HI

7.20: ARBIT Packet Config Register

The bits in this register control the operation of the Packet Memory arbiter.

Length	4 bits
Type	Clear/Set
Address	XXXX 0EB8 and BC
Power On Value	X'0'
Restrictions	None

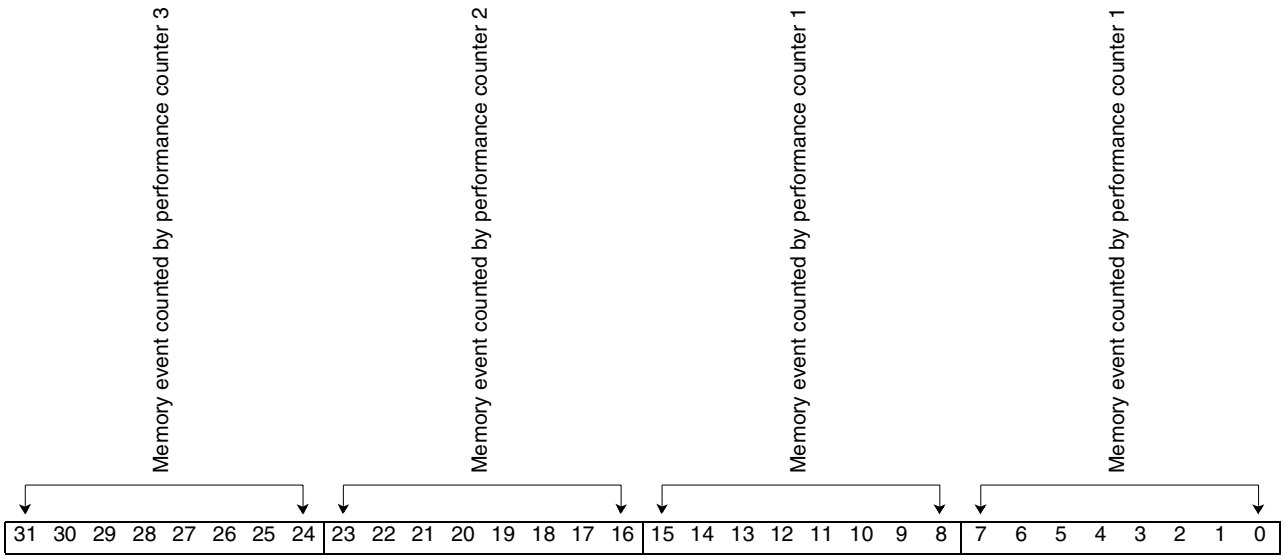


Bit(s)	Description
3-2	Reserved
1	This bit controls the arbit entity state debug mux. When set, the incoming entity requests and outgoing acknowledges are routed to the entity state pins. When reset, the internal state information is routed to the entity state pins.
0	When set, this bit forces all operations to Packet Memory to be serialized. An operation from one entity must be entirely complete before an operation from another entity will be started. When reset, if the memory operation in process can be overlapped, a second operation will be started before the first operation is complete.

7.21: ARBIT Performance Counter Control

The bits in this register determine what events are counted by the memory performance counters. This 32-bit register is divided into four 8-bit values, one value for each of the counters. The eight bits determine what memory event is counted by the associated counter.

Length	32 bits
Type	Read/Write
Address	XXXX 0EB8 and BC
Power On Value	X'9F1E 8000'
Restrictions	None

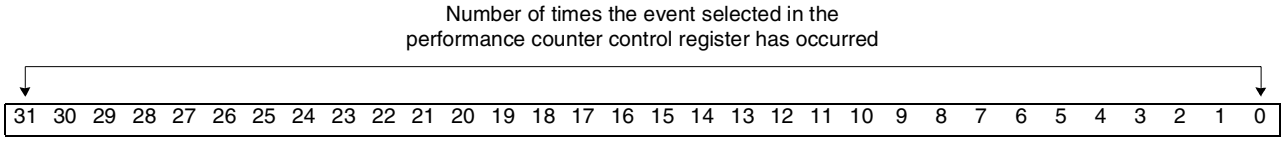


Bit(s)	Description
31-24	The value loaded into these bits defines what memory event is counted by performance counter 3. These bits are defined as follows: Bit 7: When reset, Control Memory events are counted, when set, Packet Memory events are counted by the associated counter. Bit 6: Reset the associated counter to 0. This bit will be reset by the hardware, during the same cycle that the counter is being reset. Bit 5: Reserved Bits 4-0: Cycle type, encoded as: 00000 Any Request 00001 Any Request between 0 and 4 bytes 00010 Any Request between 5 and 8 bytes 00011 Any Request between 9 and 16 bytes 00100 Any Request between 17 and 32 bytes 00101 Any Request between 33 and 64 bytes 00110 Any Request between 65 and 128 bytes 00111 Reserved 01000 Any Read Request 01001 Any Read Request between 0 and 4 bytes 01010 Any Read Request between 5 and 8 bytes 01011 Any Read Request between 9 and 16 bytes 01100 Any Read Request between 17 and 32 bytes 01101 Any Read Request between 33 and 64 bytes 01110 Any Read Request between 65 and 128 bytes 01111 Reserved 10000 Any Write Request 10001 Any Write Request between 0 and 4 bytes 10010 Any Write Request between 5 and 8 bytes 10011 Any Write Request between 9 and 16 bytes 10100 Any Write Request between 17 and 32 bytes 10101 Any Write Request between 33 and 64 bytes 10110 Any Write Request between 65 and 128 bytes 10111 Reserved 11000 Read op latency 11001 Write op latency 11010 Reserved 11011 Reserved 11100 Reserved 11101 Reserved 11110 Hold Current Count 11111 Every Cycle
23-16	The value loaded into these bits define what memory event is counted by performance counter 2.
15-8	The value loaded into these bits define what memory event is counted by performance counter 1.
7-0	The value loaded into these bits define what memory event is counted by performance counter 0.

7.22: Arbit Memory Performance Counter

These registers count memory events as defined in the ARBIT performance counter control register.

Length	32 bits	
Type	Read/Write	
Address	Counter 0	XXXX 0EC0
	Counter 1	XXXX 0EC4
	Counter 2	XXXX 0EC8
	Counter 3	XXXX 0ECC
Power On Value	X'0000 0000'	
Restrictions	None	



Bit(s)	Description
31-0	These bits count the number of times that the event selected in the performance counter control register has occurred.

Entity 8: The Bus DRAM Cache Controller (BCACH)

This entity provides the caching function for data transfers on the Control Processor bus. The array is organized in four logically separate cache lines, any of which can be used for processor accesses or master/slave DMA accesses. The cache is accessible on byte boundaries on the Control Processor side; access of this entity to COMET is performed on 64-bit (word) boundaries. The address tags of each of the four 32-byte cache lines are compared to the requesting address to select the bank to be used to satisfy the Control Processor bus operation.

Streaming accesses of the cache use a predictive look-ahead scheme to fill the cache for read operations from Packet Memory. Under normal conditions, a single cache miss will be expected at the start of each DMA read operation. This cache miss will initiate a read operation from Packet Memory to fetch the requested data and enough additional data to fill the remainder of the cache line. If the requested data is in the last N bytes (N is programmable via the BCACH control register) of the cache line, the read operation to COMET will be extended to fill the next cache line with sequential data as well. This same programmable value is used to determine when to initiate the next sequential cache line fill operation during a DMA read operation. During non-aligned write operations to Packet Memory, BCACH will perform read/modify/write cycles to PAKIT.

Processor accesses operate without predictive caching. When a cache miss occurs, a COMET read operation will be initiated to fetch the 32-byte block of data that contains the requested data. The data read from COMET will be loaded into the ‘Least Recently Used’ cache line.

This section contains descriptions of the registers used by the Bus Cache logic.

8.1: BCACH Control Register

The bits in this register control the various functions provided by the cache logic. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

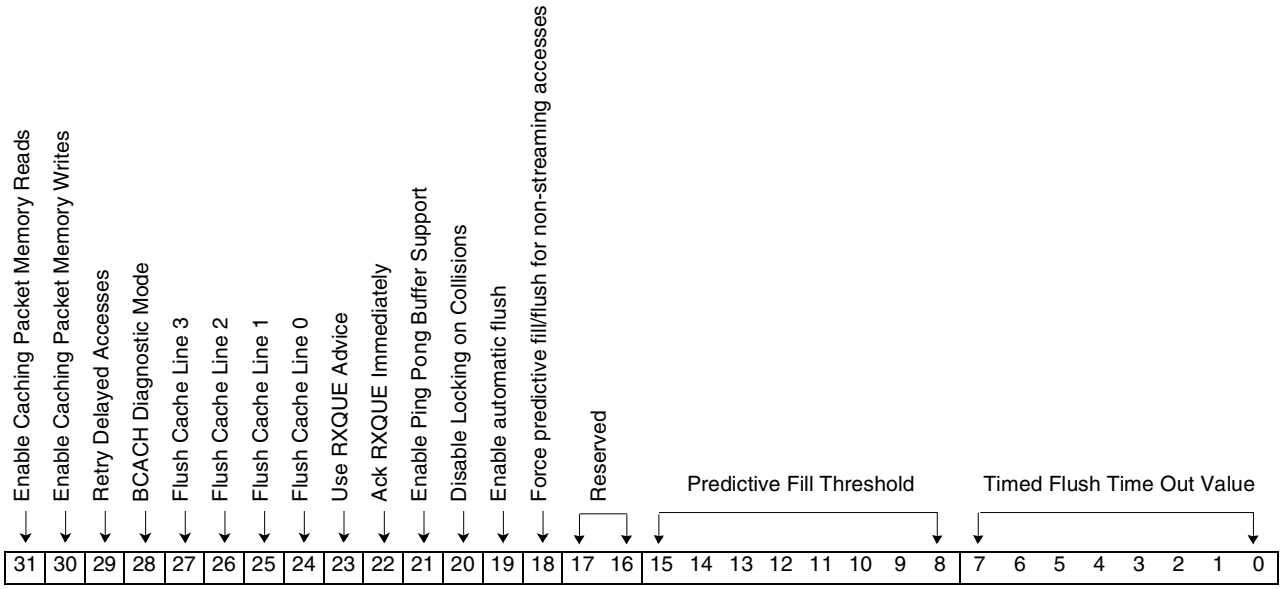
Length 32 bits

Type Clear/Set

Address XXXX 1000 and 004

Power On Value X'2000 0000'

Restrictions None



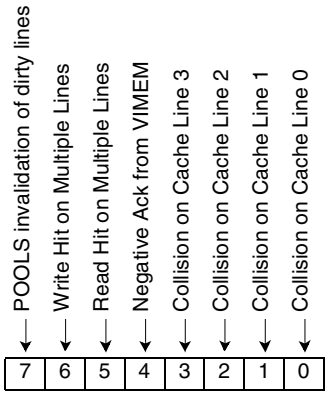
Bit(s)	Function	Description
31	Enable Caching Packet Memory Reads	When this bit is set, reads of Packet Memory will be cached.
30	Enable Caching Packet Memory Writes	When this bit is set, writes to Packet Memory will be cached.
29	Retry Delayed Accesses	When this bit is set, and the cache is enabled, any access of Packet Memory that cannot be satisfied within one cycle will be terminated by the cache with a retry indication. The accessing device is expected to allow competing devices a chance to gain access to the bus and then retry the same operation.
28	BCACH Diagnostic Mode	When this bit is set, diagnostic mode is enabled and reads and writes of the BCACH array from the processor are enabled. When reset, reads from the processor will return X'BADDBADD' and writes will have no affect. Care must be taken when performing writes from the processor: if a cache line fill operation is in process and a write is performed from the processor that writes to the same address in the array as is being written from the fill operation, results are indeterminate.
27	Flush Cache Line 3	Setting this bit forces a flush of cache line 3 if it is dirty. This bit is reset by the hardware when the flush completes.
26	Flush Cache Line 2	Setting this bit forces a flush of cache line 2 if it is dirty. This bit is reset by the hardware when the flush completes.

Bit(s)	Function	Description
25	Flush Cache Line 1	Setting this bit forces a flush of cache line 1 if it is dirty. This bit will be reset by the hardware when the flush completes.
24	Flush Cache Line 0	Setting this bit forces a flush of cache line 0 if it is dirty. This bit will be reset by the hardware when the flush completes.
23	Use RXQUE Advice	When set, advice from the receive queue entity causes the cache logic to fill a line with the data from the start of the buffer that was just dequeued by the software. This should improve performance by having the receive data available when the processor accesses the buffer after the dequeue. To make best use of this feature, the code should access the receive data shortly after the dequeue to avoid the data in the cache line from becoming stale and being invalidated due to other cache functions. When reset, advice from the receive queue entity will be ignored.
22	Ack RXQUE Immediately	When reset, advice from the receive queue entity is acknowledged immediately even if the cache is not able to perform the requested data fetch. In this case, the advice is lost, and the cache will not fetch the data until the processor requests it again. When set, the advice from the receive queue entity is not acknowledged until the cache has actually latched the advice information. This guarantees that the advice will be used, but may cause delays in the Receive Queue entity's processing.
21	Enable Ping Pong Buffer Support	When reset, this bit disables the two-line ping pong feature associated with consistent sequential cache accesses. When set, a series of sequential accesses to Packet Memory that would normally require more than two cache lines to be satisfied is limited to only two cache lines, regardless of the length of the transfer. This feature is intended to improves cache performance by preventing cache lines that contain the most recently used processor data from being flushed due to a long streaming access.
20	Disable Locking on Collisions	When set, this bit prevents detected collisions from locking up the memory control entity.
19	Enable automatic flush	When set, this bit enables the automatic flush feature of the cache. The auto flush feature forces a flush of a cache line to be performed if a sequential write of the last 2 locations in the cache line is detected.
18	Force predictive fill/flush for non-streaming accesses	When set, this bit forces the predictive fill/flush logic to operate on all accesses of the cache rather than just streaming accesses. When reset, the predictive fill logic will only be activated for streaming accesses in the cache.
17-16	Reserved	Reserved.
15-8	Predictive Fill Threshold	These bits set the threshold at which a predictive fill will be initiated. If all of these bits are set to '1', a predictive fill will be initiated on the first streaming access of a cache line, regardless of which byte in the line is accessed. If this field is set to X'3F' a predictive fill will be initiated on any streaming access of bytes at offset X'2' through X'7' in the cache line. If this field is set to X'03' a predictive fill will be initiated on any streaming access of bytes at offset X'6' or X'7' in the cache line. Setting the field to all '0's will disable predictive fills.
7-0	Timed Flush Time Out Value	These bits control the time-out value used to monitor dirty cache lines for inactivity. The value loaded into these eight bits is the number of 240 ns ticks that can occur without any activity in a dirty cache line before the cache logic will force a flush of the line to main memory. Setting these bits to all '0's disables the timed flush feature.

8.2: BCACH Status Register

The bits in this register reflect the current status of the cache. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	8 bits
Type	Clear/Set
Address	XXXX 1008 and 00C
Power On Value	X'00'
Restrictions	None



Bit(s)	Function	Description
7	POOLS invalidation of dirty lines	When this bit is set, it indicates that POOLS requested that the cache logic invalidate a line that was dirty. This is usually an indication that a buffer was freed by the software before data written out to the buffer had been flushed to memory. This may or may not be an error condition.
6	Write Hit on Multiple Lines	When this bit is set, the cache logic has detected a write hit to multiple lines. This indicates an internal logic error in the cache.
5	Read Hit on Multiple Lines	When this bit is set, the cache logic has detected a read hit to multiple lines. This indicates an internal logic error in the cache.
4	Negative Ack from VIMEM	When set, the cache logic has detected a negative acknowledgment from the Virtual Memory Logic entity. This indicates that a virtual buffer boundary was crossed and a new real buffer was needed to map the requested address space into, but no real buffer was available. In addition to setting this status bit, the cache logic writes the pattern X'zzzzzBAD' into the header of the packet at offset X'C' where zzzzz is the offset of the failing write into the packet.
3	Collision on Cache Line 3	When this bit is set, the cache logic has detected a collision in cache line 3. This is a situation where another entity in IBM3206K0424 was accessing an area of memory that was contained in one of the cache lines that was dirty. Further information for problem diagnosis is latched in the memory controller logic when this condition is detected.
2	Collision on Cache Line 2	When this bit is set, the cache logic has detected a collision in cache line 2.
1	Collision on Cache Line 1	When this bit is set, the cache logic has detected a collision in cache line 1.
0	Collision on Cache Line 0	When this bit is set, the cache logic has detected a collision in cache line 0.

8.3: BCACH Interrupt Enable Register

The low eight bits in this register allow the user to selectively determine which bits in the BCACH status register will cause processor interrupts. A ‘0’ in a bit position masks interrupts from the corresponding bit location in the BCACH status register. A ‘1’ in a bit position allows interrupts for the corresponding bit in the BCACH status register. The high eight bits in this register allow the user to selectively determine which bits in the BCACH status register will lock the cache. A ‘1’ in any bit position forces the cache to lock if the corresponding bit is set in the BCACH status register. If the cache locks, all status regarding the cache lines is maintained until the cache enable bits in the control register are turned off.

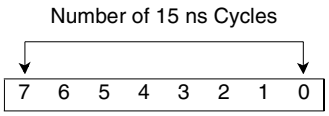
See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	16 bits
Type	Clear/Set
Address	XXXX 1010 and 014
Power On Value	X’FFFF’
Restrictions	None

8.4: BCACH High Priority Timer Value

This register defines the number of 15 ns cycles that will pass from the time that a valid PCI bus request is raised to BCACH until BCACH will raise its high priority request to the memory controllers. A value of ‘0’ in this register disables this function completely.

Length	8 bits
Type	Read/Write
Address	XXXX 1040
Power On Value	X’40’
Restrictions	None

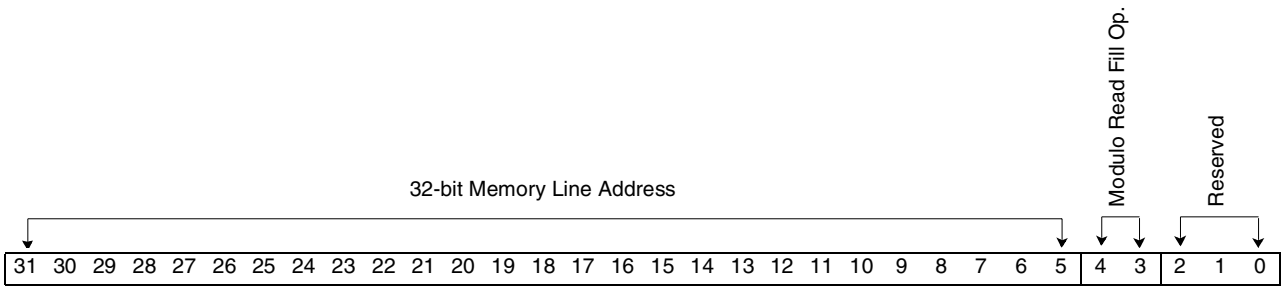


Bit(s)	Description
7-0	Specifies the number of 15 ns cycles before a high priority request. For example, if bit 3 is set to ‘1’ and all others are set to ‘0’, then 6 cycles (120 ns) will pass between receipt of request and sending request to controllers.

8.5: BCACH Line Tag Registers

These registers are useful only in diagnostic testing of the cache logic. Each register will contain the tag value for the data contained in that particular cache line.

Length	32 bits	
Type	Read Only	
Address	Tag Number 0	XXXX 1080
	Tag Number 1	XXXX 10A0
	Tag Number 2	XXXX 10C0
	Tag Number 3	XXXX 10C0
Power on Value	X'0000 0000'	
Restrictions	None	

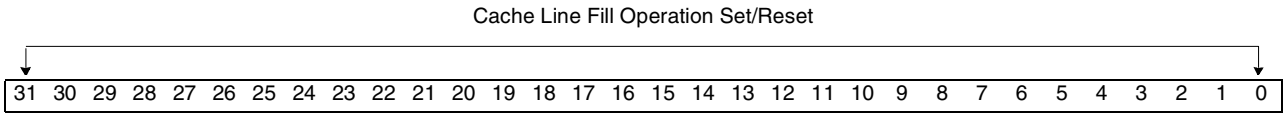


Bit(s)	Description
31-5	These bits contain the address of the 32-byte line of memory contained in the cache line.
4-3	In an attempt to provide the fastest possible access to data in memory, the 8 byte word in memory that contains the requested read data is accessed first and all other entries in the cache line are filled by wrapping back to the beginning of the cache line if required. These two bits contain the starting address for the modulo read fill operation. They will also contain the least significant address bits when a cache line is initially written to.
2-0	Will always be returned as '0'.

8.6: BCACH Line Valid Bytes Register

These registers are useful only in diagnostic testing of the cache logic. Each register will contain a bit significant flag indicating which bytes in the 32-byte cache line are valid. All of these bits will be active after a cache line fill operation has occurred, but any combination of these bits can be valid after the processor has performed a write operation to memory.

Length	32 bits	
Type	Read Only	
Address	Tag Number 0	XXXX 1084
	Tag Number 1	XXXX 10A4
	Tag Number 2	XXXX 10C4
	Tag Number 3	XXXX 10E4
Power on Value	X'0000 0000'	
Restrictions	None	

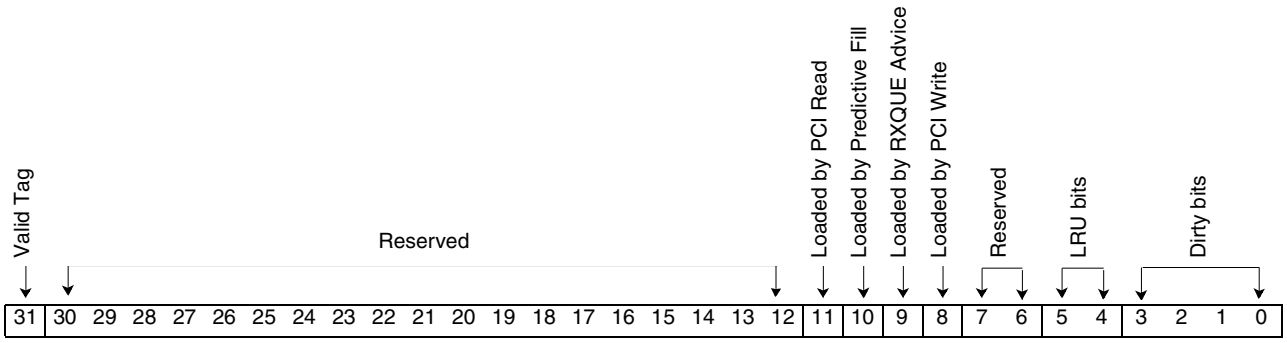


Bit(s)	Description
31-0	Each bit indicates whether the associated byte in the cache line contains valid data or not. If the bit is set, the cache line contains valid data and a fetch from main storage is not required to fulfill a request for a read from this location. If the bit is reset, a read of the associated location will require a cache line fill operation before the request can complete.

8.7: BCACH Line Status Register

These registers are useful only in diagnostic testing of the cache logic. Each register will contain a bit significant flag indicating the current status of the associated cache line.

Length	32 bits	
Type	Read Only	
Address	Tag Number 0	XXXX 1088
	Tag Number 1	XXXX 10A8
	Tag Number 2	XXXX 10C8
	Tag Number 3	XXXX 10E8
Power on Value	Tag Number 0	X'0000 0000'
	Tag Number 1	X'0000 0010'
	Tag Number 2	X'0000 0020'
	Tag Number 3	X'0000 0030'
Restrictions	None	



Bit(s)	Function	Description
31	Valid Tag	When set, this indicates that the associated tag register contains a valid tag.
30-12	Reserved	Reserved
11	Loaded by PCI Read	When set, this bit indicates that the associated tag register was loaded due to a read request from the PCI bus
10	Loaded by Predictive Fill	When set, this bit indicates that the associated tag register was loaded due to a predictive fill request
9	Loaded by RXQUE Advice	When set, this bit indicates that the associated tag register was loaded due to advice from the receive queue entity
8	Loaded by PCI Write	When set, this bit indicates that the associated tag register was loaded due to a write request from the PCI bus
7-6	Reserved	Reserved
5-4	LRU bits	These bits indicate the cache lines current position with respect to the least recently used algorithm. A value of '0' indicates it is the most recently used while a value of '3' indicates the least recently used.

Bit(s)	Function	Description
3-0	Dirty bits	These bits, when set, indicate that the associated eight-byte word of the cache line is dirty. This information is used on cache line flushes, to lower memory utilization, by eliminating non-dirty word flushes from the cache line flush operation. For example if these bits contain a X'1', only the eight-byte word at offset zero in the cache line is dirty, so the flush operation will only write this one word to memory, saving three memory access cycles. If these bits contain a X'C', only the two eight-byte words starting at offset X'10' in the cache line are dirty.

8.8: BCACH Cache Line Array

This array is divided into four 32-byte buffers used as cache lines 0, 1, 2, and 3.

Length 16 Words x 64 bits

Type Read/Write

Address XXXX 1100 - 17F

Restrictions This array can only be accessed when the diagnostic mode bit in the control register is set.

Bit(s)	Description
??	The four cache lines start at the following offsets into the array: Line 0 Offset X'00' Line 1 Offset X'20' Line 2 Offset X'40' Line 3 Offset X'60'

Entity 9: Buffer Pool Management (POOLS)

POOLS acts as a memory manager for the IBM3206K0424. Memory buffers are checked out and checked in via two operations (primitives) supported by POOLS: the get pointer primitive and the free pointer primitive. These primitives can be performed explicitly by accessing specified addresses within the POOLS entity, and they may also be done by hardware. CSKED can free a buffer upon transmission if specified by the corresponding packet header (see *ECC Syndrome Bits* on page 196), and RAALL gets buffers to store received data. In addition, POOLS contains mechanisms to control resource utilization and supports a Real Memory Mode and a Virtual Memory Mode.

Basic Operation in Real Memory Mode

If memory is viewed as a series of buffers, POOLS maintains a circular list of available buffers. There are pointers (the head and tail) to the start and the end of the list. When a get pointer primitive is executed, the buffer at the head of the list is checked out, the head pointer is advanced and the correct resource group(s) is debited. When a free pointer primitive is executed, the freed buffer is checked in at the end of the list, the tail pointer is advanced, and the correct resource group is credited.

Basic Operation in Virtual Memory Mode

With the addition of Virtual Memory, POOLS must maintain five sets of head and tail pointers, thresholds, and active counts: one for the virtual buffers themselves and the rest for the four regions of real buffers that constitute the virtual buffers. In this case the base virtual address is the item returned from a get pointer operation and returned during a free pointer operation. When the get buffer primitive is executed, POOLS creates an active buffer map (page table) for the virtual address. As the virtual address is used and buffer (page) boundaries are crossed, VIMEM will request buffers from POOLS when a buffer (page) fault occurs. VIMEM then places the buffer index in the buffer map. When the virtual buffer is no longer needed and a free pointer primitive is issued with the starting virtual address, POOLS takes the contents of the buffer map and frees the resources that were assigned to the buffer map.

Resource Controls

POOLS adds another layer of service by creating "pools" of buffers (currently a maximum of 16 pools). For each pool, a maximum number of allowable buffers can be specified. The intent is to make it possible for several applications to use an IBM3206K0424 at once without one or more applications starving the remaining applications for memory buffers. Particular pools buffers are divided into “guaranteed” and “common” buffers. All the guaranteed buffers are considered to be dedicated to their respective pool and are therefore not available for general use. The common buffers are all the memory buffers remaining after the guaranteed buffers are subtracted from the total buffers. To maintain the buffer limits on each pool, every pool has a guaranteed threshold, total threshold, and an active count. When a request is made for a buffer from a particular pool, the guaranteed threshold is first checked. If the active count of the pool is less than the guaranteed threshold, the buffer is provided. If the guaranteed threshold has been reached, then the total threshold is checked. If the active count is equal to the total threshold, no buffer is provided. If the active count is less than the total threshold, and a common buffer is available, a buffer is provided. If there are no common buffers available, a buffer cannot be provided and a null index is returned. To determine if a common buffer is available, a count is maintained for each size of buffer.

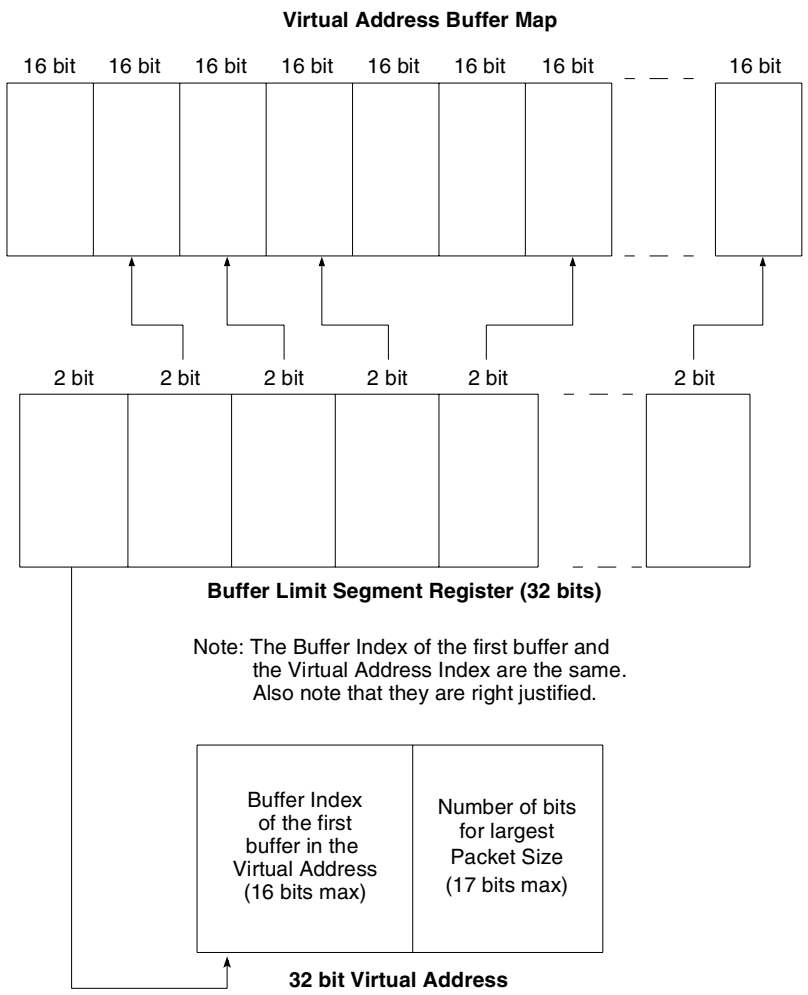
Virtual Memory Overview

Each virtual buffer consists of a number of real buffers. For each virtual buffer there is a buffer map that defines the size and number of real buffers that may be allocated to the virtual buffer. Each map is built from a common template (the VIMEM Virtual Buffer Segment Size Register) that associates 1 to n buffer indexes in the map to a real buffer in one of the four real buffer regions defined in VIMEM. In VIMEM, the Buffer Map Base Address Register defines the size of the map and therefore also the number of buffer indexes in the virtual buffer map. Each eight-byte entry of the map contains the pool ID of the pool to which the buffer is allocated plus space for three real buffer segment indexes. This implies the smallest map yields a virtual buffer of one to four real buffer segments (three real buffer segments plus the implicit real buffer that all virtual buffers are allocated). The biggest map defines a virtual buffer of 1-16 real buffer segments (15 plus the implicit one).

The intention of this structure is to allow the user to customize the value in the Virtual Buffer Segment Size Register to utilize memory in an efficient manner relative to network data traffic. For example, if network traffic contained 50% packets of < 512 bytes, 35% packets of < 1K bytes, and the rest was < 5K bytes, the user could set up Virtual Memory to use three real segments of 512 bytes, 512 bytes, and 4K bytes respectively. The incoming data would neatly fit into the segments and minimize wasted memory.

POOLS and VIMEM maintain the maps for the virtual buffers. On a write that crosses a real buffer boundary into an as yet an unresolved region of a virtual buffer, a page fault occurs. When a page fault occurs, POOLS determines whether or not a real buffer can be assigned. If it can be assigned, the index of the real buffer relative to the base address of the particular buffer size is placed by VIMEM into the buffer map. The first buffer is implicitly associated with the Virtual Memory address for a particular virtual buffer and enough real memory must be available to support the first real buffer of each virtual buffer at initialization time. There is not necessarily enough real storage for all the possible real buffers associated with a virtual buffer.

Virtual Address Buffer Map

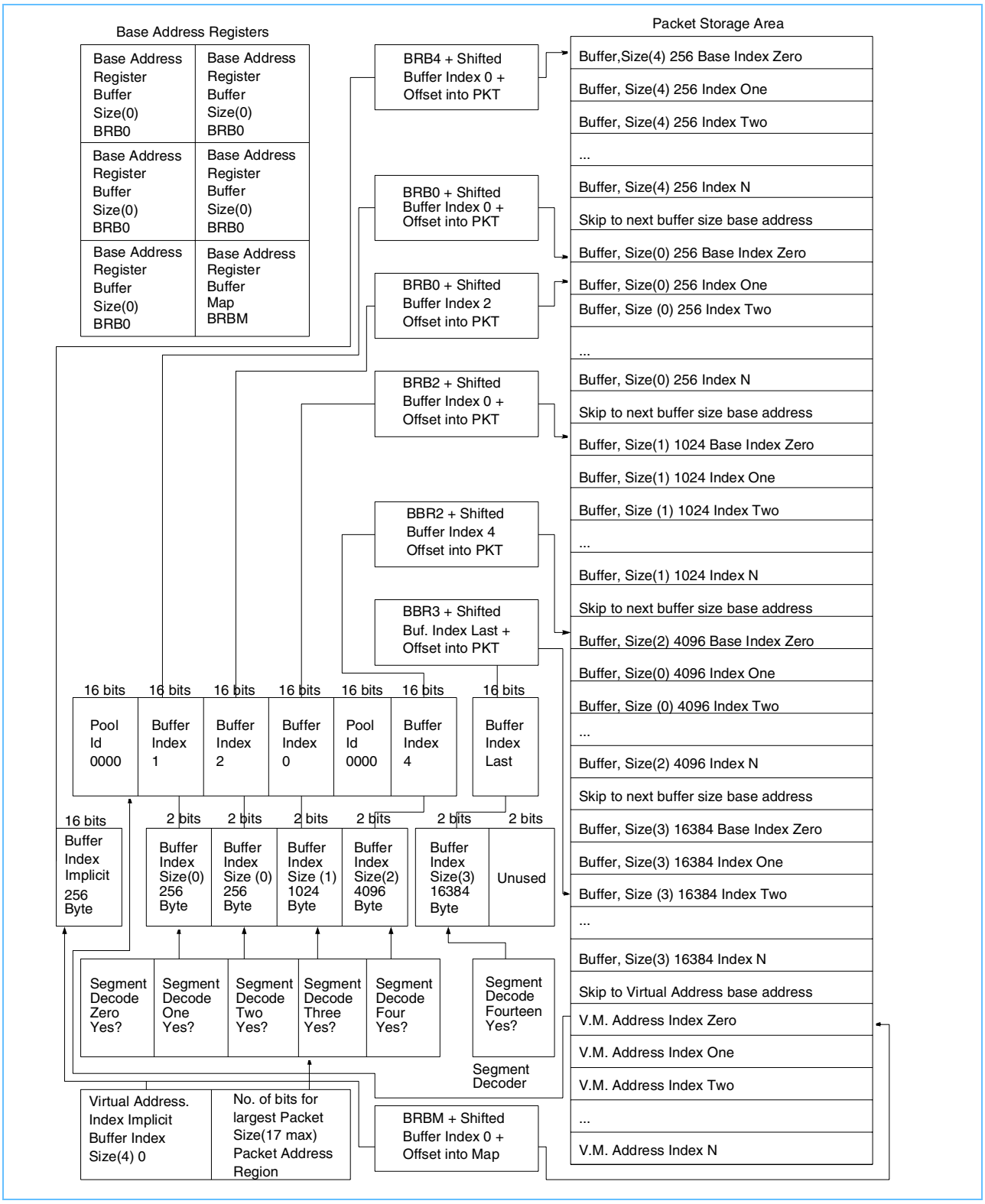


All real buffers of a particular size are stored in a contiguous region of memory. The buffer index, in conjunction with the base address for this real buffer size, points to a particular real buffer. The implicit buffers are also stored in a data structure of this type.

Buffer/Virtual Memory Allocation Structure in Memory

The following example illustrates these concepts.

Virtual Address Buffer Map



The lower seventeen bits of the virtual address are used in conjunction with the segment template in the VIMEM Virtual Buffer Segment Size Register to determine from which portion of the buffer map the buffer index is retrieved. Once the buffer index is retrieved, it is combined with the appropriate base address for that particular buffer size. The offset into the buffer is then added to get the real 32-bit address that is used in physical memory.

POOLS uses the data structures above to manage Packet Memory resources. Each LCD is associated with a particular POOL and multiple different LCDs may be associated with that same POOL. Within a POOL, there are five different resource categories and two variables to go with each resource.

Resources and Variables Example

Resource Type, Pool 0000	Guaranteed Number	Total Number
Virtual Memory Addresses	100	150
Buffer Type One	200	300
Buffer Type Two	50	100
Buffer Type Three	10	5
Buffer Type Four	0	10
Virtual Memory Addresses	100	150

9.1: POOLS Get Pointer Primitive

The POOLS Get Pointer Primitive returns a pointer to the requester. The request to the virtual packet/buffer size 4 address will always return a memory address. If in virtual mode, the address will be virtual. Requests made for buffer sizes 0 to 3 will not return an address but rather a buffer index in bits 15-0. The real address associated with this index can be generated by shifting the index by the buffer size (for example, six bit positions for a 64-byte buffer) and adding the result to the base address for this size buffer. Access to buffer sizes 0 to 3 is not permitted in operational mode.

The address of the primitive also selects the pool ID. The pool ID is contained in address bits 5-2, and it selects which pool will be charged for the pointer. The buffer size is selected with address bits 8-6.

If there are no more pointers available in the specified pool, a null pointer is returned. The active pointer count for that pool is incremented if a non-null pointer is returned. If the guaranteed threshold has been exceeded and a buffer from the common pool is returned, the common pools count for that size is decremented by 1.

Length	32 bits	
Type	Read Only	
Address	Buffer Size 0	XXXX 3200
	Buffer Size 1	XXXX 3240
	Buffer Size 2	XXXX 3280
	Buffer Size 3	XXXX 32C0
	Virtual Packets/ Buffer Size 4	XXXX 3300
Power on Value	X’00000000’	
Restrictions	During normal operations this register is to be used as a read only register. Writes to this address will be ignored.	

9.2: POOLS Free Pointer Primitive

The POOLS Free Pointer Primitive returns the pointer to the proper free list. If it is a Virtual Memory address, the Virtual Memory Buffer Map is traversed to free the indexes associated with the Virtual Memory address. In the case where it is a real memory buffer, the single index is freed.

This primitive uses address mapping to select the size of the object to be freed. The size is contained in address bits 4-2. During normal operation, only frees to buffer size four are relevant. During initialization mode, buffer sizes 0 to 3 can be used to load indexes. The indexes are loaded into bits 31-16.

In normal operations it is not necessary to read this "register".

Length	32 bits	
Type	Write Only	
Address	Buffer Size 0	XXXX 3350
	Buffer Size 1	XXXX 3354
	Buffer Size 2	XXXX 3358
	Buffer Size 3	XXXX 335C
	Virtual Packets/ Buffer Size 4	XXXX 3360
Power on Value	X'00000000'	
Restrictions	During normal operations this register is to be used as a Write only register. Reads from this address will return '0'.	

9.3: POOLS Common Pools Count Registers

The POOLS Common Pools Count Registers indicates the number of pointers in the particular common pool.

The bits are a 16-bit count. The Get Pointers that exceed the guaranteed allocation decrement this count by one assuming that the count is non-zero. When the count is zero, the Get Buffer operation will fail. The Free Pointers that operate beyond the guaranteed threshold for a particular client and free the pointer(s) increment this count by one. The microcode should initialize these registers to the value of the respective common pool that it desires to have.

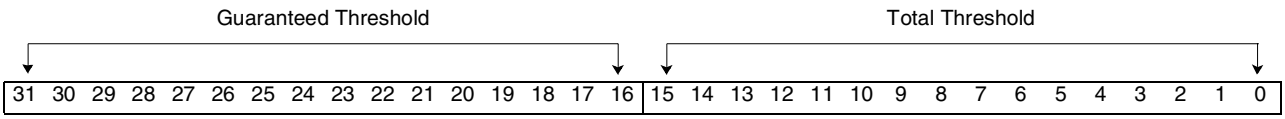
Length	16 bits	
Type	Read/Write	
Address	Buffer Size 0	XXXX 3000
	Buffer Size 1	XXXX 3004
	Buffer Size 2	XXXX 3008
	Buffer Size 3	XXXX 300C
	Virtual Packets / Buffer Size 4	XXXX 3010
Power on Value	X'0000'	
Restrictions	During normal operations these registers are to be used as a read only. Writing to these registers during operation could create a data loss situation. This register should be set up by the microcode at initialization time.	

9.4: POOLS Client Thresholds Array

The POOLS Client Thresholds Array holds the guaranteed and total threshold values for the 16 pools and the four (five) pointer sizes. This array contains the guaranteed and total thresholds for the managed POOLs.

When a Get Pointer primitive is processed, the values in this array are used to determine if a primitive can return a pointer. The active count from the Active Packet Count Array is used with these registers to determine if a threshold has been exceeded. If the guaranteed threshold has been exceeded and the total not exceeded and there is a common pointer available, then the common count will be incremented. If there are no common buffers available or the request will cause the total threshold to be exceeded, the request will be rejected. During a Free Pointer primitive processing, the pointer is returned to the free list and these thresholds are used to determine if a common count should be credited.

Length	32 bits x 16 Words	
Type	Read/Write	
Address	Buffer Size 0	XXXX 3400
	Buffer Size 1	XXXX 3440
	Buffer Size 2	XXXX 3480
	Buffer Size 3	XXXX 34C0
	Virtual Packets/ Buffer Size 4	XXXX 3500
Power on Value	X'00000000'	
Restrictions	None	



Bit(s)	Description
31-16	Guaranteed Threshold
15-0	Total Threshold.

9.5: POOLS User Threshold and Client Active Packet Count Array

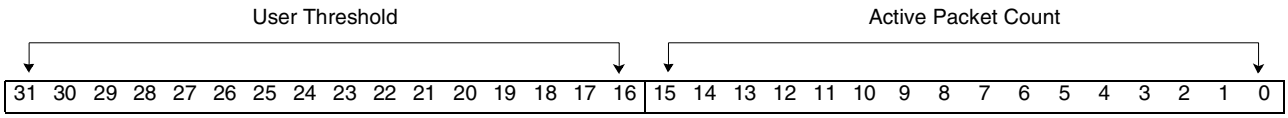
The POOLS User Threshold and Client Active Packet Count Array holds the user thresholds and active pointer counts for each of the 16 managed pools and four (five) pointer sizes.

When a Get Pointer primitive is processed, the active count is retrieved and compared with the threshold counts. If it falls within bounds and a pointer is available, the active count will be incremented by one reflecting the additional buffer charged to that queue.

When a Free Pointer primitive is processed, the active count is retrieved, and, when the pointer is returned to the free list, the active buffer count is decremented by one.

The user threshold may be used to check on resource utilization as opposed to resource allocation. The Guaranteed and Total Thresholds are used when allocating resources to make decisions. The User Threshold is not used to govern resource allocation directly. One such use is for high water mark indication. When a Free Pointer primitive is processed or a Get Pointer is processed The active packet count is compared to the user threshold. If the event interface is enabled and a boundary condition is crossed an event is issued to the event interface.

Length	32 bits x 16 Words	
Type	Read/Write	
Address	Buffer Size 0	XXXX 3600
	Buffer Size 1	XXXX 3640
	Buffer Size 2	XXXX 3680
	Buffer Size 3	XXXX 36C0
	Virtual Packets/ Buffer Size 4	XXXX 3700
Power on Value	X'00000000'	
Restrictions	None	



Bit(s)	Description
31-16	User Threshold
15-0	Active Packet Count

9.6: POOLS Pointer Queues DRAM Head Pointer Offset Address Register

The POOLS Pointer Queues DRAM Head Pointer Offset Address Register indicates the address in DRAM where the head of the queue starts. This address, however, is only relative to the DRAM portion of the queue. Unless the head of the queue portion of the cache is locked out and needs two frames, the actual head of the queue is in the cache.

These 19 bits on write represent the offset to the address in DRAM of the head of the queue relative to the DRAM base address. On a read, the address in DRAM of the pointer is returned. This pointer is adjusted every time a cache frame boundary is crossed and a cache update cycle is completed to write through the additional queue elements. Because each memory reference contains four indices, this allows for a possible 128K index locations in the queue.

Length	32 bits Read/19 bits Write	
Type	Read/Write	
Address	Buffer Size 0	XXXX 3014
	Buffer Size 1	XXXX 3018
	Buffer Size 2	XXXX 301C
	Buffer Size 3	XXXX 3020
	Virtual Packets/ Buffer Size 4	XXXX 3024
Power on Value	Buffer Size 0	X'00 01 C0 00'
	Buffer Size 1	X'00 02 00 00'
	Buffer Size 2	X'00 02 40 00'
	Buffer Size 3	X'00 02 60 00'
	Virtual Packets/ Buffer Size 4	X'00 02 70 00'
Restrictions	During normal operations this register is to be used as a read only register. This register defaults to zero at initialization. It is assumed that the queues start on a maximum size queue boundary. These registers should be set up at initialization time. This register is cleared when the POOLS Pointer Queues DRAM Lower Bound Address Register is written to.	

9.7: POOLS Pointer Queues DRAM Tail Pointer Offset Address Register

The POOLS Pointer Queues DRAM Tail Pointer Offset Address Register indicates the offset address in DRAM where the tail of the queue starts. This address, however, is only relative to the DRAM portion of the queue. Unless a 'no cache frames to be written through' state is in effect, the actual tail of the queue is in the cache.

These 19 bits on write represent the offset to the address in DRAM of the tail of the queue relative to the DRAM base address. On a read, the address in DRAM of the pointer is returned. This pointer is adjusted every time a cache frame boundary is crossed and a cache update cycle is completed to write through the additional queue elements. Since each memory reference contains four indices this allows for 128K index locations possible in the queue.

Length	16 bits	
Type	Read/Write	
Address	Buffer Size 0	XXXX 3028
	Buffer Size 1	XXXX 302C
	Buffer Size 2	XXXX 3030
	Buffer Size 3	XXXX 3034
	Virtual Packets/ Buffer Size 4	XXXX 3038
Power on Value	Buffer Size 0	X'00 01 C0 00'
	Buffer Size 1	X'00 02 00 00'
	Buffer Size 2	X'00 02 40 00'
	Buffer Size 3	X'00 02 60 00'
	Virtual Packets/ Buffer Size 4	X'00 02 70 00'
Restrictions	During normal operations this register is to be used as a read only register. This register defaults to zero at initialization. It is assumed that the queues start on the maximum size queue boundary. These registers should be setup at initialization time. This register is cleared when the POOLS Pointer Queues DRAM Lower Bound Address Register is written to.	

9.8: POOLS Pointer Queues DRAM Lower Bound Address Register

The POOLS Pointer Queues DRAM Lower Bound Address Register indicates the address in DRAM where the queue data structure is initially started. When the queue reaches the maximum address allowed for in the upper bound register, it wraps back around to the address specified in this register. This implements the queue in a circular buffer.

These 32 bits represent the address in DRAM where the queue begins and eventually wraps to. At initialization, this register and the POOLS Pointer Queues DRAM Tail Pointer Offset Address Register and the POOLS Pointer Queues DRAM Head Pointer Offset Address Register must be equal.

Length	32 bits	
Type	Read/Write	
Address	Buffer Size 0	XXXX 303C
	Buffer Size 1	XXXX 3040
	Buffer Size 2	XXXX 3044
	Buffer Size 3	XXXX 3048
	Virtual Packets/ Buffer Size 4	XXXX 304C
Power on Value	Buffer Size 0	X'00 01 C0 00'
	Buffer Size 1	X'00 02 00 00'
	Buffer Size 2	X'00 02 40 00'
	Buffer Size 3	X'00 02 60 00'
	Virtual Packets/ Buffer Size 4	X'00 02 70 00'
Restrictions	During normal operations, this register is to be used as a read only register. This register should be setup at initialization time. The size of the DRAM queue storage which is formed with the lower and upper bounds is constrained in its size. It can be written when the diagnostic mode bit is set, otherwise the write is ignored. Note that if the maximum queue length exceeds the space available in the circular buffer, data corruption will occur when the actual queue length exceeds the maximum queue space available.	

9.9: POOLS Pointer Queues DRAM Upper Bound Register

The POOLS Pointer Queues DRAM Upper Bound Register indicates the max queue length in DRAM of the queue data structure. When the queue reaches this address, it wraps back to the address specified by the lower bound register. This implements the queue in a circular buffer. This upper bound is to be provided as an encoded field. The encoded field represents the number of eight-byte addresses that can be contained by the queue.

These four bits represent the encoded maximum queue length in DRAM which, when matched, trigger the queue to wrap back to the address contained in the DRAM Lower Bound Address Register.

Length	4 bits	
Type	Read/Write	
Address	Buffer Size 0	XXXX 3050
	Buffer Size 1	XXXX 3054
	Buffer Size 2	XXXX 3058
	Buffer Size 3	XXXX 305C
	Virtual Packets/ Buffer Size 4	XXXX 3060
Power on Value	Buffer Size 0	X'B'
	Buffer Size 1	X'A'
	Buffer Size 2	X'9'
	Buffer Size 3	X'9'
	Virtual Packets/ Buffer Size 4	X'B'
Restrictions	During normal operations, this register is to be used as a read only register. This register should be setup at initialization time. The size of the DRAM queue storage which is formed with the lower and upper bounds is constrained in its size. It can be written when the diagnostic mode bit is set, otherwise the write is ignored. Note that if the maximum queue length exceeds the space available in the circular buffer, data corruption will occur when the actual queue length exceeds the maximum queue space available.	

Bit(s)	Description		
	Encoded Value	Number of 32 Bit Words	Number of Indexes
3-0	X'0'	8	16
	X'1'	16	32
	X'2'	32	64
	X'3'	64	128
	X'4'	128	256
	X'5'	256	512
	X'6'	512	1024
	X'7'	1024	2048
	X'8'	2048	4096
	X'9'	4096	8192
	X'A'	8192	16384
	X'B'	16384	32768
	X'C'	32768	65536
	X'D'	65536	131072
	X'E'	65536	131072
	X'F'	65536	131072

9.10: POOLS Pointer Queues Length Registers

The POOLS Pointer Queues Length Registers indicates the length of the queue. The bits are a 16-bit count. A primitive that adds to the queue increments this counter. Primitives that remove items from the queue decrement this counter.

Length	16 bits	
Type	Read/Write	
Address	Buffer Size 0	XXXX 3064
	Buffer Size 1	XXXX 3068
	Buffer Size 2	XXXX 306C
	Buffer Size 3	XXXX 3070
	Virtual Packets / Buffer Size 4	XXXX 3074
Power on Value	X'00 00'	
Restrictions	During normal operations, this register is to be used as a read only register. It can be written when the diagnostic mode bit is set, otherwise the write is ignored. This register is cleared when the POOLS Pointer Queues DRAM Lower Bound Address Register is written to.	

9.11: POOLS Interrupt Enable Register

This register is used to enable bits from the POOLS Status Register and potentially generate interrupts to the control processor. When both a bit in this register and the corresponding bit(s) in the POOLS Status Register are set, the POOLS interrupt to PCINT will be enabled.

See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See *POOLS Status Register* on page 265 for the bit descriptions.

Length	32 bits
Type	Clear/Set
Address	XXXX 3078 and 07C
Power On Value	X'00 03 F8 00'
Restrictions	None

9.12: POOLS Event Enables

This register is used to enable an event based on bits from the corresponding primitive transaction. If the bits are set in the enable and a transaction occurs that matches the event, an event will be sent to the RXQUE.

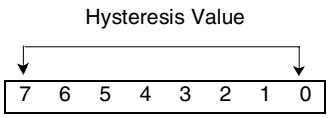
See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	16 bits	
Type	Clear/Set	
Address	GTD Event Enables	XXXX 3A00 and A04
	Total Event Enables	XXXX 3A08 and A0C
	User Event Enables	XXXX 3A10 and A14
Power on Value	X'0000'	
Restrictions	None	

9.13: POOLS Event Hysteresis Register

The POOLS Event Hysteresis Register provide the capability for hysteresis on threshold checking.

Length	8 bits
Type	Read/Write
Address	XXXX 3A18
Power on Reset value	X'0000'
Restrictions	None



Bit(s)	Function	Description
7-0	Hysteresis Value	When a free occurs, the value in this register is added to the next Active Packet Count. This value will be then tested against the threshold value. If it is equal to the threshold, an event will be issued if events are enabled and the event associated with this transaction is enabled.

9.14: POOLS Event Data Register

The POOLS Event Data Register provides the data that was sent on the last event.

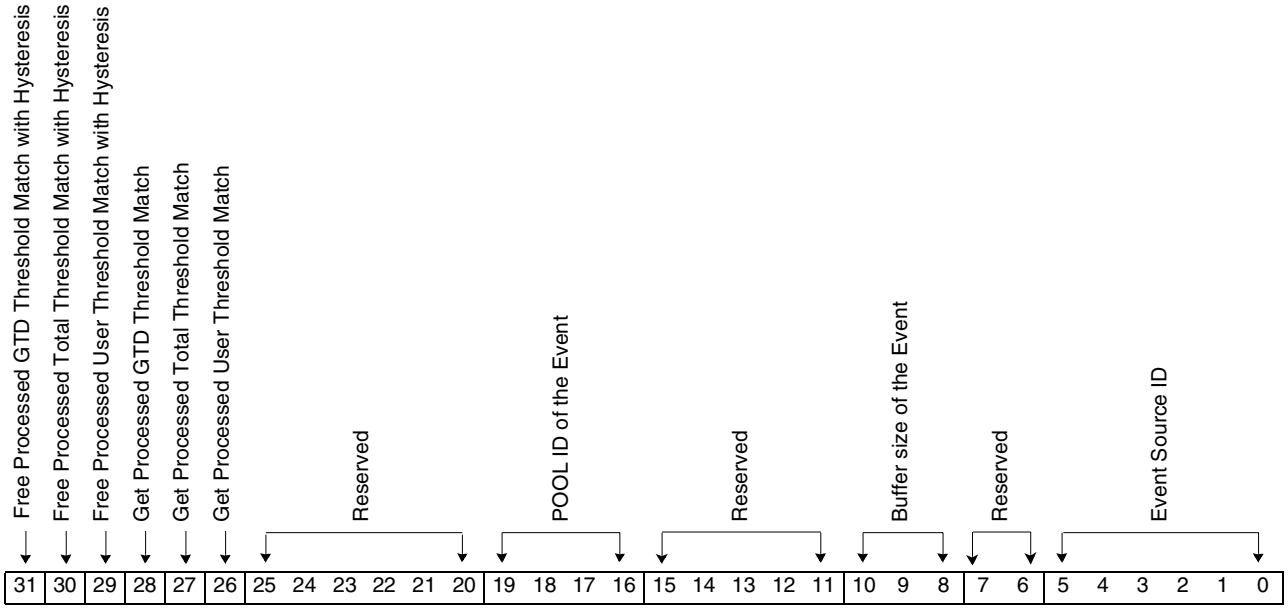
Length 32 bits

Type Read

Address XXXX 3A1C

Power on Reset value X'0000003E'

Restrictions None



Bit(s)	Function	Description
31	Free Processed GTD Threshold Match with Hysteresis	This event occurs when a free is processed and the threshold is matched. The threshold is modified by the value in the hysteresis register. The event is issued when the actual value of the Active Packet count plus the hysteresis equals the threshold.
30	Free Processed Total Threshold Match with Hysteresis	This event occurs when a free is processed and the threshold is matched. The threshold is modified by the value in the hysteresis register. The event is issued when the actual value of the Active Packet count plus the hysteresis equals the threshold.
29	Free Processed User Threshold Match with Hysteresis	This event occurs when a free is processed and the threshold is matched. The threshold is modified by the value in the hysteresis register. The event is issued when the actual value of the Active Packet count plus the hysteresis equals the threshold.
28	Get Processed GTD Threshold Match	This event occurs when a get is processed and the threshold is matched. The event is issued when the new Active Packet count equals the threshold.
27	Get Processed Total Threshold Match	This event occurs when a get is processed and the threshold is matched. The event is issued when the new Active Packet count equals the threshold.
26	Get Processed User Threshold Match	This event occurs when a get is processed and the threshold is matched. The event is issued when the new Active Packet count equals the threshold.
25-20	Reserved	Reserved

Bit(s)	Function	Description
19-16	POOL ID of the Event	This indicates which pool is associated with this event.
15-11	Reserved	Reserved
10-8	Buffer size of the Event	This indicates which size is associated with this event.
7-6	Reserved	Reserved
5-0	Event Source ID	This indicates that POOLS is associated with this event.

9.15: POOLS Status Register

The POOLS Status Register provides status information about pools operations. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

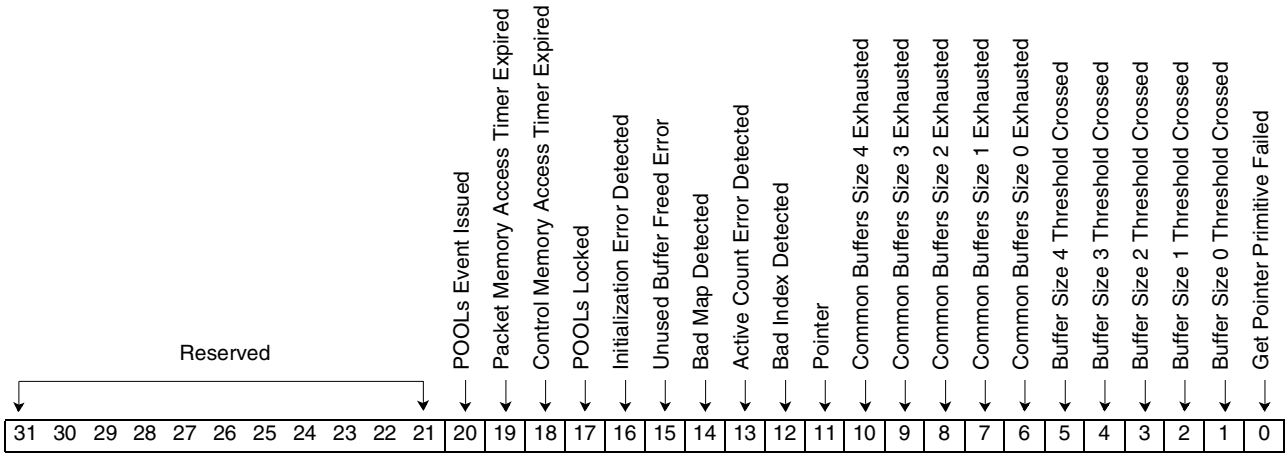
Length 32 bits

Type Clear/Set

Address XXXX 3080 and 084

Power on Reset value X'00 00 00 00'

Restrictions During normal operations, if a status bit is cleared, it will be reset if the condition that is causing it is still present.



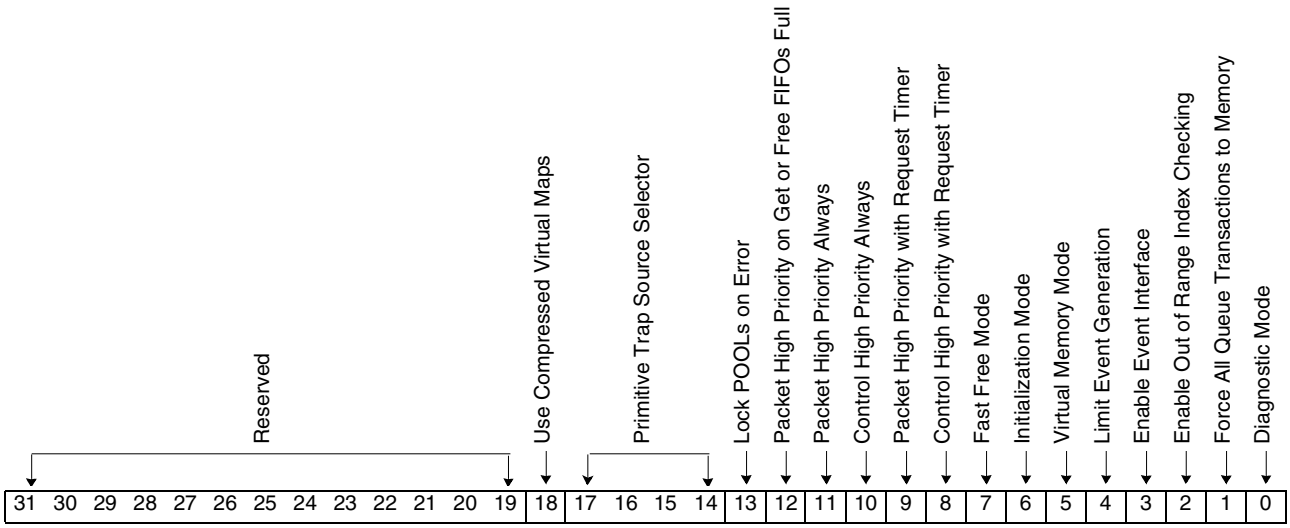
Bit(s)	Function	Description
31-21	Reserved	Reserved
20	POOLs Event Issued	This bit is set when a POOLs event is issued.
19	Packet Memory Access Timer Expired	This bit is set when the Packet Memory access timer hits the Packet Memory access threshold and the control bit in the control register is set to enable this function.
18	Control Memory Access Timer Expired	This bit is set when the Control Memory access timer hits the Control Memory access threshold and the control bit in the control register is set to enable this function.
17	POOLs Locked	This bit is set when a lock enable bit is set and the corresponding status bit is set. This causes all state machines to be held in idle once this bit is set. It is the functional equivalent to POOLS Control Register bit 0.
16	Initialization Error Detected	This bit is set when too many indexes are freed to a queue.
15	Unused Buffer Freed Error	This bit is set when a previously freed buffer is detected during a free operation. This typically would occur when the buffer was freed two or more times.
14	Bad Map Detected	This bit is set when a bad map is detected during a free operation.
13	Active Count Error Detected	This bit is set when an active packet count is decremented from '0' to X'FFFF'. This is most likely the result of a subtle map corruption where a POOL ID has been changed.
12	Bad Index Detected	This bit is set when an Index Threshold is crossed.

Bit(s)	Function	Description
11	Free Pointer Primitive Null Detected/Max Pointer Queue Length Exceeded.	This bit is set as a result of one of two detectable errors: Null index detected within free address. Total allowed storage for a particular queue has been exceeded.
10	Common Buffers Size 4 Exhausted	Common buffer count for size 4 is zero.
9	Common Buffers Size 3 Exhausted	Common buffer count for size 3 is zero.
8	Common Buffers Size 2 Exhausted	Common buffer count for size 2 is zero.
7	Common Buffers Size 1 Exhausted	Common buffer count for size 1 is zero.
6	Common Buffers Size 0 Exhausted	Common buffer count for size 0 is zero.
5	Buffer Size 4 Threshold Crossed	The number of size 4 buffers is equal to or less than the threshold that was set for size 4 buffers.
4	Buffer Size 3 Threshold Crossed	The number of size 3 buffers is equal to or less than the threshold that was set for size 3 buffers.
3	Buffer Size 2 Threshold Crossed	The number of size 2 buffers is equal to or less than the threshold that was set for size 2 buffers.
2	Buffer Size 1 Threshold Crossed	The number of size 1 buffers is equal to or less than the threshold that was set for size 1 buffers.
1	Buffer Size 0 Threshold Crossed	The number of size 0 buffers is equal to or less than the threshold that was set for size 0 buffers.
0	Get Pointer Primitive Failed	This bit is set when a null address is returned on a get.

9.16: POOLS Control Register

The POOLS Control Register provide status information about POOLs operations. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	32 bits
Type	Clear/Set
Address	XXXX 30C8 and 0CC
Power on Reset value	X'00 00 20 01'
Restrictions	Caution must be used when asserting some of the bits during operation.



Bit(s)	Function	Description
31-19	Reserved	Reserved
18	Use Compressed Virtual Maps	This bit selects compressed virtual maps when set.
17-14	Primitive Trap Source Selector	These bits will select the source of the last primitive trapped register. 0000 Free from the PCI bus 0001 Free from RAALL 0010 Free from RXQUE 0011 Free from CSKED 0100 Free from SEGBF 0101 Free from DMAQS 0110 Get from PCI Bus 0111 Get from REASM 1(RA) 1000 Get from DMAQS 1001 Get from VIMEM (POOL ID (4 bits)), Size (2 bits), blank (10 bits), index (16 bits)) '1010' Get from REASM 0(RC) '1010' Last Get Processed (Buffer Addr:(31-4)), Source (3-0) see above) '1011' Last Free Accepted (Buffer Addr:(31-4)), Source (3-0) see above) '1100' Last Primitive Processed (Buffer Addr:(31-4)), Source (3-0) see above) '1101' Last Free Processed

Bit(s)	Function	Description
13	Lock POOLs on Error	When set, this bit in conjunction with the lock mask will hold pools state machines in an idle state until cleared.
12	Packet High Priority on Get or Free FIFOs Full	When set, this bit causes pools to turn on its high priority request to Packet Memory when either the free or get FIFO is full.
11	Packet High Priority Always	When set, this bit causes pools to always use its high priority request to Packet Memory.
10	Control High Priority Always	When set, this bit causes pools to always use its high priority request to Control Memory.
9	Packet High Priority with Request Timer	When set, this bit causes pools to time the wait for Packet Memory service and when the timer expires move to high priority.
8	Control High Priority with Request Timer	When set, this bit causes pools to time the wait for Control Memory service and when the timer expires move to high priority.
7	Fast Free Mode	When this bit is set, Fast Free Mode is enabled. When pools is in Fast Free Mode it does not write out the buffer map with the modified control information that indicates that the map is unused. When in this mode unused buffer free error checking is disabled.
6	Initialization Mode	When the value of the bit is '0', initialization mode is set. When the value is '1', operational mode is set. During initialization mode indexes are in the upper 16 bits of the data word. It is assumed that when initialization mode is on other normal operations are not active such as transmit or receive. During operational mode packet addresses assumed to be on the data bus.
5	Virtual Memory Mode	When set to '0', Virtual Memory mode is enabled. When set to '1', real memory mode is enabled.
4	Limit Event Generation	When set, this bit causes pools to limit the issuance of events to RXQUE when a GTD threshold, Total Threshold or POOL Threshold is reached. It will issue the first event and disable the related event enable bit. Software must then reset the bit if it wishes to see another such event. However, it is possible that events may be lost when this bit is set on.
3	Enable Event Interface	When set, this bit causes pools to issue resource events to RXQUE when a GTD threshold, Total Threshold or POOL Threshold is reached.
2	Enable Out of Range Index Checking	When set, this bit causes pools to check the indexes that are streaming by to be checked against a maximum value for that size index. If the normal initialization sequence is used, these maximum values will auto set.
1	Force All Queue Transactions to Memory	When set, this bit disables the internal tail to head transfer path within the queue. All indexes will proceed into memory before being brought to the head of the queue. This effectively preserves the operational history in memory. However, some caution is warranted since four full entries are required for a write to memory. This could cause indexes to get "stuck" at the back of the queue. When this residue occurs, a zero pointer is returned even though the operation might have otherwise returned a valid pointer.
0	Diagnostic Mode	When set, pools is in diagnostic mode. When cleared, pools is in normal mode. When in diagnostic mode, state machines are held in idle. If they are already active, when they next go to idle they will hold there.

9.17: POOLS Buffer Threshold Registers 0-4

The POOLS Buffer Threshold Registers 0-4 is the threshold set by the software to set the threshold crossed bit in the POOLS Status Register. This register is used to compare with the queue length register.

This register consists of a 16-bit count match. The threshold count is compared to the queue length count. If the queue length is less than the value in this register, the appropriate bit is set in the status register respective to this queue.

Length	16 bits	
Type	Read/Write	
Address	Buffer Size 0	XXXX 3088
	Buffer Size 1	XXXX 308C
	Buffer Size 2	XXXX 3090
	Buffer Size 3	XXXX 3094
	Virtual Packets/ Buffer Size 4	XXXX 3098
Power on Value	X'0000'	
Restrictions	None	

9.18: POOLS Index Threshold Registers 0-4

The POOLS Index Threshold Registers 0-4 provide error checking. These are the thresholds set by the software or hardware to set the index threshold crossed bit in the POOLS Status Register. This register is used to check indexes during free operations to look for an out of bounds index.

Each register consists of a 16-bit compare value. The threshold count is compared to the index while being processed. If an index is greater than the value in this register, the appropriate bit is set in the status register.

Length	16 bits	
Type	Read/Write	
Address	Buffer Size 0	XXXX 30F0
	Buffer Size 1	XXXX 30F4
	Buffer Size 2	XXXX 30F8
	Buffer Size 3	XXXX 30FC
	Virtual Packets/ Buffer Size 4	XXXX 3100
Power on Value	X'0000'	
Restrictions	None	

9.19: POOLS Last Primitive Trap Register

The POOLS Last Primitive Trap Register provide debug assistance. It contains the 32-bit last primitive address, and it is the last primitive address to POOLS, as selected in the POOLS Control Register, while in operational mode.

Length	32 bits
Type	Read
Address	XXXX 30E8
Power on Reset values	X'00 00 00 00'
Restrictions	None

9.20: POOLS Last Buffer Map Read on Free Register

The POOLS Last Buffer Map Read on Free Register provide debug assistance. It contains the 32-bit address of the buffer map used in the last free operation, and it is the address of the last buffer map read on a free.

Length	32 bits
Type	Read
Address	XXXX 30EC
Power on Reset values	X'00 00 00 00'
Restrictions	None

9.21: POOLS Error Lock Enable Register

The POOLS Error Lock Enable Register provides the ability to halt pools when the corresponding status bit in the status register are set.

When a bit in this register that corresponds to a bit that is set in the status register, the state machines in pools will be held in idle state until the lock is disabled.

See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	21 bits
Type	Clear/Set
Address	XXXX 30D8 and DC
Power on Reset value	X'00 F8 00'
Restrictions	None

9.22: POOLS Packet and Control Memory Access Threshold

The POOLS Packet and Control Memory Access Threshold timers are used to help limit the amount of time that pools can be held off from its respective memory.

The bits are a 12-bit count. When the proper bit in the POOLS Control Register is set and a request is made to the requisite memory, a counter is loaded with this value. The counter will then count down to zero in 30 ns ticks. When it hits zero, it forces the request to high priority.

Length	12 bits	
Type	Read/Write	
Address	Packet Memory Timer Threshold	XXXX 30E4
	Control Memory Timer Threshold	XXXX 30E0
Power on Reset value	X'080'	
Restrictions	None	

9.23: POOLS Buffer Map Group

The POOLS Buffer Map Group holds the buffer map of the packet that is in the process of being freed. From this map, the pool ID and the indexes that have been used are returned to their correct queue.

This register consists of a 16-bit flag field and 16-bit indexes.

The flag field contains the pool id and the valid bit. When a packet is freed, the valid bit is set to ’0’. When a get operation occurs, the valid bit is then set. This helps to find address duplicates and other address related problems that software can generate.

Length	32 bits		
Type	Read/Write		
Address	**Upper 16 Bits:**	**Lower 16 Bits:**	
	Flag Field 0	Index 0	XXXX 309C
	Index 1	Index2	XXXX 30A0
	Flag Field 1	Index 3	XXXX 30A4
	Index 4	Index 5	XXXX 30A8
	Flag Field 2	Index 6	XXXX 30AC
	Index 7	Index 8	XXXX 30B0
	Flag Field 3	Index 9	XXXX 30B4
	Index 10	Index 11	XXXX 30B8
	Flag Field 4	Index 12	XXXX 30BC
	Index 13	Index 14	XXXX 30C0
Power on Reset value	X'FFFFFFFF'		
Restrictions	None		

Transmit Data Path Entities

Entity 10: Transmit Buffer (CSKED)

The transmit cell scheduler entity is responsible for receiving a packet from the processor, determining when cells from the packets need to be transmitted, and passing this information to the segmentation buffer entity.

The logic consists of timers and counters for determining transmit opportunities and interfaces to ARBIT (for accessing the timing data and descriptors), PCINT (for register accesses), RXQUE (for queuing events), POOLS (for returning buffers when finished transmitting), and SEGBF (for getting the data from memory to transmit).

Scheduling Overview

This entity provides traffic shaping to ensure that traffic sent by the IBM3206K0424 conforms to the Quality of Service (QoS) parameters as defined by the ATM Forum. CSKED provides support for the following QoS parameters:

- Peak Cell Rate (PCR) - The maximum number of cells per second that the connection can transfer into the network.
- Sustained Cell Rate (SCR) - The average number of cells per second that the connection can transfer into the network. The burst tolerance determines the length of time over which the network measures this average.
- Burst Tolerance - The maximum length of time that the user can transfer at the peak cell rate. Burst Tolerance can be measured in number of cells, a measurement known as maximum burst size (MBS).

CSKED will send out cells at the SCR. If transmit opportunities are missed, as is the case when there is no data to send, the actual rate will become less than SCR. When data becomes available to send, CSKED will transmit up to MBS cells at the PCR, until the transmit rate returns to SCR.

Operational Description

LCD Initialization

A Logical Channel Data Structure (LCD) containing scheduling parameters for the circuit must be initialized before segmentation can be started. The parameters that are important to the operation of this entity are:

average_interval	This field contains the minimum average spacing allowed between cells transmitted on this connection. It is the reciprocal of the Sustainable Cell Rate (SCR), as specified in the ATM Forum Traffic Management Specification. The value for this field is expressed in slot times. The length of time for a slot is defined by the Timeslot Prescaler Register and should normally be set to one cell time.
peak_interval	This field contains the minimum spacing allowed between consecutive cells on this connection. It is the reciprocal of the Peak Cell Rate as specified in the ATM Forum Traffic Management Specification. This spacing is also expressed in slot times. A connection that can transmit every slot time would have a value of '1' for this field.
max_burst_value and max_burst_mult	The values in these fields are used to limit the number of cells that can be transferred at the peak rate. The max_burst_value will be multiplied by four to the power of the max_burst_mult to yield the maximum credit time. This time is expressed in slot times and represents the time it would take to acquire the maximum number of cell credits. This maximum credit time should equal the average interval minus the peak interval, multiplied by the maximum number of cells (MBS) that can be transferred at the peak rate.
transmit_priority	This field specifies the priority of transmission on this connection. Three levels of priority are available. Connections needing the highest quality of service, such as a CBR connection, should use the highest priority. Connections with the lowest quality of service requirements, such as a UBR connection, should use the lowest priority.
drop	Data can be sent on up to four physical drops. This field specifies the drop for this connection.
max_resolution	If this bit is set, the lower eight bits of the average interval and peak interval parameters contain a fractional component. This allows a finer resolution for scheduling. For example, for a peak interval of 1.5 time units, the value written to the peak_interval field should be hex 0180. If this bit is set, the initial value of timestamp should contain the current timeslot counter shifted 16 bits to the left.

See *Transmit Logical Channel Descriptor Data Structures* on page 66 for further information.

A Scheduling Example

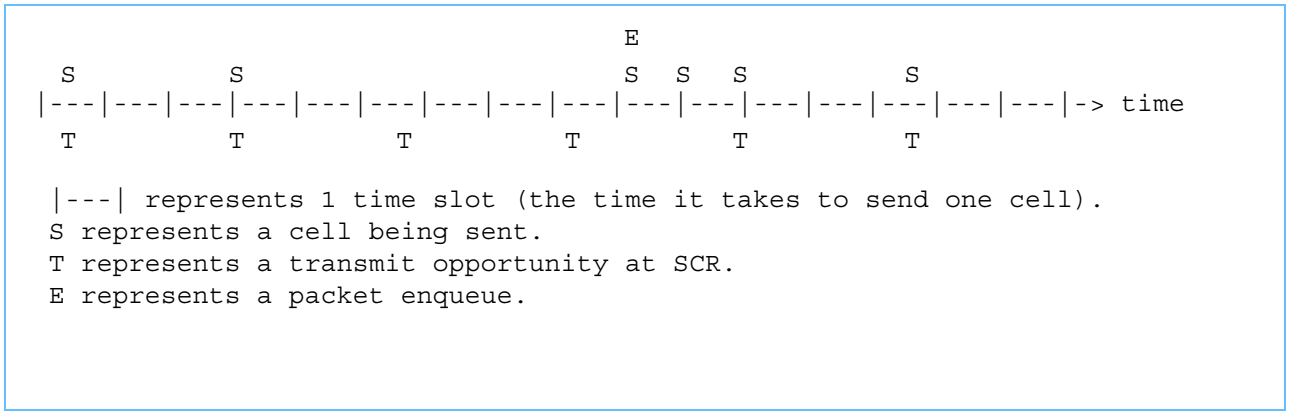
If a connection is to have an SCR of 50 Mbps and a PCR at the line rate of 150 Mbps and a MBS of 10 cells, the LCD needs to be initialized as follows:

- average_interval = 150 Mbps/50 Mbps = 3
- peak_interval = 150 Mbps/150 Mbps = 1
- max_burst_value = 10*(3-1) = 20

The following example uses a timeline to show how a connection with these parameters is scheduled. Cells are sent every third slot while there is data to send. After the first two cells are sent there is no more data to

send until another packet is enqueued. For each missed transmit opportunity, a cell can be sent at the peak interval, which is one. In the 15 timeslots after the first cell, five cells are sent for an SCR of 50 Mbps. The burst size is three, which is less than MBS. The unfilled slots can be used by other connections.

Timeline Example of Scheduling



CSKED Initialization

Before packets are enqueued for transmission, in addition to initializing the above scheduling parameters in the LCD, the following registers need to be set up.

- Timeslot Prescaler Register - The amount of time for one timeslot is defined by this register. It defaults to 707 ns which is one cell time on a 622 Mbps Sonet interface.
- CSKED Control Register - Additional scheduling options such as number of physical drops needs to be set in this register.

Packet Initialization

Packets to be segmented are written to Packet Memory, which has been allocated by POOLS. The address of the LCD describing the channel that this packet is to be transmitted on must be written to the header of the packet. Packet segmentation is started by issuing the transmit enqueue primitive to this entity. This entity will schedule segmentation of the packet according to the parameters set up in the LCD.

Scheduling Options

ABR Scheduling

CSKED has logic to assist in the processing of ABR connections. If the connection is ABR, the LCD will have a different configuration, as specified in *Transmit Logical Channel Descriptor Data Structures* on page 66. The following fields need to be initialized before the packets are sent on the connection.

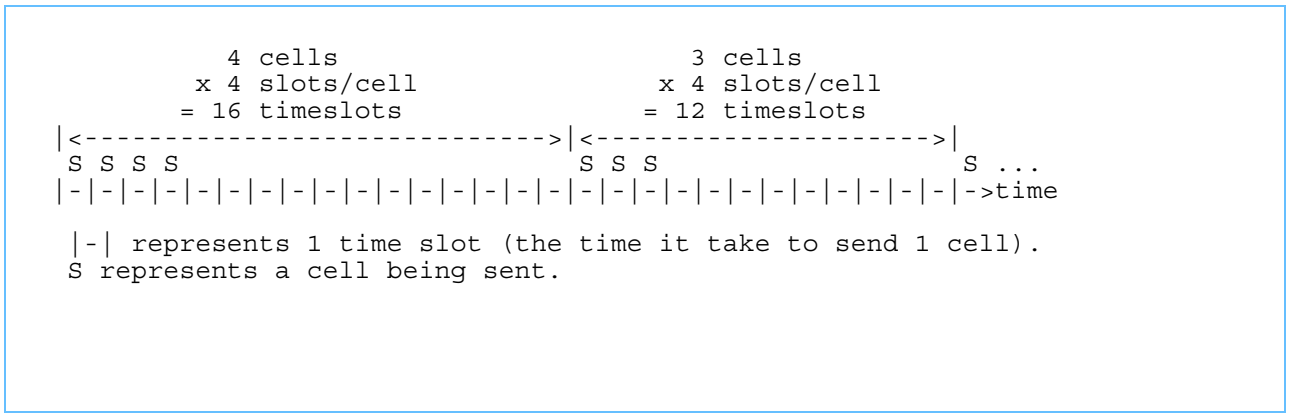
- **Scheduling type** - This field must be set to the value specifying an ABR connection.
- **Nrm -** This field should specify the maximum number of cells a source may send for each forward RM-cell. Number of cells = (2**Nrm)+1.
- **Trm** - This field provides an upper bound on the time between forward RM-cells for an active source. Time = 100*(2**-Trm) msec.
- **ADTF** - The ACR Decrease Time Factor is the time permitted between sending RM-cells before the rate is decreased to ICR. Time = ADTF* 0.01 msec.
- All other ABR fields should be initialized to ’0’.

Frame Scheduling

CSKED has logic to support frame-based scheduling. It is enabled whenever the PHY type is configured for POS-PHY in LINKC. In frame-based scheduling the packet is sent out at the line rate, but the start time of the next frame is determined by multiplying the peak interval by the number of 64-byte blocks in the packet. The average portion of the bandwidth used by a connection will be 1/(peak interval).

For example, if a connection is to use 1/4 of the bandwidth, the peak interval should be set to four. The frames will be sent out at line rate, but the spacing between the start of each frame will be four timeslots for each cell sent from the packet, so on average 1/4 of the bandwidth will be used by the connection. The following timeline example depicts sending two packets, the first contains four cells and the second contains three cells. The unfilled slots can be used by other connections.

Timeline Example of Frame Scheduling



The above example assumes that one slot time is initialized to the time it takes to send 64 bytes out on the line. The term "cells" was used in this example to mean a 64-byte block of packet data. In frame mode, the ATM header is not prepended to the data being sent.

Weighted fair queueing on a frame basis is supported on the low priority queue by setting bit 17 in the CSKED Control Register. When using frame-based scheduling and weighted fair queuing together, the average interval will be used to limit the spacing between packets, not cells.

Path Scheduling

CSKED has logic to support sharing scheduling parameters between multiple connections. In path scheduling, an LPD is set up to contain the scheduling parameters for the group of connections in the same way it is done for LCDs. All connections that wish to share this bandwidth set the alter_sched field in their LCD to indicate this VC is on a VP, and initialize the lpd_pointer field to point to the LPD. The segmentation portion of the LPD is not used since the segmentation parameters are taken from the LCD. The scheduling parameters in the LCDs are not used as they are in the LPD. The bandwidth is shared on a packet or cell basis depending the value of the alter_sched field in the LCD. Since both the LPD and LCD need to be fetched for each transmit opportunity, this scheduling method should not be used where the performance boundaries are being pushed, as in 622 Mbps. See *Transmit Logical Channel Descriptor Data Structures* on page 66 for further information on LPD descriptors.

Primitives

Enqueue

After a packet has been written to memory and the packet header updated with the offset and length of the data and the LCD address of the connection, an enqueue primitive needs to be issued to the Transmit Enqueue Primitive address.

Close Connection

When no more traffic is to be sent on a connection, this primitive can be executed to cause an event to be generated when segmentation has stopped on this connection. Segmentation will be stopped immediately, or stopped after all packets on this connection have been transmitted as specified in the CSKED Control Register.

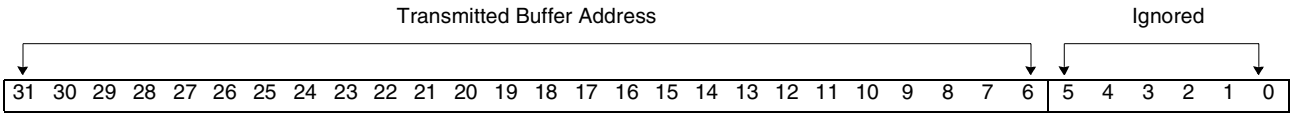
Start/Stop Timer

When this primitive is executed, a timer is started or stopped whose parameters are contained in the specified LCD. When the timer pops, a DMA descriptor specified in the LCD will be executed.

10.1: Transmit Enqueue Primitive

Enqueues a buffer for transmission.

Length: 32 bits
Type Write Only
Address XXXX 1200
Power On Value X'0000 0000'
Restrictions None

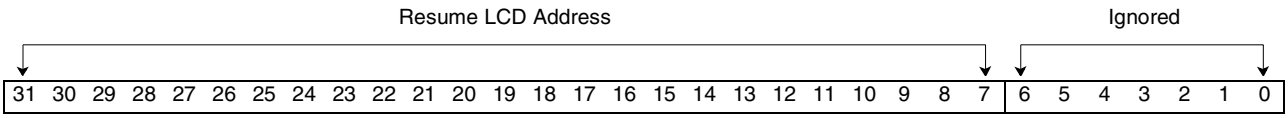


Bit(s)	Description
31-6	This must contain the address of the buffer to be transmitted. Buffers must be aligned on at least 64 byte boundaries. The lower six bits are ignored.
5-0	Ignored

10.2: Resume Transmission Primitive

Resumes transmission on an ABR connection that has been suspended. On an ABR connection, ADTF, CRM, and CCR=0 events will cause the transmission to be suspended until a rate conversion is completed, normally by the internal processor. This primitive will resume transmission on those connections, once the rate conversion is completed.

Length 32 bits
Type Read/Write
Address XXXX 1204
Power On Value X'0000 0000'
Restrictions This address should be written with care. This primitive should only be used on connections that have been suspended.

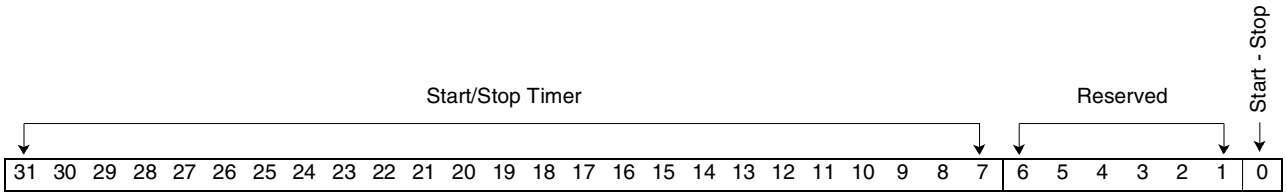


Bit(s)	Description
31-7	This must contain the address of the LCD that is to resume transmission. The lower seven bits are ignored.
6-0	Ignored

10.3: Start/Stop Timer Primitive

Start or stop a timer with the parameters in the specified LCD address. When this primitive is executed, a timer is started or stopped whose parameters are contained in the specified LCD. Bit 0 specifies whether to start (0) or stop (1) the timer. When the timer pops, a DMA descriptor specified in the LCD will be executed.

Length: 32 bits
Type: Read/Write
Address: XXXX 1208
Power On Value: X'0000 0000'
Restrictions: None

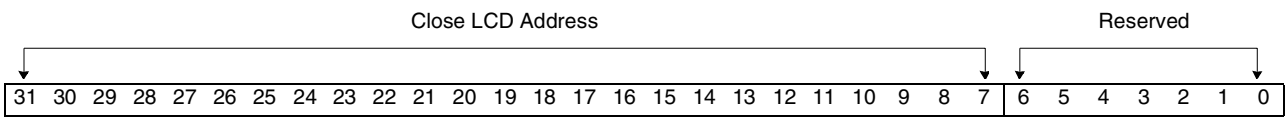


Bit(s)	Description
31-7	This must contain the address of the LCD that contains the timer parameters.
6-1	Reserved.
0	This bit specifies whether the timer is to be started (0) or stopped (1).

10.4: Close Connection Primitive

Transmission is complete on a connection specified by an LCD address. When no more traffic is to be sent on a connection, this primitive can be executed to cause an event to be generated when segmentation has stopped on this connection. Segmentation will be stopped immediately, or stopped after all packets on this connection have been transmitted, as specified in the CSKED Control Register.

Length 32 bits
Type Read/Write
Address XXXX 120C
Power On Value X'0000 0000'
Restrictions None



Bit(s)	Description
31-7	This must contain the address of the LCD that is to be closed. The lower seven bits are ignored.
6-0	Reserved.

10.5: Timeslot Prescaler Register

This register determines the length of time for one timeslot. This controls the rate that the cell scheduling counters are incremented. Each clock cycle, the value in this register is added to a 24-bit counter. When the upper bit of the counter changes state, the Current Timeslot Counter is incremented. This should normally be set to the time it takes to transmit one cell. It will be initialized to the cell time for a 622 Mb/s SONET connection (599.04 Mb/s payload). The following formula should be used to determine the value to load in this register: Timeslot Prescaler = (clock interval/timeslot interval) x 2**23.

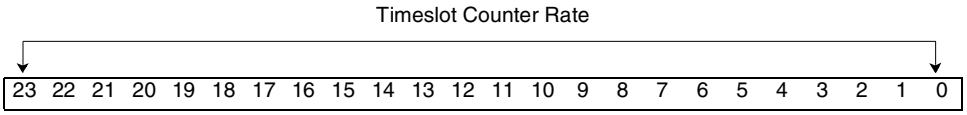
Length 24 bits

Type Read/Write

Address XXXX 1210

Power On Value X'02B67C'

Restrictions This register should be written only at initialization time.



Bit(s)	Description
23-0	This value will determine the rate at which the Current Timeslot Counter is advanced.

10.6: Current Timeslot Counter

This counter contains a count of how many prescaled intervals have elapsed. It is used to determine if scheduling needs to be done or credits exist.

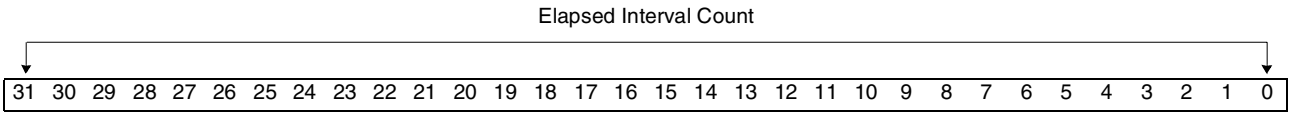
Length 32 bits

Type Read/Write

Address XXXX 1218

Power On Value X'0000 0000'

Restrictions This register is meant to be read only. It is writable for diagnostic purposes only.



Bit(s)	Description
31-0	This value represents how many expirations have occurred since the counter rolled over.

10.7: CSKED Control Register

This register is used to control the actions of CSKED. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

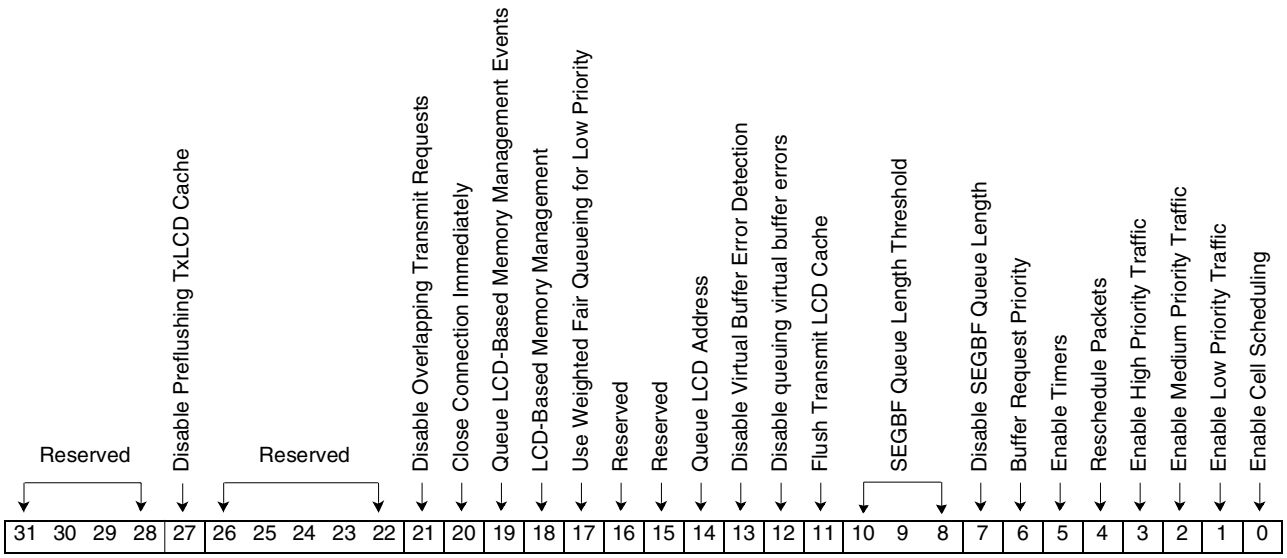
Length 32 bits

Type Read/Write

Address XXXX 1220 and 224

Power On Value X'0759'

Restrictions None



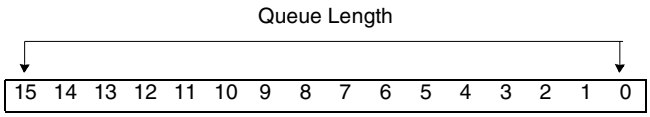
Bit(s)	Name	Description
31-28	Reserved	Reserved
27	Disable preflushing txLCD cache	This bit disables automatically, flushing a cache line when all lines are dirty
26-22	Reserved	Reserved
21	Disable overlapping transmit requests	This bit is meant for debug purposes only. It will disable the ability of CSKED to overlap requests to SEGBF.
20	Close connection immediately	Setting this bit will cause segmentation to stop immediately on any connection that has been issued a close connection primitive. If this bit is not set, an event will be generated after all traffic queued to this connection has been sent.
19	Queue LCD-based memory management events	Setting this bit will cause LCD-based memory management events to be queued to the Transmit Complete Queue, if enabled by bit 18 of this control register. If LCD-based memory management is enabled and this bit is off, the receive pool ID associated with this connection will be updated when a threshold is crossed.
18	LCD-based memory management	Setting this bit will enable LCD-based memory management for received packets. See *Definition of LCD-Based Memory Management of Transmit LCD* on page 73 for further information on this function.

Bit(s)	Name	Description
17	Use weighted fair queueing for low priority	Setting this bit will cause low priority traffic to be scheduled using weighted fair queueing. The peak interval in the LCD is used to provide a relative weight in determining the amount of bandwidth the connection will use. For example, a peak interval of one will use twice the bandwidth as a connection with a peak interval of two. The average interval specifies the maximum rate that the connection can use. For example if the average interval is set to two, the maximum rate at which it can send a cell is every two timeslot times (as defined in the Timeslot Prescaler Register).
16	Reserved	Reserved.
15	Reserved	Reserved.
14	Queue the LCD address if freeing and queueing	Setting this bit will cause the LCD address, instead of the packet address, to be queued if both freeing and queueing on transmit are complete.
13	Disable virtual buffer error detection	Setting this bit will cause the buffer enqueue logic to ignore virtual buffer errors.
12	Disable queuing virtual buffer errors	If virtual buffer error detection is not disabled, detected errors will be queued. If this bit is set, this queueing is disabled and the buffer will be freed.
11	Flush Transmit LCD Cache	If this bit is set, the transmit LCD cache will be flushed. This bit will be reset after the cache has been flushed. Flushing the cache should not be needed in normal operation.
10-8	SEGBF Queue Length Threshold	Cells can be queued in SEGBF up to the number specified in this register. The default is seven, which is above the limit for pass two. Writing these bits to ’0’ will also disable this function.
7	Disable SEGBF Queue Length in Scheduling	CSKED will normally include SEGBFs queue length in the calculations when rescheduling a cell. If this bit is on it will disable this function and the cell will be scheduled as if the cells were transferred when SEGBF accepted the cells.
6	Priority of buffer requests	If this bit is not set, scheduling requests have a higher priority than buffer requests. If this bit is set this priority is reversed. It should be set if a significant percentage of packets are only a few cells in length.
5	Enable timers	Timer descriptors can be enqueued to this entity that will cause a DMA descriptor to be executed on expiration. If these timers are used, this bit must be set. If they are not used, this bit should be reset.
4	Reschedule Packets in the Slow Queue to the Fast Queue	This function is not implemented in pass one. It is implemented in pass two. If the average or peak interval is greater than 255, the cells will be scheduled in the slow queue. The slow queues will be serviced every 64 pre-scaler time units. This means that a jitter of up to 64 pre-scaler time units should be expected for slow traffic. If this bit is set, packets in the slow queue will be rescheduled at the appropriate time to the fast queue. This will decrease the variation in the scheduling but may cause some performance degradation if traffic is heavy.
3	Enable High Priority Traffic	For each priority enabled 16KB of Control Memory must be reserved for timing data. If only one or two priorities are to be used, bits corresponding to unused priorities should be cleared to improve performance.
2	Enable Medium Priority Traffic	Enable Medium Priority Traffic.
1	Enable Low Priority Traffic	Enable Low Priority Traffic.
0	Enable Cell Scheduling	If this bit is off, no primitives will be handled or cells scheduled.

10.8: Transmit Segmentation Throttle Register

This register contains the number of cycles to wait between successive requests to transmit a cell. Its purpose is to slow segmentation on all VCIs if it is determined by software that the network can not handle the generated load. The value in this register will be loaded into the Transmit Segmentation Counter each time a cell is accepted for transmission. For normal operation the value in this register should be ’0’.

Length	16 bits
Type	Read/Write
Address	XXXX 1230
Power On Value	X’0000’
Restrictions	None



Bit(s)	Description
15-0	When the transmit complete queue length reaches this value an interrupt will be generated.

10.9: Transmit Segmentation Throttle Counter

This register is loaded with the value in the Transmit Segmentation Throttle Register after each cell is accepted for transmission and counts down until it reaches ’0’. A new cell transmission will not be requested until this counter reaches ’0’.

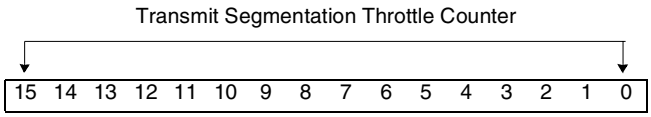
Length 16 bits

Type Read Only

Address XXXX 1234

Power On Value X’0000’

Restrictions Read Only



Bit(s)	Description
15-0	When this counter reaches ’0’, a new cell can be transmitted.

10.10: MPEG Conversion Register

This register is used to convert MPEG time units into timeslot time units. If MPEG traffic is configured in the LCD, the data stream will be monitored for PCRs. If a PCR is detected, it will be scheduled at the time specified in the PCR. A conversion factor needs to be written into this register to convert the MPEG time units into timeslot units. It will be initialized to a value that converts the MPEG time units (90 KHz) into the timeslot units (353.2 KHz, assuming one timeslot is the time it takes to send one cell over a SONET connection). The lower 12 bits of this register contain the fractional portion of this conversion factor.

Example: 353.2076 KHz / 90 KHz = 3.924528 = 3.ECB hex

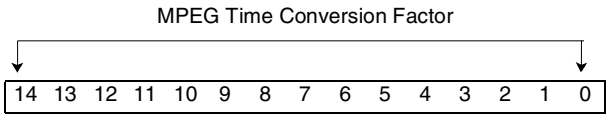
Length 15 bits

Type Read/Write

Address XXXX 125C

Power On Value X’3ECB’

Restrictions None



Bit(s)	Description
14-0	Contains the conversion factor.

10.11: ABR Timer Prescaler Register

This register determines the length of time for a tick of the RM Cell Timer. This controls the rate that the cell scheduling counters are incremented. Each clock cycle, the value in this register is added to a 24-bit counter. When the upper bit of the counter changes state, the RM Cell Timer is incremented. This should be set to value of 0.78 ms. It will be initialized to 0.78 ms assuming a 30-ns clock (as set up in SCLOCK). The following formula should be used to determine the value to load in this register:

ABR Timer Prescaler = (clock interval/0.78 ms) x 2**23.

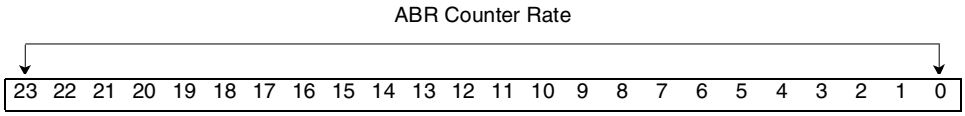
Length 24 bits

Type Read/Write

Address XXXX 127C

Power On Value X'0000A2'

Restrictions This register should be written only at initialization time.



Bit(s)	Description
23-0	This value will determine the rate at which the ABR counter is advanced.

10.12: RM Cell Timer

This register is used to keep track of the last time that an ABR RM cell was sent. Its period should be 0.78 ms.

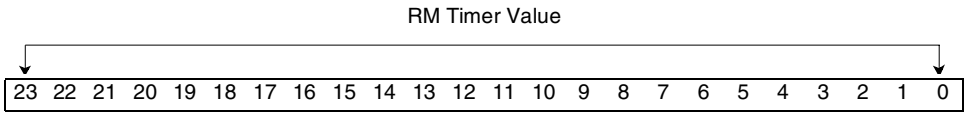
Length 24 bits

Type Read/Write

Address XXXX 126C

Power On Value X'000000'

Restrictions None



Bit(s)	Description
23-0	Timer value.

10.13: CSKED LCD Update Data Registers

Used to specify data to write into the LCD on the update LC operation. These registers contain the data used in the LC Update Operation. For more information on its use, see the *CSKED LCD Update Operation Registers* on page 287.

This register changes to contain the updated data written to the LC word while the operation is completing.

The second set of LCD update registers is meant for the core to use, but is available for general use.

Length	32 bits
Type	Read/Write
Address	Update1 XXXX 1300 Update2 XXXX 130C
Power On Value	X'000000'
Restrictions	None

10.14: CSKED LCD Update Mask Registers

Used to specify data to write into the LCD on the update LC operation. These registers contain the mask used in the LC Update Operation. For more information on its use, see the *CSKED LCD Update Operation Registers* on page 287.

The second set of LCD update registers is meant for the core to use, but is available for general use.

Length	32 bits
Type	Read/Write
Address	Update1 XXXX 1304 Update2 XXXX 1310
Power On Value	X'000000'
Restrictions	None

10.15: CSKED LCD Update Operation Registers

Used to specify the LCD word to update. This operation is used to update a portion of an LCD. If this operation is not used, software or IBM3206K0424 updates of the LCD may be lost because the LCD is cached in IBM3206K0424 while cells are being processed.

This register is written with the address of the LCD word to update. Once this register is written the update operation starts. All subsequent reads or writes to the data, mask, or update registers are held off until the operation completes. A read-modify-write will occur to update the portion specified by the mask with the masked value in the data register.

Normally this register would not be read. However, if it is read then the low order bit is read as '0' and the next lowest order bit (bit 1) is read as the busy bit. This signifies whether an operation is still going on. If an operation is still going on, then a new write to any of the data, mask, or update operation registers is held off until the original operation is complete.

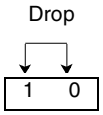
The second set of LCD update registers is meant for the core to use, but is available for general use.

Length	32 bits
Type	Read/Write
Address	Update1 XXXX 1308 Update2 XXXX 1314
Power On Value	X'000000'
Restrictions	The low order two bits are not writable

10.16: Drop Access Control Register

Each drop(4) has registers that can be used for debugging purposes and bandwidth limiting. To conserve address space, this register determines the drop for the register access. These registers are Fast Serviced Counters, Slow Serviced Counters, and Priority Bandwidth Limit Registers. This register must be rewritten whenever values for a different drop need to be read or written.

Length	2 bits
Type	Read/Write
Address	XXXX 1288
Power On Value	X'00000000'
Restrictions	None



Bit(s)	Description
1-0	This value represents the drop number or the above registers that will be accessed.

Performance Registers

This section contains registers that are for performance purposes.

10.17: High Priority Bandwidth Limit Register

This register can be used to limit the bandwidth used by high priority connections. The upper eight bits of this register specifies the number of high priority cells that can be sent in a window specified by the lower eight bits of this register. If no data needs to be sent by lower priority connections, high priority connections will not be limited.

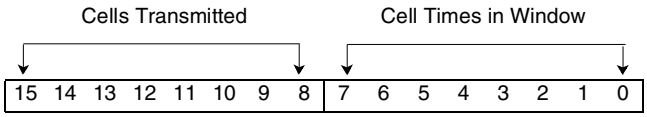
Length 16 bits

Type Read/Write

Address XXXX 1270

Power On Value X'00000000'

Restrictions None



Bit(s)	Description
15-8	This value specifies the number of cells that can be transmitted from high priority connections in one time window.
7-0	This value specifies the number of cell times in the window.

10.18: Medium Priority Bandwidth Limit Register

This register can be used to limit the bandwidth used by medium priority connections. The upper eight bits of this register specify the number of medium priority cells that can be sent in a window specified by the lower eight bits of this register. If no data needs to be sent by low priority connections, medium priority connections will not be limited.

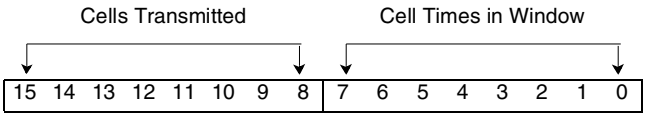
Length 16 bits

Type Read/Write

Address XXXX 1274

Power On Value X'00000000'

Restrictions None



Bit(s)	Description
15-8	This value specifies the number of cells that can be transmitted from medium priority connections in one time window.
7-0	This value specifies the number of cell times in the window.

10.19: Low Priority Bandwidth Limit Register

This register can be used to limit the bandwidth used by low priority connections. The upper eight bits of this register specify the number of low priority cells that can be sent in a window specified by the lower eight bits of this register.

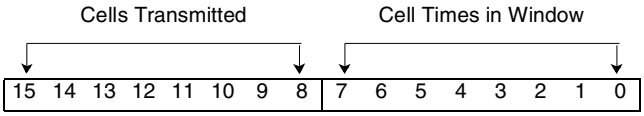
Length 16 bits

Type Read/Write

Address XXXX 1278

Power On Value X'00000000'

Restrictions None

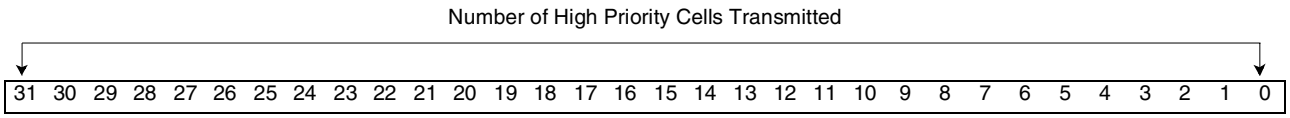


Bit(s)	Description
15-8	This value specifies the number of cells that can be transmitted from low priority connections in one time window.
7-0	This value specifies the number of cell times in the window.

10.20: High Priority Cells Transmitted Counter

This register contains the number of cells transmitted from high priority connections.

Length 32 bits
Type Read/Write
Address XXXX 1260
Power On Value X'00000000'
Restrictions None

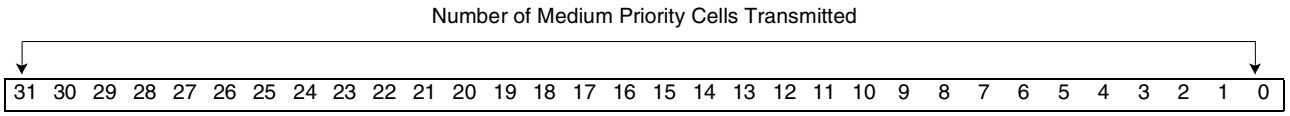


Bit(s)	Description
31-0	This value represents the number of cells transmitted from high priority connections.

10.21: Medium Priority Cells Transmitted Counter

This register contains the number of cells transmitted from medium priority connections.

Length 32 bits
Type Read/Write
Address XXXX 1264
Power On Value X'00000000'
Restrictions None



Bit(s)	Description
31-0	This value represents the number of cells transmitted from medium priority connections.

10.22: Low Priority Cells Transmitted Counter

This register contains the number of cells transmitted from low priority connections.

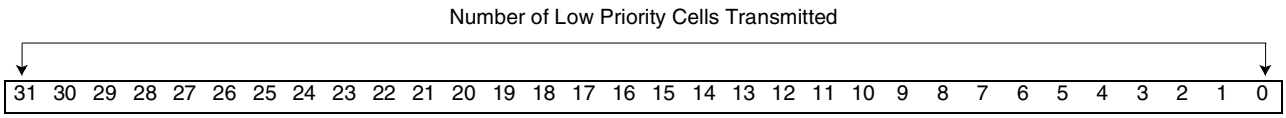
Length 32 bits

Type Read/Write

Address XXXX 1268

Power On Value X'00000000'

Restrictions None



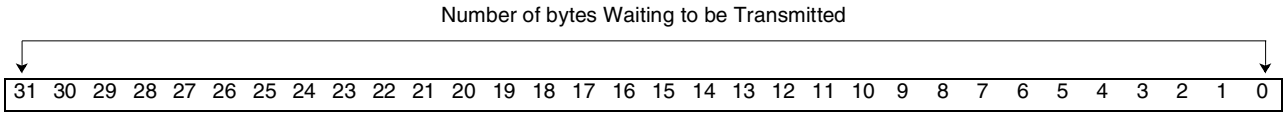
Bit(s)	Description
31-0	This value represents the number of cells transmitted from low priority connections.

10.23: Bytes Queued Counters

These registers (12) contain the number of bytes queued for transmission for each priority (3) on each drop (4). The addresses are assigned to the range in the following order:

High priority	Port 0
High priority	Port 1
High priority	Port 2
High priority	Port 3
Medium priority	Port 0
Medium priority	Port 1
Medium priority	Port 2
Medium priority	Port 3
Low priority	Port 0
Low priority	Port 1
Low priority	Port 2
Low priority	Port 3

Length	32 bits
Type	Read/Write
Address	XXXX 1290-2BC
Power On Value	X'00000000'
Restrictions	None



Bit(s)	Description
31-0	This value represents the number of bytes waiting to be transmitted for each priority on each drop.

Debugging Register Access

This section contains registers that are for debug purposes only. These registers need not be written or read during normal operations.

10.24: Fast Serviced Counters

There are three fast serviced counters, one for each transmit priority: high, medium, and low. These registers contain the value of the last fast time slot that has been serviced. When this count differs from the current timeslot count, at least one fast slot needs servicing. Each time the fast slot is serviced, this counter will increment.

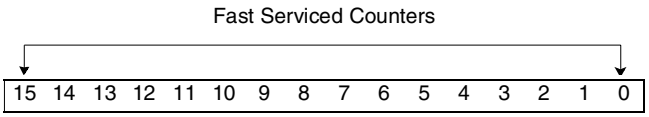
Length 16 bits

Type Read/Write

Address XXXX 1240 XXXX 1244 XXXX 1248

Power On Value X'0000'

Restrictions This register is meant to be read only. It is writable for diagnostic purposes only.



Bit(s)	Description
15-0	This value represents how many times this time wheel has been serviced since the counter rolled over.

10.25: Slow Serviced Counters

There are three slow serviced counters, one for each transmit priority: high, medium, and low. These registers contain the value of the last slow time slot that has been serviced. When this count differs from the current timeslot count, at least one slow slot needs servicing. Each time the slow slot is serviced, this counter will increment.

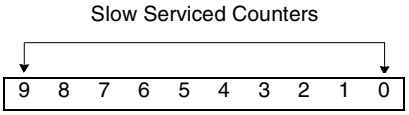
Length 10 bits

Type Read/Write

Address XXXX 124C XXXX 1250 XXXX 1254

Power On Value X'000'

Restrictions This register is meant to be read only. It is writable for diagnostic purposes only.

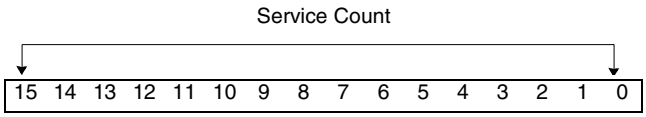


Bit(s)	Description
9-0	This value represents how many times this slow wheel has been serviced since the counter rolled over.

10.26: Timer Serviced Counters

In addition to the counters above, there is an additional counter for processing timer requests. These registers contain the value of the last timer slot that has been serviced. When this count differs from the current timeslot count (bits 22-15), at least one slow slot needs servicing. Each time a timer slot is serviced, this counter will increment.

Length	16 bits
Type	Read/Write
Address	XXXX 1280
Power On Value	X'00'
Restrictions	This register is meant to be read only. It is writable for diagnostic purposes only.



Bit(s)	Description
15-0	This value represents how many times this timer wheel has been serviced since the counter rolled over.

10.27: CSKED Status Register

This register is used to control the actions of CSKED. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

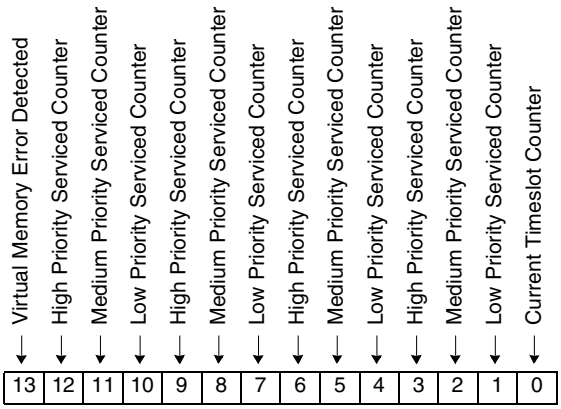
Length 14 bits

Type Clear/Set

Address XXXX 1228 and 22C

Restrictions None

Power On Value X'0'



Bit(s)	Name	Description
13	Virtual memory error detected	If a virtual memory write operation could not complete because a real buffer was not available, a signature is written to the packet header. If this signature is detected when the buffer is enqueued for transmission, this bit will be set and an event will be posted to RXQUE.
12	High priority serviced counter	High priority serviced counter has overrun on drop 3.
11	Medium priority serviced counter	Medium priority serviced counter has overrun on drop 3.
10	Low priority serviced counter	Low priority serviced counter has overrun on drop 3.
9	High priority serviced counter	High priority serviced counter has overrun on drop 2.
8	Medium priority serviced counter	Medium priority serviced counter has overrun on drop 2.
7	Low priority serviced counter	Low priority serviced counter has overrun on drop 2.
6	High priority serviced counter	High priority serviced counter has overrun on drop 1.
5	Medium priority serviced counter	Medium priority serviced counter has overrun on drop 1.
4	Low priority serviced counter	Low priority serviced counter has overrun on drop 1.
3	High priority serviced counter	High priority serviced counter has overrun on drop 0.
2	Medium priority serviced counter	Medium priority serviced counter has overrun on drop 0.
1	Low priority serviced counter	Low priority serviced counter has overrun on drop 0.
0	Current timeslot counter	Current timeslot counter has wrapped. If this bit is on, the timeslot counter has wrapped.

10.28: CSKED Interrupt Enable Register

This register is used to enable interrupts from CSKED. If a bit is on in the status register and the corresponding enable bit is on in this register then an interrupt will be generated if enabled in INTST. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

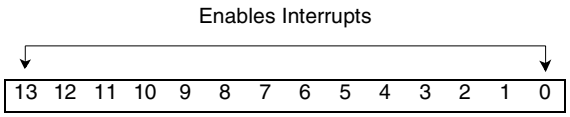
Length 14 bits

Type Clear/Set

Address XXXX 1238 and 23C

Power On Value X'0'

Restrictions None



Bit(s)	Description
13-0	Enables interrupts

10.29: CSKED Timing Data Array Pointer

The CSKED Timing Data Array contains data relevant to scheduling cells. It contains 96 32-bit words. It should only be written for diagnostic purposes to test the array. It will power up to all zeros and should be rewritten to zeros after the array has been tested. This register points to an offset in the CSKED timing data array for accesses from the PCI bus. This register must be loaded with the correct offset before the desired data can be read from the CSKED timing data array data register. This register will auto-increment after each access to the associated data register. When the last address is accessed, this register wraps to zero.

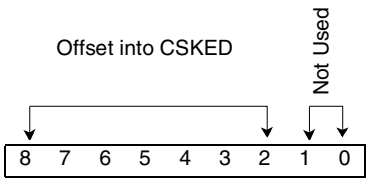
Length 9 bits

Type Read/Write

Address XXXX 12C0

Power On Value X'000'

Restrictions None



Bit(s)	Description
8-2	These bits provide an offset into the CSKED cell staging array for accesses from the PCI bus. These bits provide access to the 256 unique four-byte locations in the array. Accessing the last location in the array will cause the address to wrap back around to the beginning of the array.
1-0	These bits are not implemented. They cannot be written and will always return '0' when read.

10.30: CSKED Timing Data Array Data

This register is used to access the array pointed to by the CSKED Timing Data Array Pointer.

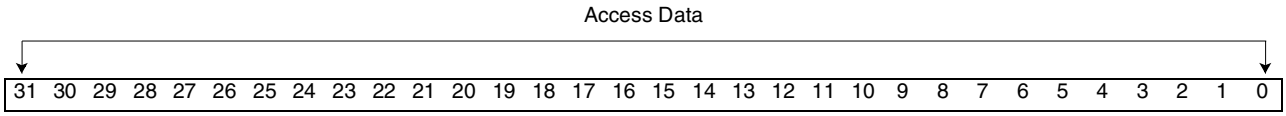
Length 32 bits

Type Read/Write

Address XXXX 12C4

Power On Value

Restrictions Should be written for diagnostic use only. Initialize back to ’0’s when through testing.



Bit(s)	Description
31-0	Access data

10.31: CSKED Time Wheel Array Pointer

The CSKED Time Wheel Array contains data relevant to scheduling cells. It contains 16K 19-bit words. It should only be written for diagnostic purposes to test the array. It will power up to all zeros and should be rewritten to zeros after the array has been tested. This register points to an offset in the CSKED time wheel array for accesses from the PCI bus. This register must be loaded with the correct offset before the desired data can be read from the CSKED Time Wheel Array Data register. This register will auto-increment after each access to the associated data register. When the last address is accessed, this register wraps to zero.

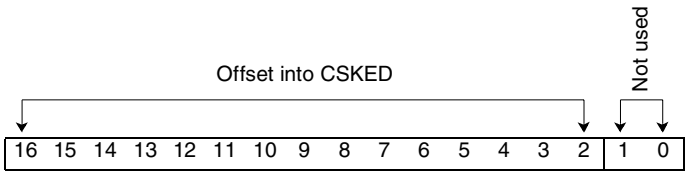
Length 17 bits

Type Read/Write

Address XXXX 12C8

Power On Value X’000’

Restrictions None



Bit(s)	Description
16-2	These bits provide an offset into the CSKED cell staging array for accesses from the PCI bus. These bits provide access to the 16K unique 19-bit locations in the array. Accessing the last location in the array will cause the address to wrap back around to the beginning of the array.
1-0	These bits are not implemented, they can not be written and will always return ’0’ when read.

10.32: CSKED Time Wheel Array Data

This register is used to access the array pointed to by the CSKED Time Wheel Array Pointer.

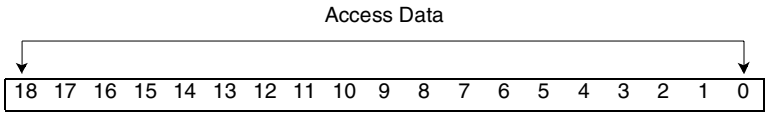
Length 19 bits

Type Read/Write

Address XXXX 12CC

Power On Value

Restrictions Should be written for diagnostic use only. Initialize back to zeros when through testing.



Bit(s)	Description
18-0	Access data

10.33: CSKED LCD Cache Array Pointer

The CSKED LCD Cache Array contains the transmit portion of LCDs used by CSKED and SEGBF. It contains 64 32-bit words. It should only be written for diagnostic purposes to test the array. It will power up to all zeros and should be rewritten to zeros after the array has been tested. This register points to an offset in the LCD Cache array for accesses from the PCI bus. This register must be loaded with the correct offset before the desired data can be read from the LCD Cache Array Data register. This register will auto-increment after each access to the associated data register. When the last address is accessed, this register wraps to zero.

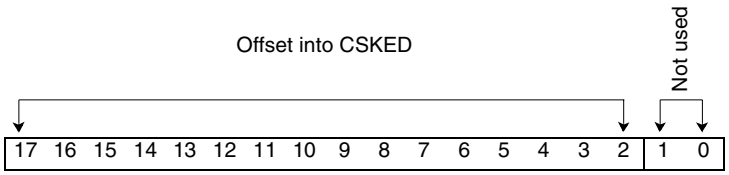
Length 18 bits

Type Read/Write

Address XXXX 1318

Power On Value X'000'

Restrictions None



Bit(s)	Description
17-2	These bits provide an offset into the CSKED cell staging array for accesses from the PCI bus. These bits provide access to the 64 unique 32-bit locations in the array. Accessing the last location in the array will cause the address to wrap back around to the beginning of the array.
1-0	These bits are not implemented. They can not be written and will always return '0' when read.

10.34: CSKED LCD Cache Array Data

This register is used to access the array pointed to by the CSKED LCD Cache Array Pointer.

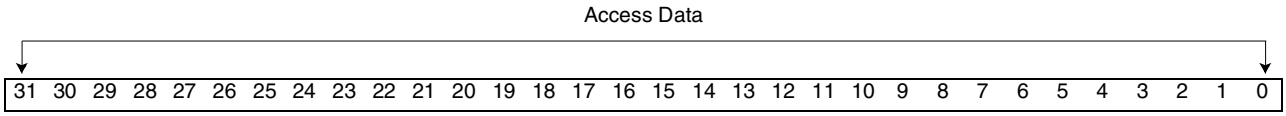
Length 32 bits

Type Read/Write

Address XXXX 131C

Power On Value

Restrictions Should be written for diagnostic use only. Initialize back to zeros when through testing.



Bit(s)	Description
31-0	Access data

10.35: CSKED Congestion Control Register

CSKED can halt scheduling on up to 160 groups of connections. A connection specifies which group it belongs to by encoding the QNR(0-39) and DP(0-3) bits in the LCD. This register contains 160 bits which specify which of the 160 groups should be halted. Bit xx in the CSKED control register must be set in order to write to these bits from software. This register is read in eight groups of 20 bits. The group is specified by bits four through two of the address, with increasing addresses corresponding to increasing group numbers.

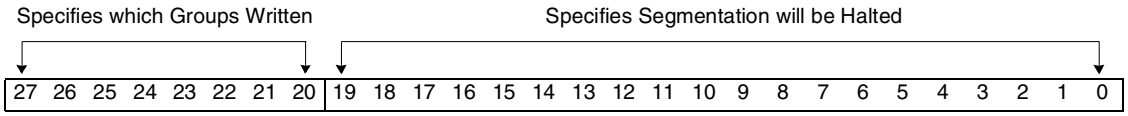
Length 28 bits

Type Read/Write

Address XXXX 120F0-2FC

Power On Value X'000'

Restrictions None

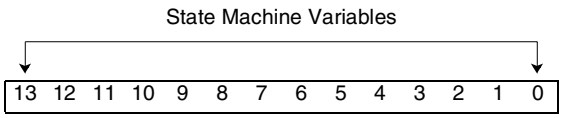


Bit(s)	Description
27-20	These bits specify which eight groups of 20 bits will be written when writing this register. From one to all eight groups can be written at the same time, but all will be written with the same data specified in bits 19-0.
19-0	When set, these bits specify that segmentation will be halted for all connections associated with the corresponding group.

10.36: State Machine Variables

This register contains the current state of the three main state machines in this entity.

Length	14 bits
Type	Read
Address	XXXX 1258
Power On Value	X'0000'
Restrictions	None



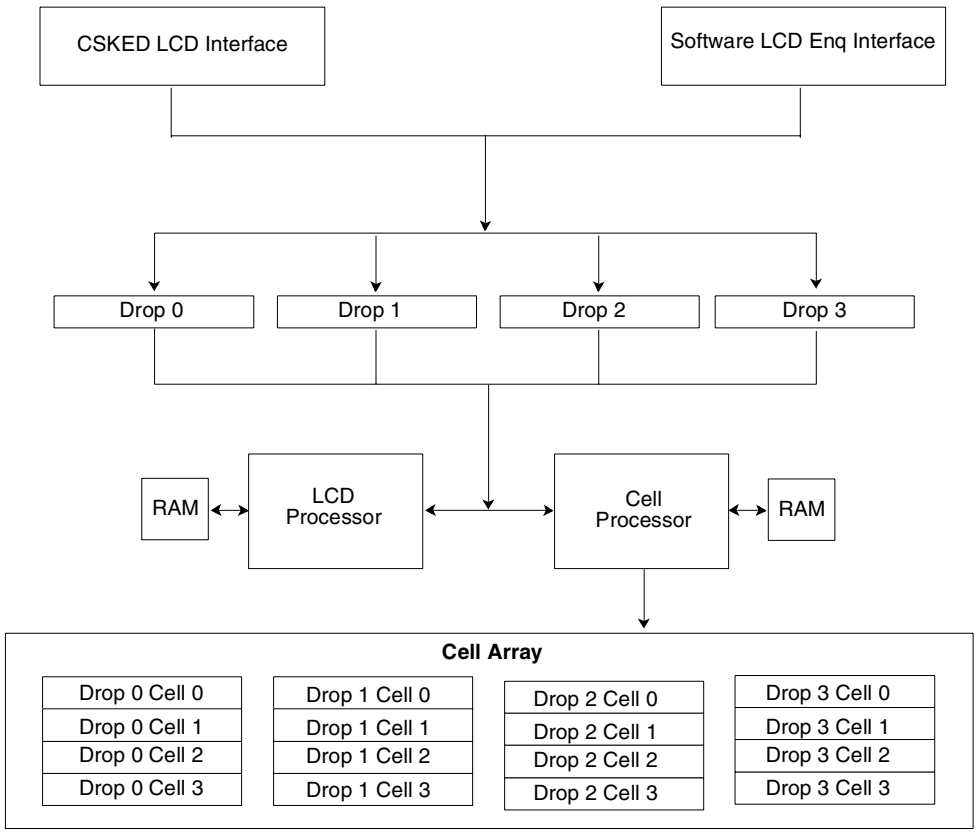
Bit(s)	Description
13-0	Value of state machine variables.

Entity 11: ATM Transmit Buffer Segmentation (SEGBF)

The segmentation buffer entity (SEGBF) accepts frames from the cell scheduler (CSKED) or software, and then generates ATM cells to send out over the external physical interface. This entity knows or cares nothing about scheduling cells over time; it will simply construct a cell when it is provided an address of a logical circuit descriptor to operate on. All rate and scheduling concerns must be addressed by the CSKED logic or software prior to queueing a frame to SEGBF.

The SEGBF logic consists of four input LCD address latches, two specialized processors and the associated ROM/RAM for each, a 16-cell array buffer, and various support logic. The input latches and cell buffers are logically divided into four different drops, with each drop consisting of an input latch and four cell buffers. Under normal operation (CSKED providing LCD), the drop is defined by the drop field in the transmit LCD. When an LCD is enqueued by software, data bits five and six of the enqueued address define which drop the enqueued LCD is associated with. The four logical drops in SEGBF can be mapped to any of the physical link level addresses via registers in LINKC (LINKC Map Transmit Ports to Configuration). The processors fetch instructions from ROM/RAM and handle normal segmentation activity such as LCD update operations and cell generation functions. The type of cell that is generated by the segmentation logic is determined by the initial instruction pointer that is contained in the LCD structure. For example, software can enqueue an LCD that has the initial instruction pointer field set for normal AAL5 cells, and SEGBF will generate a single AAL5 cell as a result of the enqueue operation. If, however, software enqueues an LCD that has the initial instruction pointer field set for POS-PHY operation, then SEGBF will continue to generate buffers to pass to the link level until all the data has been exhausted. For a more complete description of the segmentation entry points, refer to the seg_prc_Entry_point field in the transmit LCD data structure section of this document. After the buffers/cells are built in a 16-by-64 byte array, they are marked as available to the link level layer (LINKT) in the IBM3206K0424. A simplified block diagram is shown in the SEGBF Block Diagram.

SEGBF Block Diagram



The sequence of events that happens when an AAL5 frame is enqueued to SEGBF is as follows:

1. An initial check is made to determine if there is space available in the cell buffer. If no space is available, processing stops for this drop until a cell buffer is freed by the link logic (LINKT).
2. When buffer space becomes available, the enqueued LCD address is requested from the transmit LCD cache (there are four entries in the LCD cache). The segmentation logic waits for a valid indication from the cache; at this time all LCD information is available to the segmentation logic on the LCD cache interface.
3. The initial instruction pointer (IP) is fetched from the LCD and loaded for both of the segmentation processors. Software controls what type of cells are generated by the segmentation logic by initializing this field in the LCD to the entry points defined for different types of cell generation. For a more complete description of the segmentation entry points, refer to the seg_prc_Entry_point field in the transmit LCD data structure section of this document.
4. Assuming an AAL5 entry point is setup in the LCD, the segmentation logic will first initiate a memory fetch of the data required to build the cell.
5. While the data fetch is in process, the segmentation processors will update various fields in the LCD including statistics and the next segmentation pointer. Cell construction will be started using the ATM

header from the LCD.

6. When the payload data is available from memory, it is written to the cell buffer following the header.
7. If the segmentation logic determines that the current cell being assembled will be the last cell of this frame, the AAL5 trailer is appended to the cell buffer, with any unused bytes being padded with zeros.
8. If the current cell is not the last of the frame, the partial CRC is written back out to the LCD.
9. After the cell has been completed, it is marked as available to the link level logic

11.1: SEGBF Software LCD Enqueue

This register provides a mechanism for software to transmit a single cell or a group of cells making up a buffer that can contain any user-defined data at any time. To cause a cell/buffer to be transmitted, the software must write the address of a valid LCD control block to this register. The segmentation hardware will then construct a cell to match the AAL type defined in the LCD control block, using the segmentation pointer contained in the LCD to fetch data and present this cell to the next lower level of hardware to transmit. This method of cell transmission bypasses the cell scheduler completely, so it is the responsibility of the software to ensure that peak and average rates are not violated. When the segmentation logic has completed building the cell/frame and queued it for transmission, the LCD address will be loaded into the software LCD complete register. This method of cell transmission is not designed for high performance and, as such, there is only a single level of queueing underneath the complete register. It is recommended that only a single software LCD be queued to the segmentation logic at any one time to prevent hanging the segmentation logic as it attempts to queue a complete software LCD to the complete queue.

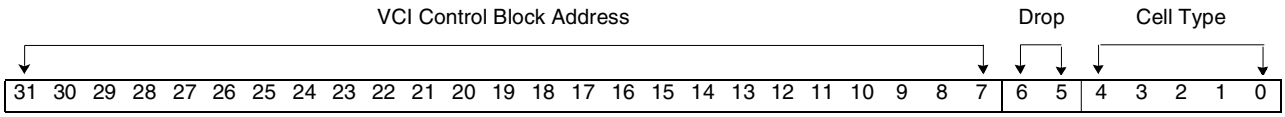
Length 32 bits

Type Read/Write

Address XXXX 1400

Power On Value X'0000 0000'

Restrictions Before enqueing a VCI, software must ensure that the previous software enqueue has been handled by the hardware. This is accomplished by reading this register before an enqueue is attempted. If a value of '0' is returned, the segmentation hardware is ready to accept an enqueue operation. If a non-zero value is returned, it will be the address of the previous VCI that was enqueued and this indicates that the segmentation hardware has not been able to enqueue the VCI to it's internal VCI buffer segmentation queue. If this mechanism shows that this interface is busy and unable to accept new VCI addresses for any appreciable amount of time (tens of µs), it is likely that a condition exists which is preventing the hardware below the segmentation logic from accepting cells for transmission, and the segmentation logics input buffer is full. This mechanism also adds the restriction that a VCI control block should never exist at address '0'.

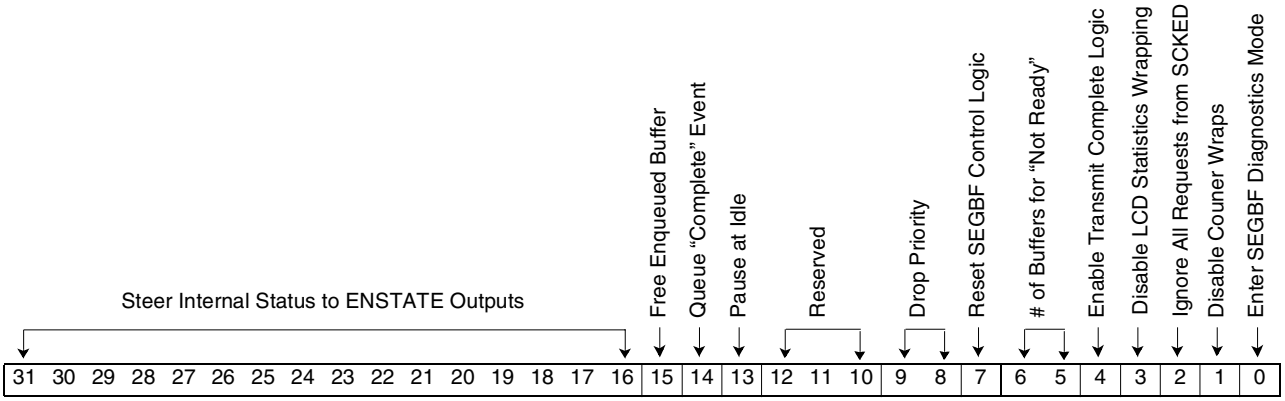


Bit(s)	Description
31-7	These bits contain the upper 25 bits of the address of the VCI control block.
6-5	These bits define which drop this enqueue operation will be associated with.
4-0	These bits control what type of cell will be built by the segmentation logic. There are currently only two valid values for these bits. If these bits are all '0', a normal cell as defined by the LCD will be built. If these bits have a value of 0x1F, an ABR cell will be built using fields defined in the LCD.

11.2: SEGBF Control Register

This register provides a mechanism to control the various programmable features of SEGBF. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length 32 bits
Type Clear/Set
Address XXXX 1408 and 40C
Power On Value X'98400000'
Restrictions None



Bit(s)	Description
31-16	These bits are used to steer internal SEGBF status to the ENSTATE outputs.
15	This bit, when set, will cause the segmentation processors to free a software enqueued buffer after the last cell has been generated.
14	This bit, when set, will cause the segmentation processors to queue a transmit complete event after the last cell has been generated for a software enqueued frame.
13	This bit, when set, will cause the segmentation logic to pause when it reaches the idle state. Segmentation will not be continued until this bit has been reset. Care must be taken to leave this bit set for a very short duration so that segmentation throughput will not be adversely affected.
12-10	Reserved
9-8	These two bits define the prioritization scheme used by the segmentation logic to determine which drop to build a cell for when running in frame mode. A value of '00' will provide for equal priority among all drops, the drops will be processed in order from zero to three as long as data is available to segment and space is available in the cell buffer for the drop. A value of '01' will provide descending priority from drop 0 to drop 3. If data exists and a cell buffer is available for drop 0, a drop 0 cell will always be built regardless of the situation on any other drops. In this mode, a cell will only be built on drop three if all other drops either have not data or no cell buffer available.
7	This bit, when set, will reset all control logic in the entity. After being set, this bit must be reset before the segmentation logic will function properly. This bit must remain set for at least one microsecond to reset the segmentation logic properly.
6-5	These two bits define the number of cell buffers that can be filled on a given drop before a not ready condition is returned to the cell scheduler. This addresses latency issues caused by multiple cells waiting for transmission by the lower link level. A value of '00' allows all four cell buffers to be used at any time; a value of '01' allows one cell buffer to be used; a value of '10' allows two cell buffers to be used, and a value of '11' allows three buffers to be used.
4	This bit, when set, enables the transmit complete event modification logic in the segmentation processors. This logic will retrieve two bits from the xmit_comp_evnt_mod field in the LCD and logically OR them with bits eight down to seven of the buffer address being enqueued to RXQUE. This logic only functions when buffer addresses are being queued; it will not modify the event if LCD addresses are being enqueued. This also adds the restriction that all buffer addresses must start on a 512-byte boundary or greater.

Bit(s)	Description
3	This bit, when set, disables the LCD statistics wrap events.
2	This bit, when set, will cause all requests from the cell scheduler to be ignored. This allows complete program control of all cells being sent out on the external interface.
1	This bit, when set, disables the programmable counter wrap events.
0	This bit, when set, causes the SEGBF entity to enter diagnostic mode. This bit must be set in order to access the internal array. When accessing the array, care must be taken that normal entity reads and writes of the array are not happening at the same time or the results will be indeterminate.

11.3: SEGBF Status Register

This register provides feedback to the user on the current status of SEGBF. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

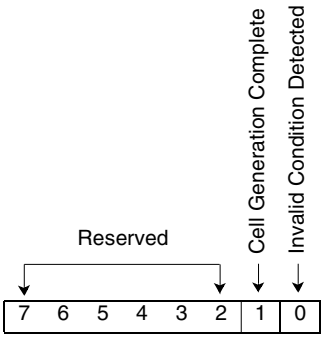
Length 8 bits

Type Clear/Set

Address XXXX 1410 and 414

Power On Value X'00'

Restrictions None

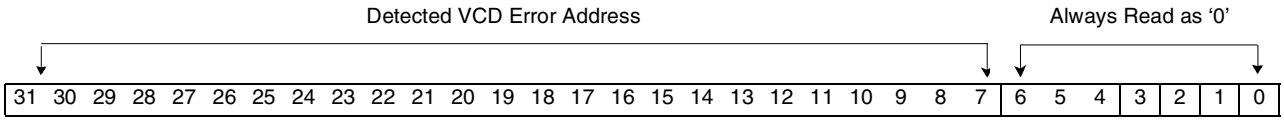


Bit(s)	Description
7-2	Reserved
1	This bit, when set, indicates that the segmentation logic has completed cell generation for an LCD that was enqueued by the software to the software LCD enqueue register.
0	This bit, when set, indicates that the segmentation logic has detected an invalid condition in one of the LCDs that it was processing. The address of the LCD in error is contained in the Invalid LCD register. Any invalid LCDs detected are not processed further by the segmentation logic, so the program must do something to clear this condition.

11.4: SEGBF Invalid LCD Register

This register provides feedback to the program when the segmentation logic detects an invalid LCD. If multiple invalid LCDs are being processed, this register will contain the address of the last one that was processed by the segmentation logic. There are several invalid LCD situations that the segmentation logic checks for: The first is the LCD address not being on the correct boundary. For example, if the chip is configured to have all LCDs on 128-byte boundaries and an LCD is encountered that is not on a 128-byte boundary. Another invalid condition is when the transmit length configured in the LCD plus the offset in the LCD when added together exceed the maximum overall packet size configured in the chip. It is up to the program to determine which of the possible conditions caused the error to be reported.

Length	32 bits
Type	Read/Write
Address	XXXX 1418
Power On Value	X'00000000'
Restrictions	None

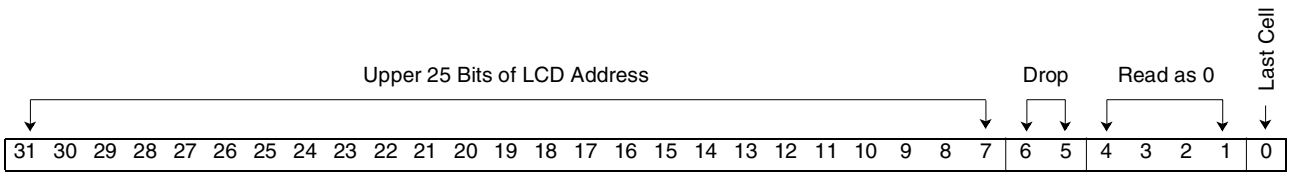


Bit(s)	Description
31-7	These bits contain the 32-bit address of the LCD detected to be in error.
6-0	Reserved. Always read as '0'.

11.5: SEGBF Software LCD Complete

This register provides feedback to the program when the segmentation logic completes cell generation for an LCD that was enqueued by the software. After the segmentation logic has updated the LCD, the address of the LCD is copied into this register providing any previous LCD addresses written to this register have been read by the software. If multiple software queued LCDs are outstanding to the segmentation logic at any time, the segmentation process can be delayed when multiple software enqueued LCDs complete without the software getting a chance to read the LCD addresses from this register. To guarantee that the segmentation logic never has to wait for the software to read this register, it is recommended that only one software LCD be enqueued at any one time.

Length	32 bits
Type	Read/Write
Address	XXXX 141C
Power On Value	X'00000000'
Restrictions	Bits 5 through 0 are not implemented and will always return '0'. To maintain future compatibility, '0's should be written to these bits.



Bit(s)	Description
31-7	These bits contain the upper 25 bits of the LCD address that the segmentation logic has finished processing.
6-5	These bits indicate the drop on which the cell was sent.
4-1	These bits will read back as '0'.
0	This bit will be set when the cell that was built was the last cell of a frame and reset if the cell that was built was not the last cell of a frame.

11.6: SEGBF Interrupt Enable Register

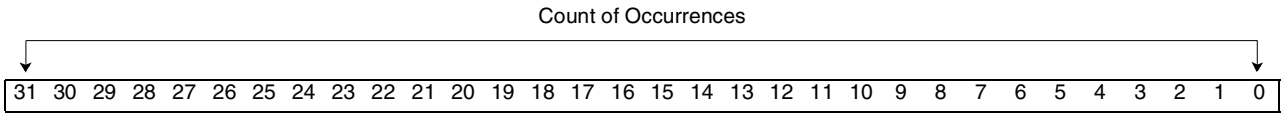
This register allows the user to selectively determine which bits in the SEGBF status register will cause processor interrupts. A '0' in a bit position masks interrupts from the corresponding bit location in the SEGBF status register. A '1' in a bit position allows interrupts for the corresponding bit in the SEGBF status register. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	12 bits
Type	Clear/Set
Address	XXXX 1420 and 424
Power On Value	X'00'
Restrictions	None

11.7: SEGBF Programmable Counters

This register provides the user with feedback on the number of times that a particular event or condition has occurred in the segmentation logic. The event or condition that causes this counter to increment is defined by the associated SEGBF Programmable Counter Source Specification register. When the counter wraps, an event is generated.

Length	32 bits	
Type	Read/Write	
Address	Counter 0	XXXX 1430
	Counter 1	XXXX 1434
	Counter 2	XXXX 1438
	Counter 3	XXXX 143C
Power On Value	X'0000 0000'	
Restrictions	None	

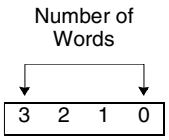


Bit(s)	Description
31-0	These bits contain a count of the occurrences of the desired event or condition.

11.8: SEGBF Transmit LCD Size

This register should be loaded with the number of eight-byte words that are needed for the maximum-sized LCD that will be setup by software. Refer to the previous section describing LCD layout to determine the number of words required to support the different modes. The minimum value is six. This is the correct value when running only AAL5 mode; it includes three words of scheduling information, two words shared between CSKED and SEGBF, and one word for SEGBF to maintain statistics. Setting this register to a value that is too small will likely cause the chip to function improperly. Setting this register to a value that is too large will adversely affect performance.

Length	4 bits
Type	Write/Read
Address	XXXX 1488
Power On Value	X'6' (This is the correct value when running AAL5 with statistics.)
Restrictions	Minimum is 6, maximum is 0xA

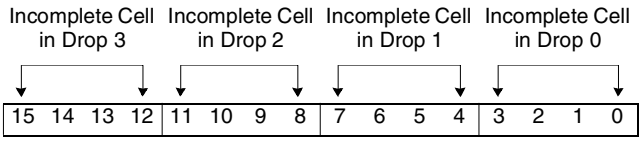


Bit(s)	Description
3-0	These bits contain the number of eight-byte words in the LCD to be used by the transmit logic.

11.9: SEGBF Cell Queue Status

This register indicates the number of cells queued up for transmission over the media. SEGBF can have a maximum of 16 cells queued up for transmission: four cells on each of four drops. When a bit is set to '1', the corresponding cell buffer contains a cell that has not been completely processed by the link logic.

Length	16 bits
Type	Read
Address	XXXX 148C
Power On Value	X'0000'
Restrictions	None

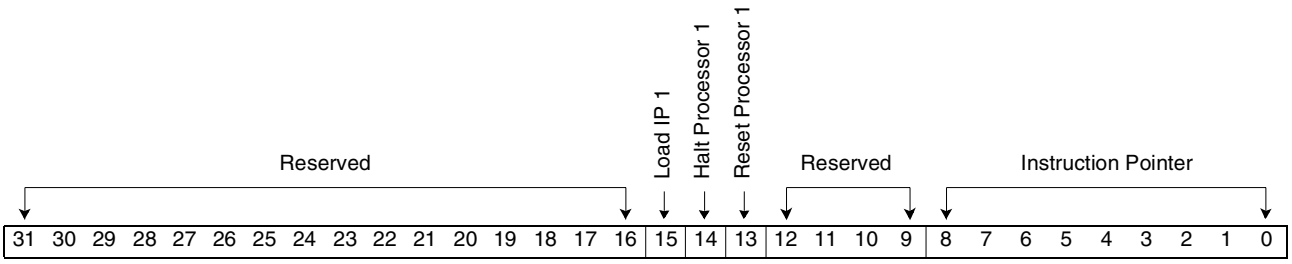


Bit(s)	Description
15-12	These bits indicate which cell buffers contain cells that have not been processed by the link level for drop 3.
11-8	These bits indicate which cell buffers contain cells that have not been processed by the link level for drop 2.
7-4	These bits indicate which cell buffers contain cells that have not been processed by the link level for drop 1.
3-0	These bits indicate which cell buffers contain cells that have not been processed by the link level for drop 0.

11.10: SEGBF Processor 1 Control/Status

Reading this register provides feedback to the user on the current state of segmentation processor 1. Writing the appropriate bits in this register causes processor 1 to begin executing at a new location specified in the data that was written.

Length	32
Type	Read/Write
Address	XXXX 14A0
Power On Value	X'0000C000'
Restrictions	None

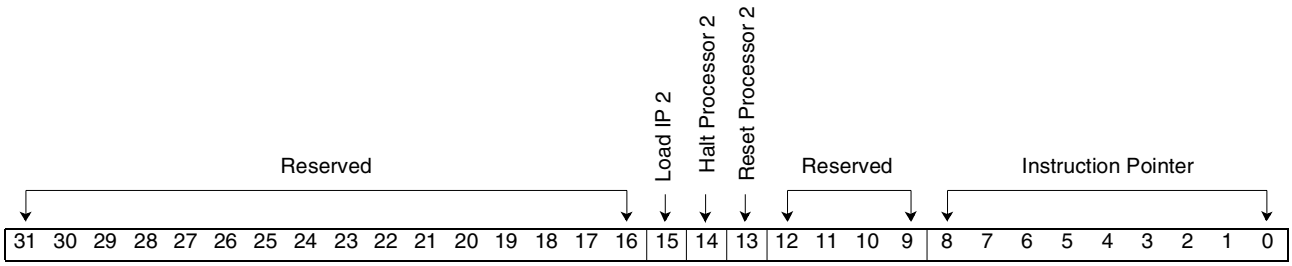


Bit(s)	Description
31-16	Reserved
15	Writing '0' to this bit causes the instruction pointer (IP) for processor 1 to be loaded with the data in bits 8 down to 0. This bit will immediately be set back to '1' after the IP load completes.
14	Writing '0' to this bit halts processor 1. Writing '1' makes the processor fetch and execute instructions.
13	Setting this bit to '1' resets processor 1.
12-9	Reserved.
8-0	When written, these 9 bits contain the new IP for processor 1. When read, these bits reflect the current IP for processor 1.

11.11: SEGBF Processor 2 Control/Status

Reading this register provides feedback to the user on the current state of segmentation processor 2. Writing the appropriate bits in this register causes processor 2 to begin executing at a new location specified in the data that was written.

Length	32
Type	Read/Write
Address	XXXX 14A4
Power On Value	X'0000C000'
Restrictions	None

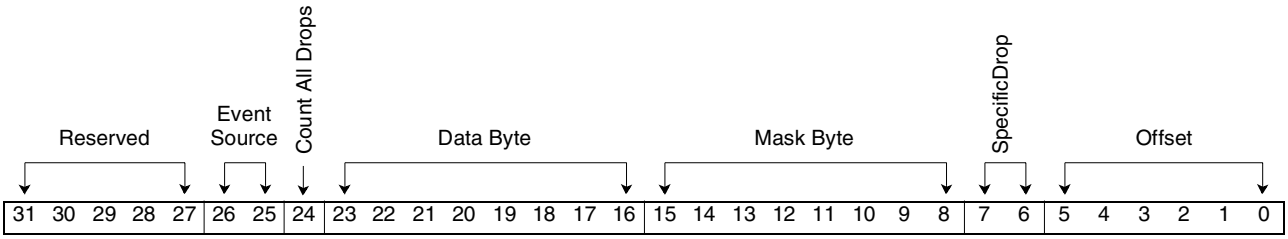


Bit(s)	Description
31-16	Reserved
15	Writing a '1' to this bit will cause the instruction pointer (IP) for processor 2 to be loaded with the data in bits 8-0. This bit will immediately be set back to a '1' after the IP load completes.
14	Writing a '0' to this bit will halt processor 2; writing a '1' will make the processor fetch and execute instructions.
13	Setting this bit to a '1' will reset processor 2.
12-9	Reserved
8-0	When written, these nine bits contain the new IP for processor 2. When read, these bits reflect the current IP for processor 2.

11.12: SEGBF Programmable Counter Source Specification

This register determines what event or condition will cause the associated counter to increment.

Length	32	
Type	Read/Write	
Address	Counter 0	XXXX 14B0
	Counter 1	XXXX 14B4
	Counter 2	XXXX 14B8
	Counter 3	XXXX 14BC
Power On Value	Counter 0	X'000 0803'
	Counter 1	X'000 0903'
	Counter 2	X'008 0803'
	Counter 3	X'002 0203'
Restrictions	None	

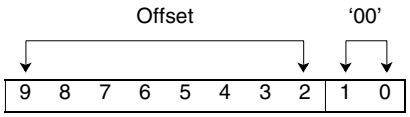


Bit(s)	Description
31-27	Reserved.
26-25	These bits select the source of the event to be counted. '00' selects the cell events that are further defined in bits 23 down to 0 of this register. '01' selects the number of bytes transmitted on a given drop. '10' selects the number of frames transmitted on a given drop. The drop is selected by bits seven and six, or all drops can be included by setting bit 24.
24	This bit, when set, causes the counter to count the specified event for all drops, not just the drop specified in bits seven and six.
23-16	These bits define the data byte that is compared to the result of logically anding the data byte being written to the cell array at the offset specified in bits 5-0 with the mask in bits 15-8. If there is an exact match, the counter will increment.
15-8	These bits define the mask byte to be logically anded with the data byte being written to the cell array.
7-6	These bits determine which drop this counter is associated with.
5-0	These bits define a byte offset into the cell being built in the segmentation cell array. Zero corresponds to the first byte in the cell and 63 corresponds to the last byte in the cell. When the segmentation logic copies a byte of data into the cell array at this offset, the logic compares the byte defined in bits 23-16 to the logical and of the data being written and the mask defined in bits 15-8 of this register. If an exact match is detected, the counter will be incremented.

11.13: SEGBF Cell Staging Array Pointer

This register points to an offset in the SEGBF cell staging array for accesses from the PCI bus. This register must be loaded with the correct offset before the desired data can be read from the SEGBF Cell Staging Array Data Register. This register will auto-increment after each access to the associated data register. When the last address is accessed, this register wraps to zero.

Length	10
Type	Read/Write
Address	XXXX 14C0
Power On Value	X'000'
Restrictions	None



Bit(s)	Description
9-2	These bits provide an offset into the SEGBF cell staging array for accesses from the PCI bus. These bits provide access to the 256 unique four-byte locations in the array. Accessing the last location in the array will cause the address to wrap back around to the beginning of the array.
1-0	These bits are not implemented, they can not be written and will always return '0' when read.

11.14: SEGBF Cell Staging Array Data

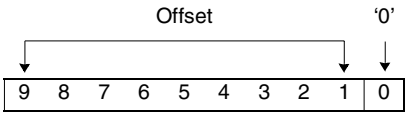
This array is divided into 16 64-byte buffers used to assemble cells that are ready for transmission on the physical interface.

Length	32
Type	Read/Write
Address	XXXX 14C4
Power On Value	Undefined
Restrictions	This array can only be accessed when the diagnostic mode bit in the control register is set. Accesses attempted when not in diagnostic mode will return 0xBADDBADD.

11.15: SEGBF Instruction SRAM Pointer

This register points to an offset in the SEGBF SRAM that is used to store processor instructions. This register must be loaded with the correct offset before the desired data can be read/written from/to the SEGBF instruction SRAM data register. This register will auto-increment after each access to the associated data register. When the last address is accessed, this register wraps to zero. The first 256 locations in the array should be loaded with the instructions for processor 1, and the second 256 locations should be loaded with the instructions for processor 2.

Length	10
Type	Read/Write
Address	XXXX 14C8
Power On Value	X’000’
Restrictions	None



Bit(s)	Description
9-1	These bits provide an offset into the SEGBF instruction SRAM for accesses from the PCI bus. These bits provide access to the 512 unique two-byte locations in the array. Accessing the last location in the array will cause the address to wrap back around to the beginning of the array.
0	This bit is not implemented; it can not be written and will always return ‘0’ when read.

11.16: SEGBF Instruction SRAM Data

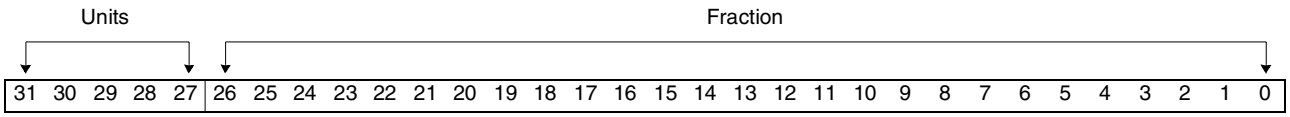
This register address can be used to read/write the instruction data that is needed by the segmentation processors. All instructions must be written before the processors can be brought out of the halt state.

Length	16
Type	Read/Write
Address	XXXX 14CC
Power On Value	Undefined
Restrictions	This array can only be accessed when the processors are in the halt state. Reads attempted when the processor is in run mode will return invalid data. Writes attempted when the processor is in run mode will be ignored.

11.17: MPEG-2 PCR Increment Register

Each tick of the time base will add the contents of this register to the MPEG PCR Reference Register. This register contains a fixed point number with 27 bits of fraction and five bits of units. This means that the external reference clock can range in speed from 22.5 Khz to the maximum speed of this entity which is TBD. Assuming that the entity will run with a 50 Mhz clock, the conversion to 720Khz can be done with an accuracy of 1.1 parts in two million. (A clock of 19.4 Mhz will give a conversion accuracy of one part in 4.9 million.) If the input clock is 19.4 Mhz, then the value to put in the increment register is (720,000 / 19,400,000) * 2**27 or 4,981,277. If the input clock is 33 Mhz, then the value to put in the increment register is (720,000 / 16,666,666) * 2**27 or 5,798,206.

Length	32
Type	Read/Write
Address	XXXX 1468
Power On Value	X'002C3C9F' (33 Mhz)
Restrictions	None



Bit(s)	Description
31-27	These bits contain the whole part of the increment value.
26-0	These bits contain the fractional part of the increment value.

Receive Data Path Entities

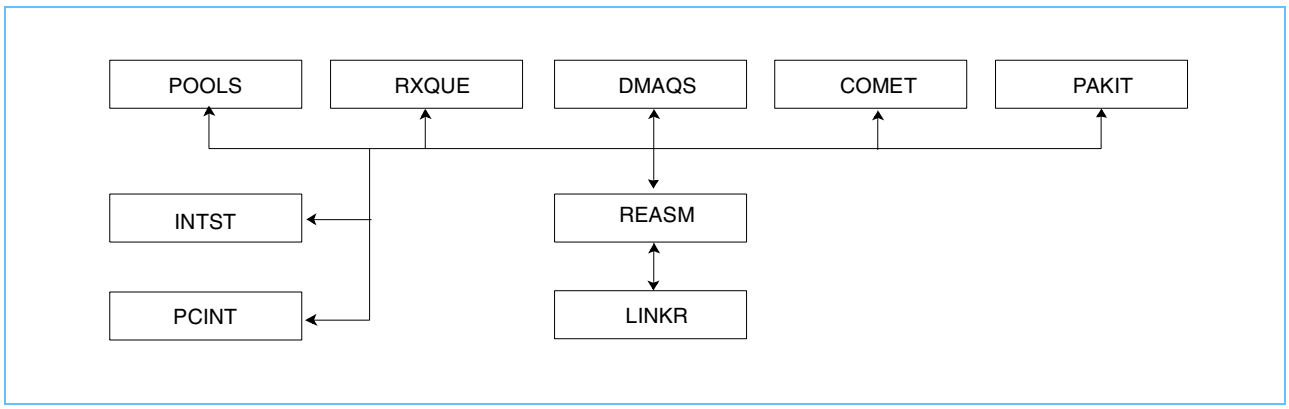
Entity 12: Cell/Packet Re-assembly (REASM)

REASM is the top level receive entity that encapsulates all of the receive sub-entities.

Note: The receive portion of the chip is very different from previous versions of the processor. REASM no longer does HEC correction or detection, since everything we interface to already does this function.

The following figure shows how REASM interacts with the other entities:

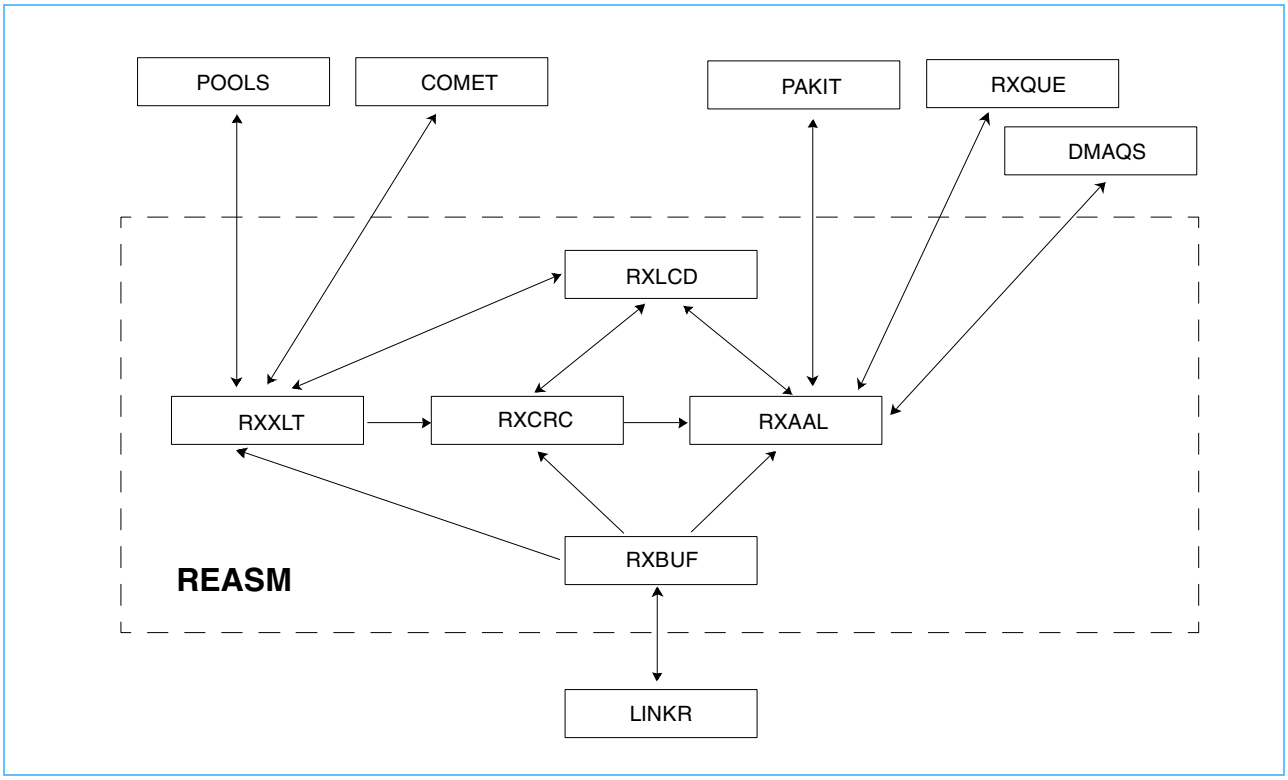
REASM Entity Interfaces



REASM is made up of a number of sub-entities to provide the overall receive functionality. REASM contains the following sub-entities:

RXBUF	Provides receive cell/packet buffering between LINKC and REASM. Also defines the port configurations and mappings.
RXXLT	Provides general LCD translation facilities.
RXCRC	Provides CRC facilities.
RXAAL	Performs the Cell And Packet reassembly functions including AAL processing and all Cell and Packet Post-Processing functions.
RXLCD	Provides RX LCD caching for the REASM sub-entities.

REASM Sub-Entity Block Diagram



RXXLT, RXCRC, and RXAAL form a cell processing pipeline. Each stage is allowed up to a cell time to process the cell it is currently working on. This means each cell has the potential to have a three-cell time latency through the REASM entity. The overall performance should be at its maximum since a single cell completes processing every cell time. The stages are allowed to run faster if there is no blocking condition. The only blocking condition is the latency of memory accesses and the total length of the nano-program that needs to run.

The following sections provide a brief description of each sub-entity and the function it provides.

Miscellaneous Reassembly Functions

ATM OAM Cell Processing

When processing an ATM data stream, OAM cell processing may be necessary. This function needs to be enabled in the REASM Reassembly Modes Register. REASM performs OAM discrimination when enabled. This function can be enabled on a per port basis. As cells arrive for processing, the nano-code works together with some dedicated logic to discriminate between different types of OAM cell traffic. Each type that is discriminated produces a different event when traffic is surfaced to a receive queue. The different types that are discriminated are:

Segment	VCI = 0003
End to End	VCI=0004
Segment	Not F4 and PTI field = "100"
End To End	Not F4 and PTI field = "100"
Forward Cell	PTI = "110" and DIR=0
Backward Cell	PTI = "110" and DIR=1
Reserved Cell	PTI field = "111"
End to End Alarm	End to End and OAM type=0001 and func=0000 or 0001
Segment Alarm	Segment and OAM type=0001 and func=0000 or 0001
End to End Loopback	End to End and OAM type=0001 and func=1000
Segment Loopback	Segment and OAM type=0001 and func=1000
End to End Pm Cell	End to End and OAM type=0010
Segment Pm Cell	Segment and OAM type=0010
End to End Activate/Deactivate Cell	End to End and OAM type=1000
Segment Activate/ Deactivate Cell	Segment and OAM type=1000

Both the RXCRC and RXAAL nano-code understand and process OAM traffic differently when enabled. In the receive LCD there are control bits that allow the user to specify on a per connection basis how the OAM traffic should be handled. There are two types of flows when processing OAM traffic. A connection can be either "routed" or "terminated." A terminated connection is any connection that terminates either cells or packets. AAL5 and packet LCDs that are fast forwarded are considered "terminated" because the packet is terminated before the packet is fast forwarded. A "routed" connection is a raw LCD that is fast forwarded including raw mode with early packet discard. When OAM traffic is processed on a "terminated" connection, the OAM traffic can be terminated or dropped based on the discrimination configuration bits in the receive LCD. A mask value of ’0’ specifies to terminate the cell, and a mask value of ’1’ specifies to drop the cell. When OAM traffic is processed on a "routed" connection, the OAM traffic can be fast forwarded or dropped based on the discrimination configuration bits in the receive LCD. A mask value of ’0’ specifies to fast forward the cell, and a mask value of ’1’ specifies to drop the cell. This allows the user to filter the OAM traffic based on the type of OAM traffic on a per connection basis.

The mask bit is determined by using the sub-type mask bit if pertinent. For example, an F5 Segment Pm Cell will use the "Segment Pm Cell" mask bit. If one of the sub-types does not match, then the basic types are

used (that is, F4 End-to-end). There is a 16-bit mask field in the receive LCD. A description of each bit is in the table below:

Bit	Description
15	Reserved
14	F4 segment
13	F4 End to End
12	F5 Segment
11	F5 End to End
10	Rm Forward Cell
9	Rm Backward Cell
8	Pti Reserved Cell
7	End to End Alarm
6	Segment Alarm
5	End to End Loopback
4	Segment Loopback
3	End to End Pm Cell
2	Segment Pm Cell
1	End to End Activate/Deactivate Cell
0	Segment Activate/Deactivate Cell

When an OAM cell is processed, the configuration information from the gpmtagE is used to make reassembly decisions like which buffer pool or receive queue to use.

TCP/IP Receive Checksum Verification

When enabled, the IBM3206K0424 is able to perform verification of the IP header checksum and the associated protocol checksum for the user. The final checksum status is raised to the user in the packet header flags. This function is accomplished by the RXCRC entity using the CRC nano-program.

To enable and use the function, several things must be done:

- TCP/IP checksum function must be enabled in the REASM Mode Register. This enables the overall function and includes the checksum state words in the receive portion of the LCD.
- Specify a frame type in the frameType field of the receive LCD. This field specifies an entry point into the checksum verification nano-code. The entry point points to a piece of code that understands how to locate the IP header based on the current frame type. For example, a connection might be native IP over ATM or LANE ETH. For the currently supported types, see the crc.txt file provided with the RXCRC code load.

Note: Other types can be added if needed.

Once properly enabled, TCP/IP checksums are verified as packets arrive. The status is raised to the user in the packet header using four bits of status (two bits for IP and two bits for protocol). The format and meanings of the bits are specified in Packet Header. For connections that have mixed traffic, the checksum operation is only run on packets that are recognized as TCP/IP. For example, on a LANE ETH connection packets of different protocols can be interleaved.

Scatter/Cut Through Receive Processing

The scatter/cut through support is very versatile and easy to use. Scatter typically refers to scattering pages from a contiguous IBM3206K0424 buffer into multiple non-contiguous host pages. Cut through refers to moving contiguous data from IBM3206K0424 buffers into contiguous storage in host memory. The functions are similar, and the terms scatter and cut through may be interchanged in the following text since cut through is really a special case of scatter. The correct term is used when context requires it.

The scatter processing is performed by RXAAL along with RXQUE and DMAQS. The following items need to be setup:

- DMAQS should be setup and ready to run.
- RXQUE should be setup and ready to run. Each queue that is used should have the proper event size selected, the direction should be set correctly, and timestamps and bcach advice should be disabled.
- Page buffers or descriptors should be placed on the configured receive queues.
- The scatter configurations need to be set in the RXALL - Scatter/Cut Through Info Registers.
- The scatter flags need to be set in the RXALL - Scatter/Cut Through Flag Registers.
- Each LCD that is to use scatter, needs to have the ppMode field set to do scatter.
- A scatter configuration needs to be selected for each connection by setting the cutThruSel field in the receive LCD.

Once set up, the scatter mechanism is performed by RXAAL for the user. There is user intervention required to process the packets when the scatter is complete, and for page recovery in error scenarios.

There are many options when setting up scatter, and the selection of the type of scatter processing to perform depends on the users environment. The basic scatter mechanism will be described followed by a discussion of the different options that can be used.

When packets are received and scattered, the following structure shows the main components of the received packet:

General Layout of a Received Packet in Scatter Mode

```
struct ScatterRxPacket {
    packetHeader;   // Charm packet header
    dmaList;        // Scatter dma list
    padbytes;       // Pad out to receive offset from lcd
    packetData;     // Actual receive packet
  };
```

The DMA list is maintained immediately after the IBM3206K0424 packet header, and before the receive packet data. For this reason, the user should allocate enough space between the packet header and the packet data to accommodate a maximum-sized DMA list based on the maximum number of pages that could be DMAed. The rxOffset field in the receive LCD is where this offset is specified. The maximum receive offset is 256 bytes, which is specified with an rxOffset value of ’0’. The following is the layout of the DMA list:

General Layout of a Scatter DMA List

```
struct dmaList {
    bit16  numHeadbytes;       // number of bytes included with header dma
    bit16  numTailbytes;       // number of bytes in last dma page

    bit1   deqLocked;          // error status
    bit1   deqInvalid;         // "
    bit1   headerTruncated;    // "
    bit1   badDmaList;         // "
    bit4   DeqLockedQueueNum;  // "
    bit16  reserved;
    bit2   cutThruSel;         // copy of cutThruSel used from receive lcd
    bit6   numPages;           // number of page entries that follow (max=63)

    bit32  pageList[N];        // Actual dma descriptors or page addresses
                               // Note: each entry can be 32, 64, or 128 bits wide
 };
```

The first eight bytes are always present and are filled in when the packet is complete or when an error occurs. The actual page list is filled in as the data is DMAed. The first location is initially skipped and is filled in later when the header is DMAed. The second and subsequent entries are filled as each page is DMAed. Each page list entry contains either physical page addresses, IBM3206K0424-based DMA descriptor addresses, or user data. Whether a physical page address or DMA descriptor address is present depends on the cut through configuration. From here on, the term page address and DMA descriptor are used interchangeably as either is valid based on the configuration, but the correct term is used when the context requires it.

A page list entry can contain one or two pieces of information. Each page list entry can be 32, 64, or 128 bits long. If using 32 bit addresses, then the entry is 32 or 64 bits. If using 64-bit addresses, then the entry is 64 or 128 bits. The first piece is always the page address or the DMA descriptor address. Each cut through configuration allows the user to specify an optional second deque operation from the receive queue being used. This allows the user to place user information associated with the particular page address in the receive queue and the page list. Thus, the corresponding virtual address of a page could be surfaced along with the physical page address. It is very important to enque information to the receive queue in the proper order if using this mode. The physical page address is first, followed by the user information.

As the receive packet data is being received, it is DMAed as soon as a page crossing occurs if there is a DMA descriptor available on the receive queue being used. If no descriptor is available, then no DMA takes place until the next receive cell is received, at which time the receive queue availability is checked again. This catch-up process continues until the packet is complete. When the packet is complete, all DMAs are scheduled if page addresses are available. If a page address is not available, then a "no DMA descriptor available" is surfaced to the user so the packet can be used and the DMA list recovered.

As each data page is DMAed to the user, a DMA descriptor is formed and enqueued to the DMAQS DMA queue specified in the cut through configuration. The DMA descriptor is formed based on if page addresses or DMA descriptor are provided on the receive queue being used. If IBM3206K0424-based DMA descriptors are being used, then the receive queue contains DMA descriptor chains where the low order bits of the DMA descriptor address specify how many descriptors are in the chain. Typically, this chain length is one, but more can be used. The first descriptor in the chain provides the destination address (page address) which is filled in by the user. The source address, the length, and the flags are filled in by the IBM3206K0424 for the first descriptor in the chain. The source address is filled in with the beginning address of the page within the current receive buffer. The length is filled in with either the page size (if a full page is present) or the number of bytes in the last page for the last page. The flags are filled in using the flags from the appropriate RXALL - Scatter/Cut Through Flag Registers. If physical page addresses are contained in the receive queue, then a

single DMA descriptor is formed by DMAQS directly in DMA queue storage using the page address and the same information that would have been filled into the DMA descriptor. Generally, using page addresses performs better, but is less versatile, which is usually a good trade off. Normally no DMA event is generated in the flags when pages are DMAed.

When the last cell of a packet is received, and all the data pages have been DMAed, the packet header and DMA list are updated and DMAed into a header buffer. The mechanism and configuration of the header buffer is similar to the pages, but separate configuration is usually necessary for correct functionality. For example, different flags are normally used in order to get an event for the header DMA so the user can process the packet. The header DMA normally frees the IBM3206K0424 buffer; another difference is the page (or buffer) size used for headers is normally different than the normal page size. Some of the optional features described later also drive having different configuration for the header DMAs.

Once the user gets the event for the header DMA, the user processes the received packet using the DMA list and the packet header. Once the packet has been processed, the pages or descriptors need to be returned to the proper receive queue when the pages can be reused, thus completing the scatter processing.

There are several ways of getting that important event to start the receive processing. Usually the event is generated when the last header DMA is complete. If DMA descriptors are being used, then there are two choices. First, the normal DMA flag that generates an event can be used. This generates an event with the DMA descriptor address in the significant bits. While this works and may be desirable in some environments, the DMA descriptor needs to be read in order to get access to the host header buffer address. The second way to generate an event when using descriptors is to provide a second DMA descriptor in the DMA descriptor chain that enqueues the header buffer address and a user-defined event in the lower order bits. Generating an event in this manner provides the user with the buffer address; the header DMA descriptor address is available in the header buffer as part of the DMA list.

When using page addresses, the event source in the cut through configuration should be set to use the destination address as the event data. When this is done, the event data contains the header buffer address as in the second case above.

Error Recovery

There are two types of error that need to be handled. First, if there is no error on the receive packet and a page address was not available, then the packet is surfaced to the user with a "no DMA descriptor available" event in the event type field and the IBM3206K0424 buffer address in the event information field. The user has a choice at this point. The packet is good so it can be used or freed by the user. In either case, the DMA list in the packet header must be recovered. So if the DMA list in the header is not used, it should be recovered by returning it to the proper receive queues. The event surfaced specifies which type of page address failed so the user can parse the DMA list properly. The event will specify whether a normal page descriptor, optional header descriptor, or packet header descriptor. The same descriptor recovery must be performed when an error event is surfaced (that is, CRC error). When there is an error event, no packet header DMA is performed so no packet header descriptor is in the DMA list.

Scatter Options

Most of the optional scatter features have to do with the header bytes and how the header DMA is performed. How the number of header bytes is determined is explained below. For now, just assume there are some number header bytes. The numHeadbytes field in the DMA header specifies the number of header bytes that are available. The location of the header bytes in the DMA buffers can be configured. The default is for them to be kept with the packet header and DMA list in the packet header buffer. For this case, the user should be sure the packet header buffer size is large enough to contain all of this data.

Alternatively, the header bytes can be placed in a separate buffer. To do this, split header mode should be

enabled in the cut through configuration. This buffer is referred to as the optional header page and uses its own page size and receive queue to allow for better storage utilization. When enabled, the second entry in the DMA list becomes the optional header page entry, and should be treated accordingly for page recovery. In split header mode, the header bytes are DMAed as soon as they are all received. This mode is useful if the user environment requires the header bytes to be in a separate buffer from the packet header and DMA list.

Another optional feature is to DMA the header only. This feature is enabled in the cut through configuration. When enabled, only the packet header, DMA list, and header bytes are DMAed. Either a single DMA or two DMAs occur based on if the optional header feature is enabled. This feature can be useful when a routing decision needs to be made for a packet and the entire packet does not need to be brought into host storage. Another possible scenario is the header bytes may determine how the user wants to DMA the packet data to the host.

For smaller packet sizes, it may be more efficient to perform a single DMA and keep the packet header, DMA list, and packet data in a single buffer. To do this, single page mode should be enabled in the cut through configuration. When enabled, the header page size is used to determine the DMA behavior. If a packet completes and the total length of the packet and headers will fit in the header page size, then a single DMA is performed. If the data length exceeds the single page size as it is being received, the data is scattered using the normal options (might need to catch up). This feature can be useful for optimizing user processing for smaller packets. For example, all packets less than 2K might be a candidate for this feature.

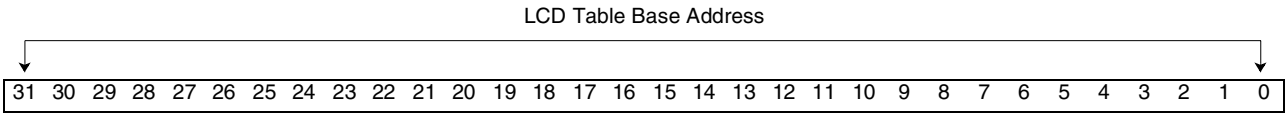
Head Bytes

There are two ways to set the number of head bytes. They are set on a per connection basis in the receive LCD using the numHeadbytes and useCrcNumHead fields. When the useCrcNumHead field is set to ’0’, the numHeadbytes field provides a fixed number of bytes that is used as the number of header bytes. When useCrcNumHead field is set to ’1’, then the RXCRC nano code will calculate the number of header bytes using the frameType field to index an IP procedure. Currently, RXCRC will only set the number of header bytes for recognized TCP/IP headers. Other headers can be recognized. Contact IBM technical support to discuss requirements.

12.1: REASM Logical Channel Descriptor Base Register

The REASM Logical Channel Descriptor Base Register indicates the starting address of the logical channel descriptor table. This register defines where the Logical Channel Descriptors are located.

Length	32 bits
Type	Read/Write
Address	XXXX 1618
Power On Value	X'00008000'
Restrictions	The value must be in the range of the physical memory allocated for Control Memory.



Bit(s)	Description
31-0	This register defines where the Logical Channel Descriptors are located.

12.2: REASM Mode Register

Used to set REASM and sub-entity modes. This register contains the mode bits that specify how REASM is to operate. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

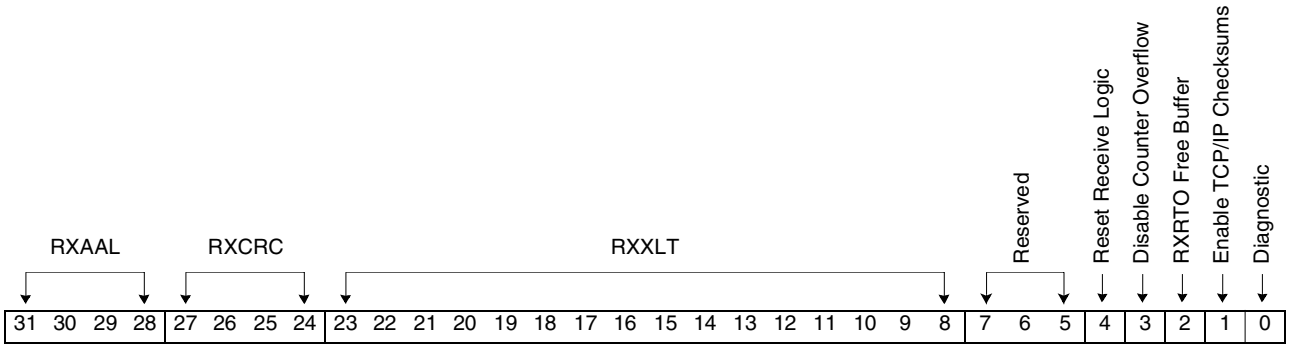
Length 32 bits

Type Read/Write

Address XXXX 1610 and 614

Power On Value X'00000000'

Restrictions None



Bit(s)	Description
31-28	RXAAL code specific mode bits. If used, documented in code load documentation.
27-24	RXCRC code specific mode bits. If used, documented in code load documentation.
23-8	RXXLT code specific mode bits. If used, documented in code load documentation. Each nibble is for a port. Bits 23-20 are RXXLT mode bits 3-0 for port 3, bits 19-16 are RXXLT mode bits 3-0 for port 2, bits 15-12 are for port 1, and bits 11-8 are for port 0.
7-5	Reserved.
4	Reset receive logic.
3	Disable counter overflow events.
2	RXRTO free buffer option. When set, a buffer that times out is freed and the event will contain the LCD address instead.
1	Enable TCP/IP checksums. When set, TCP/IP checksum verification is enabled. When enabled, the receive LCD data structure includes the IP state.
0	Diagnostic mode. When set, REASM is placed in diagnostic mode.

12.3: REASM Reassembly Modes Register

Used to set reassembly modes. This register contains the mode bits that specify different reassembly modes. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

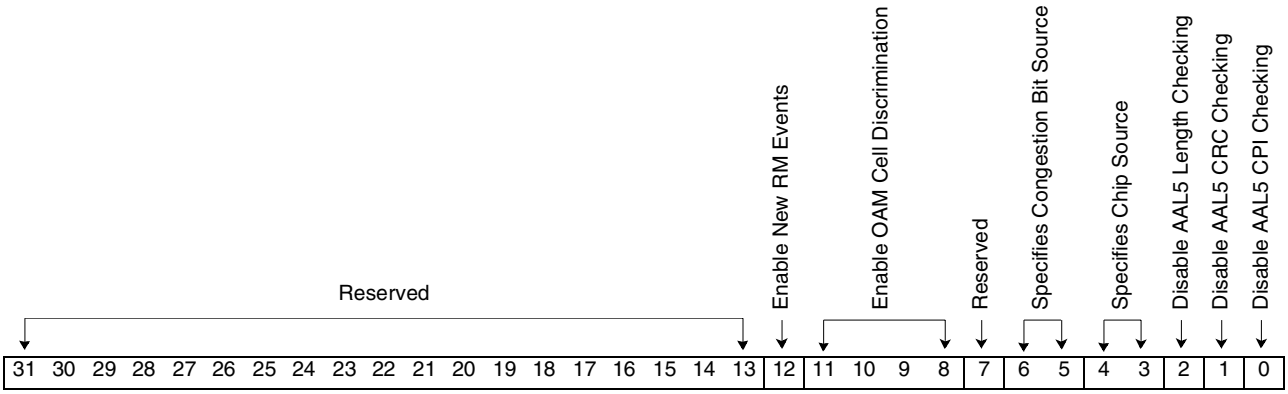
Length 32 bits

Type Read/Write

Address XXXX 1630-34

Power On Value X'00000000'

Restrictions None

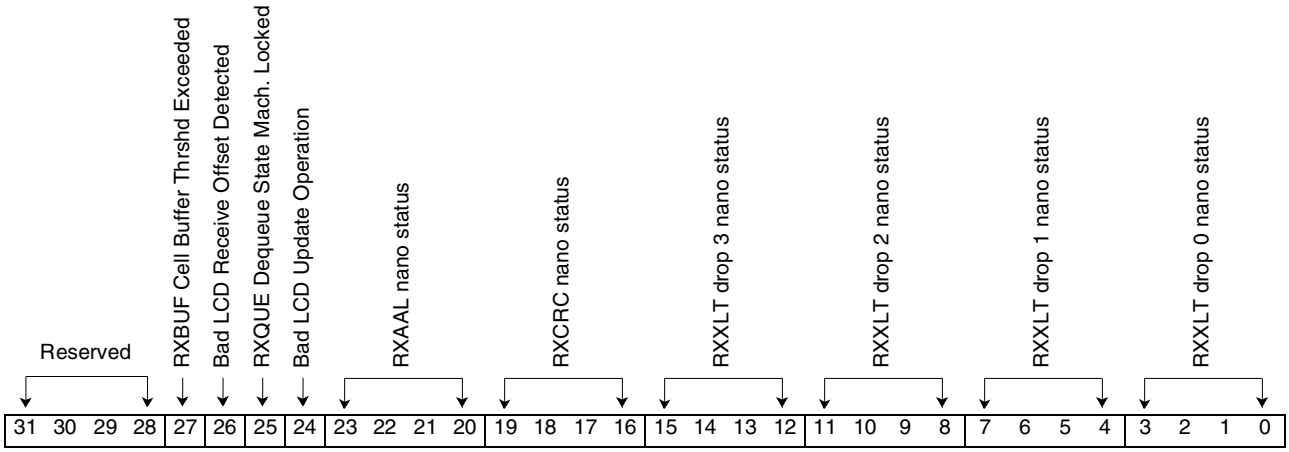


Bit(s)	Name	Description
31-13	Reserved	Reserved
12	Enable new RM events	When set, the new RM cell events are enabled, and the ABR event routing register is ignored. This allows separate events for forward and backward RM cells, but routes the events to the general OAM cell receive queue.
11-8	Enable OAM cell discrimination	When set, the OAM processing is enabled for ports 0-3. When cleared, OAM cells are NOT discriminated.
7	Reserved	Reserved
6-5		Specify which congestion bit source to use if doing routing. Values are: 00 Use value from LCD routed LCD ptr field 01 Use ORed congestion value (or last in case of cell) 10 Use value from last cell
4-3		Specify which CLP source to use if doing routing. Values are: 00 Use value from LCD routed LCD ptr field 01 Use ORed CLP value (or last in case of cell) 10 Use value from last cell.
2	Disable AAL5 length checking	When set, the AAL5 trailer length field is not checked.
1	Disable AAL5 CRC checking	When set, the AAL5 trailer CRC is not checked.
0	Disable AAL5 CPI checking	When set, the AAL5 trailer CPI bytes are not checked against zero.

12.4: REASM Status Register

Used to surface REASM and sub-entity status. This register contains the status bits for the REASM sub-entities. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	32 bits
Type	Read/Write
Address	XXXX 1600 and 604
Power On Value	X'00000000'
Restrictions	None



Bit(s)	Description
31-28	Reserved.
27	RXBUF cell buffer threshold exceeded. This bit is set when the number of receive cell buffers exceeds the threshold set in RXBUF Receive Buffer Threshold. This bit is sticky and must be reset by software after being set.
26	Bad LCD receive offset detected. This bit is set when a receive LCD is used that does not have a good receive offset. Either the receive offset does not allow for enough room for the configured packet header, or there is not enough room for the DMA list entry that would have been written.
25	RXQUE deque state machine is locked. This bit is set when an RXQUE deque operation is attempted in order to get a piece of user data for the DMA list and there is no data available. When using two pieces of data in a DMA list entry, both must be available.
24	Bad LCD update operation. This bit is set when a receive LCD update was attempted on an offset that falls in the transmit portion of the LCD, or when an LCD that is out of range is updated.
23-20	RXAAL nano status. These bits can be set from the RXAAL nano code. If they are used, they are documented in the code load instructions (that is, aal.txt). They may or may not be used.
19-16	RXCRC nano status. These bits can be set from the RXCRC nano code. If they are used, they are documented in the code load instructions (that is, crc.txt). They may or may not be used.
15-12	RXXLT drop 3 nano status. These bits can be set from the RXXLT nano code for drop 3. If they are used, they are documented in the code load instructions (that is, atm.txt). They may or may not be used.
11-8	RXXLT drop 2 nano status. These bits can be set from the RXXLT nano code for drop 2. If they are used, they are documented in the code load instructions (that is, atm.txt). They may or may not be used.
7-4	RXXLT drop 1 nano status. These bits can be set from the RXXLT nano code for drop 1. If they are used, they are documented in the code load instructions (that is, atm.txt). They may or may not be used.
3-0	RXXLT drop 0 nano status. These bits can be set from the RXXLT nano code for drop 0. If they are used, they are documented in the code load instructions (that is, atm.txt). They may or may not be used.

12.5: REASM Interrupt Enable Register

Used to enable interrupts for REASM status conditions. When set, the corresponding status condition generates an interrupt from REASM to INTST. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

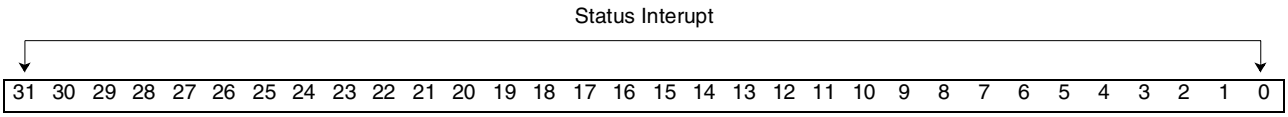
Length 32 bits

Type Read/Write

Address XXXX 1608 and 60C

Power On Value X'00000000'

Restrictions None



Bit(s)	Description
31-0	See the bitwise descriptions for the *REASM Status Register* on page 331.

12.6: REASM DEBUG State Selector Register

Selects which entity states are surfaced. This register specifies which entity states are surfaced. Use this register only under the advice of IBM technical support.

Length 2 x 32 bits

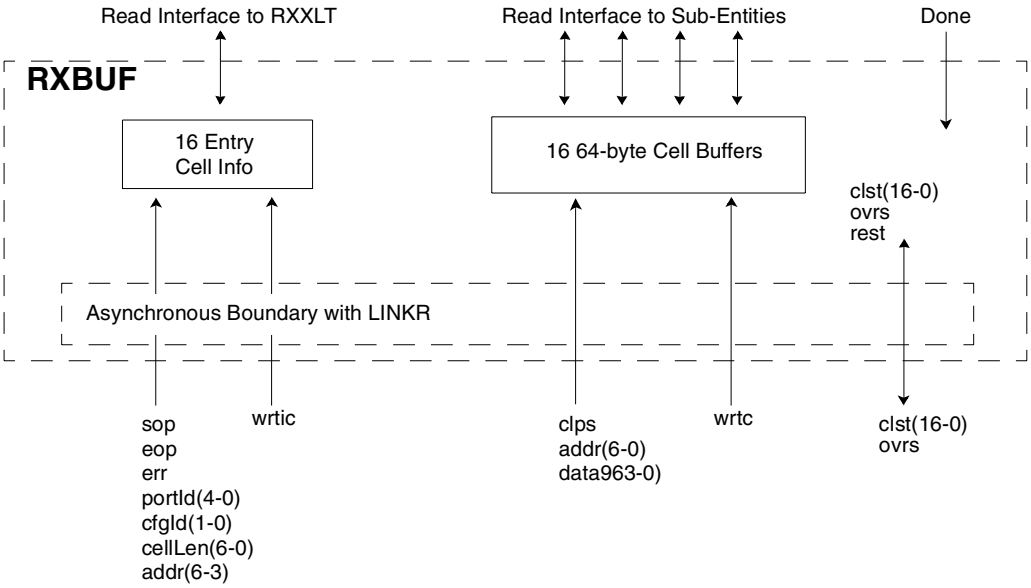
Type Read/Write

Address XXXX 1620-24

Power On Value X'00000000'

Restrictions None

RXBUF Block Diagram



RXBUF Functional Description

RXBUF provides the cell/packet buffering mechanism between LINKR and the rest of REASM. The buffering provides enough storage for 16 64-byte cells and the associated control information for each cell buffer. LINKR writes the cell/packet data and control information into the buffers. The buffering mechanism is then exposed to the remainder of the REASM sub-entities for reading.

The cell buffers are sequenced through in sequential order regardless of which port received cells arrive on. Thus, a total of 16 cells of buffering exists and is used as a shared 16-cell FIFO.

12.7: RXBUF Cell Data Buffer Address

Provides the read/write address for accessing cell data buffer. This register provides the read/write address for accessing cell data buffer. When RXBUF Cell Data Buffer Read/Write Port is read/written, this register is auto-incremented. When the last address is read/written, the address rolls over to zero.

Length	11 bits
Type	Read/Write
Address	XXXX 1640
Power On Value	X'00000000'
Restrictions	Low two bits are always zero

12.8: RXBUF Cell Data Buffer Read/Write Port

Provides read/write access to receive cell data buffer. The array is divided into 16 64-byte buffers used to buffer/assemble cells received from the line. When this register is read/written, the address provided by RXBUF Cell Data Buffer Address is used to select the array word to be accessed. RXBUF Cell Data Buffer Address is auto-incremented on each read/write. Thus, this port can be read/written multiple times to read/write the entire array.

Length	256 words x 32 bits
Type	Read/Write
Address	XXXX 1648
Power On Value	X'00000000'
Restrictions	The array can only be accessed in diagnostic mode. If not in diagnostic mode, zero is returned on reads and writes are silently ignored.

12.9: RXBUF Cell Info Buffer Address

Provides the read/write address for accessing cell information buffer. This register provides the read/write address for accessing cell information buffer. When RXBUF Cell Info Buffer Read/Write Port is read/written, this register is auto-incremented. When the last address is read/written the address rolls over to zero.

Length	6 bits
Type	Read/Write
Address	XXXX 1644
Power On Value	X'00000000'
Restrictions	Low two bits are always zero

12.10: RXBUF Cell Info Buffer Read/Write Port

Provides read/write access to the Receive Cell Info buffer. The array is divided into 16 areas used to provide information about each received cell from LINKR. When this register is read/written, the address provided by RXBUF Cell Info Buffer Address is used to select the array word to be accessed. RXBUF Cell Info Buffer Address is auto-incremented on each read/write. Thus, this port can be read/written multiple times to read/write the entire array.

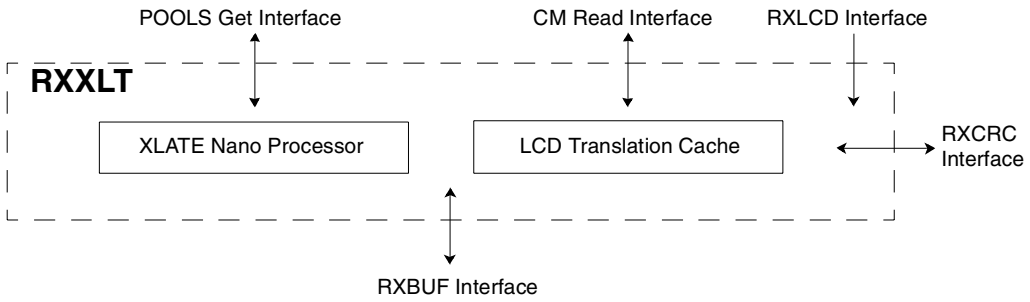
Length	16 words x 32 bits
Type	Read/Write
Address	XXXX 1650
Power On Value	X'00000000'
Restrictions	The array can only be accessed in diagnostic mode. If not in diagnostic mode, zero is returned on reads and writes are silently ignored.

12.11: RXBUF Receive Buffer Threshold

Provides a method to monitor number of cell buffers in use. The value of this register is used to compare against the number of active cell buffers. When this threshold is exceeded, the status is raised in the REASM status register.

Length	5 bits
Type	Read/Write
Address	XXXX 1660
Power On Value	X'00000000'
Restrictions	None

RXXLT Block Diagram



RXXLT Functional Description

RXXLT is the first stage in the cell processing pipeline. RXXLT provides general LCD translation facilities and link level statistics. These are provided via a nano-processor and nano-programs.

Up to four unique drops/ports can be supported, each with a unique configuration. Each port is able to run a unique nano-program to do LCD address translation. For example, one port may be a packet-based PHY using 64-byte segments with PPP LCD translation, and another port may be an ATM cell based PHY. The drop/config number (0-3) along with the port id is passed to RXXLT from RXBUF/LINKR. The drop number (0-3) is used to specify which nano-program is executed by RXXLT to translate cell/packet information into an LCD address. This is different from previous versions of the processor which had a fixed LCD translation mechanism. RXXLT instruction formats are not defined here and are IBM Confidential.

RXXLT uses the resulting LCD address to load the LCD cache for the next stage of the pipeline. The LCD address and cell buffer address are passed to RXCRC upon completion of processing.

There are a number of degrees of freedom in the LCD translation. Generally, the nano program performs the following steps:

- Gather some bytes from the cell buffer
- Do some error checking and default LCD checking
- Select and shift appropriate bits to form an LCD translate table index
- Read the LCD translate table to get an LCD index
- Generate the LCD address from the LCD index and the LCD base address

There are eight general purpose registers per PHY drop and four drops. These eight registers contain values that the nano-code uses, and are available on a per port basis. For example, the LCD translate table addresses would reside in these registers (this is different from previous versions of the processor where these registers were at fixed addresses). Other items that might reside in these registers are default/error LCD addresses, compare values, and masks. Because these are general purpose registers, multiple LCD tables or multiple default LCDs can be specified. One quarter of the total registers are available to each port, so there can be an LCD translate table (etc.) for each port.

The total LCD translate tables and LCD table sizes are only limited in size by the amount of memory that is available to the IBM3206K0424. The LCD indexes are limited to 16 bits, and LCD addresses are always 128-byte aligned.

The LCD translation code must execute in one cell time in order to run at full bandwidth. The only variable portion of the code execution time is the reads to IBM3206K0424 memory when reading the LCD translation

table. Thus, while a double lookup is possible, it may not meet the time constraints of running at full bandwidth.

The following pseudo code shows some of the types of LCD translations that are possible using the nano-processor.

Note: The following do not imply types/numbers of instructions needed, but the different variable names imply different general purpose registers are being used, so variables are specified on a per port basis.

In the following code there are shift, mask, and compare values that are not specified. These are values that would be customizable at run time. These will be customized via the general purpose registers.

The following code uses nomenclature that matches the IBM3206K0424 (zero-based big endian). So bit(1) in a comm spec might be bit(0) in the following code. The comments use spec nomenclature so you can match the two up.

Standard ATM

```
atmH = read 4 bytes from rxbuf at offset 0  -- read the atm header from cell

if non_user_data {                    -- non-user data as specified in pti
lcdAddr = defaultLcd                  -- field is handled by the deafult lcd
return
}

tblIndex  =  (atmH and vciMask) >> 4       -- gather pertinent vci bits
tblIndex |= ((atmH and vpiMask) >> X)      -- gather pertinent vpi bits
                                           -- X depends on num of vci bits used
if non-masked bits on {                 -- check for out of range
out of range counter++                  -- update counter
if mode(X) {                         -- if configured to flush out of
flush cell                        -- range cells flush and we are done
}
else {
lcdAddr = errorLcdAddr
}
return
}

if tblIndex == 0 {                      -- check for zero id
zero id counter++                       -- update counter
if mode(X) {                         -- if configured to flush zero id
flush cell                        -- cells flush and we are done
}
else {
lcdAddr = errorLcdAddr
}
return
}

lcdIndex = Lookup(lcdTable, tblIndex)      -- read index from cm using tableIndex
lcdAddr = lcdBase + lcdIndex*128           -- calc lcd addr from index
```

PPP

```
label = read 4 bytes from rxbuf at offset 0 -- read the ???
if label == 0xff030021 {                    -- what is this field called??
read 1 bytes from rxbuf at offset 4         -- read the ip header version (5th byte)
if IPVER_HEADERLGT == 0x45 {
tblIndex = read byte from offset 5       -- read QOS field (6th byte)
lcdIndex = Lookup(lcdTable, tblIndex)   -- read index from cm using tableIndex
lcdAddr = lcdBase + lcdIndex*128         -- calc lcd addr from index
}
else {
lcdAddr = defaultPortLcd                -- use port default lcd
}
}
else {
lcdAddr = defaultPortLcd                  -- use port default lcd
}
```

Q.922 2 Byte Addressing

```
head = read 2 bytes from rxbuf at offset 0  -- read the header (network bytes 0-1)
if (bit(0) == 1) && (bit(8) == 0) {         -- EA bits indicate two byte addr
-- byte 0 bit 1 == 0 && byte 1 bit 1 == 1
tblIndex  =  (head and dlciMask0) >> X    -- gather 6 of 10 dlci bits from byte 0
tblIndex |= ((head and dlciMask1) >> X)   -- gather 4 of 10 dlci bits from byte 1
lcdIndex = Lookup(lcdTable, tblIndex)     -- read index from cm using tableIndex
lcdAddr = lcdBase + lcdIndex*128          -- calc lcd addr from index
}
else {
lcdAddr = errorLcdAddr                    -- not a two byte addr
}
```

Q.922 4 Byte Addressing

```
head = read 4 bytes from rxbuf at offset 0  -- read the header (network bytes 0-3)
if (bit(0)  == 1) && (bit(8)  == 0) &&      -- EA bits indicate four byte addr
(bit(16) == 0) && (bit(24) == 0) {       -- byte 0 bit 1 == 0 && byte 1 bit 1 == 0 &&
-- byte 2 bit 1 == 0 && byte 3 bit 1 == 1
if dcBit == 1 {                          -- check for dc bit being set
lcdAddr = defaultPortLcd                 -- and surface on default port lcd
}
else {                                -- gather 16 least significant dlci bits
tblIndex  =  (head and dlciMask1) >> X  -- gather 3 dlci bits from byte 1
tblIndex |= ((head and dlciMask2) >> X) -- gather 7 dlci bits from byte 2
tblIndex |= ((head and dlciMask3) >> X) -- gather 6 dlci bits from byte 3
lcdIndex = Lookup(lcdTable, tblIndex)   -- read index from cm using tableIndex
lcdAddr = lcdBase + lcdIndex*128        -- calc lcd addr from index
}
}
else {
lcdAddr = errorLcdAddr                    -- not a four byte addr
}
```

FUNI 2.0 2 Byte Addressing

```
head = read 2 bytes from rxbuf at offset 0  -- read the header (network bytes 0-1)
if (bit(0) == 1) && (bit(8) == 0) {          -- EA bits indicate two byte addr
-- byte 0 bit 1 == 0 && byte 1 bit 1 == 1
if (bit(2) == 0) && (bit(9) == 0) {       -- FID1 & FID2 == 0 (byte 0 bit 2 & byte 1 bit 3)
tblIndex  =  (head and faMask1) >> X    -- gather 6 of 10 FA bits from byte 0
tblIndex |= ((head and faMask0) >> X)   -- gather 4 of 10 FA bits from byte 1
lcdIndex = Lookup(lcdTable, tblIndex)   -- read index from cm using tableIndex
lcdAddr = lcdBase + lcdIndex*128        -- calc lcd addr from index
}
else {
lcdAddr = defaultPortLcd                -- otherwise surface on port default lcd
}
}
else {
lcdAddr = errorLcdAddr                   -- not a two byte addr
}
```

FUNI 2.0 4 Byte Addressing

```
head = read 4 bytes from rxbuf at offset 0  -- read the header (network bytes 0-3)
if (bit(0)  == 1) && (bit(8)  == 0) &&      -- EA bits indicate four byte addr
(bit(16) == 0) && (bit(24) == 0) {       -- byte 0 bit 1 == 0 && byte 1 bit 1 == 0 &&
-- byte 2 bit 1 == 0 && byte 3 bit 1 == 1
if (bit(18) == 0) && (bit(25) == 0) {       -- FID1 & FID2 == 0 (byte 0 bit 2 & byte 1 bit 3)
-- gather 16 vpi/vci bits
tblIndex  =  (head and vciMask0) >> 1   -- gather X vci bits from byte 3
tblIndex  =  (head and vciMask1) >> X   -- gather X vci bits from byte 2
tblIndex  =  (head and vciMask2) >> X   -- gather X vci/vpi bits from byte 1
tblIndex  =  (head and vpiMask3) >> X   -- gather X vpi bits from byte 0
lcdIndex = Lookup(lcdTable, tblIndex)   -- read index from cm using tableIndex
lcdAddr = lcdBase + lcdIndex*128        -- calc lcd addr from index
}
else {
lcdAddr = defaultPortLcd                -- otherwise surface on port default lcd
}
}
else {
lcdAddr = errorLcdAddr                   -- not a four byte addr
}
```

12.12: RXXLT Register Array Address Port

Provides the read/write address for accessing register array. This register provides the read/write address for accessing register array. When RXXLT Register Array Read/Write Port is read/written, this register is auto-incremented. When the last address is read/written the address rolls over to zero.

Length	7 bits
Type	Read/Write
Address	XXXX 1680
Power On Value	X'00000000'
Restrictions	Low two bits are always zero

12.13: RXXLT Register Array Read/Write Port

Provides read/write access to the register array. The array is divided into four groups of eight registers. Each group is associated with a receive port (0-3). So each nano-program has eight registers to use. These registers have intended uses and also serve as the link level counters, so they are not general purpose registers.

When this register is read/written, the address provided by RXXLT Register Array Address Port is used to select the array word to be accessed. RXXLT Register Array Address Port is auto-incremented on each read/write. Thus, this port can be read/written multiple times to read/write the entire array.

Length	32 words x 32 bits
Type	Read/Write
Address	XXXX 1688
Power On Value	X'00000000'
Restrictions	None

12.14: RXXLT Processor State Selector

Allows user to select which data should be accessed with RXXLT Processor State Read/Write Port. This register provides the encoded selector for accessing internal processor registers and state via reads/writes to the RXXLT Processor State Read/Write Port.

Length 4 bits

Type Read/Write

Address XXXX 1698

Power On Value X'00000000'

Restrictions None

Bit(s)	Description
3-0	The following are the meanings of the encoded values: 0000 Accumulator register 0001 Header register 0010 LCD index register 0011 Reserved 0100 Reserved 0101 Flags 0110 Reserved 1111 Instruction ptr

12.15: RXXLT Processor State Read/Write Port

Provides read/write access to the internal state of the processor. The internal processor state is externalized for debug and testing reasons. See the description of *RXXLT Processor State Selector* for definitions on the addresses.

Length 32 bits

Type Read/Write

Address XXXX 169C

Power On Value X'00000000'

Restrictions Processor state can only be set in diagnostic mode.

12.16: RXXLT Instruction Array Address Port

This register provides the read/write address for accessing the instruction array. When RXXLT Instruction Array Read/Write Port is read/written, this register is auto-incremented. When the last address is read/written, the address rolls over to zero.

Length	9 bits
Type	Read/Write
Address	XXXX 1684
Power On Value	X'00000000'
Restrictions	Low two bits are always zero

12.17: RXXLT Instruction Array Read/Write Port

Provides read/write access to the instruction array. The instruction array is divided into four groups of 32 entries. Each group is associated with a receive port (0-3) so a nano-program can be loaded for each active port. When this register is read/written, the address provided by RXXLT Instruction Array Address Port is used to select the array word to be accessed. RXXLT Instruction Array Address Port is auto-incremented on each read/write. Thus, this port can be read/written multiple times to read/write the entire array.

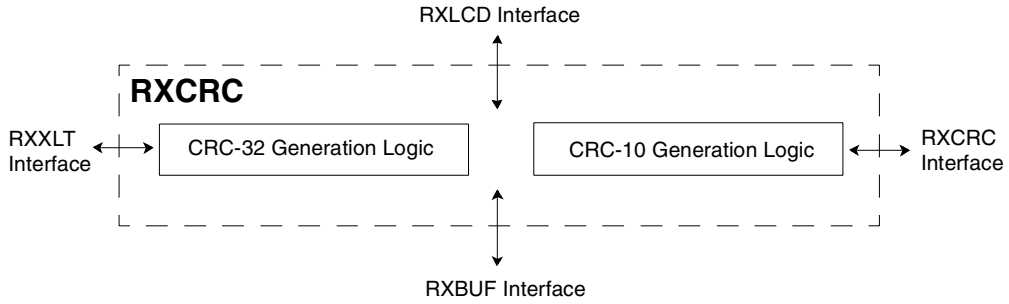
Length	128 words x 19 bits
Type	Read/Write
Address	XXXX 1690
Power On Value	X'00000000'
Restrictions	None, but the instruction stream of an active nano-program should not be written

12.18: RXXLT Last LCD Index Register

These registers provide the previous LCD index that was used for the corresponding port.

Length	16 bits
Type	Read/Write
Address	XXXX 16a0-ac
Power On Value	X'0000'
Restrictions	Can only be written in diagnostic mode

RXCRC Block Diagram



RXCRC Functional Description

RXCRC is the second stage in the cell processing pipeline. It performs the ATM CRC-32 (Ethernet FCS) and ATM CRC-10 functions if necessary. RXCRC gets LCD type, state, and seed information from the LCD cache, and updates the cache on completion. The results are passed to RXAAL upon completion, along with the LCD address.

12.19: RXCRC Instruction Array Address Port

This register provides the read/write address for accessing the instruction array. When RXCRC Instruction Array Read/Write Port is read/written, this register is auto-incremented. When the last address is read/written, the address rolls over to zero.

Length	10 bits
Type	Read/Write
Address	XXXX 16C4
Power On Value	X'00000000'
Restrictions	Low two bits are always zero

12.20: RXCRC Instruction Array Read/Write Port

Provides read/write access to the instruction array. The instruction array contains a single nano-program. When this register is read/written, the address provided by RXCRC Instruction Array Address Port is used to select the array word to be accessed. RXCRC Instruction Array Address Port is auto-incremented on each read/write. Thus, this port can be read/written multiple times to read/write the entire array.

Length 256 words x 19 bits

Type Read/Write

Address XXXX 16D0

Power On Value X'00000000'

Restrictions None, but the instruction stream of an active nano-program should not be written

12.21: RXCRC Processor State Selector

Allows user to select which data should be accessed with RXCRC Processor State Read/Write Port. This register provides the encoded selector for accessing internal processor registers and state via reads/writes to the RXCRC Processor State Read/Write Port.

Length 3 bits

Type Read/Write

Address XXXX 16D8

Power On Value X'00000000'

Restrictions None

Bit(s)	Description
2-0	The following are the meanings of the encoded values: 000 Accumulator register 001 Header register 010 Reserved 011 Reserved 100 Reserved 101 Flags 110 Reserved 111 Instruction ptr

12.22: RXCRC Processor State Read/Write Port

Provides read/write access to the internal state of the processor. The internal processor state is externalized for debug and testing reasons. See the description of *RXCRC Processor State Selector* on page 345 for definitions on the addresses.

Length	32 bits
Type	Read/Write
Address	XXXX 16Dc
Power On Value	X'00000000'
Restrictions	Processor state can only be set in diagnostic mode

12.23: RXCRC Last LCD Index Register

These registers provide the previous LCD index that was used for the corresponding port.

Length	16 bits
Type	Read/Write
Address	XXXX 16E0
Power On Value	X'0000'
Restrictions	Can only be written in diagnostic mode

12.24: RXCRC Checksum Protocol Registers

These registers provide additional protocol bytes for which checksum calculations should be enabled. The first register allows up to four protocols to be enabled with headers that are similar to UDP and TCP which use a pseudo-header. The second register allows up to four protocols to be enabled with headers that are similar to ICMP (no pseudo header). IP, UDP, TCP, V4 ICMP, and V6 ICMP are automatically recognized and should not be specified again in these registers.

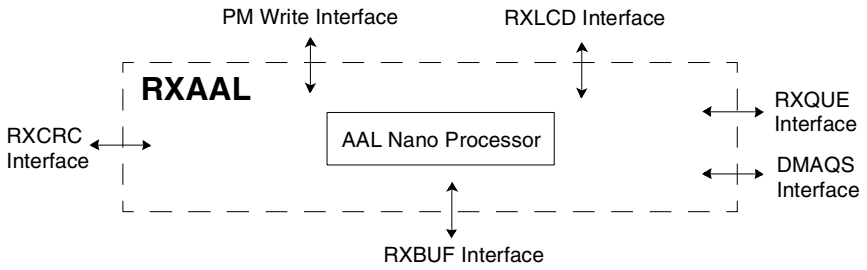
Each byte specifies a different protocol.

Length	2x32 bits
Type	Read/Write
Address	Checksum Type: With header XXXX 16E4 Checksum Type: With no header XXXX 16E8
Power On Value	X'00000000'
Restrictions	None

RXAAL Functional Description

The following is a block diagram of RXAAL:

RXAAL Block Diagram



RXAAL performs the cell and packet reassembly functions. This includes the AAL processing and moving cell data and packet headers to Packet Memory.

The nano-program uses state and configuration informations from the LCD cache to perform the necessary function for each cell. The nano-processor is capable of executing programs to run the following types of reassembly:

- AAL5
- AAL3/4
- Raw cells
- Non-user data (this might be the same as raw)
- Packets
- MPEG FIFO Mode

RXAAL also performs the cell/packet post processing step. This includes event generation to RXQUE, cut through processing, scatter processing, and DMA enqueues.

12.25: RXAAL Instruction Array Address Port

This register provides the read/write address for accessing the instruction array. When RXAAL Instruction Array Read/Write Port is read/written, this register is auto-incremented. When the last address is read/written, the address rolls over to zero.

Length	11 bits
Type	Read/Write
Address	XXXX 1704
Power On Value	X'00000000'
Restrictions	Low two bits are always zero

12.26: RXAAL Instruction Array Read/Write Port

Provides read/write access to the instruction array. The instruction array contains a single nano-program. When this register is read/written, the address provided by RXAAL Instruction Array Address Port is used to select the array word to be accessed. RXAAL Instruction Array Address Port is auto-incremented on each read/write. Thus, this port can be read/written multiple times to read/write the entire array.

Length	512 words x 24 bits
Type	Read/Write
Address	XXXX 1710
Power On Value	X'00000000'
Restrictions	None, but the instruction stream of an active nano-program should not be written

12.27: RXAAL Processor State Selector

Allows user to select which data should be accessed with RXAAL Processor State Read/Write Port. This register provides the encoded selector for accessing internal processor registers and state via reads/writes to the RXAAL Processor State Read/Write Port.

Length 3 bits

Type Read/Write

Address XXXX 1718

Power On Value X'00000000'

Restrictions None

Bit(s)	Description
2-0	The following are the meanings of the encoded values: 000 Accumulator register 001 Header register 010 Reserved 011 Reserved 100 Reserved 101 Flags 110 Reserved 111 Instruction ptr

12.28: RXAAL Processor State Read/Write Port

Provides read/write access to the internal state of the processor. The internal processor state is externalized for debug and testing reasons. See the description of *RXAAL Processor State Selector* on page 349 for definitions on the addresses.

Length 32 bits

Type Read/Write

Address XXXX 171C

Power On Value X'00000000'

Restrictions Processor state can only be set in diagnostic mode.

12.29: RXAAL Last LCD Index Register

This register provides the previous LCD index that was used.

Length	16 bits
Type	Read/Write
Address	XXXX 1720
Power On Value	X'0000'
Restrictions	None

12.30: RXAAL Transmit Queue Length Compression Configuration

This register allows the user to configure how the transmit queue lengths should be compressed for use in the receive packet header. CSKED provides twelve transmit queue lengths specified in bytes. A 32-bit register (the *Bytes Queued Counters* in CSKED) is available for the high, medium, and low priority queues for each of the four PHY ports. Using the full counts in the receive packet header generally uses too much room. This register allows the user to configure how this information should be compressed for use in the receive packet header.

Length 4 bits

Type Read/Write

Address XXXX 1730

Power On Value X'00000000'

Restrictions None

Bit(s)	Description
3-0	The following are the options: 0000 Use the full register representation for port zero only (3 - 32 bit words) 0001 Use the 2K scaled representation for port zero only (1 - 32 bit words) 0010 Use the 4K scaled representation for port zero only (1 - 32 bit words) 0011 Use the 8K scaled representation for port zero only (1 - 32 bit words) 0100 Use the 16K scaled representation for port zero only (1 - 32 bit words) 0101 Use the 32K scaled representation for port zero only (1 - 32 bit words) 0110 Use the 64K scaled representation for port zero only (1 - 32 bit words) 0111 Use the 128K scaled representation for port zero only (1 - 32 bit words) 1000 Reserved 1001 Use the 2K scaled representation for all ports (3 - 32 bit words) 1010 Use the 4K scaled representation for all ports (3 - 32 bit words) 1011 Use the 8K scaled representation for all ports (3 - 32 bit words) 1100 Use the 16K scaled representation for all ports (3 - 32 bit words) 1101 Use the 32K scaled representation for all ports (3 - 32 bit words) 1110 Use the 64K scaled representation for all ports (3 - 32 bit words) 1111 Use the 128K scaled representation for all ports. (3 - 32 bit words) For example, using option "0001," a single 32-bit word is used. The most significant byte contains the transmit queue length for the high priority queue for port zero divided by 2K bytes. If the scaled count overflows (greater than 2K*0xff), a value of 0xff is used. The next byte contains the scaled count for the medium priority queue, and the third byte contains the scaled count for the low priority queue. The least significant byte is not used. Using option "1010" three 32-bit words are used. The first word contains the scaled counts for the high priority queue. The second word contains the scaled counts for the medium priority queue. The third word contains the scaled counts for the low priority queue. Within each word, the first byte contains the scaled count for port zero, and the subsequent bytes are used for the other ports (1, 2, 3). The counts are divided by 4K in this case.

12.31: RXAAL Packet Header Configuration

Allows user to configure the contents of each optional packet header word, and specify how many optional packet words are used.

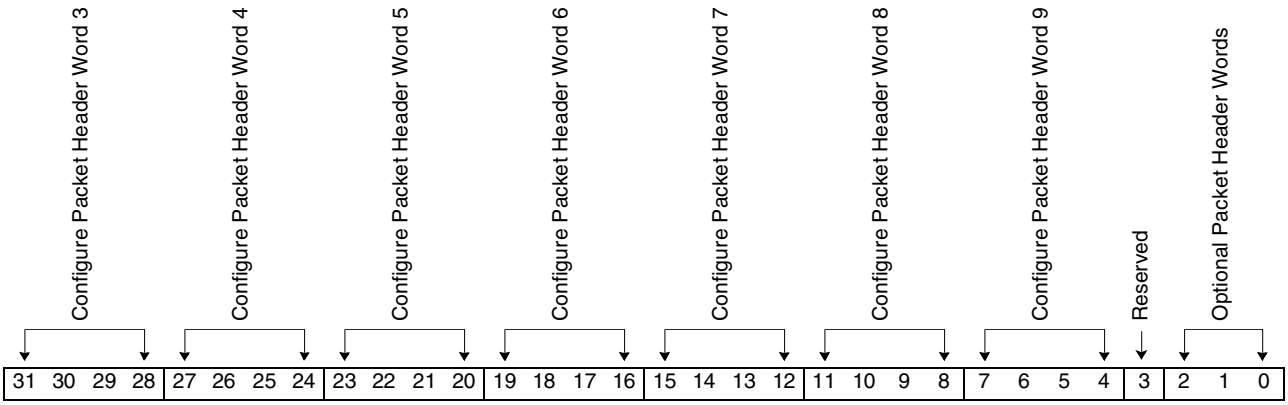
This register configures the contents of each optional packet header word in the optional portion of the packet header. There are four possible configurations, and the configuration used is selected with the packHeadSel field in the receive LCD.

The first three words of the packet header are fixed, and up to seven additional words can be configured. The low nibble of this register specifies how many optional packet header words are used, and the remaining nibbles of the register configure each packet header word if used.

Note: The base receive and transmit packet headers must be compatible if internal cell or packet routing is being used. The receive packet header becomes the transmit packet header in this scenario.

User 0 and user 1 values can be used to place non-standard values in the packet header. These values are built by the nano-code, and are then placed in the packet header. In order to use these values, the nano-code must be customized. Do this only under the advisement of IBM technical support.

Length	32 bits	
Type	Read/Write	
Address	Config 0	XXXX 1740
	Config 1	XXXX 1744
	Config 2	XXXX 1748
	Config 3	XXXX 174C
Power On Value	X'00000000'	
Restrictions	None	



Bit(s)	Description
31-28	Configure packet header word 3
27-24	Configure packet header word 4
23-20	Configure packet header word 5
19-16	Configure packet header word 6
15-12	Configure packet header word 7

Bit(s)	Description
11-8	Configure packet header word 8
7-4	Configure packet header word 9
3	Reserved
2-0	Specifies how many optional packet header words to use. Each nibble specifies what should be selected for the corresponding packet header word. The following are the options: 0000 Start Timestamp 0001 End Timestamp 0010 ATM Header 2 0011 Host data 0100 VBA - Virtual buffer address 0101 Transmit queue length word 0 0110 Transmit queue length word 1 0111 Transmit queue length word 2 1000 User 0 1001 User 1 1010 AAL5 trailer, two user bytes and the length 1011 VBA with the number of header bytes in the low 10 bits 1100 Reserved

12.32: RXAAL Error Count Register

Maintains a count of detected errors. This register maintains a count of error conditions that are detected. For example, CRC errors and other protocol types of errors are counted. This count is useful when the chip is configured to only surface good packets.

When this counter overflows, a counter overflow event is generated to RXQUE.

Length 32 bits

Type Read/Write

Address XXXX 1734

Power On Value X'00000000'

Restrictions None

12.33: RXAAL Dropped Count Register

This register maintains a count of packets that are dropped due to a lack of resource. For example, if no POOLS buffer (real or virtual mode) is available, the packet is dropped and this counter is incremented. This count is useful when the chip is configured to only surface good packets.

When this counter overflows, a counter overflow event is generated to RXQUE.

Length	32 bits
Type	Read/Write
Address	XXXX 1738
Power On Value	X'00000000'
Restrictions	None

12.34: RXAAL Maximum SDU Length Register

Specifies the maximum SDU size for a packet. This register contains the maximum SDU size for a packet. This size includes only the protocol data length of the packet. For example, this length would be compared with the AAL5 length field in the AAL5 trailer. When a packet is completely received, this register is used to make sure it does not exceed the MSDU specified.

Length	18 bits
Type	Read/Write
Address	XXXX 173C
Power On Value	X'0000FFFF'
Restrictions	None

12.35: RXAAL OAM LCD Information Register

This register specifies the reassembly information for OAM cells. The format of this register is equivalent to word zero of the receive LCD. The following fields are valid: ppMode, size, storeCrc10, rxqNum, rxPoolId, rxOffset, and cutThruSel. Refer to the format of LCD word zero for the Raw receive LCDs.

Length	32 bits
Type	Read/Write
Address	XXXX 172C
Power On Value	X'0000FFFF'
Restrictions	None

12.36: RXALL - Scatter/Cut Through Info Registers

These registers specify the scatter/cut through configurations. A configuration is selected in the LCD via the cut through selector field when doing cut through/scatter mode. A configuration consists of three registers. The first two registers, described here, define the four possible configurations for scatter/cut through. The third register, RXALL - Scatter/Cut Through Flag Registers, is described in "RXALL - Scatter/Cut Through Flag Registers.

The first register, one of cti(0-3), is defined as follows:

12.36.1: Scatter/Cut Through Info Register 1

Length 32 bits

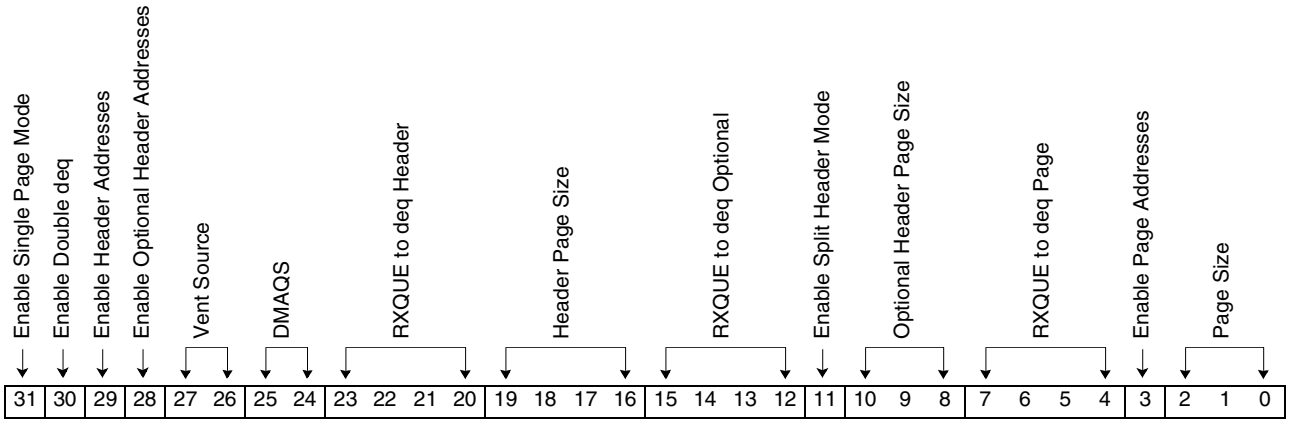
Type Read/Write

Address

Config 0	XXXX 1750
Config 1	XXXX 1754
Config 2	XXXX 1758
Config 3	XXXX 175C
Config 4	XXXX 1770
Config 5	XXXX 1774
Config 6	XXXX 1778
Config 7	XXXX 177C

Power On Value X'00000000'

Restrictions None



Bit(s)	Description
31	Enable single page mode
30	Enable double deq for packet header buffers to get virtual addresses
29	Enable header addresses vs. descriptor (1->addresses 0->descriptor)
28	Enable optional header addresses vs. descriptor (1->addresses 0->descriptor)

Bit(s)	Description
27-26	Event source: Specify how the event data (if any) is formed. Normally, the DMA descriptor is used for DMA events, but when the entire DMA descriptor is built there is no descriptor address to use. The following sources are possible: 00 Use DMA descriptor address (DMA queue address where built) 01 Use source address 10 Use destination address
25-24	DMAQS DMA queue to enqueue DMA descriptor to
23-20	RXQUE to deq header DMA descriptor addresses from
19-16	Header page size: Specifies the page size of the buffer that the header is DMAed into. The following encodings are used: 0 128 bytes 1 256 bytes 2 512 bytes 3 1K bytes 4 2K bytes 5 4K bytes 6 8K bytes 7 6K bytes 8 32K bytes 9 64K bytes
15-12	RXQUE to deq optional header DMA descriptor addresses from
11	Enable Split Header Mode (optional header buffer)
10-8	Optional header page size: Specifies the page size of the buffer that the header is DMAed into. The following encodings are used: 0 64 bytes 1 128 bytes 2 256 bytes 3 512 bytes 4 1K bytes 5 2K bytes 6 4K bytes 7 8K bytes
7-4	RXQUE to deq page DMA descriptor addresses from
3	Enable page addresses vs. page descriptor
2-0	Page size: Specifies the page size of the page buffers that the scatter pages are DMAed into. The following encodings are used: 0 512 bytes 1 1K bytes 2 2K bytes 3 4K bytes 4 8K bytes 5 16K bytes 6 32K bytes 7 64K bytes

12.36.2: Scatter/Cut Through Info Register 2

Length 32 bits

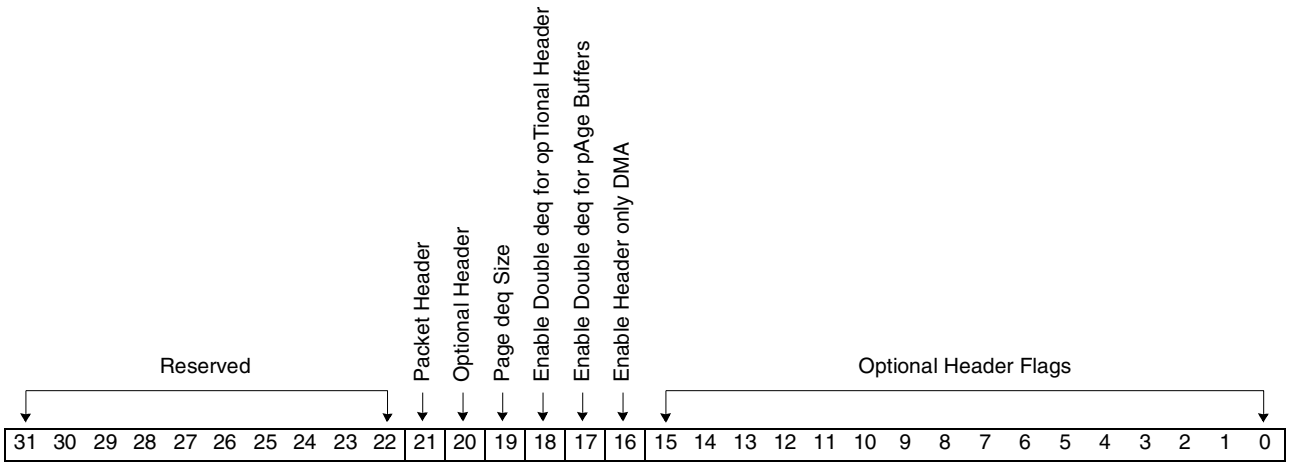
Type Read/Write

Address

Config 0	XXXX 1750
Config 1	XXXX 1754
Config 2	XXXX 1758
Config 3	XXXX 175C
Config 4	XXXX 1770
Config 5	XXXX 1774
Config 6	XXXX 1778
Config 7	XXXX 177C

Power On Value X'00000000'

Restrictions None



Bit(s)	Description
31-22	Reserved.
21	Packet Header deq size: 0 = 32 bit, 1 = 64 bit
20	Optional Header deq size: 0 = 32 bit, 1 = 64 bit
19	Page deq size: 0 = 32 bit, 1 = 64 bit
18	Enable double deq for optional header buffers to get virtual addresses
17	Enable double deq for page buffers to get virtual addresses
16	Enable header only DMA. When set, only the header bytes as specified in the LCD or from RXCRC will be DMAed along with the DMA list and packet header
15-0	Optional Header Flags. These flags are used when DMAing the optional header page.

12.37: RXALL - Scatter/Cut Through Flag Registers

Use to specify the scatter/cut through flags. These registers specify the scatter/cut through flags. A flag register is selected in the LCD via the cut through selector field when doing cut through/scatter mode. The flags are used when building DMA descriptors for scatter pages and the first scatter buffer (packet header, DMA list, etc.).

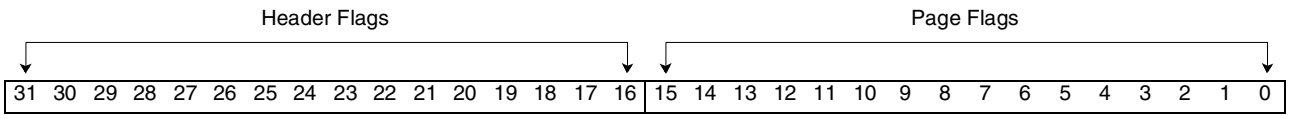
Length 32 bits

Type Read/Write

Address
Config 0 XXXX 1760
Config 1 XXXX 1764
Config 2 XXXX 1768
Config 3 XXXX 176C

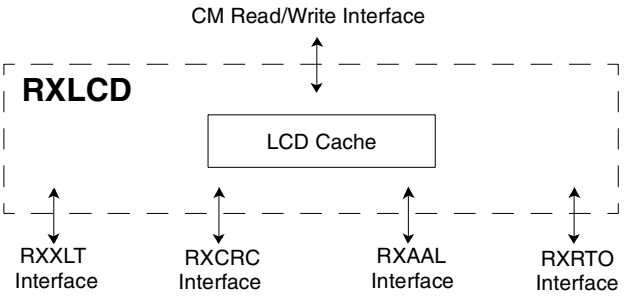
Power On Value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-16	Header Flags	These flags are used when DMAing the packet header, DMA list, and header bytes from the packet.
15-0	Page Flags	These flags are used when DMAing a scatter page (other than the header page).

RXLCD Block Diagram



RXLCD Functional Description

RXLCD provides an LCD cache for REASM sub-entities. This cache holds the last four receive LCDs. The sub-entities can request to load an LCD, read the LCD words, and update parts of the LCD.

12.38: RXLCD Cache Data Array Address Port

This register provides the read/write address for accessing the LCD cache data array. When the RXLCD Cache Data Array Read/Write Port is read/written, this register is auto-incremented. When the last address is read/written, the address rolls over to zero.

The cache is organized as 64 32-bit words. Each 16 words comprises an LCD. The first four words and the last word of each LCD do not contain valid data and can not be written. These locations return zero on reads.

Length	6 bits
Type	Read/Write
Address	XXXX 1780
Power On Value	X'00000000'
Restrictions	The low two bits are always zero

12.39: RXLCD Cache Data Array Read/Write Port

Provides read/write access to the LCD cache data array. When this register is read/written, the address provided by RXLCD Cache Data Array Address Port is used to select the array word to be accessed. RXLCD Cache Data Array Address Port is auto-incremented on each read/write so this port can be read/written multiple times to read/write the entire array.

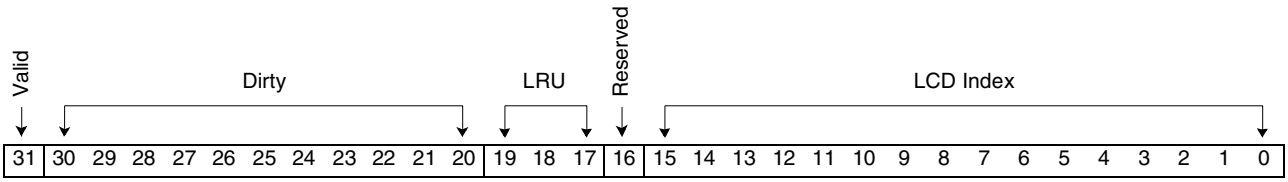
The cache is organized as 64 32-bit words. Each 16 words comprises an LCD. The first four words and the last word of each LCD do not contain valid data and can not be written. These locations return zero on reads.

Length	64 words x 32 bits
Type	Read/Write
Address	XXXX 1788
Power On Value	X'00000000'
Restrictions	Must be in diagnostic mode to read/write the cache

12.40: RXLCD Cache Line Info Registers

These registers provide the cache line tags, valid, and dirty bits. There is a register for each of the four cache lines.

Length	4 x 32 bits
Type	Read/Write
Address	XXXX 17a0-7ac
Power On Value	X'00000000'
Restrictions	Must be in diagnostic mode to write these registers



Bit(s)	Description
31	Valid
30-20	Dirty bits
19-17	LRU bits
16	Reserved
15-0	LCD index

12.41: RXLCD Mode Register

This register provides a means to control cache operation.

Length 1 bit

Type Read/Write

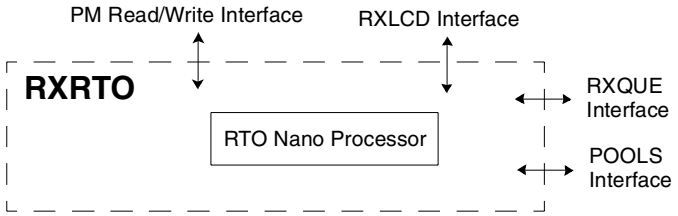
Address XXXX 17b8-7bc

Power On Value X'00000000'

Restrictions None

Bit(s)	Name	Description
0	Flush all entries	When set, all dirty entries are flushed to memory but remain in the cache. This bit will reset when the operation is complete.

RXRTO Block Diagram



RXRTO Functional Description

RXRTO performs periodic reassembly timeout processing and LCD update operations.

Reassembly Timeout (RTO) Processing

Reassembly timeout processing is generally an AAL5 operation. It is supported for other LCD types as well. It can be enabled on an LCD basis by turning on the RTO enable bit in the LCD. The following registers also need to be properly set up to run RTO processing:

- RXRTO RTO LCD Table Bound Registers
- RXRTO Reassembly Timeout Value Register
- RXRTO Reassembly Timeout Pre-Scaler Register

See the register descriptions for more register details.

The LCD table registers define the LCD table that the RTO process examines. The value register is used as a compare value against a counter that counts based on a pre-scaler. Each time the registers compare, RTO processing is started for a single LCD and the time base is reset. RTO processing checks the RTO test and set bit. If it is reset, it sets it and continues. If it is set, then a timeout occurs and the LC is placed in error state and the current packet is surfaced to the user via an event. Any resource associated with the packet must be recovered by software. For example, if the LCD is setup to use scatter mode, then there may be scatter DMA pages in the DMA list that need to be returned to the proper receive queue. The RTO bit is reset with each inbound cell received. An LCD needs to be touched twice to cause a timeout (once to set it and once to detect that it is already set).

The time base starts running as soon as the RTO processing is complete. Thus, RTO processing is a low priority task.

Shutting Down an LCD

To shut down a receive LCD, the following steps should be followed:

- Clear the entry for this LCD in the LCD table (to stop receiving cells for this LCD)
- Do an LCD update operation that sets LCD state to down
- Read the LCD REASM ptr
- If REASM ptr is non-zero and LCD is set up to do cut through, be sure to free any DMA descriptor that was added with cut through operation
- If REASM ptr is non-zero and LCD is set up to do scatter, be sure to free any pages in the DMA list
- If REASM ptr is non-zero, free it to POOLS

12.42: RXRTO LCD Update Data Registers

These two registers are used to specify data to write into the receive LCD on the update LC operation. They contain the data used in the LC update operation. For more information on their use, see the *RXRTO LCD Update Op Registers* on page 364.

The Update Date Register changes to contain the updated data written to the LC word while the operation is completing.

The second set of LCD update registers is meant for the core to use, but is available for general use.

Length	32 bits
Type	Read/Write
Address	Update 1 register XXXX 17C0 Update 2 register XXXX 17D0
Power On Value	X'00000000'
Restrictions	None

12.43: RXRTO LCD Update Mask Registers

These two registers are used to specify which data to write into the LCD on the update LC operation. They contain the mask used in the LC update operation. For more information on their use, see the *RXRTO LCD Update Op Registers* on page 364.

The second set of LCD update registers is meant for the core to use, but is available for general use.

Length	32 bits
Type	Read/Write
Address	Update 1 register XXXX 17C4) Update 2 register XXXX 17D4
Power On Value	X'00000000'
Restrictions	None

12.44: RXRTO LCD Update Op Registers

Used to specify the LCD word to update. This operation is used to update a portion of the receive LCD. If this operation is not used, then software or the IBM3206K0424 updates of the LCD may be lost because the receive LCD is cached in the IBM3206K0424.

This register is written with the address of the LCD word to update. Once this register is written the update operation starts. All subsequent reads or writes to the data, mask, or update registers are held off until the operation completes. A read-modify-write will occur to update the portion specified by the mask with the masked value in the data register.

Normally this register would not be read. However, if it is read then the low order bit is read as '0' and the next lowest order bit (bit 1) is read as the busy bit. This signifies whether an operation is still going on. If an operation is still going on, then a new write to any of the data, mask, or update operation registers is held off until the original operation is complete.

The second set of LCD update registers is meant for the core to use, but is available for general use.

Length	32 bits
Type	Read/Write
Address	Update 1 register XXXX 17C8 Update 2 register XXXX 17D8
Power On Value	X'00000000'
Restrictions	The low order two bits are not writable

12.45: RXRTO RTO LCD Table Bound Registers

Used to specify the lower/upper bounds of the LCD table. The lower bound should be initialized to the LCD index of the first LCD in the LCD table if reassembly timeout processing is to be done. The upper bound should be initialized to the LCD index of the last LC in the LC table if reassembly timeout processing is to be done.

Length	16 bits
Type	Read/Write
Address	Lower bound 1 register XXXX 17E0 Upper bound 1 register XXXX 17E4 Lower bound 2 register XXXX 17F0 Upper bound 2 register XXXX 17F4
Power On Value	X'00000000'
Restrictions	None

12.46: RXRTO Reassembly Timeout Value Register

Used to specify the time interval used for reassembly timeout processing. This register is the number of pre-scaler intervals between reassembly processing. The pre-scaler interval is determined by RXRTO Reassembly Timeout Pre-Scaler Register. A single LC is checked for reassembly timeout during each reassembly processing interval.

When this register is set to ’0’, reassembly timeout processing is disabled.

For more information on how reassembly timeout conditions are processed see *Reassembly Timeout (RTO) Processing* on page 362.

Length	32 bits	
Type	Read/Write	
Address	Timeout 1 register	XXXX 17E8
	Timeout 2 register	XXXX 17F8
Power On Value	X'00000000'	
Restrictions	None	

12.47: RXRTO Reassembly Timeout Pre-Scaler Register

Used to specify the time interval of each RTO timer tick. This register determines the number of 15 ns intervals between RTO timer ticks. The value in the register plus 1 is the number of 15 ns intervals between RTO timer ticks. Thus, the default value of ’0’ means that the RTO timer ticks every 15 ns. If a value of nine is placed in this register, the RTO timer ticks every 150 ns (10 * 15 ns).

For more information on how reassembly timeout conditions are processed, see *Reassembly Timeout (RTO) Processing* on page 362.

Length	16	
Type	Read/Write	
Address	Prescale 1	XXXX17EC
	Prescale 2	XXXX 17FC
Power On Value	X’00000000’	
Restrictions	None	

Entity 13: Receive Queues (RXQUE)

Functional Description

RXQUE has a single function: to manage the receive queues for software by providing an easy to use primitive interface. When talking about the receive queues, the term rxq is used to talk about a receive queue, and the term deq is used to refer to a dequeue operation, and the term enq is used to refer to an enqueue operation.

Receive Queue Interface

A group of sixteen receive queues is available for software use. The receive queues hold events or user specified data.

Each queue entry (event) is either 32 or 64 bits and contains two fields: event-identifier and event-information. The seven least significant bits in the entry contain the event-identifier field. The most significant bits in the entry comprise the event-information field.

Warning: In order to maintain the atomicity of 64-bit atomic transfers, the user must ensure that 64-bit transfers are bus atomic within the particular bus system in which IBM3206K0424 is being used.

The event information typically contains a pointer (when low order bits are zeroed) to a packet buffer, a cell buffer, or an LCD. It can also contain a DMA descriptor address or user-specified data. *Event Summary and Routing Information* on page 367 lists the different event types.

Event Summary and Routing Information (Page 1 of 3)

Event Number	Description	Event Information	Error	Count	Tx Comp	ABR	POOLS
0000000 = 00	AAL5 packet event (packet complete)	Packet/LCD					
0000001 = 01	AAL5 packet header event (packet start)	Packet					
0000010 = 02	AAL5 packet with bad CRC	Packet/LCD	X				
0000011 = 03	AAL5 packet with bad length field	Packet/LCD	X				
0000100 = 04	AAL5 packet that exceeds maximum length in LC	Packet/LCD	X				
0000101 = 05	AAL5 packet timeout	Packet/LCD	X				
0000110 = 06	AAL5 packet forward abort	Packet/LCD	X				
0000111 = 07	AAL5 packet CPI field not equal to zero	Packet/LCD	X				
0001110 = 0e	AAL5 FIFO packet	Packet					
0001000 = 08	Cell event (user data)	Packet/LCD					
0001001 = 09	NUD cell event (non-user data)	Packet/LCD					
0001010 = 0a	NUD cell with bad CRC-10	Packet/LCD	X				
0001011 = 0b	Bad cell - bad HEC	Packet	X				
0001100 = 0c	Bad cell - out of range	Packet	X				
0001101 = 0d	Bad cell - index equal zero	Packet	X				
0010000 = 10	AAL0 cell dropped - lack of POOLS buffers	LCD	X				
0010001 = 11	AAL5 cell dropped - lack of POOLS buffers	LCD	X				
0010010 = 12	OAM cell dropped - lack of POOLS buffers	LCD	X				
0011000 = 18	Total user cells receive counter overflow	LCD		X			
0011001 = 19	Total user cells rx clp=0 counter overflow	LCD		X			
0011010 = 1a	Total user cells tx counter overflow	LCD		X			
0011011 = 1b	Total user cells tx clp=0 counter overflow	LCD		X			
0011100 = 1c	Threshold 1 crossed - down	LCD			X		
0011101 = 1d	Threshold 1 crossed - up	LCD			X		
0011110 = 1e	Threshold 1 crossed - down	LCD			X		
0011111 = 1f	Threshold 2 crossed - up	LCD			X		
0100000 = 20	Transmit complete	Packet			X		
0100001 = 21	Transmit complete buffer freed	Packet/LCD			X		
0100010 = 22	"bad" found in first word of packet	Packet			X		
0100011 = 23	Connection closed	LCD			X		
0100100 = 24	Transmit DMA complete	Descriptor					
0100101 = 25	Receive DMA complete	Descriptor					
0100110 = 26	Transmit DMA complete with error	Descriptor					
0100111 = 27	Receive DMA complete with error	Descriptor					
0101000 = 28	Transmit DMA complete with virtual error	Descriptor					

Event Summary and Routing Information (Page 2 of 3)

Event Number	Description	Event Information	Error	Count	Tx Comp	ABR	POOLS
0101001 = 29	Zero address in DMA descriptor SRC/DST address	Descriptor					
0101010 = 2a	Transmit buffer allocated						
0101100 = 2c	ADTF Event	LCD				X	
0101101 = 2d	CRM Event	LCD				X	
0101110 = 2e	CCR=0 Event	LCD				X	
0101111 = 2f	RM Cell Event	Packet				X	
0110000 = 30	User event						
0110001 = 31	User event						
0110010 = 32	User event						
0110011 = 33	User event						
0110100 = 34	User event						
0110101 = 35	User event						
0110110 = 36	User event						
0110111 = 37	User event						
0111000 = 38	Virtual memory resource event	Packet/LCD	X				
0111001 = 39	Buffer overflow event	Packet/LCD	X				
0111010 = 3a	No DMA descriptor for AAL7 packet	Packet/LCD	X				
0111011 = 3b	DMA canceled for AAL7 packet due to error	Descriptor					
0111100 = 3c	No scatter pages available and packet complete	Packet/LCD	X				
0111101 = 3d	Entity counter overflow event	Counter		X			
0111110 = 3e	POOLS status event	Status					X
1000000 = 40	Frame event (good frame)						
1000001 = 41	Frame event (error)		X				
1000010 = 42	Frame event (protocol error)		X				
1000011 = 43	Frame event (dropped - lack of buffers)		X				
1000100 = 44	Frame event (reserved)						
1000101 = 45	Frame event (reserved)						
1000110 = 46	Frame event (reserved)						
1000111 = 47	Frame event (reserved)						
1010000 = 50	PCORE event						
1010001 = 51	PCORE event						
1010010 = 52	PCORE event						
1010011 = 53	PCORE event						
1010100 = 54	PCORE event						
1010101 = 55	PCORE event						

Event Summary and Routing Information (Page 3 of 3)

Event Number	Description	Event Information	Error	Count	Tx Comp	ABR	POOLS
1010110 = 56	PCORE event						
1010111 = 57	PCORE event						
1011000 = 58	REASM counter-overflow event	Counter		X			
1011001 = 59	SEGBF counter-overflow event	Counter		X			
1011100 = 5c	System - receive queue event (start of buffer)	Previous lower bound					
1011101 = 5d	System - receive queue event (end of buffer)	Next lower bound					
1011110 = 5e	Timestamp event	Timestamp					
1011111 = 5f	64-bit timestamp event	Timestamp					
1100100 = 64	Tx DMA complete	Descriptor					
1100101 = 65	Rx DMA complete	Descriptor					
1100110 = 66	Tx DMA complete with error	Descriptor					
1100111 = 67	Rx DMA complete with error	Descriptor					
1101000 = 68	Tx DMA complete with virtual error	Descriptor					
1101001 = 69	Zero address in DMA descriptor SRC/DST address	Descriptor					

AAL5 Packet Events

For AAL5 packet events, the event specifies the packet buffer address, and the event type field specifies the type of packet event. The following event types are defined:

Bit(s)	Name	Description
0x00=0000000	Normal AAL5 Packet Event (Packet Complete)	This event specifies that an AAL5 packet was received and has passed all AAL5 protocol checks (CRC, length). The event information contains a pointer to the packet.
0x0e=0001110	Normal AAL5 FIFO Packet Event	This event specifies that an AAL5 FIFO packet was received and has passed all AAL5 protocol checks (CRC, length,...). The event information contains a pointer to the packet.
0x01=0000001	Normal AAL5 Packet Header Threshold Event (Packet Start)	This event specifies that the AAL5 packet header threshold was exceeded as set in the LCD. The event information contains a pointer to the packet header, and the user can access up to the packet header threshold bytes of data.
0x02=0000010	AAL5 Packet with Bad CRC was RX on LC	This event specifies that an AAL5 packet was received and the AAL5 CRC is bad. The event information contains either a pointer to the packet if receiving bad frames, or a pointer to the LCD on which this packet was received.
0x03=0000011	AAL5 Packet with Bad Length Field was RX on LC	This event specifies that a AAL5 packet was received and the AAL5 length field is bad. For example, there is too much data or not enough, but typically the bad CRC is detected first. The event information contains either a pointer to the packet if receiving bad frames, or a pointer to the LCD on which this packet was received.
0x04=0000100	AAL5 Packet that exceeds Max Len in LC was RX	This event specifies that an AAL5 packet was received but the amount of data has exceeded the maximum length as specified in the LCD or in the MSDU register. The event information contains either a pointer to the packet if receiving bad frames, or a pointer to the LCD on which this packet was received.
0x05=0000101	AAL5 Packet Timeout on this LC	This event specifies that a reassembly timeout has occurred for an AAL5 packet that was being reassembled. The event information contains either a pointer to the packet if receiving bad frames, or a pointer to the LCD on which this packet was received.
0x06=0000110	AAL5 Packet Forward Abort	This event specifies that an AAL5 packet was terminated with a forward abort. The event information contains either a pointer to the packet if receiving bad frames, or a pointer to the LCD on which this packet was received.
0x07=0000111	AAL5 Packet CPI Field not equal to Zero	This event specifies that a AAL5 packet was received and the AAL5 CPI field was not set to '0' which is an AAL5 protocol violation. The event information contains either a pointer to the packet if receiving bad frames, or a pointer to the LCD on which this packet was received.

Cell Events

For AAL0 events, the event specifies a cell buffer address, and the event type field specifies type of AAL0 event. The following event types are defined:

Bit(s)	Name	Description
0x08=0001000	AAL0 Cell Event	This event specifies that an AAL0 (non-FIFO mode) cell was received. The event information contains a pointer to the cell.
0x09=0001001	Non-User Data Cell Event	This event specifies that a non-user data cell was received and the CRC-10 was good if checking was enabled. The event information contains a pointer to the cell.
0x0a=0001010	Non-User Data Cell with Bad CRC-10 Event	This event specifies that a non-user data cell was received and the CRC-10 was bad. The event information contains either a pointer to the cell if receiving bad frames, or a pointer to the LCD on which this cell was received.
0x0b=0001011	Cell with Bad HEC Event	This event specifies that a cell was received with a bad HEC. The event information contains a pointer to the cell.
0x0c=0001100	Cell with VP/VC Out Of Range Event	This event specifies that a cell was received with a VP/VC that was out of range. The event information contains a pointer to the cell.
0x0d=0001101	Cell with VC Index Equal Zero	This event specifies that a cell was received with a VC index equal zero. The event information contains a pointer to the cell.

LC Events

For LC events, the event specifies a LC, and the event type field specifies what happened on the LC. The following event types are defined:

Bit(s)	Name	Description
0x10=0010000	AAL0 Cell was Dropped due to Lack of POOLS Buffers	This event specifies that a AAL0 (non-FIFO mode) cell was received, but was discarded because no POOL buffers were available. The event information contains a pointer to the LCD on which this cell was received.
0x11=0010001	AAL5 Cell was Dropped due to Lack of POOLS Buffers	This event specifies that the first AAL5 cell for a packet was received, but was discarded because no POOL buffers were available. The event information contains a pointer to the LCD on which this cell was received.
0x12=0010010	Non-user Data Cell was Dropped due to Lack of POOLS Buffers	This event specifies that a non-user data cell was received, but was discarded because no POOL buffers were available. The event information contains a pointer to the LCD on which this cell was received.
0x17=0010111	Reserved	Reserved
0x18=0011000	LC Total User Cells RX Counter Overflow	This event specifies that the TUC RX counter in the LCD has overflowed. The event information contains a pointer to the LCD.
0x19=0011001	LC Total User Cells CLP=0 RX Counter Overflow	This event specifies that the TUC w/CLP=0 RX counter in the LCD has overflowed. The event information contains a pointer to the LCD.
0x1a=0011010	LC Total User Cells TX Counter Overflow	This event specifies that the TUC TX counter in the LCD has overflowed. The event information contains a pointer to the LCD.
0x1b=0011011	LC Total User Cells CLP=0 TX Counter Overflow	This event specifies that the TUC w/CLP=0 TX counter in the LCD has overflowed. The event information contains a pointer to the LCD.

ABR Events

The following events are used for ABR processing and are routed to the receive queue specified in the ABR event routing register.

Bit(s)	Name	Description
0x2c=0101100	ADTF Event	This event specifies that the ADTF timer expired. The event information contains a pointer to the LCD.
0x2d=0101101	CRM Event	This event specifies that the CRM count has been exceeded. The event information contains a pointer to the LCD.
0x2e=0101110	CCR = 0 Event	This event specifies that CCR = 0. The event information contains a pointer to the LCD.
0x2f=0101111	RM Cell Rx-ed	This event specifies that a RM cell was received. The event information contains a pointer to the receive buffer.

Miscellaneous Events

Bit(s)	Name	Description
0x1c=0011100	Thresh 1 Event - Down	This event specifies that a memory management threshold was crossed downwards. The event information contains the LCD address.
0x1d=0011101	Thresh 1 Event - Up	This event specifies that a memory management threshold was crossed upwards. The event information contains the LCD address.
0x1e=0011110	Thresh 2 Event - Down	This event specifies that a memory management threshold was crossed downwards. The event information contains the LCD address.
0x1f=0011111	Thresh 2 Event - Up	This event specifies that a memory management threshold was crossed upwards. The event information contains the LCD address.
0x20=0100000	Transmit Complete	This event specifies that a packet/cell was successfully transmitted. The event information contains a pointer to the buffer transmitted.
0x21=0100001	Transmit Complete Buffer Freed	This event specifies that a packet/cell was successfully transmitted and the buffer was freed back to POOLS. The event information contains a pointer to the buffer transmitted.
0x22=0100010	Transmit Bad	This event specifies that a packet was to be transmitted, but the buffer was marked as bad, so was canceled. This is caused by getting a page fault when DMAing into the transmit packet buffer. The event information contains a pointer to the bad buffer.
0x23=0100011	Connection Closed	This event specifies that all packets for the given LCD have been transmitted. The event information contains the LCD address.
0x24=0100100	TX DMA Complete (into IBM3206K0424)	This event specifies that a TX DMA completed successfully. The event information depends on how the DMA was set up. Event-identifier 0x64 is an alias.
0x25=0100101	RX DMA Complete (out of IBM3206K0424)	This event specifies that an RX DMA completed successfully. The event information depends on how the DMA was set up. Event-identifier 0x65 is an alias.
0x26=0100110	TX DMA Complete with Error (into IBM3206K0424)	This event specifies that a TX DMA had errors. The event information depends on how the DMA was set up. Event-identifier 0x66 is an alias.
0x27=0100111	RX DMA Complete with Error (out of IBM3206K0424)	This event specifies that an RX DMA had errors. The event information depends on how the DMA was set up. Event-identifier 0x67 is an alias.
0x28=0101000	TX DMA Complete with Virtual Error	This event specifies that a TX DMA had a virtual error. The remainder of the DMA descriptor was cancelled. The event information contains the DMA descriptor address. Event-identifier 0x68 is an alias.
0x29=0101001	DMA Desc has Zero Address	This event specifies that a DMA descriptor source destination address was zero. The remainder of the DMA descriptor was cancelled. The event information contains the DMA descriptor address. Event-identifier 0x69 is an alias.
0x2a=0101010	Transmit-Buffer Allocated	
0x30=0110000	User Defined	
0x31=0110001	User Defined	
0x32=0110010	User Defined	
0x33=0110011	User Defined	
0x34=0110100	User Defined	
0x35=0110101	User Defined	
0x36=0110110	User Defined	
0x37=0110111	User Defined	

Miscellaneous Events (Continued)

Bit(s)	Name	Description
0x38=0111000	Virtual Memory Resource Event	This event specifies that an AAL5 cell was received, but could not be written into the buffer because a virtual memory boundary was crossed and a buffer was not available to fill in the next segment. This can also happen if the cell crosses a boundary that would make the buffer larger than the virtual buffer size. The event information contains either a pointer to the packet if receiving bad frames, or a pointer to the LCD on which this packet was received.
0x39=0111001	Buffer Overflow Event	This event specifies that an AAL5 cell was received, but could not be written into the buffer because it would exceed the real buffer size. The event information contains either a pointer to the packet if receiving bad frames, or a pointer to the LCD on which this packet was received.
0x3a=0111010	No DMA Desc for AAL7 Packet Event	This event specifies that an AAL7 (AAL5 with cut through) packet was completed, but no DMA descriptors were available to DMA the packet header. The event information contains either a pointer to the packet if receiving bad frames, or a pointer to the LCD on which this packet was received.
0x3b=0111011	DMA Cancelled for AAL7 Packet Due to Error Event	This event specifies that a DMA descriptor enqueued with a cut-through operation was cancelled because an error condition was detected with the packet (CRC, length,...). The event information contains the system descriptor address that was cancelled.
0x3c=0111100	No Pages Available for AAL5 Scatter Packet	This event specifies that an AAL5 packet completed with no errors, but there was a lack of scatter buffers to complete the scatter DMA. The user must interrogate the partial DMA list in the packet header to recover the system pages. Once this is done, the packet should be freed. The user can alternately treat this as a good packet and complete the packet processing by setting up additional DMAs to move the remaining data to system pages. The event information contains the IBM3206K0424 buffer address.
0x3d=0111101	IBM3206K0424 Counter Overflow Event	This event specifies that a counter overflow event has been raised. The event information specifies which counter overflowed. Multiple counter overflows can be specified with a single event. The following is the definition of the event information: Bit 27 GPDMA Write DMA Byte Count Overflow Bit 26 GPDMA Read DMA Byte Count Overflow Bit 25 RXQUE Timestamp Counter Overflow Bit 24 PCINT Performance Counter 1 Overflow Bit 23 PCINT Performance Counter 2 Overflow
0x3e=0111110	POOLS Status Event	This event specifies that a POOLS status event has been raised. The event information contains the status. See *Buffer Pool Management (POOLS)* on page 247 for the definition.
0x5e=1011110	Timestamp Event	This event specifies that a timestamp event has been placed ahead of next event. The event information contains the timestamp.
0x5f=1011111	64-bit Timestamp Event	This event specifies that a timestamp event has been placed ahead of next event. The most significant word in the event-information field contains the full 32-bit timestamp.

Frame-Based Events

Bit(s)	Name	Description
0x40=1000000	Frame Event (good frame)	
0x41=1000001	Frame Event (error)	
0x42=1000010	Frame Event (protocol error)	
0x43=1000011	Reserved (dropped - lack of buffers)	
0x44=1000100	Reserved	
0x45=1000101	Reserved	
0x46=1000110	Reserved	
0x47=1000111	Reserved	

PCORE Events

Bit(s)	Name	Description
0x50=1010000	PCORE Event	
0x51=1010001	PCORE Event	
0x52=1010010	PCORE Event	
0x53=1010011	PCORE Event	
0x54=1010100	PCORE Event	
0x55=1010101	PCORE Event	
0x56=1010110	PCORE Event	
0x57=1010111	PCORE Event	
0x58=1011000	REASM Counter-Overflow Event	This specifies that REASM raised counter-overflow event. The event information specifies which counter overflowed.
0x59=1011001	SEGBF Counter-Overflow Event	This specifies that SEGBF raised counter-overflow event. The event information specifies which counter overflowed.

System-Receive-Queue Events

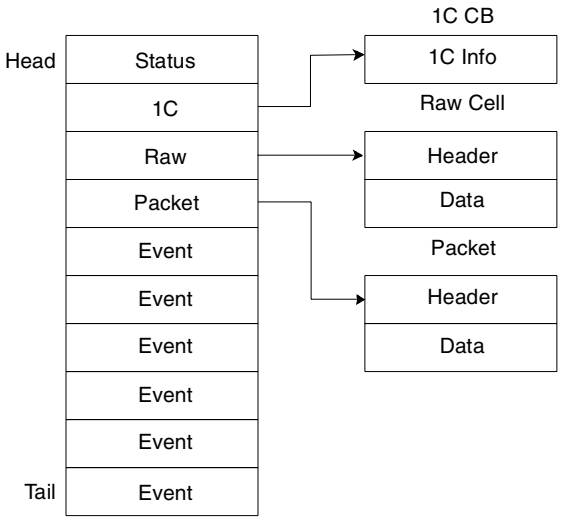
Bit(s)	Name	Description
0x5c=1011100	System-Receive-Queue Start-of-Buffer Event	
0x5d=1011101	System-Receive-Queue End-of-Buffer Event	

The receive queues are maintained by RXQUE with the following operations being available to software:

- **dequeue** - Remove the entry at the head of the queue
- **enqueue** - Add arbitrate entry at the tail of the queue

The following figure shows how events in a receive queue link to other data structures including LC control blocks, packet buffers, and cell buffers.

General Queue, Event, and Data Structure Linkage



RXQUE Structure

Each queue has a number of registers that define the queue and its behavior:

Lower bound	Pointer to starting address of queue's buffer
Properties	Indicates the queue's size, type, and behavior
Head pointer	Pointer to head of queue
Tail pointer	Pointer to the next free entry in queue - points to head if queue is full or empty
Queue length	Current length of the queue
Threshold	Length threshold used to generate interrupts
Next lower bound	Pointer to starting address of a system receive queue's next buffer

RXQUE Initialization

To set up a receive queue, at least two pieces of information are needed. The first is the receive queue's set of properties, and the second is its base address.

The following restrictions should be taken into account when setting up a queue:

- The properties register must be set up before the lower-bound and next-lower-bound registers can be set up.
- The lower bound and next lower bound must be at least 1K aligned. The low order 10 bits of these registers are not writable, so the minimum physical size of a receive queue is 1024 bytes (256 32-bit entries). The alignment should correspond to the size specified in the properties register.
- The head and tail pointers are initialized when the lower bound register is written. These registers are only writable for diagnostic purposes.
- The threshold is level sensitive, so as long as the queue length is greater than or equal to the threshold, the appropriate status bit is driven.
- All registers, except the threshold and next-lower-bound registers, can only be written in diagnostic mode and are intended to only be written once when they are set up.

RXQUE Event Routing

Events are routed to a receive queue based on the current event type and mode of the chip. Events fall into these categories:

- Normal Events
- Error Events
- Counter Events
- Transmit Complete Events
- DMA Events (tx/rx)
- ABR Events
- POOLS Status Events

See *Event Summary and Routing Information* on page 367 to see how the different events are categorized.

All events other than normal, DMA, and error events are routed using the corresponding RXQUE Event Routing Registers.

Normal events are always routed to the receive queue specified in the receive portion of the LCD.

DMAQS specifies the route for all DMA events.

Error events are special, in that they are routed based on the values of the "receive bad frame mode” bit and the "always route error events" mode bit in RXQUE Control Register. If the “always route error events” bit is on, then the error events are always routed to the error queue. Otherwise, if the receive bad frame mode is on, then the error events are routed to the receive queue specified in the receive portion of the LCD just like a normal event would be. When receive bad frames is off, then the error events are routed to the error queue.

RXQUE Normal Operation

This section describes how to use the receive queues (rxq) and the rxq operations.

The receive queue contains events for the end user to process. These events are obtained by the user by executing the rxq deq operation. The user can be notified of new events by setting up the threshold and interrupt enable registers appropriately. Otherwise, the rxq length register can be polled to check for events.

The deq operation is executed by reading the deq register address for the appropriate rxq. The event at the head of the queue is returned and the event is removed from the queue. Some events have a packet/cell buffer associated with it. This buffer is owned by the user, and it is the users responsibility to free this buffer.

The following pseudo code illustrates how an rxq could be processed:

RXQUE Dequeue Event Loop

```
// rxq was polled or int ocurred to get here

Event = RXQUE->Deq;              // read an event from rxq

if (Event neq 0) {               // need to check for null event
   EventType = Event & 0x7f;     // calc event type
   Event = Event & ~(0x7f);   // calc lc or buffer ptr

   switch (EventType) {
      case(Event1):
         ProcessSimpleEvent1(Event);
         break;
      case(Event2):
         ProcessSimpleEvent2(Event);
         break;


      case(EventX):
         ProcessSimpleEventX(Event);
         break;
   }
}
```

RXQUE Queue Full Operation

When a receive queue is full (length is equal to maximum length), the appropriate status bit is set. When a queue is full, all subsequent events are flushed until room is available in the receive queue. If a buffer was associated with the event and the RXQUE-Properties-Register bit Disable Auto-Free is not set, then that buffer is freed back to POOLS.

When an event is dropped, the event dropped status bit is set and the event data that was dropped can be found in RXQUE Last Event Dropped Register. The RXQUE Last Event Dropped Register will not be changed until the event dropped status bit is cleared.

It is not good to let a receive queue become full.

RXQUE Event Timestamping

When timestamp mode is set in the RXQUE Control Register, events are timestamped. When timestamping is enabled, a timestamp event is placed in the corresponding rxq followed by the actual event. The event information of the timestamp event carries the timestamp. The timestamp is determined from the RXQUE Timestamp Register, RXQUE Timestamp Pre-Scaler Register, and the RXQUE Timestamp Shift Register.

If the corresponding rxq is full, both events are dropped. It is possible to lose only the timestamp event or lose the actual event depending on the length of the queue and the timing of the dequeue operations.

RXQUE System Receive Queues

To set up a system receive queue, set the "Diagnostic-Mode" bit in the RXQUE Control Register. Next, set the system receive queue bit in the RXQUE Properties Register. Load the upper bound and size of event fields, as well. After this, allocate two identical buffers in system memory. Let each buffer be large enough to contain N events (the upper bound field prescribes the value N) and initialize both to all zeros. Write one buffer's starting address into the RXQUE Lower Bound Register (LOBR), and write the other's starting address into the RXQUE Next Lower Bound Register (NLBR). Finally, reset the diagnostic mode bit in the RXQUE Control Register. System-receive-queue setup is complete.

The first event enqueued to a system receive queue causes RXQUE to begin filling the buffer which LOBR references. RXQUE fills the buffer's first entry with a "System-Receive-Queue Start-of-Buffer" event whose information is the value '0'. After this, RXQUE fills the buffer's second entry with the enqueued event and writes the value '2' into the RXQUE Length Register (LENR).

The second event enqueued to the system receive queue causes RXQUE to fill the buffer's third entry and write the value '3' into LENR.

The third event enqueued to the system receive queue causes RXQUE to fill the buffer's fourth entry and write the value '4' into LENR.

This continues until LENR contains the value 'N-1'. At that time, RXQUE fills the buffer's Nth entry with a System-receive-queue end-of-buffer event whose information is the value in NLBR. After this, RXQUE begins filling the buffer which NLBR references. RXQUE fills its first entry with a "System-Receive-Queue Start-of-Buffer" event whose information is the value in LOBR. After this, RXQUE copies the contents of NLBR into LOBR and writes the value '1' into LENR. Finally, RXQUE writes the value '0' into NLBR. Subsequent events enqueued to the system receive queue fill the buffer which LOBR references. To prevent the system receive queue from becoming full, write a non-zero value into NLBR.

The system receive queue becomes full when the value in LENR becomes 'N-1' while NLBR contains the value '0'. At that time, RXQUE fills the buffer's Nth entry with a system-receive-queue end-of-buffer event whose information is the value '0'. RXQUE writes the value N into LENR and preserves the contents of LOBR while dropping subsequent enqueued events. To restart the system receive queue from the "full" state, write a non-zero value into NLBR. After restarting, the next event enqueued to the system receive queue causes RXQUE to begin filling the buffer which NLBR references. RXQUE fills the buffer's first entry with a system-receive-queue start-of-buffer event whose information is the value preserved in LOBR. RXQUE fills the buffer's second entry with the enqueued event. After this, RXQUE copies the contents of NLBR into LOBR and writes the value '2' into LENR. Finally, RXQUE writes the value '0' into NLBR. Subsequent events enqueued to the system receive queue fill the buffer which LOBR references. To prevent the system receive queue from becoming full, write a non-zero value into NLBR.

The RXQUE Queues' Status Register bit "Threshold Exceeded" indicates that the value in LENR is greater than or equal to the value in the RXQUE Threshold Register while NLBR contains the value '0'. To reset this

status bit, write a non-zero value into NLBR.

Events enqueued to a system receive queue may not be dequeued via the RXQUE Dequeue Register. To dequeue from a system receive queue, poll system memory directly. A buffer's entry is filled if its value is non-zero.

RXQUE synchronizes its internal-register operations with the initiation, rather than completion, of its system-memory operations. Therefore, the state of system memory lags the state of RXQUE.

13.1: RXQUE Lower Bound Registers

These registers specify the lower bound of the corresponding receive queue data structure. The head and tail of the receive queue are initialized when this register is written. When the receive queue wraps past the upper bound, it wraps back to the value in the lower bound register, thus implementing the receive queue as a circular buffer.

When this register is written, the corresponding receive queue is essentially reset. This is because the head, tail, and length of the queue are all reset.

The length of the RXQUE Lower Bound Register is 64 bits if all three conditions, below, are met; otherwise, the length is 32 bits.

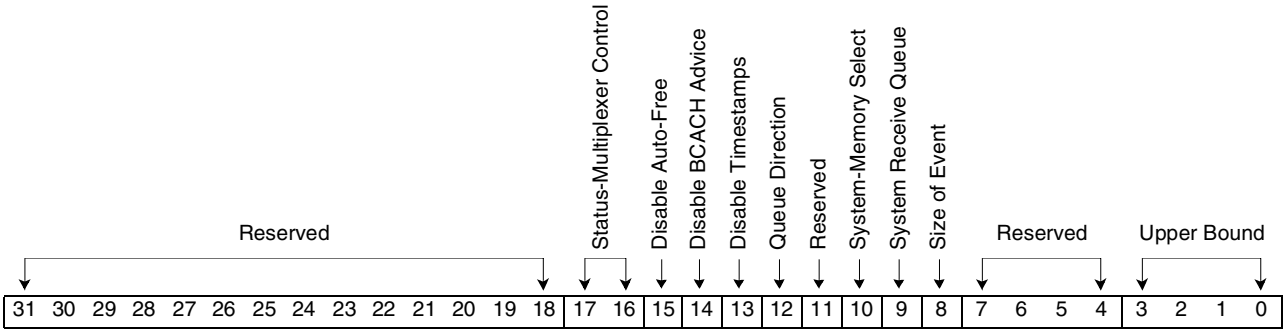
- **System Receive-Queue** in the RXQUE Properties Register is set.
- **System-Memory Select** in the RXQUE Properties Register indicates "PCI Memory."
- **Enable Master 64-bit Addressing** in the PCINT 64bit Control Register is set.

Length	32 or 64 bits	
Type	Read/Write	
Address	Queue 0	XXXX 1800
	Queue 1	XXXX 1840
	Queue 2	XXXX 1880
	Queue 3	XXXX 18C0
	Queue 4	XXXX 1900
	Queue 5	XXXX 1940
	Queue 6	XXXX 1980
	Queue 7	XXXX 19C0
	Queue 8	XXXX 1A00
	Queue 9	XXXX 1A40
	Queue 10	XXXX 1A80
	Queue 11	XXXX 1AC0
	Queue 12	XXXX 1B00
	Queue 13	XXXX 1B40
	Queue 14	XXXX 1B80
	Queue 15	XXXX 1BC0
Power on Value	X'0000000000000000'	
Restrictions	During normal operations, these registers are read only. These registers can only be written when the diagnostic bit has been set in the control register. The lower bound registers must be at least 1K aligned (low order 10 bits not writable). The alignment should also correspond to the size specified in the upper bound register. For example, it should be 4K aligned if the upper bound specifies 4K size.	

13.2: RXQUE Properties Registers

These registers specify the properties of the corresponding receive queue.

Length	32 bits	
Type	Read/Write	
Address	Queue 0	XXXX 1808
	Queue 1	XXXX 1848
	Queue 2	XXXX 1888
	Queue 3	XXXX 18C8
	Queue 4	XXXX 1908
	Queue 5	XXXX 1948
	Queue 6	XXXX 1988
	Queue 7	XXXX 19C8
	Queue 8	XXXX 1A08
	Queue 9	XXXX 1A48
	Queue 10	XXXX 1A88
	Queue 11	XXXX 1AC8
	Queue 12	XXXX 1B08
	Queue 13	XXXX 1B48
	Queue 14	XXXX 1B88
	Queue 15	XXXX 1BC8
Power on Value	X'00010001'	
Restrictions	Bits 11-0 may only be written when the diagnostic bit has been set in the control register. There are no restrictions for bits 32-12.	



Bit(s)	Name	Description
31-18	Reserved	Reserved.
17-16	Status-Multiplexer control	00 Select “Full/Empty” 01 Select “Threshold Exceeded” (power-on value) 10 Select “Head Valid” 11 Reserved
15	Disable Auto-Free	Inhibits freeing of packet-buffers when the receive queue is full.
14	Disable BCACH Advice	When set, the bcach advice is disabled for this queue. This is necessary in order to use a queue as a general purpose container for user data.
13	Disable Timestamps	When set, timestamps are disabled for this queue. This is necessary in order to run cut through modes.
12	Queue Direction	When set, the direction of the queue is assumed to be reversed. This only affects the full condition and the threshold exceeded condition. When this bit is set, the polarity of these status signals changes. Thus, the full condition becomes an empty condition, and the two threshold conditions trigger when the length of the queue is less than the corresponding threshold instead of greater than or equal. This mode is mainly used for queues that relay information from the system to the IBM3206K0424. Also, event enqueues to a queue that is reversed do not start the event latency timer (since no new event for system has arrived).
11	Reserved	Reserved
10	System-Memory Select	This is valid only when bit 9 (System Receive Queue) is set. 0 PCI Memory (power-on value) 1 On-Chip Memory
9	System Receive Queue	
8	Size of Event	0 32 bits (power-on value) 1 64 bits
7-4	Reserved	Reserved

Bit(s)	Name	Description
3-0	Upper Bound	This specifies the encoded upper bound of the corresponding receive queue data structure. The actual upper bound is calculated by adding the decoded queue size to the lower bound. When the receive queue wraps past the upper bound, it wraps back to the lower bound register, thus implementing the receive queue as a circular buffer. 0000 Reserved 0001 256 entries (power-on value) 0010 512 entries 0011 1024 entries 0100 2K entries 0101 4K entries 0110 8K entries 0111 16K entries 1000 32K entries 1001 64K entries 101- Reserved 11-- Reserved

13.3: RXQUE Head Pointer Registers

These registers point to the head element of the corresponding receive queue. During normal operations, these registers do not need to be read or written, as they are used by the IBM3206K0424 to implement the receive queues. These registers are initialized when the lower bound register for the corresponding receive queue is written.

Length	32 bits	
Type	Read/Write	
Address	Queue 0	XXXX 1810
	Queue 1	XXXX 1850
	Queue 2	XXXX 1890
	Queue 3	XXXX 18D0
	Queue 4	XXXX 1910
	Queue 5	XXXX 1950
	Queue 6	XXXX 1990
	Queue 7	XXXX 19D0
	Queue 8	XXXX 1A10
	Queue 9	XXXX 1A50
	Queue 10	XXXX 1A90
	Queue 11	XXXX 1AD0
	Queue 12	XXXX 1B10
	Queue 13	XXXX 1B50
	Queue 14	XXXX 1B90
	Queue 15	XXXX 1BD0
Power on Value	X'00000000'	
Restrictions	During normal operations, these registers are read only. These registers can only be written when the diagnostic bit has been set in the control register. The head pointer registers are four-byte aligned (low order two bits not writable). Bits 31-19 are calculated internally, and are not writable.	

13.4: RXQUE Tail Pointer Registers

These registers point to the next free element of the corresponding receive queue. During normal operations, these registers do not need to be read or written, as they are used by the IBM3206K0424 to implement the receive queues. These registers are initialized when the lower bound register for the corresponding receive queue is written.

Length	32 bits	
Type	Read/Write	
Address	Queue 0	XXXX 1814
	Queue 1	XXXX 1854
	Queue 2	XXXX 1894
	Queue 3	XXXX 18D4
	Queue 4	XXXX 1914
	Queue 5	XXXX 1954
	Queue 6	XXXX 1994
	Queue 7	XXXX 19D4
	Queue 8	XXXX 1A14
	Queue 9	XXXX 1A54
	Queue 10	XXXX 1A94
	Queue 11	XXXX 1AD4
	Queue 12	XXXX 1B14
	Queue 13	XXXX 1B54
	Queue 14	XXXX 1B94
	Queue 15	XXXX 1BD4
Power on Value	X'00000000'	
Restrictions	During normal operations, these registers are read only. These registers can only be written when the diagnostic bit has been set in the control register. The tail pointer registers are four-byte aligned (low order two bits not writable). Bits 31-19 are calculated internally, and are not writable.	

13.5: RXQUE Length Registers

These registers specify the length (number of valid entries) of the corresponding receive queue. They can be used to query the status of a receive queue.

This register is cleared when the corresponding lower bound is written.

Length	17 bits	
Type	Read	
Address	Queue 0	XXXX 1818
	Queue 1	XXXX 1858
	Queue 2	XXXX 1898
	Queue 3	XXXX 18D8
	Queue 4	XXXX 1918
	Queue 5	XXXX 1958
	Queue 6	XXXX 1998
	Queue 7	XXXX 19D8
	Queue 8	XXXX 1A18
	Queue 9	XXXX 1A58
	Queue 10	XXXX 1A98
	Queue 11	XXXX 1AD8
	Queue 12	XXXX 1B18
	Queue 13	XXXX 1B58
	Queue 14	XXXX 1B98
	Queue 15	XXXX 1BD8
Power on Value	X’00000’	
Restrictions	These registers can only be written in diagnostic mode.	

13.6: RXQUE Threshold Registers

These registers specify a queue length threshold at which the corresponding status bit is generated. These registers should be set equal to the number of queue entries that should cause status to be generated. For example, if the value was set to five, then no interrupt would be generated until five or more events were queued on the corresponding receive queue. The threshold is level sensitive, so as long as the length is greater than or equal to the threshold, the corresponding status bit is set. When this register is set to '0', no thresholding is done.

When the direction bit is set for a receive queue, the threshold has the opposite polarity. For example, as long as there are more events in the queue than specified in the threshold register, no status would be raised.

Length	17 bits	
Type	Read/Write	
Address	Queue 0	XXXX 181C
	Queue 1	XXXX 185C
	Queue 2	XXXX 189C
	Queue 3	XXXX 18DC
	Queue 4	XXXX 191C
	Queue 5	XXXX 195C
	Queue 6	XXXX 199C
	Queue 7	XXXX 19DC
	Queue 8	XXXX 1A1C
	Queue 9	XXXX 1A5C
	Queue 10	XXXX 1A9C
	Queue 11	XXXX 1ADC
	Queue 12	XXXX 1B1C
	Queue 13	XXXX 1B5C
	Queue 14	XXXX 1B9C
	Queue 15	XXXX 1BDC
Power on Value	X'00000'	
Restrictions	None	

13.7: RXQUE Dequeue Registers

These registers are used to retrieve the event at the head of the corresponding receive queue. These registers are used to retrieve the event at the head of the corresponding receive queue.

The length of an RXQUE Dequeue Register is 64 bits if Size of Event is set in its corresponding RXQUE Properties Register; otherwise, the length is 32 bits.

Length	32 or 64 bits	
Type	Read	
Address	Queue 0	XXXX 1820
	Queue 1	XXXX 1860
	Queue 2	XXXX 18A0
	Queue 3	XXXX 18E0
	Queue 4	XXXX 1920
	Queue 5	XXXX 1960
	Queue 6	XXXX 19A0
	Queue 7	XXXX 19E0
	Queue 8	XXXX 1A20
	Queue 9	XXXX 1A60
	Queue 10	XXXX 1AA0
	Queue 11	XXXX 1AE0
	Queue 12	XXXX 1B20
	Queue 13	XXXX 1B60
	Queue 14	XXXX 1BA0
	Queue 15	XXXX 1BE0
Power on Value	X’00000000’	
Restrictions	This is a read only register, and all writes will be ignored. Events are only returned when the diagnostic bit is reset in the control register, otherwise zero will be returned.	

13.8: RXQUE Enqueue Registers

These registers are used to enqueue user events at the tail of the corresponding receive queue.

The length of a RXQUE Enqueue Register is 64 bits if Size of Event is set in its corresponding RXQUE Properties Register; otherwise, the length is 32 bits.

Length	32 or 64 bits	
Type	Read/Write	
	Queue 0	XXXX 1828
	Queue 1	XXXX 1868
	Queue 2	XXXX 18A8
	Queue 3	XXXX 18E8
	Queue 4	XXXX 1928
	Queue 5	XXXX 1968
	Queue 6	XXXX 19A8
	Queue 7	XXXX 19E8
	Queue 8	XXXX 1A28
	Queue 9	XXXX 1A68
	Queue 10	XXXX 1AA8
	Queue 11	XXXX 1AE8
	Queue 12	XXXX 1B28
	Queue 13	XXXX 1B68
	Queue 14	XXXX 1BA8
	Queue 15	XXXX 1BE8
Power on Value	X’00000000’	
Restrictions	All reads result in zero. RXQUE should be enabled to do this.	

13.9: RXQUE Next Lower Bound Registers

These registers specify the next lower bound of the corresponding system receive queue data structure. See *RXQUE System Receive Queues* on page 380 for instruction about managing system receive queues.

The length of the RXQUE Next Lower Bound Register is the same as the length of the RXQUE Lower Bound Register. See *RXQUE Lower Bound Registers* on page 382 for conditions which determine the length.

Length	32 or 64 bits	
Type	Read/Write	
	Queue 0	XXXX 1830
	Queue 1	XXXX 1870
	Queue 2	XXXX 18B0
	Queue 3	XXXX 18F0
	Queue 4	XXXX 1930
	Queue 5	XXXX 1970
	Queue 6	XXXX 19B0
	Queue 7	XXXX 19F0
	Queue 8	XXXX 1A30
	Queue 9	XXXX 1A70
	Queue 10	XXXX 1AB0
	Queue 11	XXXX 1AF0
	Queue 12	XXXX 1B30
	Queue 13	XXXX 1B70
	Queue 14	XXXX 1BB0
	Queue 15	XXXX 1BF0
Power on Value	X’0000000000000000’	
Restrictions	None	

13.10: RXQUE Last Event Dropped Register

This register contains the last event that was dropped. It holds its value until the event dropped status bit is cleared.

The length of the RXQUE Last Event Dropped Register is 64 bits if Size of Dropped Event is set in the RXQUE Status Register; otherwise, the length is 32 bits.

Length	32 or 64 bits
Type	Read/Write
Address	XXXX 1C10
Power On Value	X'0000000000000000'
Restrictions	None

13.11: RXQUE Timestamp Register

Used to specify the current timestamp measured using the timestamp pre-scaler ticks. It counts based on the value in the RXQUE Timestamp Pre-Scaler Register. It can be read or written at any time. It is cleared when the pre-scaler register is written.

Length	32 bits
Type	Read/Write
Address	XXXX 1C30
Power On Value	X’00000000’
Restrictions	None

13.12: RXQUE Timestamp Pre-Scaler Register

Used to specify the time interval of each timestamp timer tick. This register determines the number of 15 ns intervals between timestamp timer ticks. The value in the register plus one is the number of 15 ns intervals between timestamp timer ticks. So, the default value of ’0’ means that the timestamp timer ticks every 15 ns. If a value of four is placed in this register, the timestamp timer ticks every 75 ns (5 x 15 ns).

Length	16 bits
Type	Read/Write
Address	XXXX 1C38
Power On Value	X’00000000’
Restrictions	None

13.13: RXQUE Timestamp Shift Register

This register determines the number of bits that the timestamps are shifted. For example, if a value of ’0’ is placed in this register, then the timestamp is not shifted, and the low order seven bits are lost. If a value of ‘2’ is placed in this register, then the timestamps are shifted two places and only the low order five bits of the timestamp are lost. This allows the user to control what portion of the timestamp is lost due to the low order event bits.

Length	3 bits
Type	Read/Write
Address	XXXX 1C3C
Power On Value	X’00000002’
Restrictions	None

13.14: RXQUE Event Routing Registers

Used to specify which receive queue different types of events should be routed to. These registers contain the receive queue that different types of events should be routed to. See *Event Summary and Routing Information* on page 367 for event type mappings.

Length	3 bits	
Type	Read/Write	
Address	Event Tx Complete	XXXX 1C40
	Event Counter Overflow	XXXX 1C44
	Event Error	XXXX 1C48
	Event POOLS Status	XXXX 1C54
	Event ABR	XXXX 1C58
Power on Value	X’00000000”	
Restrictions	None	

13.15: RXQUE Event Latency Timer Register

Used to specify the event latency time interval. This register is specified in 15 ns intervals. When a new event is placed on an rxq, the event latency timer is started (if not already started). When this timer expires, the event latency timer expired status bit is set, and the timer is stopped. The status bit must be reset before the timer is started again. Every time the status register (or prioritized status) is accessed, the timer is stopped. If this register is written while the timer is running, the new value takes effect immediately. If this register is set to '0', the latency timer does not run.

Length	32 bits
Type	Read/Write
Address	XXXX 1C20
Power On Value	X'00000000'
Restrictions	None

13.16: RXQUE Queues Status Register

Indicates the status for all receive queues.

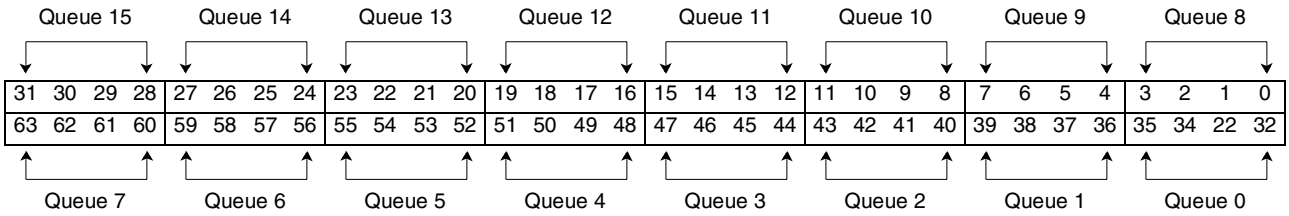
Length 64 bits

Type Read only

Address XXXX 1D40

Power On Value X'0000000000000000'

Restrictions This is a read only register



Bit(s)	Name	Description
63-60	Statue-nibble for queue 15	For each 4-bit range, bit encoding is as follows: 3 Reserved — Reserved 2 Head valid — This is set when the head is valid for the queue. If this bit is set, then a deque operation should complete successfully. 1 Threshold exceeded — This is set when the queue-length register equals or exceeds the value in the queue-threshold register. 0 Queue Full/Empty — This is set when the queue-length register is equal to the queue-maximum-length register. When the direction of the queue is reversed, this bit is set when the queue is empty.
59-56	Statue-nibble for queue 14	
55-52	Statue-nibble for queue 13	
51-48	Statue-nibble for queue 12	
47-44	Statue-nibble for queue 11	
43-40	Statue-nibble for queue 10	
39-36	Statue-nibble for queue 9	
35-32	Statue-nibble for queue 8	
31-28	Statue-nibble for queue 7	
27-24	Statue-nibble for queue 6	
23-20	Statue-nibble for queue 5	
19-16	Statue-nibble for queue 4	
15-12	Statue-nibble for queue 3	
11-8	Statue-nibble for queue 2	
7-4	Statue-nibble for queue1	
3-0	Statue-nibble for queue 0	

13.17: RXQUE Interrupt Enable Registers

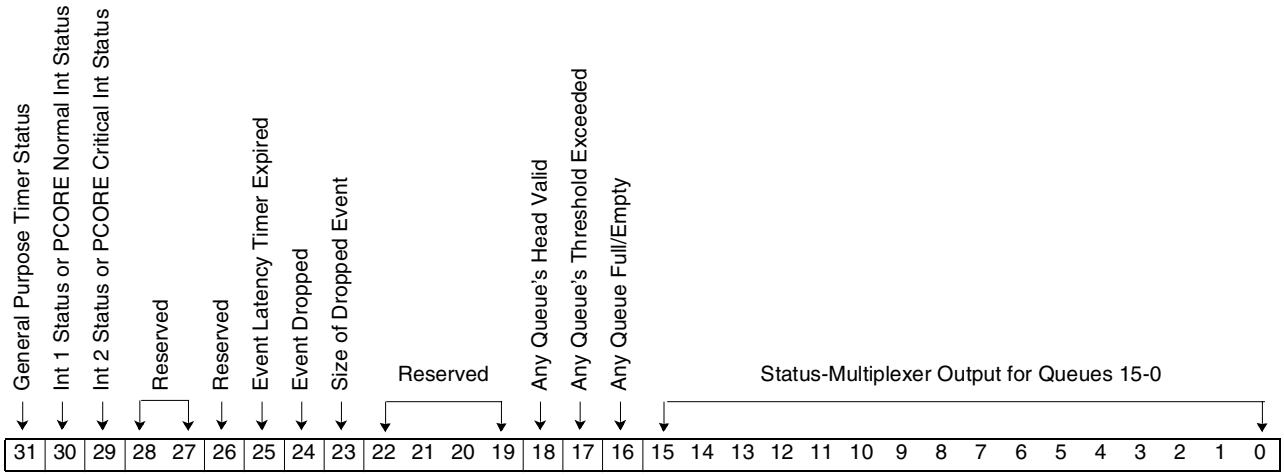
Used to specify which status register bits should be used to generate interrupts. Each mask register is used to drive a different RXQUE status bit in intst. The different masks and status bits allow two RXQUE interrupts on both the interrupt A and B pins. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See *RXQUE Status and Enabled Status Registers* on page 398 for the bit descriptions.

Length	32 bits	
Type	Clear/Set	
Address	Enable 1	XXXX 1A80 and C84
	Enable 2	XXXX 1C88 and C8C
Power on Value	Enable 1 - 2	X'00000000'
Restrictions	None	

13.18: RXQUE Status and Enabled Status Registers

This register contains the status bits used to relay RXQUE status information. The enabled version of these registers provides a version of the status register that is masked with the corresponding interrupt enable register. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	32 bits	
Type	Clear/set (None) Read only (Enable 1 - 2)	
Address	None	XXXX 1CA0 and CA4
	Enable 1	1CB0
	Enable 2	XXXX 1CB4
Power on Value	None and enable 1 - 2	X'00000000'
Restrictions	Only bits 26 down to 23 are writable. The RXQUE Enabled Status Registers are read only.	



Bit(s)	Name	Description
31	General Purpose Timer Status	This is a mirror of the general purpose timer status bit in interrupt status.
30	Int 1 Status or PCORE Normal Int Status	When read from the PCI bus, this bit indicates if there is status other than RXQUE status and general purpose timer status in the [TBD] using [TBD] as a mask. When read from the PCORE polling interface, the mask used is the [TBD].
29	Int 2 Status or PCORE Critical Int Status	When read from the PCI bus, this bit indicates if there is status other than RXQUE status and general purpose timer status in the [TBD] using [TBD] as a mask. When read from the PCORE polling interface, the mask used is the [TBD].
28-27	Reserved	Reserved
26	Sequence-Error in 64-Bit Register-Access	Reserved
25	Event Latency Timer Expired	When this bit is set, the event latency timer has expired. This indicates that new events are waiting to be processed on some queue(s) and the queue has not been processed for a period equal to the latency timer. This bit must be reset to re-enable the event latency timer.
24	Event Dropped	When this bit is set, at least one event has been dropped. RXQUE Last Event Dropped Register contains the event that was dropped.

Bit(s)	Name	Description
23	Size of Dropped Event	0 32 bits 1 64 bits This bit does not generate interrupts.
22-19	Reserved	Reserved
18	Any Queue's Head Valid	
17	Any Queue's Threshold Exceeded	
16	Any Queue Full/Empty	
15-0	15-0: Status-Multiplexer Output for Queues 15-0	Reports either "Head Valid," "Threshold Exceeded," or "Queue Full/Empty" according to the setting of the Status-Multiplexer-Control field in the RXQUE Properties Register.

13.19: RXQUE Control Register

Used to set RXQUE modes. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. This register contains the mode bits that specify how RXQUE is to operate.

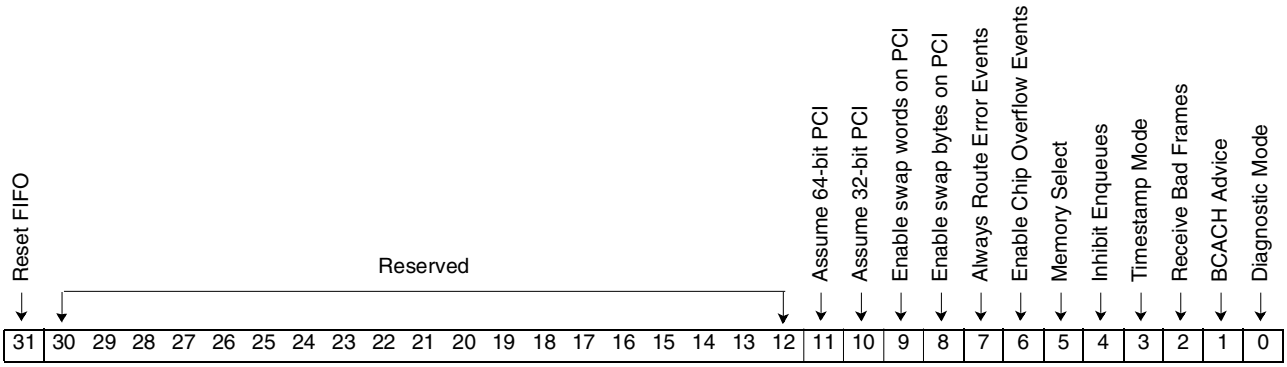
Length 32 bits

Type Clear/Set

Address XXXX 1C00 and C04

Power On Value X'00000300'

Restrictions None



Bit(s)	Name	Description
31	Reset FIFO	When this bit is set, the internal FIFO is flushed, and this bit is reset. The result is this bit will always be read as a '0'. This bit can only be set in diagnostic mode.
30-12	Reserved	Reserved.
11	Assume 64-bit PCI	
10	Assume 32-bit PCI	
9	Enable swap words on PCI	This bit is automatically set at power-on.
8	Enable swap bytes on PCI	This bit is automatically set at power-on.
7	Always route error events	When this bit is set, all error events are routed to the error queue even if rx bad frames (bit2) is turned on. When cleared, error events are only routed to the error queue if rx bad frames is turned off. This bit allows the user to keep bad frames in time sequence with good frames or to route them to the error queue. The clear state of this bit is code compatible with previous versions of the processor.
6	Enable chip overflow events	When set, the chip level counter overflow events are surfaced.
5	Memory select	When this bit is set, RXQUE will use Packet Memory instead of Control Memory to store the event queues.
4	Inhibit enqueues	When this bit is set, the enq state machine will not accept any new enq requests. This should be used in extreme cases as it holds off all enqueues indefinitely.
3	Timestamp mode	When this bit is set, timestamp events are inserted before each real event. The timestamps correspond to when the event happened on chip. When this bit is off, timestamps can still be read from the timestamp register. The timestamps would correspond to when the event was dequeued in this scenario.

Bit(s)	Name	Description
2	Receive bad frames	When this bit is set, bad frame events (all error events), will be received in the normal rxq defined in the LCD. All buffers are not freed, and the packet address is raised in the event data. When this bit is reset, bad frame events are routed to the rxq specified by the Error Event Receive Queue Register. All packet based events will carry the LC address in the event data instead of the packet address. All buffers are freed back to POOLS. **Note:** This bit should only be changed sparingly because it changes the way packets are freed and what is surfaced in an event (LCD vs. frame ptr). It should really only be changed when the receive side is inactive.
1	BCACH advice	This bit, when set, allows RXQUE to give BCACH cache fill advice based on events that are dequeued.
0	Diagnostic mode	When this bit is set or when the chip is disabled, the RXQUE entity is in diagnostic mode and primitive execution is disabled.

Debugging Register Access

This section is a very brief documentation of access that has been put in for the internal registers of RXQUE. These addresses need not be written or read during normal operations.

13.20: RXQUE RXQ State Machine Variable Register

Main state variable for RXQUE processing state machine.

Length 4 bits
Type Read/Write
Address XXXX 1E80
Power On Value X'00000000'
Restrictions None

13.21: RXQUE RXQ ENQ State Machine Variable Register

Main state variable for RXQUE processing state machine.

Length 3 bits
Type Read/Write
Address XXXX 1E84
Power On Value X'00000000'
Restrictions None

13.22: RXQUE Enq FIFO Head Ptr Register

Used to maintain the enqueue FIFO. Points to the head FIFO entry in the FIFO array. The MSB bit is used to determine if the head is chasing the tail, and is inverted each time the head pointer wraps.

Length	5 bits
Type	Read/Write
Address	XXXX 1E88
Power On Value	X'00000000'
Restrictions	Can only be written in diagnostic mode.

13.23: RXQUE Enq FIFO Tail Ptr Register

Used to maintain the enqueue FIFO. Points to the next free FIFO entry in the FIFO array. The MSB bit is used to determine if the head is chasing the tail, and is inverted each time the tail pointer wraps.

Length	5 bits
Type	Read/Write
Address	XXXX 1E8C
Power On Value	X'00000000'
Restrictions	Can only be written in diagnostic mode.

13.24: RXQUE Enq FIFO Array

Holds events waiting to be placed on an rxq. Array is organized as a 16x36 array. To access the upper four bits of each word (holds the receive queue number for event), the array word number should be used as the address. To access the low order 32 bits (the event portion), the array word number times two plus four should be used. For example, address zero accesses the receive queue portion of array word zero, and address four accesses the event portion of the array word zero.

Note: The most significant bit is not used. Only the three bits are needed for the receive queue number.

Length	16 words X 36 bits
Type	Read/Write
Address	XXXX 1F00-FF8
Power On Value	X'00000000'
Restrictions	Can only be read/written in diagnostic mode. When read in non-diagnostic mode, zero is returned.

PHY Level Interfaces

Entity 14: The PHY Interface (LINKC)

Functional Description

LINKC provides the interface between the IBM3206K0424 and either an ATM PHY device or, when the internal framer is selected, a serializer/deserializer device. LINKC is composed of three pieces. LINKX, which contains all the registers described below, is clocked with the same clock as other parts of the chip. LINKT, the transmit logic, is clocked on the transmit clock, which is selected via the Clock Control Register (described in *Clock Control Register (Nibble Aligned)* on page 516). LINKR, the receive logic, is clocked on the receive clock, which is also selectable via the Clock Control Register. Transmit and receive transfers are synchronized via their respective interface transfer clock. The data path size is 8- or 16-bits wide and is selectable through bit 3 of the control register. The PHY devices that the IBM3206K0424 interfaces to are:

- PMC SIERRA PM5346 SUNI LITE FOR SONET STS-3c 155.52 MB/s
- UTOPIA 8 or 16 bit interface
- PMC SIERRA POS-PHY

New features added to LINKC are:

- Multi-drop Utopia support
- PMC SIERRA POS-PHY interface support

Multi-Drop

When the IBM3206K0424 is in multi-drop Utopia mode, it supports four external PHY devices. Each port is associated with a configuration. Four configurations are provided so up to four different types of PHYs can be connected to the IBM3206K0424. This allows the user to mix cell and POS-PHY devices on the transmit and/or receive interface.

The multi drop PHY devices supported are Utopia Level 2 (cell based) and PMC Sierra POS-PHY (packet/frame based). The IBM3206K0424 will select which PHY device will transfer data next by polling each of the devices to determine which PHYs can transfer data. A round-robin switching scheme is used to determine which PHY has the priority if more then one wants to transmit/receive data. The IBM3206K0424 will switch to a new drop when a cell has been received/transmitted (for a cell-based PHY) or when 64 bytes or EOP has been received/transmitted (for POS-PHY PHYs). The transmit and receive sides of LINK are separately configurable for multi-drop mode (bits 1 and 0 of the global control register).

POS-PHY

The POS-PHY interface complies with the PMC Sierra POS-PHY Level 2 Specification. The IBM3206K0424 polls each POS-PHY device to determine its status on both the receive and transmit side. It looks to switch to a different port when 64 bytes or EOP (End of Packet) have been transferred between the POS-PHY and itself. The IBM3206K0424 does not support direct status indication or byte-level transfers. Therefore, the PHY must be programmed to always be able to send/receive at least 64 bytes of information. The RMOD signal will only be looked at when REOP is b'1'; at all other times it will be ignored. POS-PHY devices should be configured so that they will only signal they are ready for a transfer if they have 64 bytes free in their receive buffer and 64 bytes or EOP in their transmit FIFO.

Moving Cells To and From the IBM3206K0424

Interface	Data Payload	Cell	Cycles for Eight-bit bus	Cycles for 16-bit bus
Utopia Cell	48	52	52	26
Utopia Cell	48	53	53	27
Utopia Cell	48	54	54	27
Utopia Cell	48	55	55	28
Utopia Cell	48	56	56	28

14.1: LINKC Global Control Register

This register contains the information which controls the operation of LINKC. These controls affect all configurations. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

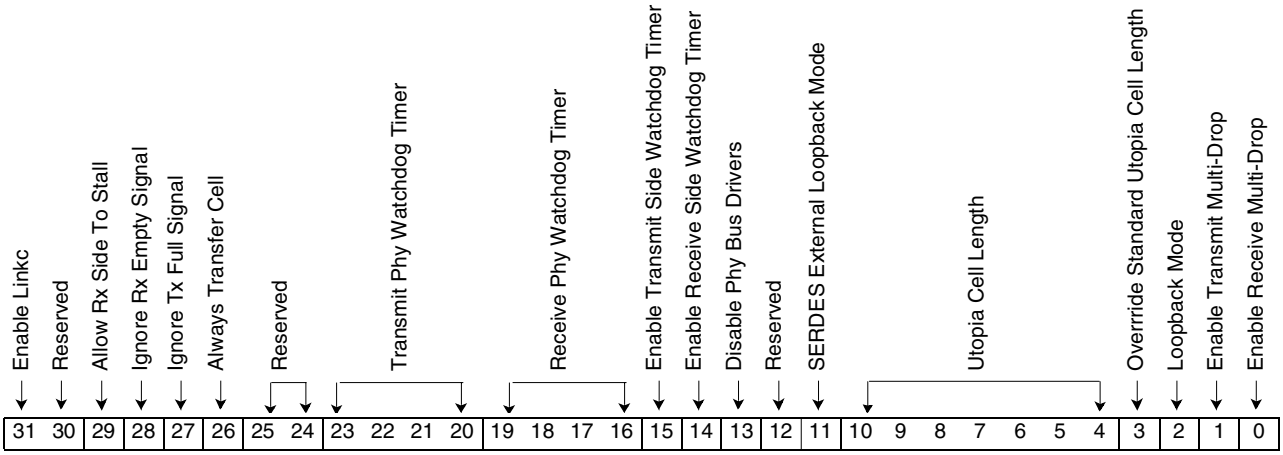
Length 32 bits

Type Clear/Set

Address XXXX 0B30 AND 34

Power On Value X'C0000344'

Restrictions None



Bit(s)	Name	Description
31	Enable LINKC	This bit, when set to '1' will enable LINKC. The default for this bit is '1'.
30	Reserved	Reserved.
29	Allow RX side to stall	When this bit is set, the RX side is allowed to stall. This means that the IBM3206K0424 is allowed to park on a port until the port has data for the IBM3206K0424. This bit should be set if you are talking to a single drop POS-PHY device that doesn't have address pins and the IBM3206K0424 is being run in multi-drop mode.
28	Ignore RX Empty signal	When this bit is set the EMPty signal will be ignored and it is always assumed the PHY has data. The default for this bit is '0'.
27	Ignore TX Full signal	When this bit is set the Full signal will be ignored and it is always assumed the PHY has room. The default for this bit is '0'.

Bit(s)	Name	Description
26	Always Transfer cell	When this bit is set to '1', the TCA (transmit cell available) will be ignored until the current transfer ends. This mode is recommended when talking to a single drop Utopia device. The default for this bit is '0'.
25-21	Reserved	Reserved
23-20	Transmit PHY WatchDog Timer	These four bits are the cycles the transmit side (LINKT) will wait before switching to another PHY. If the timer times out, a status bit in LINKC Interrupt/Status Register will be set for the offending PHY.
19-16	Receive PHY WatchDog Timer	These four bits are the cycles the receive side (LINKR) will wait before switching to another PHY. If the timer times out, a status bit in LINKC Interrupt/Status Register will be set for the offending PHY.
15	Enable Transmit Side WatchDog Timer	This bit, when set to '1', causes LINKT to timeout if the PHY has started to take data but is now unable to take data for the number of cycles determined by the Transmit PHY Watchdog Timer (bits 23-20). The Timer is only valid if the transmit side is in multi-drop mode.
14	Enable Receive Side WatchDog Timer	This bit, when set to '1', causes LINKR to timeout if the PHY is receiving data and is unable to provide data for the number of cycles determined by the Receive PHY watchdog timer (bits 19-16). The timer is only valid if the receive side of LINKC is in multi-drop mode and the PHY is a Utopia device.
13	Disable PHY Bus Drivers	This bit, when set to '1', tri-states the drivers of the PHY bus. When set to '0', the drivers are enabled.
12	Reserved	Reserved
11	SERDES External Loopback Mode	When this bit is set, the Receive Side SERDES input will be routed to the Transmit Side SERDES output.
10-4	Utopia Cell Length	These seven bits will define what the Utopia cell length (in bytes) will be if the override standard utopia cell length bit (bit 3) is set to '1'. The upper limit of this register is 64 and the lower limit is one. The default value of these bits is x'0110100'.
3	Override Standard Utopia Cell Length	When this bit is set to '1', the standard utopia cell length (52 or 53 bytes) will be replaced by the value in bits 10-4. This bit will have no affect on PHYs that aren't Utopia and it will disable the HEC generation for any Utopia PHY.
2	Loop back mode	This bit set to '1' places the IBM3206K0424 in an internal loop back mode. The PHY interface will be disabled. The clocks to LINKT and LINKR should be set to the same source in the Clock Control Register. This bit is flushed to a '1' after POR. For loopback to work in multi-drop mode the transmit and receive configurations must be the same. When a configuration is set up for Utopia Cell-based transmission the receive and transmit sides should be identical in all ways. These include odd/even parity, data path length, 52-byte cell mode, null cell generation, and HEC generation of null cells. The additional header bytes should be set to '00' when in loopback mode. Please see the table*Legal Loopback Configurations on page 406*.
1	Enable Transmit Multi-Drop	Setting this bit will put the transmit side into multi-drop mode. In multi-drop mode the IBM3206K0424 will support four configurations and four unique ports. This bit should not be set if the transmit side is connected to the Internal SONET Framer. Configuration 0 Transmit Control Register will control the transmit interface. If this bit is not set, the IBM3206K0424 is in single drop mode.
0	Enable Receive Multi-Drop	Setting this bit will put the receive side into multi-drop mode. In multi-drop mode, the IBM3206K0424 will support four configurations and four unique PHY ports. This bit should not be set if the receive side is connected to the Internal SONET Framer. Configuration 0 Receive Control Register will control the receive interface. If this bit is not set, the IBM3206K0424 receive side is in single drop mode.

Legal Loopback Configurations

TX Config 0	RX Config 0	TX Config 1	RX Config 1	TX Config 2	RX Config 2	TX Config 3	RX Config 3
SONET	SONET	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used
Utopia Cell	Utopia Cell	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used
Not Used	Not Used	Utopia Cell	Utopia Cell	Not Used	Not Used	Not Used	Not Used
Not Used	Not Used	Not Used	Not Used	Utopia Cell	Utopia Cell	Not Used	Not Used
Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Utopia Cell	Utopia Cell
Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell
POS-PHY	POS-PHY	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used
Not Used	Not Used	POS-PHY	POS-PHY	Not Used	Not Used	Not Used	Not Used
Not Used	Not Used	Not Used	Not Used	POS-PHY	POS-PHY	Not Used	Not Used
Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	POS-PHY	POS-PHY
POS-PHY	POS-PHY	POS-PHY	POS-PHY	POS-PHY	POS-PHY	POS-PHY	POS-PHY
Utopia Cell	Utopia Cell	POS-PHY	POS-PHY	POS-PHY	POS-PHY	POS-PHY	POS-PHY
POS-PHY	POS-PHY	Utopia Cell	Utopia Cell	POS-PHY	POS-PHY	POS-PHY	POS-PHY
POS-PHY	POS-PHY	POS-PHY	POS-PHY	Utopia Cell	Utopia Cell	POS-PHY	POS-PHY
POS-PHY	POS-PHY	POS-PHY	POS-PHY	POS-PHY	POS-PHY	Utopia Cell	Utopia Cell
Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	POS-PHY	POS-PHY	POS-PHY	POS-PHY
Utopia Cell	Utopia Cell	POS-PHY	POS-PHY	Utopia Cell	Utopia Cell	POS-PHY	POS-PHY
Utopia Cell	Utopia Cell	POS-PHY	POS-PHY	POS-PHY	POS-PHY	Utopia Cell	Utopia Cell
POS-PHY	POS-PHY	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	POS-PHY	POS-PHY
POS-PHY	POS-PHY	Utopia Cell	Utopia Cell	POS-PHY	POS-PHY	Utopia Cell	Utopia Cell
POS-PHY	POS-PHY	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell
Utopia Cell	Utopia Cell	POS-PHY	POS-PHY	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell
Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	POS-PHY	POS-PHY	Utopia Cell	Utopia Cell
Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	Utopia Cell	POS-PHY	POS-PHY

14.2: LINKC Configuration 0 Transmit & Receive Control Register

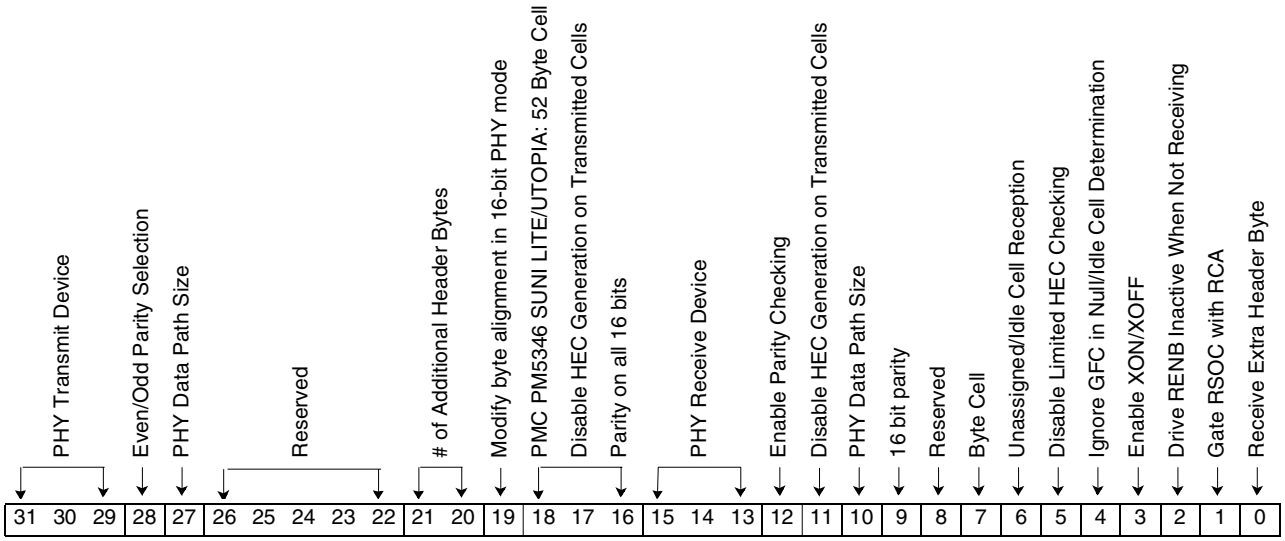
This register contains the information which controls the operation of Configuration 0 on the transmit and receive. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length 32 bits

Type Clear/Set

Address XXXX 0B50 and 0B54

Power On Value X'78016E00'



Bit(s)	Name	Description
31-29	PHY Transmit Device	Bits 31, 30, and 29 indicates to which PHY the IBM3206K0424's Transmit Config 0 will be interfacing. If the configuration's port address is all ones, then the configuration is unused and the value above bits doesn't matter. '000' Reserved '001' PMC POS-PHY (Frame-based Utopia) '010' Reserved '011' PMC PM5346 SUNI LITE/UTOPIA interface (STS-3c/STM-1 OR STS-1) '100' Reserved '101' Internal SONET(sts-3c)/SDH(STM-1) Framer with SERDES (Serial interface) '111' Reserved
28	Even/Odd Parity Selection	Even parity is selected when this bit is cleared. The default value is for odd parity. Parity will always be generated when the IBM3206K0424 is transmitting data. If the PHY doesn't check parity, then don't connect the lines.
27	PHY Data Path Size	This bit, when set to '0', selects a 16-bit wide data path to the PHY device. When set to '1', the data path width to the PHY will be eight bits. This bit has no affect on the internal SONET/SDH framer except if the internal framer has been selected as the Rx PHY device but not as the Tx PHY device. In this case, a '1' on this bit will allow FYTDAT(1-13) to be used for the 16-bit external Tx PHY device, while a '0' will allow FYTDAT(15-13) to be used for the Rx HDLC controller. This implies that it is not possible to use the internal RX framer, the RX HDLC interface, and an external 16-bit TX framer at the same time.
26-19	Reserved	Reserved.

Bit(s)	Name	Description
21-20	Number of Additional Header Bytes	These bits indicate the number of additional header bytes that will be read from SEG-BUF and added to the beginning of each cell as each is transmitted to the PHY. The bytes are meant to be used for additional routing information. These control bits have no affect in IBM 25 Mb/s PHY mode, and should be set to ’0’s when in internal SONET/SDH framer mode. If used in conjunction with 52-byte mode, the byte normally containing the cell HEC will not be transmitted and the total number of cells transmitted will be the value of this field plus 52. If 16-bit PHY mode is selected, by default, the byte alignment will follow that of normal 52- or 53-byte 16-bit mode, with the additional header bytes contiguously prepended. As a result, a mode with three additional header bytes cannot be obtained in 53-byte, 16-bit mode (LSB is normally padded with zeros so MSB gets truncated). Bit 3 of this register is therefore provided to adjust the align-ment in 16-bit, 53-byte mode so all five header bytes will be transmitted with up to three additional router bytes prepended.
19	Modify byte alignment in 16-bit PHY mode	When set to ’1’, this bit changes the default byte alignment in 16-bit PHY mode if this register also contains a non-zero value in bits 21-20. See the description of those bits for further details.
18-16	Bits 18-16 only have meaning if a PMC PM5346 SUNI LITE/UTOPIA (bits 31-29 = "011") is selected. If any other device is chosen, these bits will be ignored.	
18	52 Byte Cell	When set, the cell sent to the PHY will be 52 bytes. No HEC byte will be sent.
17	Disable HEC Generation on Transmitted Cells	If bit 17 is set to ’1’, X'00' will be placed in the HEC byte of Utopia cells. If bit 17 is set to ’0’, then the value of LINKC Transmitted HEC Control byte will be sent. If bit 18 is set to ’1’, then no HEC will be sent and this bit will be ignored.
16	Parity on all 16 bits	When set, this bit enables the IBM3206K0424 to produce a single parity bit across the 16-bit transmit data bus. When set to ’0’, the IBM3206K0424 will produce two parity bits, one across generation and the lower half of the 16 bits (parity bit 0) and one against the upper (parity bit 1). The default for this bit is ’1’.
15-13	PHY Receive Device	These bits indicate to which PHY the IBM3206K0424’s Receive Config 0 will be inter-facing. If the configuration's port address is all ‘1’s, then the configuration is unused and the value of the above bits doesn't matter. '000' Reserved '001' PMC POS-PHY (Frame-based Utopia) '010’ Reserved '011' PMC PM5346 SUNI LITE/UTOPIA interface (STS-3c/STM-1 OR STS-1) '100' Reserved '101' Internal SONET (sts-3c)/SDH(STM-1) Framer with SERDES (Serial Interface) '111' Reserved
12	Enable Parity Checking	When set, this bit will enable checking of parity on data from the receive path. The default is parity checking is disabled. The upper bit of the transmit parity is not valid when the internal SONET/SDH Framer has been selected as the receive PHY device. The upper bit of the receive parity is also not valid when the internal SONET/SDH Framer has been selected as the transmit PHY device. This is only a concern if a com-bination of the internal framer and an external PHY is being used and that external PHY has a 16-bit data interface. In this case, parity cannot be checked/generated on the upper byte.
11	Even/Odd Parity Selection	Even parity is selected when this bit is cleared. The default value is for odd parity.
10	PHY Data Path Size	This bit, when set to ’0’, selects a 16-bit wide data path to the PHY device. When set to ’1’, the data path width to the PHY will be eight bits. This bit has no affect on the inter-nal SONET/SDH framer except if the internal framer has been selected as the Rx PHY device but not as the Tx PHY device. In this case, a ’1’ on this bit will allow FYT-DAT(15-13) to be used for the 16-bit external Tx PHY device, while a zero will allow FYTDAT(15-13) to be used for the Rx HDLC controller. This implies that it is not possi-ble to use the internal RX framer, the RX HDLC interface, and an external 16-bit TX framer at the same time.
9	16 bit parity	When this bit is set to ’1’ in 16-bit mode, parity will be calculated across all 16 bits and checked against FYRPAR(1). When in 8-bit mode with bit 9 set to ’0’, the parity will be compared against FYRPAR(1). This bit has no affect if receive device is POS-PHY. The default setting of this bit is ’1’.

Bit(s)	Name	Description
8	Reserved	Reserved
7-0	Bits 7-0 only have meaning if a PMC PM5346 SUNI LITE/UTOPIA (bits 15-13 = "011") is selected. Otherwise, they are ignored. Bits 2-0 are used to make adjustments to the Utopia interface for compatibility with the Suni-PHD PHY.	
7	Byte Cell	When set, the cell received from the PHY will be 52 bytes. No HEC byte will be received.
6-5	Reserved	Reserved
6	Unassigned/Idle Cell Reception	When set to ’1’, this bit will enable unassigned/idle cell reception. This should be set to ’0’ when using the internal SONET Framer.
5	Disable Limited HEC Checking on Received Idle/Unassigned Cells	If bit 5 is set to ’1’, the receive logic will ignore the HEC byte of the header of idle and unassigned cells. Idle is defined as a header of X'00000001' and unassigned is defined as a header of X'0000000n' where n is 'xxx0'. If bit 6 is set to enable unassigned/idle cell reception, all cells will be passed to REASM regardless of how this bit is set. If bit 6 is set to disable unassigned/idle cell reception and this bit is set to ’0’, the HEC byte of cells with an apparent idle header will be completely checked before deciding whether or not to pass the cell to REASM. If a cell appears to have an unassigned header, HEC bits 7, 6, and 0 will be checked because they are a constant, regardless of the value of bits 3, 2, and 1 of the header. If other HEC bits are bad, REASM will detect the HEC error and discard the cell. If there is a correctable HEC error and the cell is indeed unassigned, an out of range error will occur in REASM.
4	Ignore GFC in Null/Idle Cell Determination	Bit 4, when set, will cause the receive logic to ignore the first four bits of the ATM header in determining whether a cell being received is a null or idle cell.
3	Enable XON/XOFF	Bit 3, when set, will allow the XON/XOFF bit of the header of a received cell to suspend/continue transmission from the IBM3206K0424's transmit logic for all ports associated with Config 0.
2	Drive RENB Inactive When Not Receiving	When set to ’1’, this bit forces the receive logic to deactivate RENB when in the idle state.
1	Gate RSOC with RCA	When set to ’1’, this bit forces the receive logic to see both RSOC and RCA before considering RSOC valid.
0	Receive Extra Header Byte	When set to ’1’, this bit allows an extra header byte to be accepted at the start of a cell by the receive logic. The extra byte is discarded.

14.3: LINKC Configuration 1 Transmit & Receive Control Register

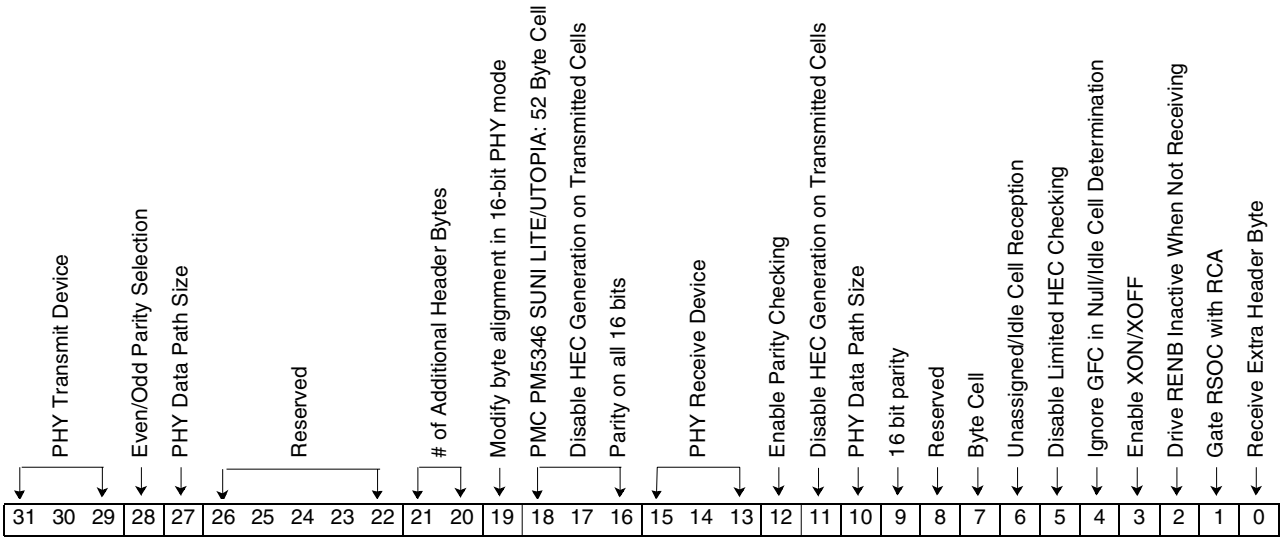
This register contains the information which controls the operation of Configuration 1 on the transmit and receive. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length 32 bits

Type Clear/Set

Address XXXX 0B58 AND 0B5C

Power On Value X'78016E00'



Bit(s)	Name	Description
31-29	PHY Transmit Device	Bits 31, 30, and 29 indicates to which PHY the IBM3206K0424's Transmit Config 1 will be interfacing. If the configuration's port address is all ones, then the configuration is unused and the value of the above bits doesn't matter. '000' Reserved '001' PMC POS-PHY (Frame based Utopia) '010' Reserved '011' PMC PM5346 SUNI LITE/UTOPIA interface (STS-3c/STM-1 OR STS-1) '100' Reserved '101' Reserved '111' Reserved
28	Even/Odd Parity Selection	Even parity is selected when this bit is cleared. The default value is for odd parity. Parity will always be generated when the IBM3206K0424 is transmitting data. If the PHY doesn't check parity, then don't connect the lines.
27	PHY Data Path Size	This bit, when set to '0', selects a 16-bit wide data path to the PHY device. When set to '1', the data path width to the PHY will be eight bits. This bit has no affect on the internal SONET/SDH framer except if the internal framer has been selected as the Rx PHY device but not as the Tx PHY device. In this case, a '1' on this bit will allow FYTDAT(15-13) to be used for the 16-bit external Tx PHY device, while a zero will allow FYTDAT(15-13) to be used for the Rx HDLC controller. This implies that it is not possible to use the internal RX framer, the RX HDLC interface, and an external 16-bit TX framer at the same time.
26-19	Reserved	Reserved

Bit(s)	Name	Description
18-16	These bits only have meaning if a PMC PM5346 SUNI LITE/UTOPIA (bits 31-29 = "011") is selected. If any other device is chosen these bits will be ignored.	
21-20		The value of bits 21-20 indicates the number of additional header bytes that will be read from SEGBUF and added to the beginning of each cell as each is transmitted to the PHY. The bytes are meant to be used for additional routing information. These control bits have no affect in IBM 25 Mb/s PHY mode, and should be set to '0's when in internal SONET/SDH framer mode. If used in conjunction with 52-byte mode, the byte normally containing the cell HEC will not be transmitted and the total number of cells transmitted will be the value of this field plus 52. If 16-bit PHY mode is selected, by default, the byte alignment will follow that of normal 52- or 53-byte 16-bit mode, with the additional header bytes contiguously prepended. As a result, a mode with three additional header bytes cannot be obtained in 53-byte, 16-bit mode (LSB is normally padded with zeros so MSB gets truncated). Bit 3 of this register is therefore provided to adjust the alignment in 16-bit, 53-byte mode so all five header bytes will be transmitted with up to three additional router bytes prepended.
19	Modify byte alignment in 16-bit PHY mode	When set to '1', this bit changes the default byte alignment in 16-bit PHY mode if this register also contains a non-zero value in bits 21-20. See the description of those bits for further details.
18	52 Byte Cell	When set, the cell sent to the PHY will be 52 bytes. No HEC byte will be sent.
17	Disable HEC Generation on Transmitted Cells	If bit 17 is set to '1', X'00' will be placed in the HEC byte of Utopia cells. If bit 17 is set to '0', the value of LINKC Transmitted HEC Control byte will be sent. If bit 18 is set to '1', then no HEC will be sent and this bit will be ignored.
16	Parity on all 16 bits	When set, this bit enables the IBM3206K0424 to produce a single parity bit across the 16-bit transmit data bus. When set to '0', the IBM3206K0424 produces two parity bits, one across generation and the lower half of the 16 bits (parity bit zero) and one against the upper (parity bit 1). The default for this bit is '1'.
15-13	PHY Receive Device	Bits 15, 14, and 13 indicate which PHY the IBM3206K0424's Receive Config 1 will be interfacing. If the configuration's port address is all ones, then the configuration is unused and the value above bits doesn't matter. '000' Reserved '001' PMC POS-PHY (Frame based Utopia) '010' Reserved '011' PMC PM5346 SUNI LITE/UTOPIA interface (STS-3c/STM-1 OR STS-1) '100' Reserved '101' Reserved '111' Reserved
12	Enable Parity Checking	When set, this bit enables checking of parity on data from the receive path. The default parity checking is disabled. The upper bit of the transmit parity is not valid when the internal SONET/SDH Framer has been selected as the receive PHY device. The upper bit of the receive parity is also not valid when the internal SONET/SDH Framer has been selected as the transmit PHY device. This is only a concern if a combination of the internal framer and an external PHY is being used and that external PHY has a 16-bit data interface. In this case, parity cannot be checked/generated on the upper byte.
11	Even/Odd Parity Selection	Even parity is selected when this bit is cleared. The default value is for odd parity.
10	PHY Data Path Size	This bit, when set to '0', selects a 16-bit wide data path to the PHY device. When set to '1', the data path width to the PHY is eight bits. This bit has no affect on the internal SONET/SDH framer except if the internal framer has been selected as the Rx PHY device but not as the Tx PHY device. In this case, a '1' on this bit will allow FYTDAT(15-13) to be used for the 16-bit external Tx PHY device, while a zero will allow FYTDAT(15-13) to be used for the Rx HDLC controller. This implies that it is not possible to use the internal RX framer, the RX HDLC interface, and an external 16-bit TX framer at the same time.
9	16 bit parity	When this bit is set to '1' and it is in 16-bit mode, that parity will be calculated across all 16 bits and checked against FYRPAR(1). When in eight-bit mode with bit 9 set to '0', the parity will be compared against FYRPAR(1). This bit has no affect if the receive device is POS-PHY. The default setting of this bit is '1'.

Bit(s)	Name	Description
8	Reserved	Reserved
7-0	Bits 7-0 only have meaning if a PMC PM5346 SUNI LITE/UTOPIA (bits 15-13 = "011") is selected. Otherwise, these bits will be ignored. Bits 2-0 are used to make adjustments to the Utopia interface for compatibility with the Suni-PHD PHY.	
7	52 Byte Cell	When set, the cell received from the PHY is 52 bytes. No HEC byte will be received.
6-5	Reserved	Reserved
6	Unassigned/Idle Cell Reception	When set to ’1’, this bit will enable unassigned/idle cell reception. This should be set to ’0’ when using the internal SONET Framer.
5	Disable Limited HEC Checking on Received Idle/Unassigned Cells	If bit 5 is set to ’1’, the receive logic will ignore the HEC byte of the header of idle and unassigned cells. Idle is defined as a header of X'00000001' and unassigned is defined as a header of X'0000000n' where n is 'xxx0'. If bit 6 is set to enable unassigned/idle cell reception, all cells are passed to REASM regardless of how this bit is set. If bit 6 is set to disable unassigned/idle cell reception and this bit is set to ’0’, the HEC byte of cells with an apparent idle header will be completely checked before deciding whether or not to pass the cell to REASM. If a cell appears to have an unassigned header, HEC bits seven, six, and zero will be checked because they are a constant regardless of the value of bits 3-1 of the header. If other HEC bits are bad, REASM will detect the HEC error and discard the cell. If there is a correctable HEC error and the cell is indeed unassigned, an out of range error will occur in REASM.
4	Ignore GFC in Null/Idle Cell Determination	Bit 4, when set, will cause the receive logic to ignore the first four bits of the ATM header in determining whether a cell being received is a null or idle cell.
3	Enable XON/XOFF	Bit 3, when set, will allow the XON/XOFF bit of the header of a received cell to suspend/continue transmission from the IBM3206K0424's transmit logic for all ports associated with Config 1.
2	Drive RENB Inactive When Not Receiving	When set to ’1’, this bit forces the receive logic to deactivate RENB when in the idle state.
1	Gate RSOC with RCA	When set to ’1’, this bit forces the receive logic to see both RSOC and RCA before considering RSOC valid.
0	Receive Extra Header Byte	When set to ’1’, this bit allows an extra header byte to be accepted at the start of a cell by the receive logic. The extra byte is discarded.

14.4: LINKC Configuration 2 Transmit & Receive Control Register

This register contains the information which controls the operation of configuration 2 on the transmit and receive. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

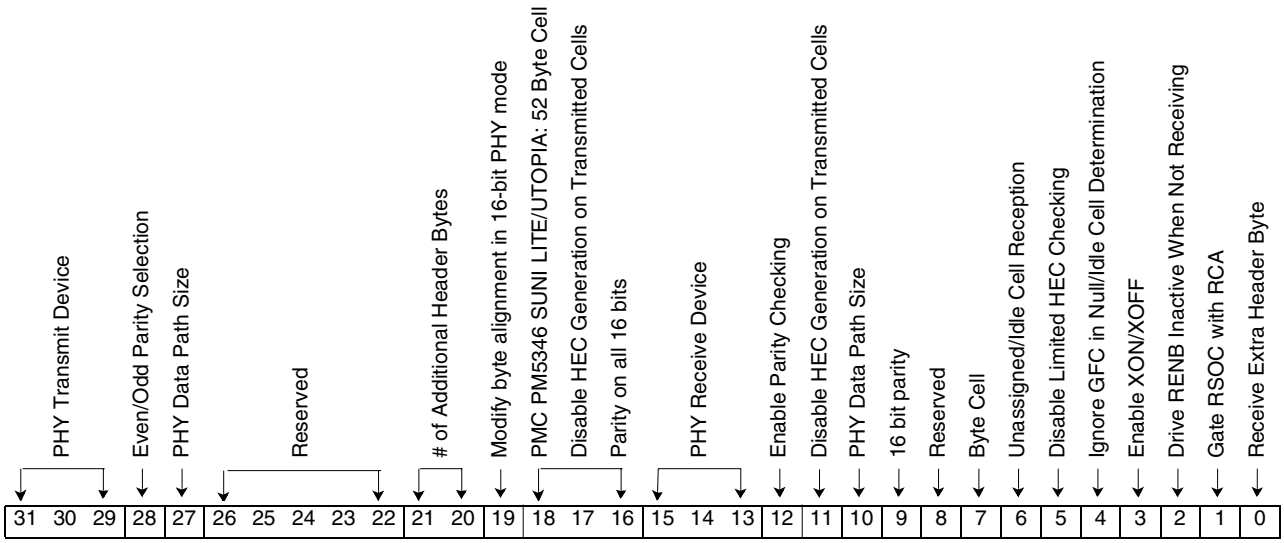
Length 32 bits

Type Clear/Set

Address XXXX 0B60 AND 0B64

Power On Value X'78016E00'

Restrictions



Bit(s)	Name	Description
31-29	PHY Transmit Device	Bits 31, 30, and 29 indicate which PHY the IBM3206K0424's Transmit Config 2 will be interfacing. If the configuration's port address is all ones then the configuration is unused and the value above bits doesn't matter. '000' Reserved '001' PMC POS-PHY (Frame based Utopia) '010' Reserved '011' PMC PM5346 SUNI LITE/UTOPIA interface (STS-3c/STM-1 OR STS-1) '100' Reserved '101' Reserved '111' Reserved
28	Even/Odd Parity Selection	Even parity is selected when this bit is cleared. The default value is for odd parity. Parity will always be generated when the IBM3206K0424 is transmitting data. If the PHY doesn't check parity then don't connect the lines.
27	PHY Data Path Size	This bit, when set to '0', selects a 16-bit wide data path to the PHY device. When set to '1', the data path width to the PHY will be eight bits. This bit has no affect on the internal SONET/SDH framer except if the internal framer has been selected as the Rx PHY device but not as the Tx PHY device. In this case, a '1' on this bit allows FYT-DAT(15-13) to be used for the 16-bit external Tx PHY device, while a zero allows FYT-DAT(15-13) to be used for the Rx HDLC controller. This implies that it is not possible to use the internal RX framer, the RX HDLC interface, and an external 16-bit TX framer at the same time.

Bit(s)	Name	Description
26-19	Reserved	Reserved
18-16	These bits only have meaning if a PMC PM5346 SUNI LITE/UTOPIA (bits 31-29 = "011") is selected. If any other device is chosen these bits will be ignored.	
21-20		The value of bits 21-20 indicate the number of additional header bytes that will be read from SEGBUF and added to the beginning of each cell as each is transmitted to the PHY. The bytes are meant to be used for additional routing information. These control bits have no affect in IBM 25 Mb/s PHY mode, and should be set to ’0’s when in internal SONET/SDH framer mode. If used in conjunction with 52-byte mode, the byte normally containing the cell HEC will not be transmitted and the total number of cells transmitted will be the value of this field plus 52. If 16-bit PHY mode is selected, by default, the byte alignment will follow that of normal 52- or 53-byte 16-bit mode, with the additional header bytes contiguously prepended. As a result, a mode with three additional header bytes cannot be obtained in 53-byte, 16-bit mode (LSB is normally padded with zeros so MSB gets truncated). Bit 3 of this register is therefore provided to adjust the alignment in 16-bit, 53-byte mode so all five header bytes will be transmitted with up to three additional router bytes prepended.
19	Modify byte alignment in 16-bit PHY mode	When set to ’1’, this bit changes the default byte alignment in 16-bit PHY mode if this register also contains a non-zero value in bits 21-20. See the description of those bits for further details.
18	52 Byte Cell	When set, the cell sent to the PHY is 52 bytes. No HEC byte will be sent.
17	Disable HEC Generation on Transmitted Cells	If bit 17 is set to ’1’, X'00' will be placed in the HEC byte of Utopia cells. If bit17 is set to ’0’, the value of LINKC Transmitted HEC Control byte will be sent. If bit18 is set to ’1’, then no HEC is sent and this bit will be ignored.
16	Parity on all 16 bits	When set, this bit enables the IBM3206K0424 to produce a single parity bit across the 16-bit transmit data bus. When set to ‘0’, the IBM3206K0424 will produce two parity bits, one across generation and the lower half of the 16 bits (parity bit zero) and one against the upper (parity bit 1). The default for this bit is ’1’.
15-13	PHY Receive Device	Bits 15, 14, and 13 indicate which PHY the IBM3206K0424’s Receive Config 2 will be interfacing. If the configuration's port address is all ones, then the configuration is unused and the value above bits doesn't matter. '000' Reserved '001' PMC POS-PHY (Frame based Utopia) '010' Reserved '011' PMC PM5346 SUNI LITE/UTOPIA interface (STS-3c/STM-1 OR STS-1) '100' Reserved '101' Reserved '111' Reserved
12	Enable Parity Checking	When set, this bit will enable checking of parity on data from the receive path. The default parity checking is disabled. The upper bit of the transmit parity is not valid when the internal SONET/SDH Framer has been selected as the receive PHY device. The upper bit of the receive parity is also not valid when the internal SONET/SDH Framer has been selected as the transmit PHY device. This is only a concern if a combination of the internal framer and an external PHY is being used and that external PHY has a 16-bit data interface. In this case, parity cannot be checked/generated on the upper byte.
11	Even/Odd Parity Selection	Even parity is selected when this bit is cleared. The default value is for odd parity.
10	PHY Data Path Size	This bit, when set to ’0’, selects a 16-bit wide data path to the PHY device. When set to ’1’, the data path width to the PHY will be eight bits. This bit has no affect on the internal SONET/SDH framer except if the internal framer has been selected as the Rx PHY device but not as the Tx PHY device. In this case, a ’1’ on this bit will allow FYT-DAT(15-13) to be used for the 16-bit external Tx PHY device, while a zero allows FYT-DAT(15-13) to be used for the Rx HDLC controller. This implies that it is not possible to use the internal RX framer, the RX HDLC interface, and an external 16-bit TX framer at the same time.

Bit(s)	Name	Description
9	16 bit parity	When this bit is set to ’1’ and it is in 16-bit mode, that parity will be calculated across all 16 bits and check against FYRPAR(1). When in eight-bit mode with bit 9 set to ’0’, the parity will be compared against FYRPAR(1). This bit has no affect if receive device is POS-PHY. The default setting of this bit is ’1’.
8	Reserved	Reserved
7-0	These bits only have meaning if a PMC PM5346 SUNI LITE/UTOPIA (bits 15-13 = "011") is selected. If any other device is chosen, these bits are ignored. Bits two through zero are used to make adjustments to the Utopia interface for compatibility with the Suni-PHD PHY.	
7	52 Byte Cell	When set, the cell received from the PHY is 52 bytes. No HEC byte is received.
6-4	Reserved	Reserved
6	Unassigned/Idle Cell Reception	When set to ’1’, this bit will enable unassigned/idle cell reception. This should be set to ’0’ when using the internal SONET Framer.
5	Disable Limited HEC Checking on Received Idle/Unassigned Cells	If bit 5 is set to ’1’, the receive logic will ignore the HEC byte of the header of idle and unassigned cells. Idle is defined as a header of X'00000001' and unassigned is defined as a header of X'0000000n' where n is 'xxx0'. If bit 6 is set to enable unassigned/idle cell reception, all cells are passed to REASM regardless of how this bit is set. If bit 6 is set to disable unassigned/idle cell reception and this bit is set to ’0’, the HEC byte of cells with an apparent idle header will be completely checked before deciding whether or not to pass the cell to REASM. If a cell appears to have an unassigned header, HEC bits seven, six, and zero will be checked because they are a constant regardless of the value of bits 3-1 of the header. If other HEC bits are bad, REASM will detect the HEC error and discard the cell. If there is a correctable HEC error and the cell is indeed unassigned, an out of range error will occur in REASM.
4	Ignore GFC in Null/Idle Cell Determination	Bit 4, when set, causes the receive logic to ignore the first four bits of the ATM header in determining whether a cell being received is a null or idle cell.
3	Enable XON/XOFF	Bit 3, when set, allows the XON/XOFF bit of the header of a received cell to suspend/continue transmission from the IBM3206K0424's transmit logic for all ports associated with Config 2.
2	Drive RENB Inactive When Not Receiving	When set to ’1’, this bit forces the receive logic to deactivate RENB when in the idle state.
1	Gate RSOC with RCA	When set to ’1’, this bit forces the receive logic to see both RSOC and RCA before considering RSOC valid.
0	Receive Extra Header Byte	When set to ’1’, this bit allows an extra header byte to be accepted at the start of a cell by the receive logic. The extra byte is discarded.

14.5: LINKC Configuration 3 Transmit & Receive Control Register

This register contains the information which controls the operation of configuration 3 on the transmit and receive. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

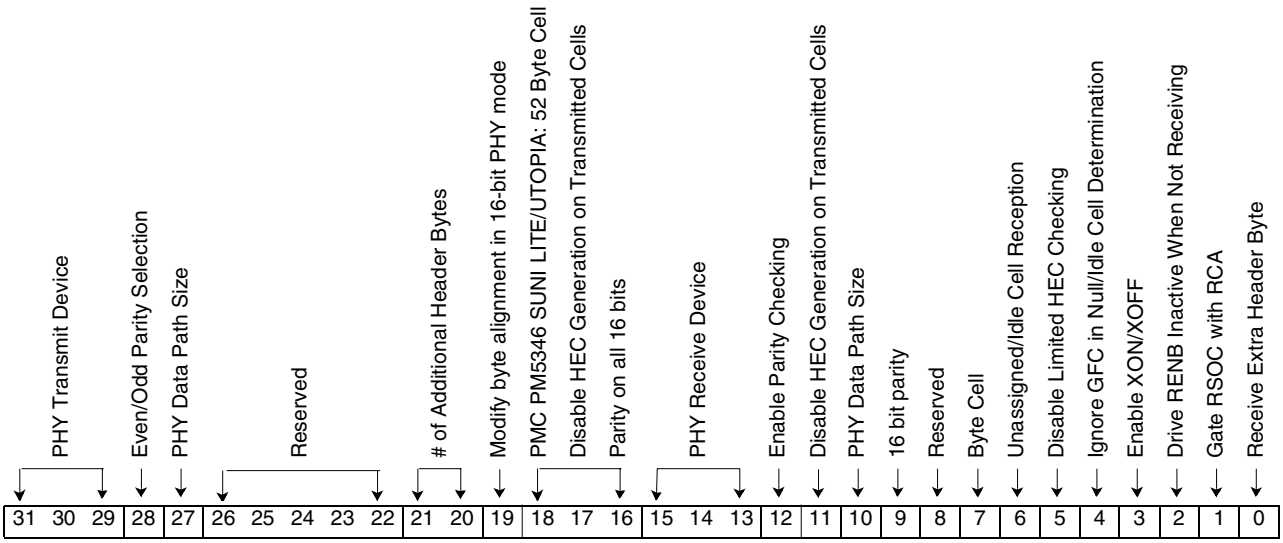
Length 32 bits

Type Clear/Set

Address XXXX 0B68 AND 0B6C

Power On Value X'78016E00'

Restrictions



Bit(s)	Name	Description
31-29	PHY Transmit Device	Bits 31, 30, and 29 indicate to which PHY the IBM3206K0424's Transmit Config 3 will be interfacing. If the configuration's port address is all ones then the configuration is unused and the value of the above bits doesn't matter. '000' Reserved '001' PMC POS-PHY (Frame based Utopia) '010' Reserved '011' PMC PM5346 SUNI LITE/UTOPIA interface (STS-3c/STM-1 OR STS-1) '100' Reserved '101' Reserved '111' Reserved
28	Even/Odd Parity Selection	Even parity is selected when this bit is cleared. The default value is for Odd parity. Parity will always be generated when the IBM3206K0424 is transmitting data. If the PHY doesn't check parity then don't connect the lines.
27	PHY Data Path Size	This bit, when set to '0', selects a 16-bit wide data path to the PHY device. When set to '1', the data path width to the PHY will be eight bits. This bit has no affect on the internal SONET/SDH framer except if the internal framer has been selected as the Rx PHY device but not as the Tx PHY device. In this case, a '1' on this bit will allow FYTDAT(15-13) to be used for the 16-bit external Tx PHY device, while a zero will allow FYTDAT(15-13) to be used for the Rx HDLC controller. This implies that it is not possible to use the internal RX framer, the RX HDLC interface, and an external 16-bit TX framer at the same time.

Bit(s)	Name	Description
26-19	Reserved	Reserved
21-20		The value of bits 21-20 indicate the number of additional header bytes that will be read from SEGBUF and added to the beginning of each cell as each is transmitted to the PHY. The bytes are meant to be used for additional routing information. These control bits have no affect in IBM 25 Mb/s PHY mode, and should be set to ’0’s when in internal SONET/SDH framer mode. If used in conjunction with 52-byte mode, the byte normally containing the cell HEC will not be transmitted and the total number of cells transmitted will be the value of this field plus 52. If 16-bit PHY mode is selected, by default, the byte alignment will follow that of normal 52- or 53-byte 16-bit mode, with the additional header bytes contiguously prepended. As a result, a mode with three additional header bytes cannot be obtained in 53-byte, 16-bit mode (LSB is normally padded with zeros so MSB gets truncated). Bit 3 of this register is therefore provided to adjust the alignment in 16-bit, 53-byte mode so all five header bytes will be transmitted with up to three additional router bytes prepended.
19	Modify byte alignment in 16-bit PHY mode	When set to ’1’, this bit changes the default byte alignment in 16-bit PHY mode if this register also contains a non-zero value in bits 21-20. See the description of those bits for further details.
18-16	These bits only have meaning if a PMC PM5346 SUNI LITE/UTOPIA (bits 31-29 = "011") is selected. If any other device is chosen these bits will be ignored.	
18	52 Byte Cell	When set, the cell sent to the PHY is 52 bytes. No HEC byte will be sent.
17	Disable HEC Generation on Transmitted Cells	If bit 17 is set to ’1’, X'00' will be placed in the HEC byte of Utopia cells. If bit 17 is set to ’0’, the value of LINKC Transmitted HEC Control byte will be sent. If bit 18 is set to ’1’, no HEC is sent and this bit will be ignored.
16	Parity on all 16 bits	When set, this bit enables the IBM3206K0424 to produce a single parity bit across the 16-bit transmit data bus. When set, the IBM3206K0424 produces two parity bits, one across generation and the lower half of the 16 bits (parity bit 0) and one against the upper (parity bit 1). The default for this bit is ’1’.
15-13	PHY Receive Device	Bits 15, 14, and 13 indicate to which PHY the IBM3206K0424's Receive Config 3 will be interfacing. If the configuration's port address is all ones, then the configuration is unused and the value above bits doesn't matter. '000' Reserved '001' PMC POS-PHY (Frame based Utopia) '010' Reserved '011' PMC PM5346 SUNI LITE/UTOPIA interface (STS-3c/STM-1 OR STS-1) '100' Reserved '101' Reserved '111' Reserved
12	Enable Parity Checking	When set, this bit enables checking of parity on data from the receive path. The default parity checking is disabled. The upper bit of the transmit parity is not valid when the internal SONET/SDH Framer has been selected as the receive PHY device. The upper bit of the receive parity is also not valid when the internal SONET/SDH Framer has been selected as the transmit PHY device. This is only a concern if a combination of the internal framer and an external PHY is being used and that external PHY has a 16-bit data interface. In this case, parity cannot be checked/generated on the upper byte.
11	Even/Odd Parity Selection	Even parity is selected when this bit is cleared. The default value is for odd parity.
10	PHY Data Path Size	This bit, when set to ’0’, selects a 16-bit wide data path to the PHY device. When set to ’1’, the data path width to the PHY will be eight bits. This bit has no affect on the internal SONET/SDH framer except if the internal framer has been selected as the Rx PHY device but not as the Tx PHY device. In this case, a ’1’ on this bit will allow FYTDAT(15-13) to be used for the 16-bit external Tx PHY device, while a zero will allow FYTDAT(15-13) to be used for the Rx HDLC controller. This implies that it is not possible to use the internal RX framer, the RX HDLC interface, and an external 16-bit TX framer at the same time.

Bit(s)	Name	Description
9	16 bit parity	When this bit is set to ’1’ and it is in 16-bit mode, that parity will be calculated across all 16 bits and checked against FYRPAR(1). When in eight-bit mode with bit 9 set to ’0’, the parity is compared against FYRPAR(1). This bit has no affect if receive device is POS-PHY. The default setting of this bit is ’1’.
8	Reserved	Reserved
7-0	These bits only have meaning if a PMC PM5346 SUNI LITE/UTOPIA (bits 15-13 = "011") is selected. If any other device is chosen, these bits are ignored. Bits 2-0 are used to make adjustments to the Utopia interface for compatibility with the Suni-PHD PHY.	
7	52 Byte Cell	When set, the cell received from the PHY is 52 bytes. No HEC byte will be received.
6-4	Reserved	Reserved
6	Unassigned/Idle Cell Reception	When set to ’1’, this bit will enable unassigned/idle cell reception. This should be set to ’0’ when using the internal SONET Framer.
5	Disable Limited HEC Checking on Received Idle/Unassigned Cells	If bit 5 is set to ’1’, the receive logic will ignore the HEC byte of the header of idle and unassigned cells. Idle is defined as a header of X'00000001' and unassigned is defined as a header of X'0000000n' where n is B'xxx0'. If bit 6 is set to enable unassigned/idle cell reception, all cells are passed to REASM regardless of how this bit is set. If bit 6 is set to disable unassigned/idle cell reception and this bit is set to ’0’, the HEC byte of cells with an apparent idle header will be completely checked before deciding whether or not to pass the cell to REASM. If a cell appears to have an unassigned header, HEC bits 7, 6, and 0 are checked because they are a constant regardless of the value of bits 3-1 of the header. If other HEC bits are bad, REASM detects the HEC error and discards the cell. If there is a correctable HEC error and the cell is indeed unassigned, an out of range error occurs in REASM.
4	Ignore GFC in Null/Idle Cell Determination	Bit 4, when set, causes the receive logic to ignore the first four bits of the ATM header in determining whether a cell being received is a null or idle cell.
3	Enable XON/XOFF	Bit 3, when set, allows the XON/XOFF bit of the header of a received cell to suspend/continue transmission from the IBM3206K0424's transmit logic for all ports associated with Config 3.
2	Drive RENB Inactive When Not Receiving	When set to ’1’, this bit forces the receive logic to deactivate RENB when in the idle state.
1	Gate RSOC with RCA	When set to ’1’, this bit forces the receive logic to see both RSOC and RCA before considering RSOC valid.
0	Receive Extra Header Byte	When set to ’1’, this bit allows an extra header byte to be accepted at the start of a cell by the receive logic. The extra byte is discarded.

14.6: LINKC Map Transmit Configurations to Port Addresses

This register contains the port address for each of the transmit configurations. If the port address is B"11111" then the configuration is unused. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

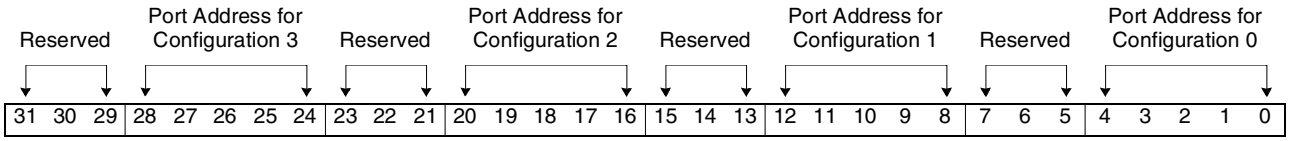
Length 32 bits

Type Clear/Set

Address XXXX 0B70 AND 0B74

Power On Value X'1F1F1F1F'

Restrictions None



Bit(s)	Description
31-29	Reserved
28-24	Port Address for Configuration 3.
23-21	Reserved
20-16	Port Address for Configuration 2.
15-13	Reserved
12-8	Port Address for Configuration 1.
7-5	Reserved
4-0	Port Address for Configuration 0.

14.7: LINKC Map Receive Configurations to Port Addresses

This register contains the port address for each of the receive configurations. If the port address is "11111" then the configuration is unused. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

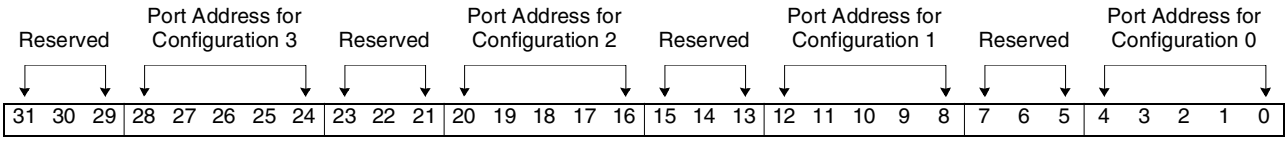
Length 32 bits

Type Clear/Set

Address XXXX 0B80 AND 0B84

Power On Value X'1F1F1F1F'

Restrictions None

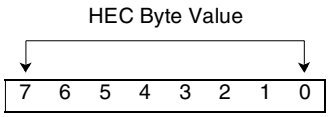


Bit(s)	Description
31-29	Reserved.
28-24	Port Address for Configuration 3.
23-21	Reserved
20-16	Port Address for Configuration 2.
15-13	Reserved
12-8	Port Address for Configuration 1.
7-5	Reserved
4-0	Port Address for Configuration 0.

14.8: LINKC Transmitted HEC Control Byte

When the IBM3206K0424 is transmitting to a 53-byte Utopia PHY the HEC byte (byte five of the ATM header) will be sent as x'00', if bit 17 of the transmitting configuration is a '1'. Otherwise the value of the LINKC Transmitted HEC Control Byte Register will be sent.

Length	8 bits
Type	Read/Write
Address	XXXX 0B04
Power On Value	X'00'
Restrictions	None



Bit(s)	Description
7-0	Value of the HEC byte to be sent when talking to a 53-byte Utopia PHY.

14.9: LINKC Interrupt/Status Register

This register reports the status of LINKC. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

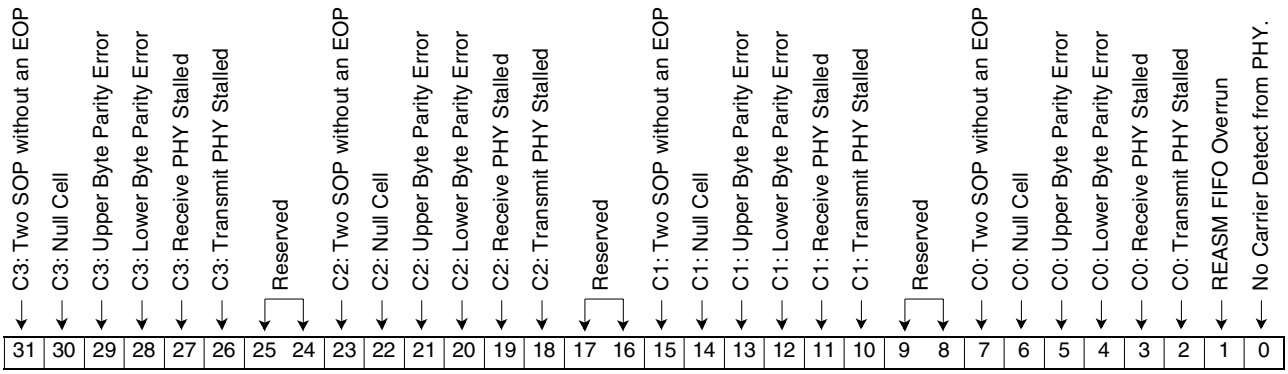
Length 32 bits

Type Clear/Set

Address XXXX 0B10 AND 14

Power On Value X'00'

Restrictions None



Bit(s)	Description
31	Indicates that Config 3 has seen two SOP in a 64-byte boundary without an EOP. Multiple SOP outside the 64-byte boundary will be passed up, and it's up to the higher levels to deal with this case.
30	Indicates that Config 3 has received a null cell.
29	Indicates a parity error on the upper byte of receive data from the Config 3 PHY.
28	Indicates a parity error on the lower byte of receive data from the Config 3 PHY.
27	Indicates Config 3's receive PHY has stalled out.
26	Indicates Config 3's transmit PHY has stalled out.
25-24	Reserved
23	Indicates that Config 2 has seen two SOP in a 64-byte boundary without an EOP. Multiple SOP outside the 64-byte boundary will be passed up, and it's up to the higher levels to deal with this case.
22	Indicates that Config 2 has received a null cell.
21	Indicates a parity error on the upper byte of receive data from the Config 2 PHY.
20	Indicates a parity error on the lower byte of receive data from the Config 2 PHY.
19	Indicates Config 2's receive PHY has stalled out.
18	Indicates Config 2's transmit PHY has stalled out.
17-16	Reserved
15	Indicates that Config 1 has seen two SOP in a 64-byte boundary without an EOP. Multiple SOP outside the 64 byte boundary will be passed up, and it's up to the higher levels to deal with this case.
14	Indicates that Config 1 has received a null cell.
13	Indicates a parity error on the upper byte of receive data from the Config 1 PHY.

Bit(s)	Description
12	Indicates a parity error on the lower byte of receive data from the Config 1 PHY.
11	Indicates Config 1's receive PHY has stalled out.
10	Indicates Config 1's transmit PHY has stalled out.
9-8	Reserved
7	Indicates that Config 0 has seen two SOP in a 64-byte boundary without an EOP. Multiple SOP outside the 64 byte boundary will be passed up, and it's up to the higher levels to deal with this case.
6	Indicates that Config 0 has received a null cell.
5	Indicates a parity error on the upper byte of receive data from the Config 0 PHY.
4	Indicates a parity error on the lower byte of receive data from the Config 0 PHY.
3	Indicates Config 0's receive PHY has stalled out.
2	Indicates Config 0's transmit PHY has stalled out.
1	REASM FIFO has been overrun.
0	No carrier detect from the PHY.

14.10: LINKC Interrupt Enable Register

This register enables the LINKC interrupt to INTST. When a bit is set in this register and the corresponding bit is set in the Interrupt Status Register, the LINKC interrupt will be driven. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See the *LINKC Interrupt/Status Register* on page 422 for the bitwise description of this register.

Length	8 bits
Type	Clear/Set
Address	XXXX 0B18 AND 1C
Power On Value	X'00'
Restrictions	None

14.11: LINKC Prioritized Interrupts

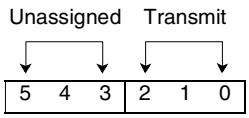
Used to access the prioritized encoding of LINKC interrupts. Reading this location will give a decimal number that is the prioritized encoding of bits 7-0 in COMET/PAKIT Status Register (seven being the most significant bit) assuming the corresponding enable bit is on.

Length	8 bits
Type	Read
Address	XXXX 0B2C
Power On Value	X'00'
Restrictions	None

14.12: LINKC Transmit State Machine Register

This register indicates the state of the transmit sequencer.

Length 6 bits
Type Read/Write
Address XXXX 0B24
Power On Value X'0'
Restrictions None



Bit(s)	Description
5-3	Unassigned cell state machine.
2-0	Transmit state machine.

14.13: LINKC Receive State Machine Register

This register indicates the state of the receiver sequencer.

Length 2 bits
Type Read/Write
Address XXXX 0B28
Power On Value X'0'
Restrictions None

Bit(s)	Description
1-0	Receive state machine.

14.14: LINKC LAN Address Register

If using an IBM built adapter that utilizes CRISCO, this register contains the ROM level in bits 63-48 and the LAN address of the adapter in bits 47-0. The lower address selects bits 63-32 and the higher address selects bits 31-0.

Length	64 bits
Type	Write/Read
Address	XXXX 0B38 AND 3C
Power On Value	X'AAAA55555555AAAA'
Restrictions	None

14.15: LINKC Canonical LAN Address Register

This register contains the same data as the LINKC LAN Address Register except each byte is bit reversed. This allows the user to obtain the LAN address in canonical format.

Length	64 bits
Type	Read
Address	XXXX 0B08 AND 0C
Power On Value	X'5555AAAAAAAA5555'
Restrictions	None

14.16: LINKC Passed TX Data Register

The bits in this register are passed over PHY transmit data I/O 15-8 when using an eight bit wide PHY data bus.

Length 8 bits

Type Read/Write

Address XXXX 0B40

Power On Value X'00'

Restrictions None

Bit(s)	Name	Description
7-0	Passed to FYTDAT(15 down to 8)	Pass to I/O line FYTDAT(15 down to 8), except in the case of the internal SONET/SDH framer. When the internal SONET/SDH framer is selected as the Rx PHY device, signals FYTDAT(15 down to 13) are used as the Rx HDLC interface, unless a 16-bit wide external Tx PHY is also selected, then they are used as data lines.

Entity 15: Nodal Processor Bus Interface (NPBUS)/CRISCO Processor for Register Initialization from EPROM Data

This entity controls the signals of the NPBUS. The PHY registers are accessible to the processor by way of the address space of the IBM3206K0424. In addition, the operation of the front end is affected by the NPBUS Status Register.

This entity also contains the CRISCO processor which can initialize chip registers at boot time by reading a data stream from EPROM which specifies the address of registers and data values to which the registers are to be initialized. In general, the data stream consists of a series of single-byte instruction operation codes, followed by an address and data values.

15.1: NPBUS Control Register

This register is used to report PHY level hardware errors and interrupts. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

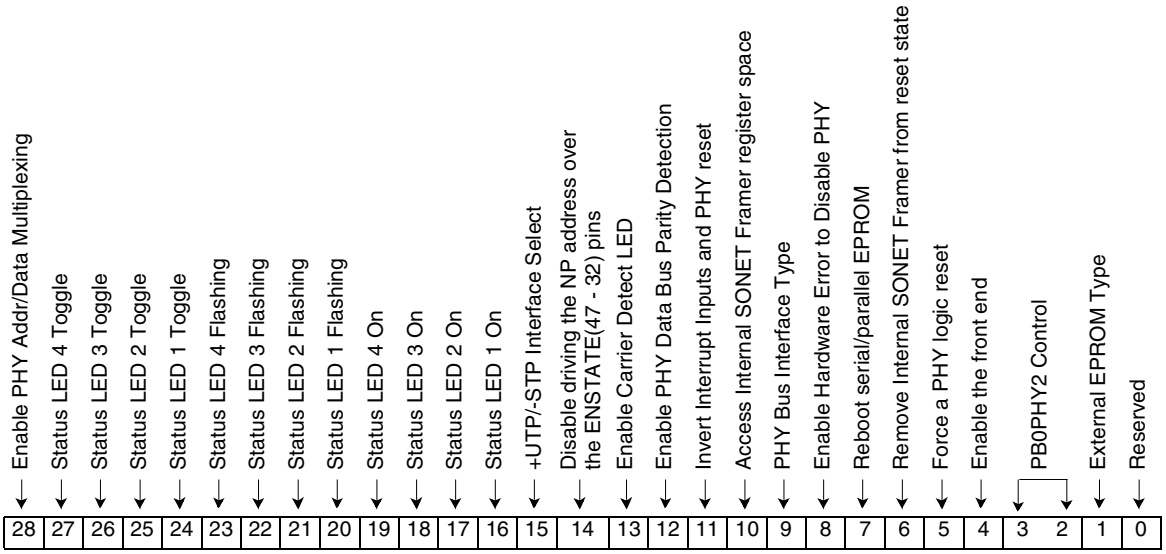
Length 29 bits

Type Clear/Set

Address XXXX 2000 and 004

Power On Value X'0002010'

Restrictions None



Bit(s)	Name	Description
28	Enable PHY Addr/Data multiplexing	This bit set to '1' will enable the ALE1, ALE2, and ALE3 control lines for PHY and Parallel EPROM accesses so that additional address bytes can be latched for up to 24Meg of addressing. Since there is an access speed penalty for this, the default is a '0' for this function.
27	Status LED 4 Toggle	When this bit is set, the state of bit 19 of this register will be toggled by repeatedly setting bit 19.

Bit(s)	Name	Description
26	Status LED 3 Toggle	When this bit is set, the state of bit 18 of this register will be toggled by repeatedly setting bit 18.
25	Status LED 2 Toggle	When this bit is set, the state of bit 17 of this register will be toggled by repeatedly setting bit 17.
24	Status LED 1 Toggle	When this bit is set, the state of bit 16 of this register will be toggled by repeatedly setting bit 16.
23	Status LED 4 Flashing	When set to '1', this bit will flash status indicator LED 4. Bit 19 of the register will override this bit.
22	Status LED 3 Flashing	When set to '1', this bit will flash status indicator LED 3. Bit 18 of the register will override this bit.
21	Status LED 2 Flashing	When set to '1', this bit will flash status indicator LED 2. Bit 17 of the register will override this bit.
20	Status LED 1 Flashing	When set to '1', this bit will flash status indicator LED 1. Bit 16 of the register will override this bit.
19	Status LED 4 On	When set to '1', this bit will turn on status indicator LED 4.
18	Status LED 3 On	When set to '1', this bit will turn on status indicator LED 3.
17	Status LED 2 On	When set to '1', this bit will turn on status indicator LED 2.
16	Status LED 1 On	When set to '1', this bit will turn on status indicator LED 1.
15	+UTP/-STP Interface Select	This bit controls a chip output pin to switch high or low and can be used to select different PHY interfaces, etc. When this bit is off, or a logical '0', the chip output is high, or a logical '1'.
14	Disable driving the NP address over the ENSTATE(47 - 32) pins	For debug reasons, the driven of the address for EPROM and PHY fetches can be turned off with this bit.
13	Enable Carrier Detect LED	When set to '1', this bit allow indicator LED 1 to reflect the status of Carrier Detect. This is a chip input.
12	Enable PHY Data Bus Parity Detection	When set to '1', if a parity error occurs on the PHY Data bus during a PHY register access, bit 1 of the NPBUS Status Register will be set.
11	Enable 16 data bit mode for PHY reg accesses	When this bit is '1', the upper eight bits of a 16-bit PHY data (bits 15-8) bus will be transferred over 47- 40 bits of the ENSTATE chip I/O bus.
10	Access Internal SONET Framer register space in memory mapped mode	When this bit is '0', the external PHY register space can be accessed through PHY 1 Registers or PHY 2 Registers. Also, the SONET Framer register space can be accessed through the EPROM access registers, NPBUS EPROM Address/Command Register and NPBUS EPROM Data Register. By providing the byte framer address (see *Sonet Framer Core (FRAMR Chiplet Address Mapping)* on page 525) in the NPBUS EPROM Address/Command Register, the byte data can be read or written from the NPBUS EPROM Data Register. When this bit is set to '1', the internal SONET framer registers can be accessed (see *Sonet Framer Core (FRAMR Chiplet Address Mapping)* on page 525). The full offset range for this access is X'2100' to X'2FFF'.
9	PHY Bus Interface Type	When this bit is '0', PHY access speed is 200 ns (SUNI-like interface). When a '1', access requires an acknowledge input response. This is to support a UTOPIA-like micro-processor interface.
8	Enable Hardware Error to Disable PHY	Allows bit 4 (Master enable) of the INTST Control Register to reset bit 4 of this register (Disables Front End logic). This function assumes that bit 4 of the INTST Control Register has already been enabled and that either a hardware or software event has turned the bit off.
7	Reboot serial/parallel EPROM	This bit will restart the external serial or parallel EPROM initialization code.
6	Remove Internal SONET Framer from reset state	This bit powers up to a zero and keeps the internal SONET Framer in reset mode. Setting this bit to a 1 will enable normal operation.

Bit(s)	Name	Description
5	Do PHY Reset	Force a PHY logic reset. Before any software reset, turn this bit on and off for the PHY specified amount of time. If the IBM ATM-TC (25 Mbps ENDEC) is used, this bit will power-up to an active reset (since the input to the ENDEC is positive reset). This bit must then be turned off for normal operation.
4	Enable	Enable the front end. When this bit is '0', no data will be transmitted to or received from the PHYs or the IBM3206K0424. See bit 8 for more information on control of this bit.
3-2	PB0PHY2 Control	Encoded control for the PB0PHY2 output pin. The enabled of PIBSELO overrides these bits and is controlled by PCINT Cascade Control Register. X'0' Enable PB0PHY2 pin X'1' Enable PBDATAP pin and its detection of valid parity. X'2' Enable MPMDSEL pin X'3' Reserved
1	External EPROM Type	This bit will set at reset time as to what type of EPROM is detected. When set, a serial EPROM has been detected. When a '0', parallel EPROM is assumed (or none at all). This will also indicate from what type of device a PCI ROM access will retrieve VPD data.
0	Reduce Serial EPROM clock	When set to '1', this bit is used for speeding up sim time for the serial EPROM. It will change the time period for the serial EPROM clock from 10 µs to 85 ns.

15.2: NPBUS Status Register

This register is used to report PHY level hardware errors and interrupts. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

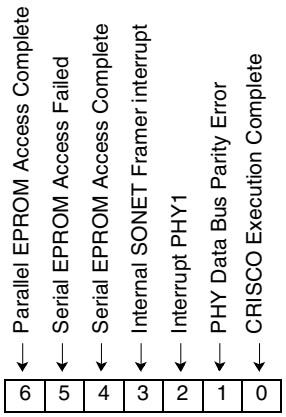
Length 7 bits

Type Clear/Set

Address XXXX 2028 and 02C

Power On Value X'x1', where x is determined by which type of CRISCO IPL EPROM is used

Restrictions None



Bit(s)	Name	Description
6	Parallel EPROM Access Complete	The requested action to the parallel EPROM has been completed. (See *NPBUS EPROM Address/Command Register* on page 433).
5	Serial EPROM Access Failed	The requested action to the serial EPROM has missed an acknowledge sequence while trying to complete the action. (See *NPBUS EPROM Address/Command Register* on page 433).
4	Serial EPROM Access Complete	The requested action to the serial EPROM has been completed. (See *NPBUS EPROM Address/Command Register* on page 433).
3	Internal SONET Framer interrupt	The internal Framer has signaled an interrupt.
2	Interrupt PHY1	This bit indicates that an interrupt occurred on PHY 1.
1	PHY Data Bus Parity Error	When set to '1', a data parity was detected over the PHY Data eight-bit bus. Parity checked is odd.
0	CRISCO Execution Complete	External serial/parallel EPROM initialization has run and is completed.

15.3: NPBUS Interrupt Enable Register

This register is used to mask bits from the NPBUS Status Register and potentially generate interrupts to the control processor. When both a bit in this register and the corresponding bit in the NPBUS Status Register are set, the NPBUS status bit will be set in INTST Interrupt Source. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See *NPBUS Status Register* on page 431 for the bit descriptions.

Length	6 bits
Type	Clear/Set
Address	XXXX 2008 and 00C
Power On Value	X'00'
Restrictions	None

15.4: NPBUS EPROM Address/Command Register

Used to accessed a maximum of 2K external serial EPROM or 16 Meg of parallel EPROM or the Sonet Framer Core. This register is used access bytes from the external EPROM or the Sonet Framer Core.

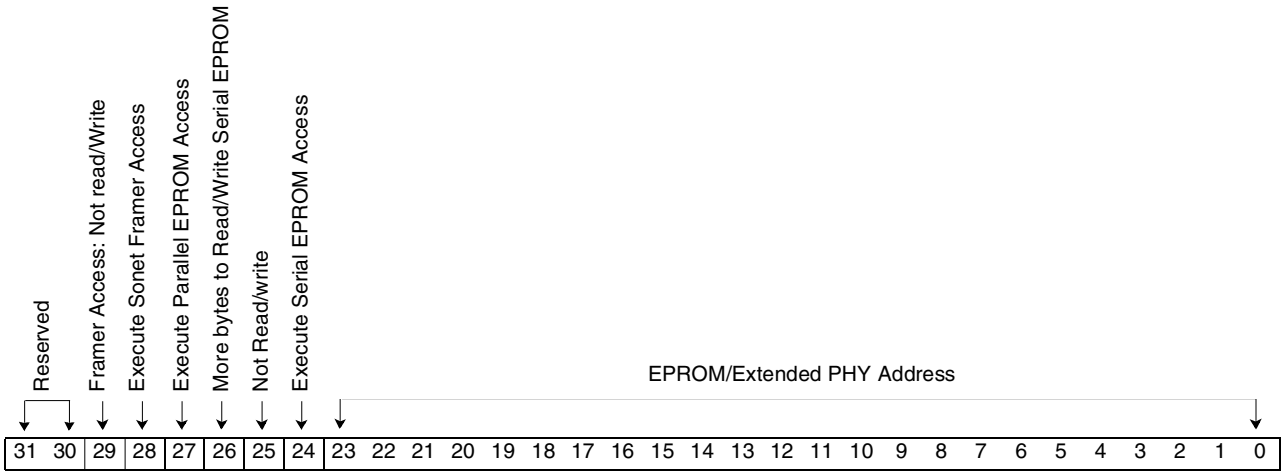
Length 32 bits

Type Read/Write

Address XXXX 2010

Power On Value X'00000100'

Restrictions None



Bit(s)	Name	Description
31-30	Reserved	Reserved.
29	Framer Access: Not read/Write	This bit set to '1' will cause a write function to the Sonet Framer Core. This bit set to '0' will cause a read function to the Sonet Framer Core.
28	Execute Sonet Framer Access	This bit will start a read or write function to the Sonet Framer Core. This bit will auto reset after the command is issued.
27	Execute Parallel EPROM Access	This bit will start a read or write function to the parallel EPROM. This bit will auto reset after the command is issued.
26	More bytes to Read/Write Serial EPROM	This bit set to '1' will help speed up sequential accesses to the serial EPROM. If writing, there is a limit as to how many bytes can be written before the serial EPROM write buffer is full. Typical range is from two to eight bytes, depending on the device.
25	Not Read/Write	This bit set to '1' will cause a write function to the serial/parallel EPROM. This bit set to '0' will cause a read function to the serial/parallel EPROM.
24	Execute Serial EPROM Access	This bit will start a read or write function to the serial EPROM. This bit will auto reset after the command is issued.
23-0	EPROM/Extended PHY Address	This holds the address field that will be used to address the serial/parallel EPROM. It is also where the 15 - 8 address bits will be held if addressing a PHY with more addressability than eight bits.

15.5: NPBUS EPROM Data Register

Used to accessed a maximum of 2K external serial EPROM or 16 Meg of parallel EPROM.

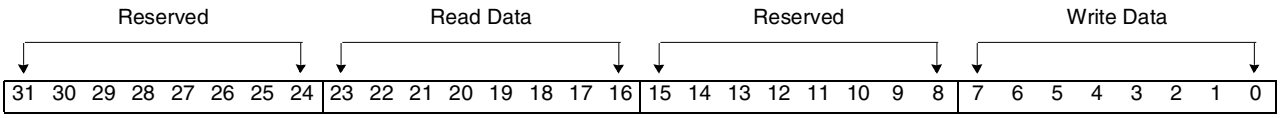
Length 32 bits

Type Read/Write

Address XXXX 2018

Power On Value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-24	Reserved	Reserved.
23-16	Read Data	Holds the data that was read back from the serial/parallel EPROM.
15-8	Reserved	Reserved.
7-0	Write Data	Holds the data that is destined to be written to the serial/parallel EPROM.

15.6: PHY 1 Registers

This address range provides access to the PHY 1 hardware. The details of the registers can be found in the specific publication for the PHY hardware.

Length 256 Doublewords (only lowest eight bits valid)

Type Read/Write

Address XXXX 2400 - 7FF

Power On Value Reference the PHY-specific publication.

Restrictions Reference the PHY-specific publication.

15.7: PHY 2 Registers

This address range provides access to the PHY 2 hardware. It should be noted that not all applications of IBM3206K0424 will use this access port. The details of the registers can be found in the specific publication for the PHY hardware.

Length 256 Doublewords (only lowest eight bits valid)

Type Read/Write

Address XXXX 2800 - BFF

Power On Value Reference the PHY-specific publication.

Restrictions Reference the PHY-specific publication.

Hardware Protocol Assist Entities

Entity 16: On-chip Checksum and DRAM Test Support (CHKSM)

Functional Description

The CHKSM entity has two functions: First, it is capable of initializing and/or testing packet and Control Memory; second, it can perform TCP checksums (Two's complement, 16-bit sum with "end-around-carry").

16.1: CHKSM Base Address Register

The CHKSM Base Address Register indicates the starting address of a test operation or checksum calculation. The base address can be set up with any alignment and valid address.

This register increments with each read or write to memory. On a test mode error, the register holds the address of the 64-bit word which was read in error.

Length	32 bits
Type	Read/Write
Address	XXXX 0A00
Power On Value	X'00000000'
Restrictions	Can only be written when CHKSM is not enabled (see EE bit in *CHKSM Control Register* on page 440)

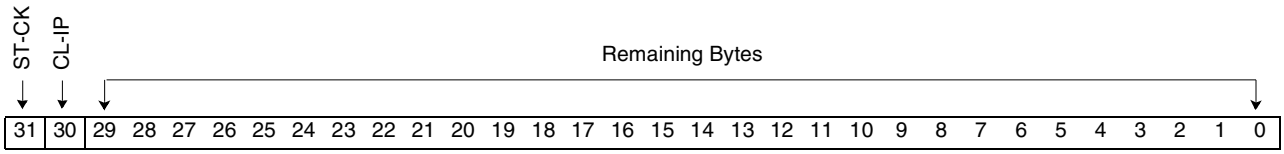
16.2: CHKSM Read/Write Count Register

The CHKSM Read/Write Count Register indicates the count of remaining bytes of a checksum operation. This register keeps track of how many bytes remain in the current checksum operation and is decremented with each read or write operation.

Any length can be set in the 30 lower significant bits.

The upper two bits of this register can be written when starting a checksum operation instead of writing the control register.

Length	32 bits
Type	Read/Write
Address	XXXX 0A04
Power On Value	X'00000000'
Restrictions	Can only be written when CHKSM is not enabled (see EE bit in *CHKSM Control Register* on page 440)



Bit(s)	Name	Description
31	ST-CK	Start a checksum operation. When this bit is written, bits 0 and 1 in the control register are set, and a checksum operation is started. This should only be done when the rest of the control register bits are already set up. (That is, memory select, invert checksum,...)
30	CL-IP	When this bit is written, it will clear the CHKSM TCP/IP Checksum Data Register. This is the same as writing a '1' to bit 6 of the CHKSM Control Register.

16.3: CHKSM TCP/IP Checksum Data Register

The CHKSM TCP/IP Checksum Data Register collects the 16-bit, two's complement, end-around-carry sum of the bytes. The source data is zero padded if it starts/ends on an odd byte boundary. The CHKSM TCP/IP Checksum Data Register collects the 16-bit, two's complement, end-around-carry sum of the bytes. It can be seeded with an initial value. If it is not cleared before running a checksum, the previous value will act as a seed. This register can be cleared when starting a checksum operation by writing the CLIP bit in the CHKSM Control Register or by writing upper bits of CHKSM Read/Write Count Register.

See *CHKSM Control Register* on page 440) for description of how to get/set current checksum alignment.

Length	16 bits	
Type	Read/Write	
Address	Normal sum	XXXX 0A08
	Inverted sum	XXXX 0A0c
	Swapped sum	XXXX 0A34
	Inverted swapped sum	XXXX 0A38
Power On Value	X'0000'	
Restrictions	Can only be written when CHKSM is not enabled (see EE bit in *CHKSM Control Register* on page 440)	

16.4: CHKSM Ripple Base Register

This register is used as base of a ripple pattern when in test mode. This register forms the base for a ripple pattern. Each consecutive byte will be incremented by one or eight in the pattern. The ripple mode must be set in the Control Register to use this feature. The value of this register will change during the test operation. If a write and compare operation are being performed, this register needs to be setup again for the second operation.

Length	8 bits
Type	Read/Write
Address	XXXX 0A14
Power On Value	X'00'
Restrictions	Can only be written when CHKSM is not enabled (see EE bit in *CHKSM Control Register* on page 440)

16.5: CHKSM Ripple Limit Register

This register is used to determine when the ripple base register overflows to zero. This register forms the compare value for the ripple base register. When the value of the ripple base register is greater than or equal to this register, the base register will overflow to zero. For example, when this register is set to four, the ripple base register would count from zero through four if counting by one.

When set to 0x00, no overflows to zero occur. For example, when the ripple value is 0xff, and you are counting by eight, the next value would still be seven. If counting by one, then the next value would be zero.

This register should be written before the ripple base.

Length	8 bits
Type	Read/Write
Address	XXXX 0A10
Power On Value	X'ff'
Restrictions	Can only be written when CHKSM is not enabled (see EE bit in *CHKSM Control Register* on page 440)

16.6: CHKSM Interrupt Enable Register

Used to specify which status register bits should be used to generate interrupts. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See *CHKSM Control Register* on page 440) for the bit descriptions.

Length	12 bits
Type	Clear/Set
Address	XXXX 0A20 and 24
Power On Value	X'ffe'
Restrictions	None

16.7: CHKSM Status Register

Used to relay CHKSM status information. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

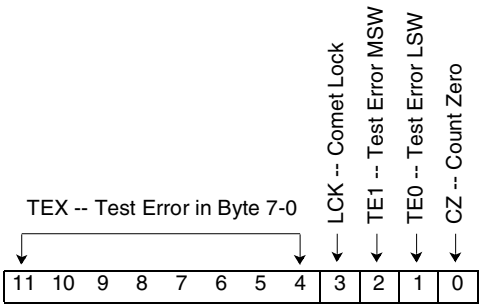
Length 12 bits

Type Clear/Set

Address XXXX 0A18 and 1c

Power On Value X'01'

Restrictions The count zero bit is not writable.



Bit(s)	Name	Description
11-4	TEX -- Test Error in Byte 7-0	When these bits are set, the checksum has determined that a read comparison error was encountered in the corresponding byte. Byte 0 is bits 63 - 56 in a 64-bit long word.
3	LCK -- Comet Lock	When this bit is set, the checksum has determined that COMIT has ceased operation for some reason, or a virtual error was detected.
2	TE1 -- Test Error MSW	When this bit is set, checksum has determined that a read comparison error was encountered in the most significant 32-bit word.
1	TE0 -- Test Error LSW	When this bit is set, the checksum has determined that a read comparison error was encountered in the least significant 32-bit word.
0	CZ -- Count Zero	This bit is set when the terminal count of an operation is reached.

16.8: CHKSM Control Register

The various bits in this register control the mode in which the checksum entity operates. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

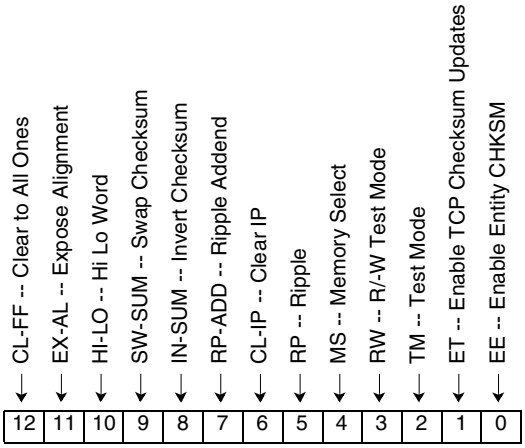
Length 13 bits

Type Clear/Set

Address XXXX 0A28 and 2c

Power On Value X'00'

Restrictions None



bit(s)	Name	Description
12	CL-FF -- Clear to All Ones	When this bit is set, the CHKSM TCP/IP Checksum Data Register is set to 0xffff when it is cleared. When this bit is cleared, the CHKSM TCP/IP Checksum Data Register is set to '0'. This option should be used if the TCP/IP checksum should never be set to '0' (0xffff is '0' also).
11	EX-AL -- Expose Alignment	When this bit is set, the internal checksum alignment is exposed for reading/writing. For writes, bit 16 of the write data is used to set the internal alignment. For reads, the alignment is exposed in bit 16 or bit 0 depending on the value of the HI-LO bit in this register. This can be useful if doing non-consecutive multiple part check sums (need to preserve alignment between chunks). When this bit is cleared, the internal checksum aliment is not exposed. It is always cleared when the CL-IP bit in this register is set. Normally, the internal alignment is calculated and maintained across consecutive check sums.
10	HI-LO -- Hi Lo Word	When this bit is set, the checksum data register data is placed in the most significant 16 bits of the 32-bit value read. When this bit is cleared, the checksum data register data is placed in the least significant 16 bits of the 32-bit value read. This bit does not affect how writes to the checksum data register occur; the data from the least significant 16 bits is always used.
9	SW-SUM -- Swap Checksum	When this bit is set, the checksum data register data is byte-swapped when read. When this bit is cleared, the checksum data register data is read normally. There are also new checksum data register addresses that can be read that do the same thing as this control bit. This bit is depreciated.
8	IN-SUM -- Invert Checksum	When this bit is set, the checksum data register data is inverted when read. When this bit is cleared, the checksum data register data is read normally. There are also new checksum data register addresses that can be read that do the same thing as this control bit. This bit is depreciated.

bit(s)	Name	Description
7	RP-ADD -- Ripple Addend	When this bit is set, the ripple base register counts up by one. When this bit is cleared, the ripple base register counts up by eight.
6	CL-IP -- Clear IP	When this bit is written, it will clear the CHKSM TCP/IP Checksum Data Register and itself. The result of this will be that this bit will never be read as a '1'. The internal alignment is also cleared.
5	RP -- Ripple	When this bit is set, a ripple pattern will be used in both the read and write test modes. The ripple pattern is used instead of the constant test pattern. When this bit is reset, the constant test pattern is used for the test mode data.
4	MS -- Memory Select	When this bit is set, all CHKSM memory accesses are to the Control Memory. When this bit is cleared, all CHKSM memory accesses are to the Packet Memory.
3	RW -- R/-W Test Mode	When this bit is set, the entity will take the data that is read and compare it to the test/ripple pattern. When this bit is reset, the checksum entity will write data using the test/ripple pattern to the DRAM.
2	TM -- Test Mode	When this bit is set, the entity will take the data that is read and compare it to the test/ripple pattern, or will write data using the test/ripple pattern to the DRAM depending on the setting of the RW bit. In both cases, the reading or writing will continue until either an error is encountered or the CHKSM Read/Write Count Register counts down to '0'. When this bit is reset, the checksum entity will operate as described by the other bits. Test and CHKSM modes are mutually exclusive, and test mode takes precedence.
1	ET -- Enable TCP Checksum Updates	When this bit is set, the entity will collect the TCP checksum in the CHKSM TCP/IP Checksum Data Register. When this bit is reset, the CHKSM TCP/IP Checksum Data Register will not be changed by data that is read from the DRAM. Test and CHKSM modes are mutually exclusive, and test mode takes precedence.
0	EE -- Enable Entity CHKSM	When this bit is set, the entity will run as specified. When this bit is reset, the entity will not run.

Debugging Register Access

16.9: CHKSM Internal State

Internal state of checksum.

Note: This register should not be written unless specifically directed to do so by IBM technical support.

Length 3 bits

Type Read/Write

Address XXXX 0A3c

Power On Value X'00000000'

Restrictions None

Software Use of CHKSM

This section outlines some ways CHKSM can be set up and used.

Test Mode Possible Patterns

In test mode, a 64-bit pattern is written/compared to/with memory. There are several different patterns that can be used:

Constant Test Pattern

When in test mode, and the RP bit is cleared, the CHKSM Ripple Base Register is replicated eight times to form a 64-bit pattern.

Ripple Pattern with increment of 1

When in test mode and the RP bit is set and RP-ADD is set and CHKSM Ripple Limit Register is set to '0', a 64-bit pattern is generated using the CHKSM Ripple Base Register as a base. For example, if the CHKSM Ripple Base Register is set to '1', the following pattern is generated:

```
0102030405060708
0203040506070809
030405060708090a
0405060708090a0b
…
```

Ripple Pattern with increment of 8

When in test mode and the RP bit is set and RP-ADD is cleared and CHKSM Ripple Limit Register is set to '0', a 64-bit pattern is generated using the CHKSM Ripple Base Register as a base. For example, if the CHKSM Ripple Base Register is set to '1', the following pattern is generated:

```
0102030405060708
090a0b0c0d0e0f10
1112131415161718
191a1b1c1d1e1f20
…
```

Ripple Pattern with Ripple Limit

Each of the above ripple patterns can also make use of the CHKSM Ripple Limit Register. By setting this register, the user can control when the ripple pattern rolls over to zero. For example, when the CHKSM Ripple Limit Register is set to three in increments by one mode the ripple pattern looks like:

```
0102030405060708
0203040506070809
030405060708090a
0001020304050607
0102030405060708
0203040506070809
030405060708090a
…
```

Similarly, when the CHKSM Ripple Limit Register is set to ten, in increment-by-eight mode, the ripple pattern looks like:

```
0102030405060708
090a0b0c0d0e0f10
1112131415161718
0001020304050607
…
```

Initializing Packet/Control Memory

The following list shows the steps to use CHKSM to initialize Packet or Control Memory:

- Make sure CHKSM is in diagnostic mode, and other mode bits are reset
- Set the start address by writing the base address
- Set up the read/write count with number of bytes to initialize
- Set up the test pattern register (ripple pattern register) with pattern to use
- Set up the Control Register to enable test mode, enable checksum entity, and set the memory select bit correctly based which memory is to be initialized
- Now busy wait until operation is done (or set up Interrupt Enable Register and wait for interrupt)

Testing Packet/Control Memory

The following list shows the steps to use CHKSM to test packet or Control Memory:

- First initialize memory with a pattern using above sequence
- Make sure CHKSM is in diagnostic mode, and other mode bits are reset
- Set the start address by writing the base address
- Set up the read/write count with number of bytes to test (same as initialization value)
- The test pattern register (ripple pattern register) already contains the pattern
- Set up the Control Register to enable test mode, turn on RW bit, enable checksum entity, and set the memory select bit correctly based which memory is to be initialized
- Now busy wait until operation is done (or set up interrupt Enable register and wait for interrupt)
- When done, check the status register for any errors

Using Ripple Pattern Generation/Checking in Packet/Control Memory

The procedures to use the ripple pattern generation and checking are the same as using test write/read modes. The only difference is that the use ripple pattern mode bit must be set and the ripple pattern base register must be set up.

Running a TCP/IP Checksum in Packet/Control Memory

The following list shows the steps to use CHKSM to generate/verify a TCP/IP checksum:

- Make sure CHKSM is in diagnostic mode (not enabled)
- Set the start address by writing the base address
- Set up the read/write count with number of bytes to run checksum over, and set the upper two bits of the read/write count register. Writing these upper two bits assumes other mode bits are set correctly (that is, memory bank select).
- Now busy wait until operation is done (or set up interrupt Enable register and wait for interrupt)

Entity 17: Processor Core (PCORE)

PCORE contains the on board processor and its local subsystems. The primary intent is to run Available Bit Rate(ABR) control software and user application code such as protocol termination/assist code.

DCR Interface

The Device Control Register (DCR) interface is a special processor bus to access local registers. These include serial port registers and various other registers.

Interrupt Controller

This logic manages the interrupts that are passed on to the Cobra Core. There are two levels of interrupt for the core: Critical Interrupts and Normal Interrupts. Interrupts can be taken from both on chip and off chip sources. PCORE has a variety of interrupt source and enable registers.

Bridge-Address Translation

Cobra Core can access a variety of memory subsystems. Facilities are provided that allow multiple subsystem accesses. Processor address space is translated into target memory system addresses. For the most part, this allows the processor to be unaware of target memory address space considerations while running mainline code.

OCM SRAM

OCM is provided for the use of the processor. Typical access time to the OCM is a single cycle, the same as for cache. Address translation facilities have been added to make more efficient use of this memory via additional and extended BAT registers.

Control Memory

Control Memory can be accessed by the processor. This memory may be mapped into the processor space in a number of different ways.

Packet Memory

Packet memory can be accessed by the processor. This memory may be mapped into the processor space in a number of different ways. Packet memory space also includes the virtual memory space of the IBM3206K0424.

PCI Master Interface-External

The processor can access the PCI bus through this interface. Parts of PCI space are mapped into processor space. There are a number of different ways that this can be mapped into processor space.

Processor Register Space

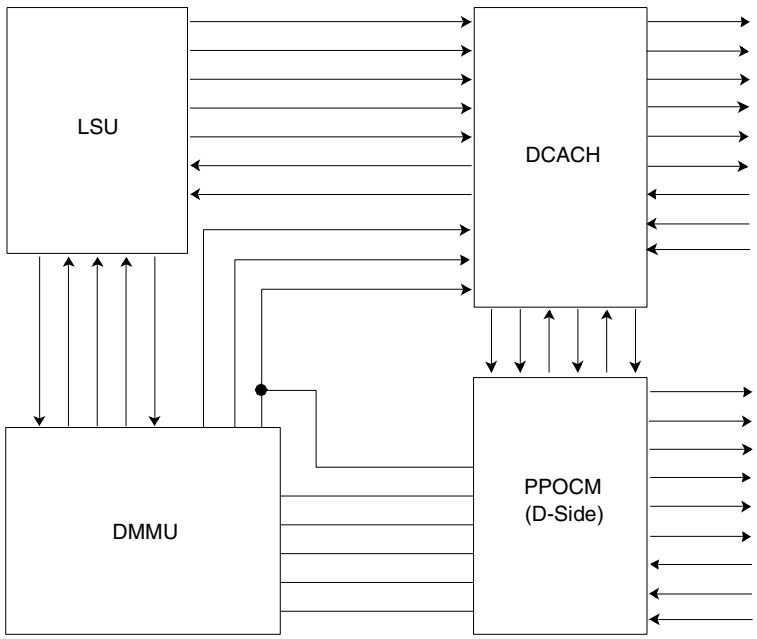
This access mode of the PCI master interface allows access to the internal IBM3206K0424 registers. This access is handled internally and does not affect the external PCI bus.

Address Translation Examples

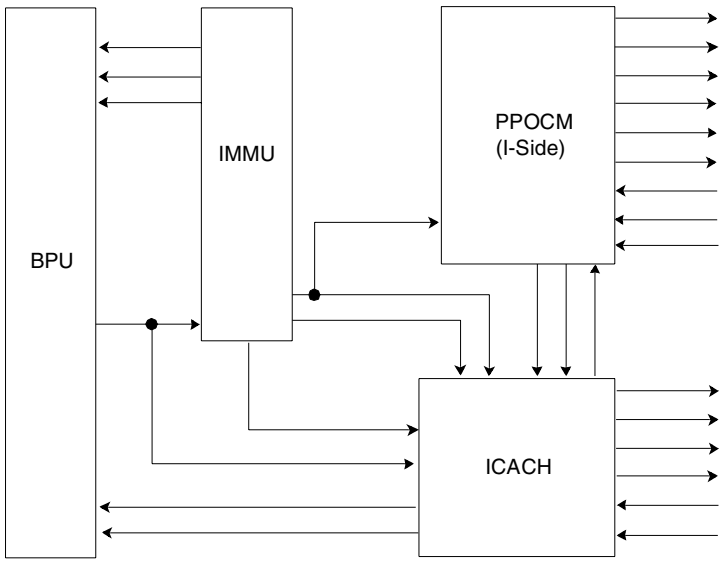
TBD

Cobra Structure

Cobra Core Data Side



Cobra Core Instruction Side



Cobra Core "Glossy" Description

The Cobra Core is a 32-bit PowerPC RISC embedded controller. It is fully compatible with the PowerPC User Instruction Set Architecture. Details about the exact instruction set are described below. The Cobra Core has a PowerPC instruction execution complex, separate 32k instruction and data caches, separate instruction and data 604-style MMUs (not supported this pass), and 401-style interrupts, timers and debug facilities. The Cobra Core has a direct connection to 96k of on-chip memory which can be used for both instruction and data storage, as well as interfaces to the IBM3206K0424's PCI and register buses and both of the IBM3206K0424's memory controllers. The DCR bus provides fastand private access to specific performance-sensitive registers.

Features

Instruction Execution

- Compatible with PowerPC User Instruction Set Architecture (UISA)
- Separate Branch, Condition Register, Integer, and Load/Store units for super-scalar execution
- Support for limited out-of-order execution
- Dispatches/Executes up to 2/4 instructions per clock cycle
- Four stage pipeline allowing single cycle execution for most instructions
- 32x32-bit general purpose registers (GPRs)
- Single cycle loads and stores
- Byte, halfword, word, and string accesses to any byte alignment supported in hardware
- Hardware multiply and divide (multiply up to 10 cycles, divide up to 32 cycles)
- No FPU hardware - FPU instructions result in interrupts

Timers

- 32-bit decrementer
- 64-bit time base
- Programmable and fixed interval timers
- Watchdog timer

Cobra Core Exceptions

- Two priority levels
  - Normal - uses SRR0/1 (603)
  - Critical - uses SRR2/3 (401 extension)
- Exception Types
  - System Reset (boot vector)
  - Machine Check
  - Data Storage Interrupt
  - Instruction Storage Interrupt
  - External
  - Alignment
  - Trap
  - Invalid Opcode
  - Privilege Violation
  - Floating Point Unavailable
  - Decrementer
  - System Call
  - Trace
  - System Management (603)
  - Programmable Interval Timer (401)
  - Fixed Interval Timer (401)
  - Watch Dog Timer (401)
  - Critical Interrupt (401)
  - Debug Interrupt (401)

Optional Architecture Extensions

- Programmable boot address (system interrupt vector)
- Interrupt enhancements
  - Individually re-locatable interrupts
  - Individually programmable interrupt level (normal/critical)

Caches

- 32-byte cache lines (eight words/line)
- Four-way set associative write back 32k instruction and data caches

Memory Management

- Real Flat Address Mode supported

Interrupt Controller

- Two interrupt levels external to COBRA: normal and critical
- Three-way interrupts
  - From IBM3206K0424 to COBRA
  - From COBRA to PCI
  - From PCI to COBRA

Debug Support

- PowerPC JTAG debugger support (401 RiscWatch)
- 401 debug instructions
- Serial Port Debugger support
- PCI Debug access to JTAG debug facilities

Interfaces

On-Chip Memory

- 96k of on-chip memory
- Can be used simultaneously by instruction and data accesses
- OCM Basic DMA controller provides bulk data moves to/from OCM

IBM3206K0424 Registers

- Read/Write access to the IBM3206K0424 register bus
- Some critical registers are mapped to COBRA Core DCR register space

PCI Bus

- Read/Write master access to PCI Bus
- Currently no actual streaming/bursting supported
- Pseudo bursting supported (multiple back to back single transfers)
- Interrupt sink
- No arbitration supported (we are not a complete bridge)

Comet/Pakit Memory

- Able to use both memory controllers for both instruction and data accesses

Performance

- 133 MHz operating frequency
- Performance estimates unavailable

Instruction Set

Instruction Set (with 401 as a base)

- Supports ALL 401 instructions with the following additions/subtractions:
- Added from 603
  - mfsr, mfsrin, mtsr, mtsrin, tlbie, tlbld, tlbli, tlbsync (**Note**: virtual memory not supported this pass)
  - mftb (for 603 style time base support)
- Removed from 401
  - dcread, icread (replaced with SPR access to the caches)
  - Changed mfspr, mtspr (to accommodate new SPRs)

Instruction Set (with 603 as a base)

- Supports ALL 603 instructions with the following additions/subtractions
- Added from 401
  - dccci, icbt, iccci, mfdcr, mtdcr, rfci, wrtee, wrteei
- Removed from 603
  - fXXXXX, lfXXX, stfXXX, mffXXX (floating point instructions),
  - eciwx, ecowx (PowerPC I/O addressing not supported - only memory space addressing)
  - Changed mfspr, mtspr (to accommodate new SPRs)

Cobra Instruction Overview

Cobra Core supports all of the instructions in either the PowerPC 603 User's Manual (MPR603UMU-01,MPC603UM/AD) or the Power PC 401 Core User's Manual (v0.07 1/28/978) with the following changes. If an instruction does not exist in both user's manuals, it is described below.

Cobra Instruction Changes

Instruction(s)	Source	Description
dccci, icbt, iccci	401	Added for cache management
mfdcr, mtdcr	401	Added for DCR bus support
rfci, wrtee, wrteei	401	Added for 40x interrupt support
mfsr, mfsrin, mtsr, mtsrin	603	Added for 60x MMU support (not supported this pass)
mftb	603	Added for 60x style time base support
tlbie	603	Added for 60x MMU support (not supported this pass)
tlbsync	603	Added for 60x compatibility, acts as a no-op
mfbus, mtbus	Cobra Core	Added - See Appendix: New instructions
dcread, icread	401	Removed - cache layout is different
dcba	405	Not Supported - 405 only instruction
tlbre, tlbsx(.), tlbwe	405	Removed - 40x style MMU not supported
eciwx, ecowx	603	Removed - only memory space addressing supported
fxxxxx	603	Removed floating point support
lfd, lfdu, lfdux, lfdx, lfs, lfsu, lfsux, lfsx	603	Removed floating point support
mffs, mffsb0, mffsb1, mffsf, mffsfi	603	Removed floating point support
stfd, stfdu, stfdux, stfdx, stfiwx, stfs, stfsu, stfsux, stfsx	603	Removed floating point support
tlbld, tlbli	603	Removed - 603 style MMU support, may add next pass

Cobra Facilities Overview

The following registers are patterned after the registers in either the PowerPC 603 User's Manual (MPR603UMU-01,MPC603UM/AD) or the Power PC 401 Core User's Manual (v0.07 1/28/978). Differences between the two implementations and the Cobra Core implementations are detailed below.

The registers are split into these functional groupings: Machine Control/Status Registers, Branch Control Registers, Debug Control Registers, Special Purpose Facilities, Interrupt and Exception Registers, Timer Registers, Cache Control Registers, and Translation Control Registers.

Source Key:

- BOTH - same bit definitions as 603 and 401
- 401 - same bit definitions as 401
- 603 - same bit definitions as 603
- PPC - same bit definitions as general Power PC architecture
- COBRA - COBRA Configuration

Machine Control/Status Registers

Register Name	SPR Number	Source	POR value	Notes
cr	NA	BOTH	X'00 00 00 00'	
hid0	1023	COBRA	X'00 00 00 FC'	
ear	282	603		Not Supported, writes ignored, reads as zeros.
msr	NA	COBRA	X'00 00 00 40'	
xer	1	BOTH	X'00 00 00 00'	
mchk-on	688/689	COBRA	X'00 00 00 00'	Machine Check Enable - set bits off/ set bits on
pvr	287	COBRA	X'10 10 00 00'	

Branch Control Registers

Register Name	SPR Number	Source	POR value	Notes
ctr	9	BOTH	X'00 00 00 00'	
lr	8	BOTH	X'00 00 00 00'	

Debug

Register Name	SPR Number	Source	POR value	Notes
dabr	1013	PPC		Not implemented, use DAC1
dac1	1014	401	X'00 00 00 00'	
dbcr	1010	401	X'00 00 00 00'	
dbdr	1011	401	X'00 00 00 00'	
dbsr	1008	401	X'00 00 00 00'	
iabr	1010	603		Not implemented, use IAC1
iac1	1012	401	X'00 00 00 00'	

Special Purpose Facilities

Register Name	SPR Number	Source	POR value	Notes
sprg0	272	BOTH	X'00 00 00 00'	
sprg1	273	BOTH	X'00 00 00 00'	
sprg2	274	BOTH	X'00 00 00 00'	
sprg3	275	BOTH	X'00 00 00 00'	
sprg4-7	276-279	COBRA	X'00 00 00 00'	Read/Write
sprg4-7	68-71	COBRA	X'00 00 00 00'	Read Only

Interrupt and Exception Registers

Register Name	SPR Number	Source	POR value	Notes
dar	19	603	X'00 00 00 00'	
dear	19	401		Same as dar with different address
dsisr	18	603	X'00 00 00 00'	
esr	980	401	X'00 00 00 00'	Only MCI, PIL, PPR, PTR, DST, and PFEU bits implemented
evpr	982	401	X'00 00 00 00'	
srr0	26	BOTH	X'00 00 00 00'	
srr1	27	BOTH	X'00 00 00 00'	When HID0(27) = 0 (603 mode), upper 16 bits act as status; the lower 16 bits are set from the MSR. When HID0(27) = 1 (401 mode), all bits are set from the MSR.
srr2	990	401	X'00 00 00 00'	Behaves like srr0; used for critical level interrupts.
srr3	991	401	X'00 00 00 00'	Behaves like srr; used for critical level interrupts.

Timer Registers

Register Name	SPR Number	Source	POR value	Notes
dec	22	603	X'FF FF FF FF'	
pit	987	401	X'00 00 00 00'	
tbhi	988	401	X'00 00 00 00'	Upper half of 64-bit time base register. R/W, privileged spr.
tblo	989	401	X'00 00 00 00'	Lower half of 64-bit time base register. R/W, privileged spr.
tbhu	972	401	X'00 00 00 00'	Upper half of 64-bit time base register. R-only, non-privileged spr.
tblu	973	401	X'00 00 00 00'	Lower half of 64-bit time base register. R-only, non-privileged spr.
tbl	284	603	X'00 00 00 00'	Same as tblo, but write only
tbu	285	603	X'00 00 00 00'	Same as tbhi, but write only
tbl	268	603	X'00 00 00 00'	tbr (not spr) same as tblu
tbu	269	603	X'00 00 00 00'	tbr (not spr) same as tbhu
tcr	986	401	X'00 00 00 00'	
tsr	984	401	X'00 00 00 00'	

Cache Control Registers

Register Name	SPR Number	Source	POR value	Notes
dccr	1018	401	X'00 00 00 00'	
dlcr	917	COBRA	X'00 00 00 00'	Bits 0-15 correspond to target tags 0-15, enabling auto data cache line locking
dtwf	919	COBRA	X'00 00 00 00'	Bits 0-15 correspond to target tags 0-15, enabling target word first data cache fills
iccr	1019	401	X'00 00 00 00'	
ilcr	916	COBRA	X'00 00 00 00'	Bits 0-15 correspond to target tags 0-15, enabling auto instruction cache line locking
itwf	918	COBRA	X'00 00 00 00'	Bits 0-15 correspond to target tags 0-15, enabling target word first instruction cache fills
cdcbr	983	401		Not implemented
dcwr	954	401		Not implemented
icdbdr	979	401		Not implemented

Translation Registers

Register Name	SPR Number	Source	POR value	Notes
sdr1	25	603	X'00 00 00 00'	Not Supported - translation disabled this pass
sgr	53	401	X'00 00 00 00'	Has no affect this pass. All storage is non-guarded.
sler	55	401		Not Supported - writes ignored, reads as '0'
stt0	912	COBRA	X'00 00 00 00'	When IR=0 or DR=0, these registers are used to assign target tags to instruction or data memory accesses. The mapping is stt0(0:3) = 0x00000000 - 0x07FFFFFF through stt3(28:31) = 0xF8000000 - 0xFFFFFFFF.
stt1	913	COBRA	X'00 00 00 00'	
stt2	914	COBRA	X'00 00 00 00'	
stt3	915	COBRA	X'00 00 00 00'	
sr0-15	NA	603	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat0u	536	603	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat0l	537	603	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat1u	538	603	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat1l	539	603	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat2u	540	603	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat2l	541	603	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat3u	542	603	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat3l	543	603	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat4u	568	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat4l	569	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat5u	570	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat5l	571	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass

Translation Registers (Continued)

Register Name	SPR Number	Source	POR value	Notes
dbat6u	572	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat6l	573	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat7u	574	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
dbat7l	575	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat0u	528	603	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat0l	528	603	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat1u	530	603	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat1l	531	603	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat2u	532	603	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat2l	533	603	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat3u	534	603	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat3l	535	603	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat4u	560	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat4l	561	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat5u	562	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat5l	563	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat6u	564	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat6l	565	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat7u	566	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass
ibat7l	567	COBRA	X'00 00 00 00'	Not Supported - translation disabled this pass

Exception Vector Override Registers

Register Name	SPR Number	Source	POR value	Notes
evc-off	700	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evc-on	701	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evov-off	702	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evov-on	703	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-mc	720	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-dsi	721	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-isi	722	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-ex1	723	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-aln	724	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-inv	725	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-prv	726	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-trp	727	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-fpu	728	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.

Exception Vector Override Registers (Continued)

Register Name	SPR Number	Source	POR value	Notes
evr-dec	729	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-sc	730	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-trc	731	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-smi	732	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-pit	733	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-fit	734	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-wdt	735	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-ex2	752	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.
evr-dbg	753	COBRA	X'00 00 00 00'	See *Cobra Core Exceptions* on page 447.

Internal Debug Access Address Map

Unit	Debug Bus Address Range
Bpu Internals	X'000' - X'07F'
Reg Internals	X'080' - X'0FF'
Lsu Internals	X'100' - X'17F'
Fxu Internals	X'180' - X'1BF'
ICache Internals	X'1C0' - X'1FF'
Immu Internals	X'200' - X'2BF'
DCache Internals	X'2C0' - X'2FF'
Dmmu Internals	X'300' - X'3BF'
Reserved	X'3C0' - X'3FF'

Cobra Specific Register Definitions

For most of the registers named above, either a 40x or 60x spec will give the bit definition. There are a few registers which are different enough that they are described in detail below.

17.1: Hardware Implementation Detail 0 Register (HID0)

Enables caches and controls architecture specific functionality. This register controls whether Cobra Core acts as a 40x or 60x series processor and allows the different features of each architecture to be individually selected.

Length 32 bits

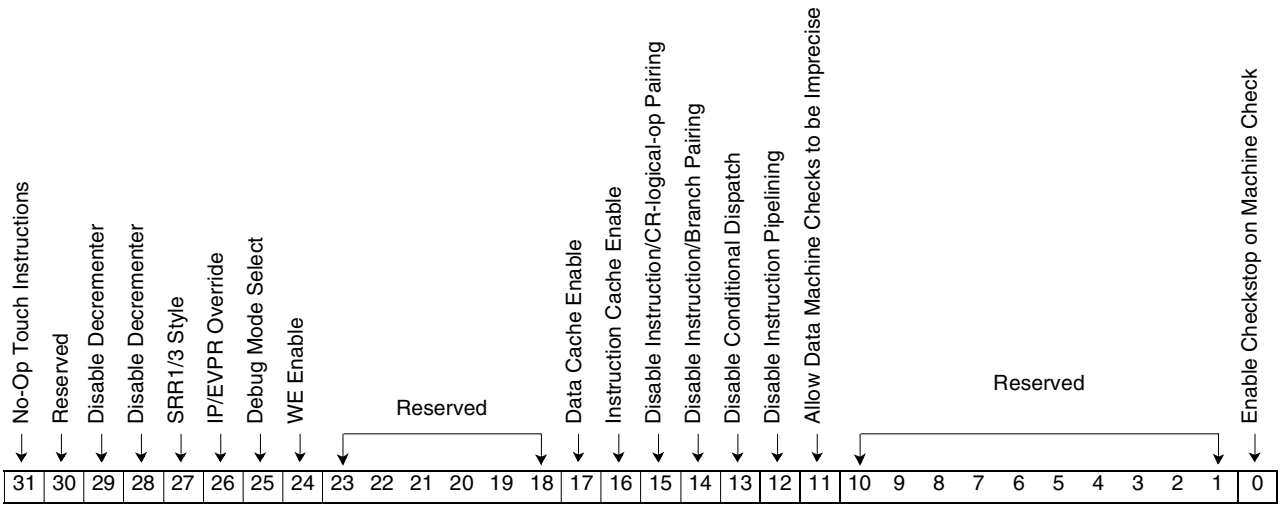
Type Read/Write

Address 1023

Power on Reset value X'00 00 00 FC' (40x mode)

X'80 00 00 00' - Alternative value (60x mode suggested value)

Restrictions



Bit(s)	Name	Description
31	No-Op Touch Instructions	0 = touch instructions work; 1 = touch instructions disabled
30	Reserved	Reserved
29	Disable decrementer	When '0', the decrementer register is a free running countdown register which causes a decrementer interrupt when it decrements through '0'. When '1', the decrementer register does not decrement.
28	Disable alignment interrupts on lmw/stmw	When '0', lmw/stmw instructions to non-word aligned addresses cause alignment interrupts (60x mode). When '1', lmw/stmw instructions never cause alignment interrupts (40x mode).
27	SRR1/3 Style	When '0', interrupts cause srr1/3 to receive interrupt codes in bits 0:15, and MSR contents in bits 16:31. Likewise, rfi/rfci restore bits 16:31 of srr1/3 to the MSR. When '1', interrupts cause srr1/3 to receive MSR contents in bits 0:31. Likewise, rfi/rfci restore bits 0:31 of srr1/3 to the MSR. Status is stored in ESR register regardless.

Bit(s)	Name	Description
26	IP/EVPR Override	When ’1’, EVPR contents are used as the upper 16 bits of the exception vector, regardless of the value of MSR(IP). When ’0’, MSR(IP) determines the upper 16 bits of the exception vector. HID0(IPO=26) MSR(IP=25) Exception Vector 0 0 0x0000vvvv 0 1 0xFFF0vvvv 1 - 0xeeeevvvv **Note:** eeee = EVPR register contents, vvvv = exception vector offset (for example, 0700)
25	Debug Mode Select	A ’0’ enables the BE and SE bits of the MSR to function as a 60x, a ’1’ enables the MSR(DE) bit as a 40x. **Note**: All 40x debug facilities (IAC, DAC, etc.) are disabled when in 60x mode.
24	WE Enable	Enables the MSR(WE) bit to cause a 40x style WE. 0 = disabled, 1 = enabled
23-18	Reserved	Reserved
17	Data Cache Enable	0 = disable, 1 = enable
16	Instruction Cache Enable	0 = disable, 1 = enable
15	Disable Instruction/CR-logical-op Pairing	1 = disable, 0 = enable
14	Disable Instruction/Branch Pairing	1 = disable, 0 = enable
13	Disable Conditional Dispatch	1 = disable, 0 = enable
12	Disable Instruction Pipelining	1 = disable, 0 = enable
11	Allow Data Machine Checks to be Imprecise	Setting this to ’1’ will result in a very small performance gain, at the expense of accuracy in the SRR0/2 of a machine check.
10-1	Reserved	Reserved
0	Enable Checkstop on Machine Check	When ’1’, a machine check which occurs when MSR(ME)=0 will cause the hardware to freeze, maintaining much of the state of the machine. When ’0’, the processor will attempt to continue execution when a machine check occurs when MSR(ME)=0. When MSR(ME)=1 and a machine check occurs, a machine check exception is taken regardless of the setting of this bit.

17.2: Machine State Register (MSR)

Controls the run time state of the Cobra Core.

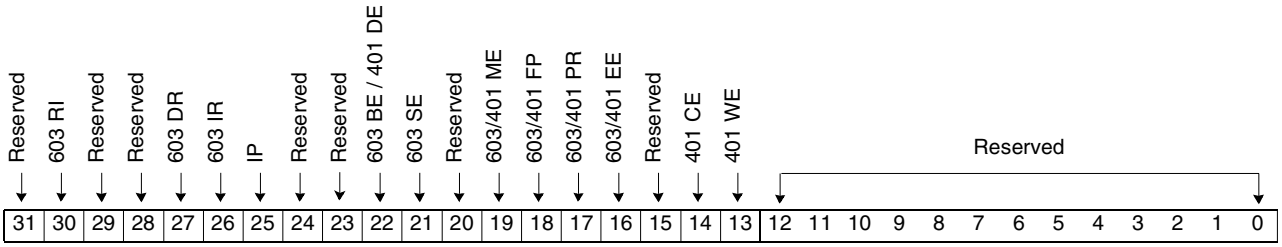
Length 32 bits

Type Read/Write

Address Accessible via the mtmsr/mfmsr instructions

Power on Reset value X'00 00 00 40'

Restrictions None



Bit(s)	Name	Description
31	Reserved	Unimplemented (603/401 LE). Cobra Core doesn't support Little Endian execution.
30	603 RI	Recoverable Interrupt. This bit is cleared when an exception is taken.
29	Reserved	Reserved
28	Reserved	Reserved
27	603 DR	Translation is not supported. Do not set this bit.
26	603 IR	Translation is not supported. Do not set this bit.
25	IP	In combination with HID0(IPO = 26), this bit determines the upper 16 bits of an exception vector. See *Hardware Implementation Detail 0 Register (HID0)* on page 456 for full details.
24	Reserved	Reserved
23	Reserved	Unimplemented (603 FE1). Cobra Core doesn't support floating point.
22	603 BE / 401 DE	If HID0(25) = 0 this is 603 BE otherwise it is 401 DE.
21	603 SE	If HID0(25) = 0 this is 603 SE otherwise it is a read/write bit with no affect.
20	Reserved	Unimplemented (603 FE0). Cobra Core doesn't support floating point.
19	603/401 ME	Enables Machine Check Exceptions. If this bit is '1', a machine check will cause a machine check exception to occur. If this bit is '0', a machine check will cause Cobra Core to halt execution (if HID0(0) = 1), or the machine check will be ignored (if HID0(0) = 0).
18	603/401 FP	If FP=0, all floating point instructions cause a floating point disabled interrupts. If FP=1, all floating point instructions cause illegal instruction interrupts.
17	603/401 PR	Privilege Instruction Restricted. If this bit is '1', attempting to execute a privileged (supervisor) instruction will result in a privilege violation exception. If this bit is '0', all instructions may be executed normally.
16	603/401 EE	External Interrupt Enable. Used to mask off non-critical level exceptions.
15	Reserved	Unimplemented (603 ILE) Cobra Core doesn't support Little Endian execution.

Bit(s)	Name	Description
14	401 CE	Enables critical level exceptions. Critical level exceptions are higher priority than normal exceptions. Any exception can be assigned critical level in Cobra Core. This bit can only mask off exceptions programmed to be critical level through other Cobra Core facilities. By default, no exception is critical level. Note: The 60x doesn't have the concept of critical level exceptions. The 40x has both critical level exceptions and a critical interrupt. The critical interrupt is just a second external interrupt. In the 40x, specific exceptions were hard wired to be critical level, and the rest were hard wired to be normal level.
13	401 WE	If HID0(24) = 0, this is a r/w bit with no affect (603 POW), otherwise it is the (401 WE) bit. When this bit is set, the processor halts operation until this bit is cleared (via an interrupt).
12-0	Reserved	Reserved

17.3: Exception Status Register (ESR)

When an exception occurs, this register is updated to indicate which condition caused the exception. If an exception vector can only be reached by one exception, this register is cleared.

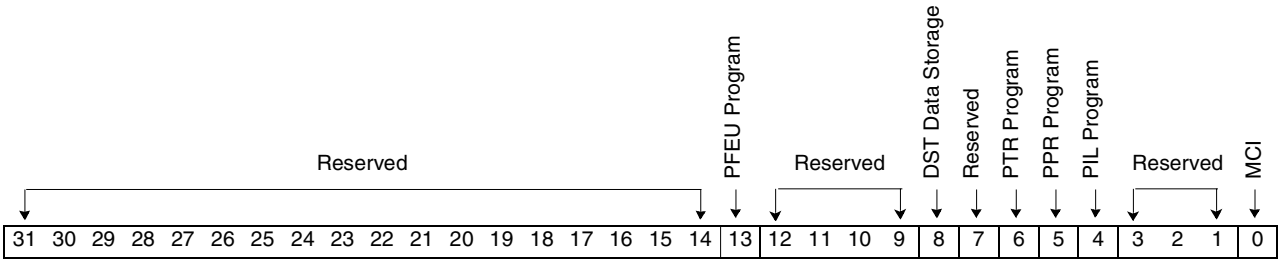
Length 32 bits

Type Read/Write

SPR Address 980

Power on Reset value X'00 00 00 00'

Restrictions None

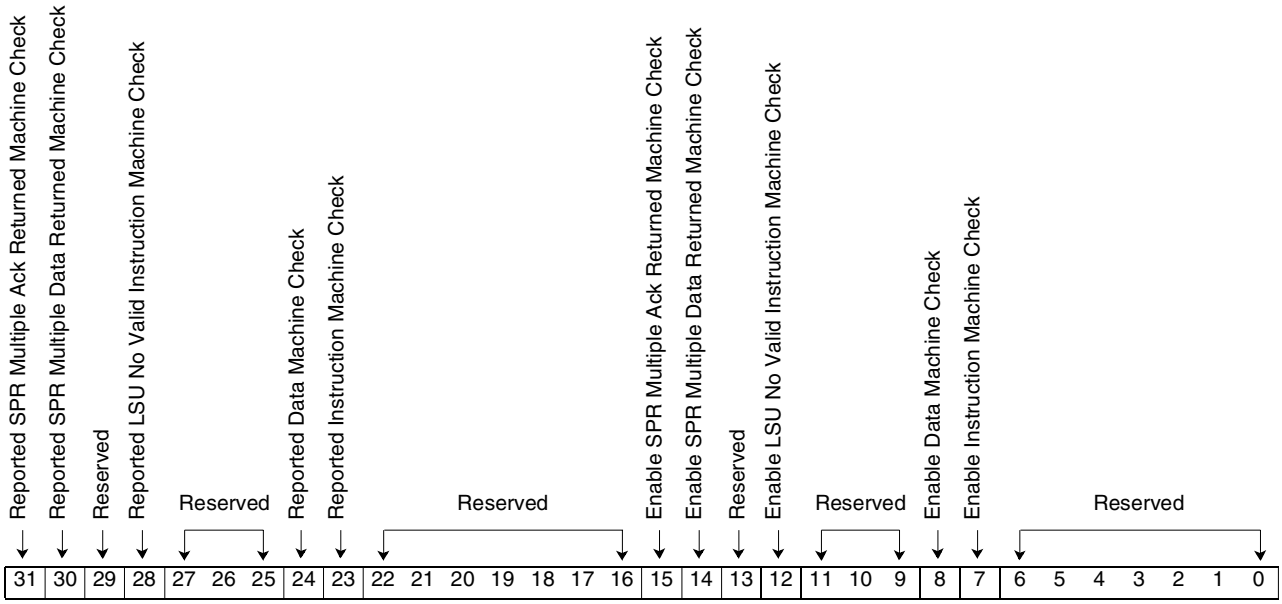


Bit(s)	Name	Description
31-14	Reserved	Reserved
13	PFEU Program	Floating point enabled, but instruction unimplemented exception.
12-9	Reserved	Reserved
8	DST Data Storage	A store instruction or (dcbz/dcbi) caused the data exception.
7	Reserved	Reserved
6	PTR Program	Program - Trap Instruction Exception
5	PPR Program	Program - Privileged Instruction Exception
4	PIL Program	Program - Illegal Instruction Exception
3-1	Reserved	Reserved
0	MCI	Machine Check - Instruction

17.4: Machine Check Enable Register (MCHK)

When an exception occurs, this register is updated to indicate which condition caused the exception. If an exception vector can only be reached by one exception, this register is cleared.

Length	32 bits
Type	Clear/Set
SPR Address	688/689
Power on Reset value	X'00 00 00 00'
Restrictions	None



Bit(s)	Name
31	Reported SPR Multiple Ack Returned Machine Check
30	Reported SPR Multiple Data Returned Machine Check
29	Reserved
28	Reported LSU No Valid Instruction Machine Check
27-25	Reserved
24	Reported Data Machine Check
23	Reported Instruction Machine Check
22-16	Reserved
15	Enable SPR Multiple Ack Returned Machine Check
14	Enable SPR Multiple Data Returned Machine Check
13	Reserved
12	Enable LSU No Valid Instruction Machine Check
11-9	Reserved

Bit(s)	Name
8	Enable Data Machine Check
7	Enable Instruction Machine Check
6-0	Reserved

PCORE Register Definitions

17.5: PCORE Control Register

The PCORE Control Register provides control information about PCORE operations. This is the PCORE control register. It is used to control operation.

See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

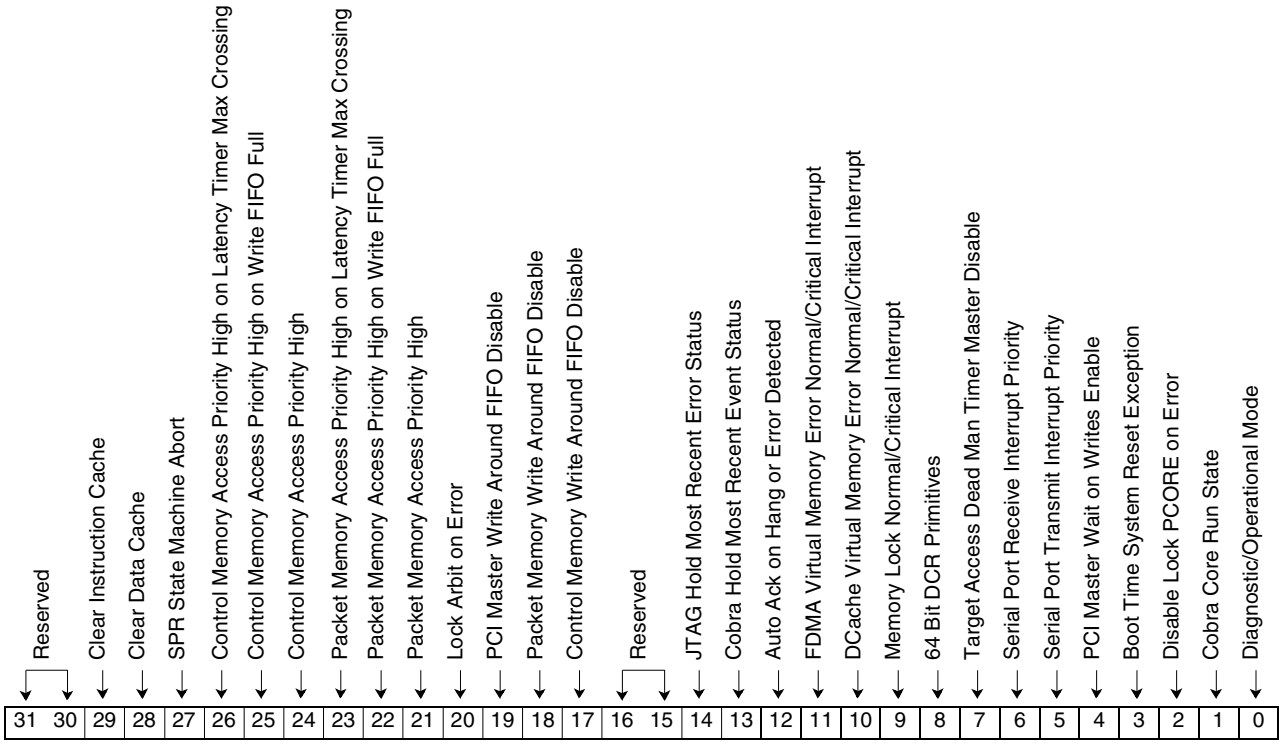
Length 32 bits

Type Clear/Set

Address XXXX 4000 and 004

Power on Reset value X'00 0E 40 09'

Restrictions Caution must be used when asserting some of the bits during operation.



Bit(s)	Name	Description
31-30	Reserved	Reserved
29	Clear Instruction Cache	This bit will cause the instruction cache to clear itself to a known state. It will also cause the processor to be in stop state when this happens. It will be cleared when the cache has finished.
28	Clear Data Cache	This bit will cause the data cache to clear itself to a known state. It will also cause the processor to be in stop state when this happens. It will be cleared when the cache has finished.
27	SPR State Machine Abort	This bit will put the SPR Access Machine into a reset state, so it should be used by setting and then clearing.

Bit(s)	Name	Description
26	Control Memory Access Priority High on Latency Timer Max Crossing	When set, Control Memory accesses will switch to high priority when the Control Memory latency counter crosses the high priority crossover register value.
25	Control Memory Access Priority High on Write FIFO Full	When set, Control Memory accesses will switch to high priority when the Control Memory write around FIFO is full.
24	Control Memory Access Priority High	When set, Control Memory accesses will be at high priority always.
23	Packet Memory Access Priority High on Latency Timer Max Crossing	When set, Packet Memory memory accesses will switch to high priority when the Packet Memory latency counter crosses the high priority crossover register value.
22	Packet Memory Access Priority High on Write FIFO Full	When set, Packet Memory accesses will switch to high priority when the Packet Memory write around FIFO is full.
21	Packet Memory Access Priority High	When set, Packet Memory accesses will be at high priority always.
20	Lock Arbit on Error	When set to '1', will cause a lock command to be issued to arbit to halt the memory subsystem.
19	PCI Master Write Around FIFO Disable	Disables the write around buffer for PCI Master. When enabled, write data from either the ICACH or DCACH is buffer through this FIFO on writes.
18	Packet Memory Write Around FIFO Disable	Disables the write around buffer for Packet Memory. When enabled, write data from either the ICACH or DCACH is buffer through this FIFO on writes.
17	Control Memory Write Around FIFO Disable	Enables the write around buffer for Control Memory. When enabled write data from either the ICACH or DCACH is buffer through this FIFO on writes.
16-15	Reserved	Reserved
15	Disable Xfer Abort on Pseudo Core Reset	When this bit is written to '1', transfer aborts on Pseudo resets are disabled; when '0', the core master state machines will be put into idle.
14	JTAG Hold Most Recent Error Status	When this bit is written to '0', the JTAG error status register will free run. When set to '1' it will hold the most recent error status.
13	Cobra Hold Most Recent Event Status	When this bit is written to '1', Cobra Core hold it most recent internal event status. When set to '0' it will free run.
12	Auto Ack on Hang or Error Detected	When this bit is written to '0', PCORE will not auto ack on hang or error. This may leave the processor in a totally stuck state. However corrupted information may be stopped from entering the processor. When set to '1' and hang or error conditions manifest reads may return garbage and write data may be lost BUT the processor should not be stopped cold.
11	FDMA Virtual Memory Error Normal/Critical Interrupt	When this bit is written to '0', PCORE will treat an IBM3206K0424 virtual memory write error as a critical interrupt. When it is '1', this condition will be treated as a normal interrupt.
10	DCache Virtual Memory Error Normal/Critical Interrupt	When this bit is written to '0', PCORE will treat an IBM3206K0424 virtual memory write error as a critical interrupt. When it is '1', this condition will be treated as a normal interrupt.
9	Memory Lock Normal/Critical Interrupt	When this bit is written to '0', PCORE will treat memory locked as a critical interrupt. When it is '1', this condition will be treated as a normal interrupt.
8	64 Bit DCR Primitives	When set, the three DCR primitives work in 64-bit mode. In this mode, the first access to the primitive register is to the upper 32 bits. The second reference is to the lower 32 bits. The second access triggers the completion of the operation at the destination.
7	Target Access Dead Man Timer Master Disable	When set, this will disable all of the Target Access Dead Man Timers: PCI Master, Control Memory, Packet Memory, IBM3206K0424 Registers and DCR.
6	Serial Port Receive Interrupt Priority	When set, this will cause a the receive interrupt to be a Critical Interrupt. When not set, it is a regular interrupt.

Bit(s)	Name	Description
5	Serial Port Transmit Interrupt Priority	When set, this will cause a the transmit interrupt to be a Critical Interrupt. When not set, it is a regular interrupt.
4	PCI Master Wait on Writes Enable	When set, this will cause the PCI access machine to wait until the data is actually confirmed written to the device. Normally, this bit is set to off since it decreases performance when enabled.
3	Boot Time System Reset Exception	When set, this bit will issue a system boot reset exception to the Cobra Core Processor. It is reset by the Cobra Core Processor after it has received the system reset exception.
2	Disable Lock PCORE on Error	When this bit is set and an error occurs and the corresponding lock enable bit is set PCORE will lock. This state is equivalent to being in diagnostic mode.
1	Cobra Core Run State	When set, this will place the Cobra Core into run state. When not set, the processor will quiesce and hold at idle.
0	Diagnostic/Operational Mode	When set, PCORE is in diagnostic mode. When cleared, PCORE is in operational mode. When in diagnostic mode, state machines are held in idle. If they are already active, when they next go to idle they will hold there.

17.6: PCORE Reset Control Register

The PCORE Reset Control Register provides control information about PCORE reset operations. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

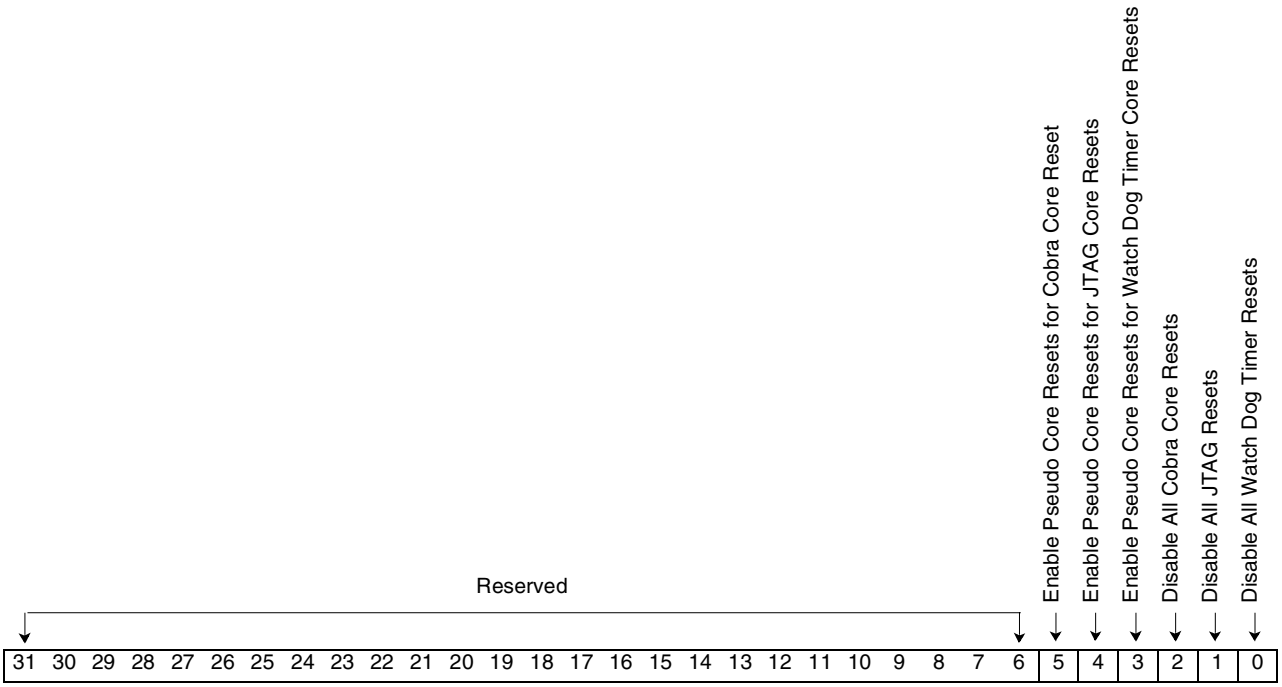
Length 32 bits

Type Clear/Set

Address XXXX 4000 and 004

Power on Reset value X'00 0E 00 09'

Restrictions Caution must be used when asserting some of the bits during operation.



Bit(s)	Name	Description
31-6	Reserved	Reserved
5	Enable Pseudo Core Resets for Cobra Core Reset	When this bit is written to '0', Cobra Core resets are converted to chip resets. When this bit is written '1', a pseudo core reset is issued instead.
4	Enable Pseudo Core Resets for JTAG Core Resets	When this bit is written to '0', Cobra Core resets are converted to chip resets. When this bit is written '1', a pseudo core reset is issued instead.
3	Enable Pseudo Core Resets for Watch Dog Timer Core Resets	When this bit is written to '0', Cobra Core Watch Dog Timer core resets are converted to chip resets. When this bit is written '1', a pseudo core reset is issued instead.
2	Disable All Cobra Core Resets	When this bit is written to '1', all Cobra Core resets are disabled.
1	Disable All JTAG Resets	When this bit is written to '1', JTAG resets are disabled.
0	Disable All Watch Dog Timer Resets	When this bit is written to '1', Watch Dog Timer resets are disabled.

17.7: PCORE Status Register

The PCORE Status Register provides status information about PCORE operations. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

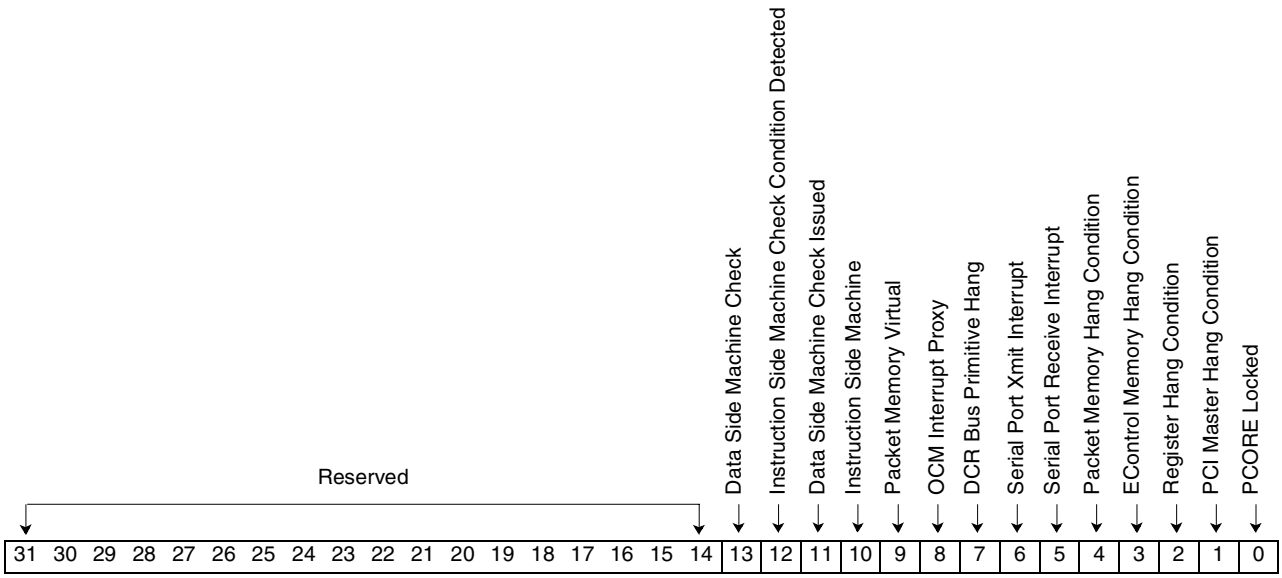
Length 32 bits

Type Clear/Set

Address XXXX 4008 and 00C

Power on Reset value X'00 00 80 00'

Restrictions During normal operations, if a status bit is cleared, it will be reset if the condition that is causing it is still present.



Bit(s)	Name	Description
31-14	Reserved	Reserved
13	Data Side Machine Check Condition Detected	A Data Side Machine Check Condition was detected but not necessarily sent to the Cobra Core.
12	Instruction Side Machine Check Condition Detected	An Instruction Side Machine Check Condition was detected but not necessarily sent to the Cobra Core.
11	Data Side Machine Check Issued	A machine check has been issued to the Cobra Core due to a Data Side PCORE error.
10	Instruction Side Machine Check Issued	A machine check has been issued to the Cobra Core due to an Instruction Side PCORE error.
9	Packet Memory Virtual Write Failure	When this is set, a virtual write has failed to virtual memory. Either a NAK was returned during the write or while holding for error checking after the write. In either case, it indicates the required storage to complete the operation was not available.
8	OCM Interrupt Proxy	When set, OCM is indicating an interrupt condition.
7	DCR Bus Primitive Hang Condition	One of the DCR primitive accesses has timed out.
6	Serial Port Xmit Interrupt	When the serial port surfaces a Xmit Interrupt it will be reflected here.
5	Serial Port Receive Interrupt	When the serial port surfaces a Receive Interrupt, it will be reflected here.

Bit(s)	Name	Description
4	Packet Memory Hang Condition	Packet memory interface Dead Man Timer has expired.
3	Control Memory Hang Condition	Control Memory interface Dead Man Timer has expired.
2	Register Hang Condition	Register interface Dead Man Timer has expired.
1	PCI Master Hang Condition	PCI Master interface Dead Man Timer has expired.
0	PCORE Locked	This bit is set when locking is enabled; an error has occurred and the lock mask bit is set that matches the error.

17.8: PCORE User Status Register

The PCORE User Status Register provides user-defined status information about PCORE software operations. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

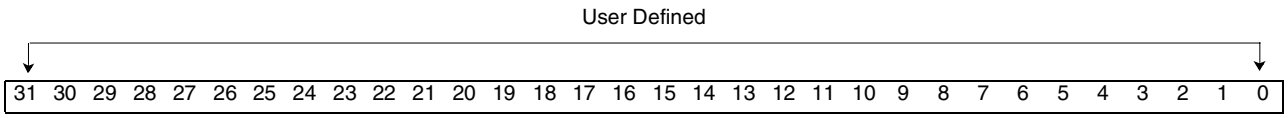
Length 32 bits

Type Clear/Set

DCR Address X'200 and 201'

Power on Reset value X'00 00 00 00'

Restrictions During normal operations, if a status bit is cleared, it will be reset if the condition that is causing it is still present.



Bit(s)	Name	Description
31-0	User defined	Reserved

17.9: PCORE Cobra Core External Status Register

The PCORE Cobra Core External Status Register provides Cobra Core-defined status information about PCORE. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

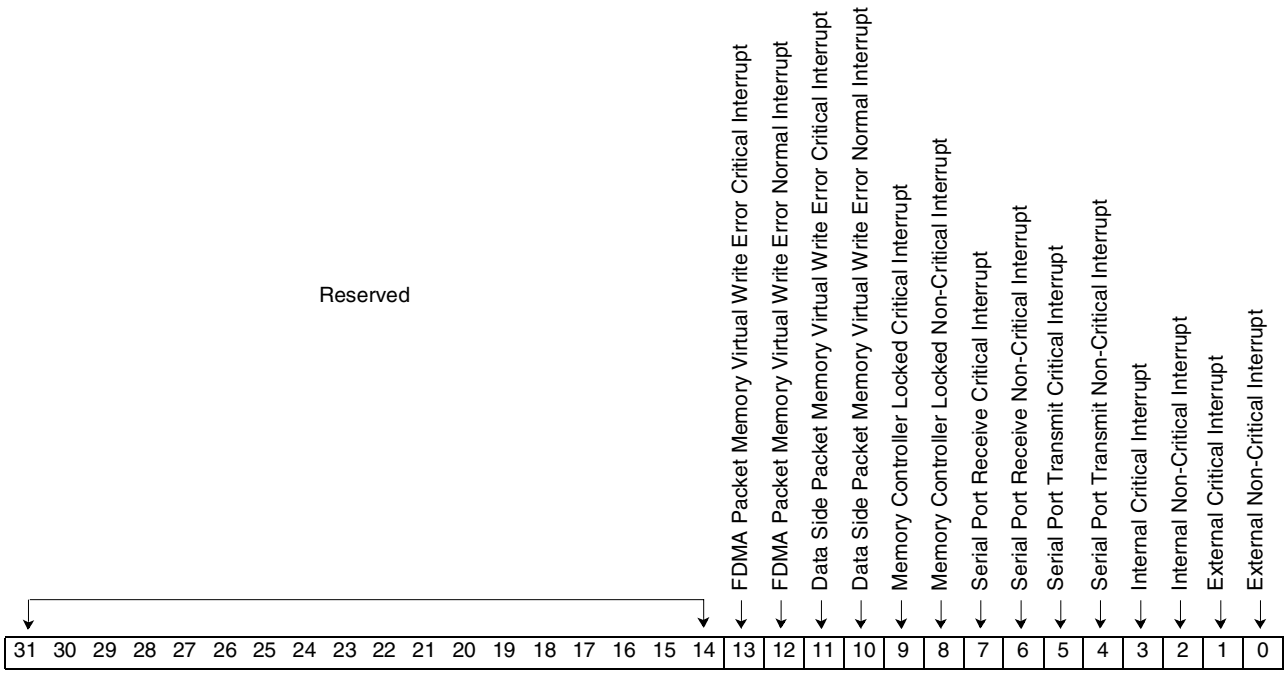
Length 32 bits

Type Clear/Set

DCR Address X'202 and 203'

Power on Reset value X'00 00 00 00'

Restrictions During normal operations, if a status bit is cleared, it will be reset if the condition that is causing it is still present.



Bit(s)	Name	Description
31-14	Reserved	Reserved
13	FDMA Packet Memory Virtual Write Error Critical Interrupt	This occurs when the Packet Memory controller returns an error on a packet virtual memory write and the FDMA-side is accessing and this condition is set as critical.
12	FDMA Packet Memory Virtual Write Error Normal Interrupt	This occurs when the Packet Memory controller returns an error on a packet virtual memory write and the FDMA-side is accessing this condition is set as normal.
11	Data Side Packet Memory Virtual Write Error Critical Interrupt	This occurs when the Packet Memory controller returns an error on a packet virtual memory write and the D-side is accessing and this condition is set as critical.
10	Data Side Packet Memory Virtual Write Error Normal Interrupt	This occurs when the Packet Memory controller returns an error on a packet virtual memory write and the D-side is accessing this condition is set as normal.
9	Memory Controller Locked Critical Interrupt	This occurs when the memory controller is locked and this condition is set as critical.
8	Memory Controller Locked Non-Critical Interrupt	This occurs when the memory controller is locked and this condition is set as non-critical.

Bit(s)	Name	Description
7	Serial Port Receive Critical Interrupt	This occurs when the serial controller has a transmit interrupt and the corresponding critical interrupt enable is on in the control register.
6	Serial Port Receive Non-Critical Interrupt	This occurs when the serial controller has a transmit interrupt and the corresponding critical interrupt enable is on in the control register.
5	Serial Port Transmit Critical Interrupt	This occurs when the serial controller has a transmit interrupt and the corresponding critical interrupt enable is on in the control register.
4	Serial Port Transmit Non-Critical Interrupt	This occurs when the serial controller has a transmit interrupt and the corresponding critical interrupt enable is on in the control register.
3	Internal Critical Interrupt	This occurs when a bit in the IBM3206K0424 primary status register is set and the corresponding critical interrupt enable is on.
2	Internal Non-Critical Interrupt	This occurs when a bit in the IBM3206K0424 primary status register is set and the corresponding non-critical interrupt enable is on.
1	External Critical Interrupt	This occurs when an off chip interrupt is received and the non-critical enable for off chip interrupts is set.
0	External Non-Critical Interrupt	This occurs when an off chip interrupt is received and the non-critical enable for off chip interrupts is set.

17.10: PCORE Cobra Core External Machine Check Status Register

The PCORE Cobra Core External Machine Check Status Register provides Cobra Core machine check status information about PCORE. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

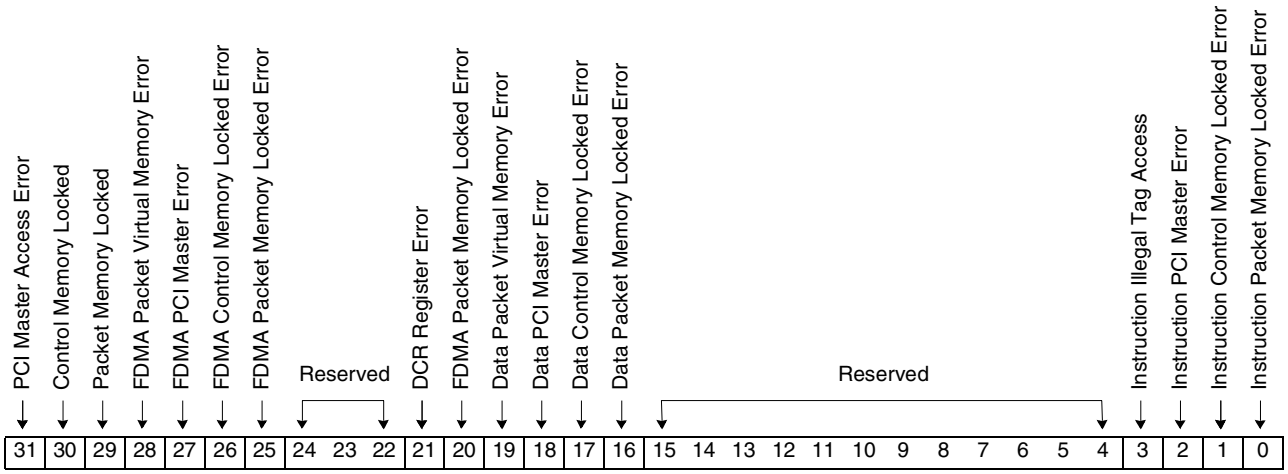
Length 32 bits

Type Clear/Set

DCR Address X'25C and 25D'

Power on Reset value X'00 00 00 00'

Restrictions During normal operations, if a status bit is cleared, it will be reset if the condition that is causing it is still present.



Bit(s)	Name	Description
31	PCI Master Access Error	This occurs when the PCI Master Machine returns an error and no requestor currently owns the machine.
30	Control Memory Locked	This occurs when Packet Memory is in a locked state.
29	Packet Memory Locked	This occurs when Packet Memory is in a locked state.
28	FDMA Packet Virtual Memory Error	This occurs when Packet Memory indicates a write error during access of a virtual buffer.
27	FDMA PCI Master Error	This occurs when a PCI master access has an error returned while the data path is accessing it.
26	FDMA Control Memory Locked Error	This occurs when Control Memory locks while the data path is actively accessing it.
25	FDMA Packet Memory Locked Error	This occurs when Packet Memory locks while the data path is actively accessing it.
25-21	Reserved	Reserved.
21	DCR Register Error	This occurs when a fatal error, typically a hang, occurs on a DCR register timeout.
20	FDMA Packet Memory Locked Error	This occurs when Packet Memory locks while the data path is actively accessing it.
20	Data Register Error	This occurs when a fatal error, typically a hang, occurs on an IBM3206K0424register timeout.

Bit(s)	Name	Description
19	Data Packet Virtual Memory Error	This occurs when Packet Memory indicates a write error during access of a virtual buffer.
18	Data PCI Master Error	This occurs when a PCI master access has an error returned while the data path is accessing it.
17	Data Control Memory Locked Error	This occurs when Control Memory locks while the data path is actively accessing it.
16	Data Packet Memory Locked Error	This occurs when Packet Memory locks while the data path is actively accessing it.
15-4	Reserved	Reserved.
3	Instruction Illegal Tag Access	This occurs when the Instruction side specifies the register space or OCM for the instruction source via the Icache. Since there are no physical connections, these accesses can not occur.
2	Instruction PCI Master Error	This occurs when PCI master access has an error returned while the instruction path is accessing it.
1	Instruction Control Memory Locked Error	This occurs when Control memory locks while the instruction path is actively accessing it.
0	Instruction Packet Memory Locked Error	This occurs when Packet Memory locks while the instruction path is actively accessing it.

17.11: PCORE JTAG Debug Control Register

The PCORE JTAG Debug Control Register enables the JTAG port or a PCI interface processor debugger to control the processor core.

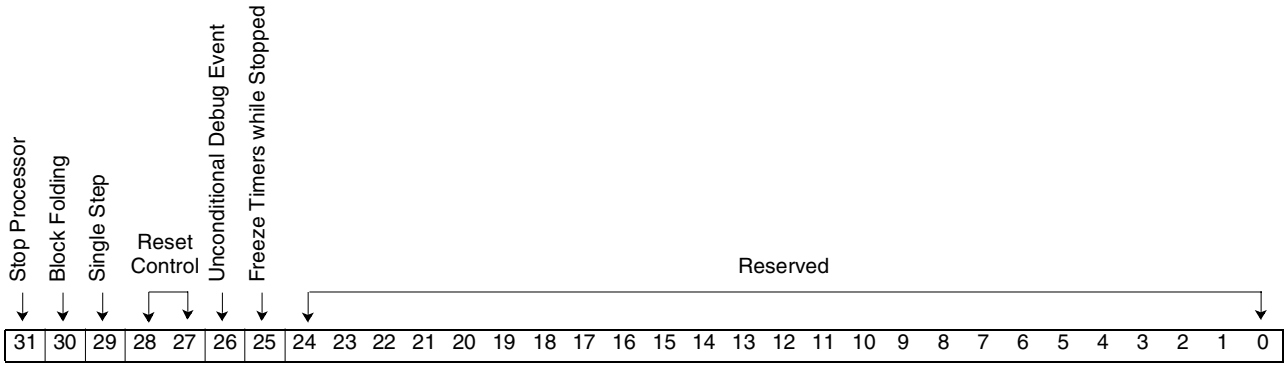
Length 32 bits

Type Read/Write

Address XXXX 4200

Power on Reset value X'00 00 00 00'

Restrictions None



Bit(s)	Name	Description
31	Stop Processor	This bit forces the processor to halt execution.
30	Block Folding	The processor normally dispatches two instructions at a time. Setting this bit forces instruction dispatches to be serialized.
29	Single Step	Setting this bit when the processor is stopped causes the processor to execute one or two instructions depending on the value of bit 30. This bit automatically clears itself after one cycle.
28-27	Reset Control	These bits potentially generate one of three resets depending on their value. They automatically reset to B'00' after one cycle. The bits decode as follows: '00' No reset '01' Core reset '10' Chip reset '11' System reset
26	Unconditional Debug Event	This bit generates an interrupt to the processor. It automatically resets to B'0' after one cycle.
25	Freeze Timers while Stopped	This bit freezes the state of all timers in the core if the processor is stopped.
24-0	Reserved	Reserved.

17.12: PCORE JTAG Debug Status Register

The PCORE JTAG Debug Status Register returns core status.

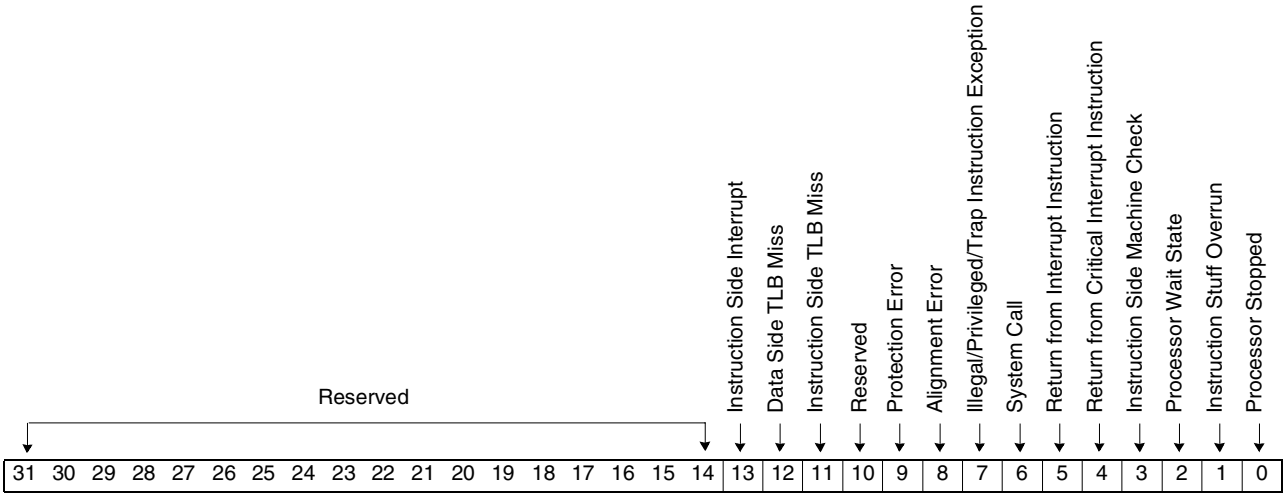
Length 32 bits

Type Read Only

Address XXXX 4204

Power on Reset value X'00 00 00 00'

Restrictions None



Bit(s)	Name	Description
31-14	Reserved	Reserved.
13	Instruction Side Interrupt	
12	Data Side TLB Miss	
11	Instruction Side TLB Miss	
10	Reserved	Reserved.
9	Protection Error	
8	Alignment Error	
7	Illegal/Privileged/Trap Instruction Exception	
6	System Call	
5	Return from Interrupt Instruction	
4	Return from Critical Interrupt Instruction	
3	Instruction Side Machine Check	
2	Processor Wait State	The processor is in a wait state.
1	Instruction Stuff Overrun	The debug port has attempted to insert an instruction into the processor via the JTAG Instruction Stuff Buffer (JISB) and it was not accepted by the processor. The instruction must be reloaded into the JISB.
0	Processor Stopped	

17.13: PCORE JTAG Instruction Stuff Buffer

The PCORE JTAG Instruction Stuff Buffer is used to insert an instruction into the processor for execution.

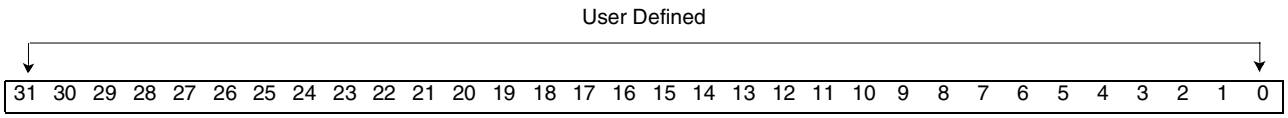
Length 32 bits

Type Read/Write

Address XXXX 4208

Power on Reset value X'00 00 00 00'

Restrictions None



Bit(s)	Name	Description
31-0	Instruction	Instruction to be passed from the JTAG/PCI debug interface to the core GPRs.

17.14: PCORE JTAG Debug Data Register

The PCORE JTAG Debug Data Register enables passing data between the JTAG/PCI debug port and the general purpose registers of the processor core.

Length 32 bits

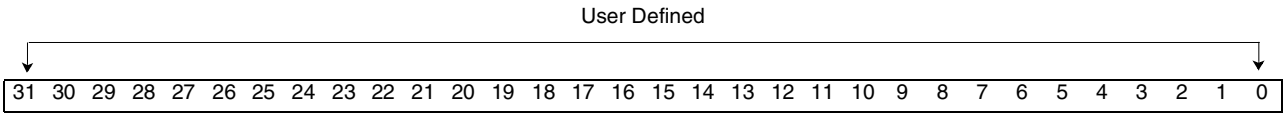
Type Read/Write

Address XXX 420C

DCR Address TBD

Power on Reset value X'00 00 00 00'

Restrictions None



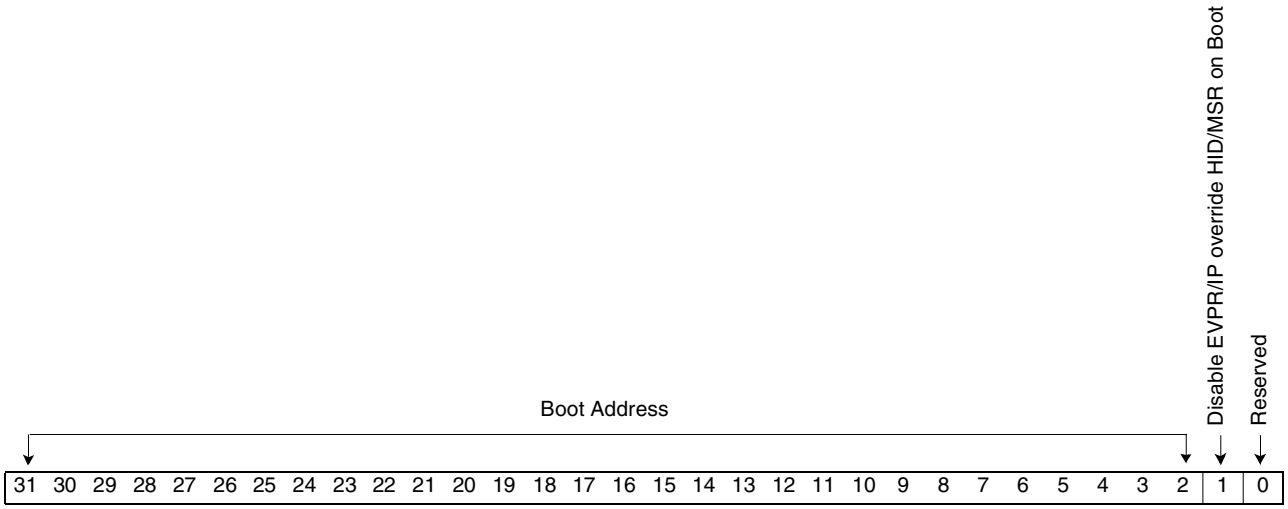
Bit(s)	Name	Description
31-0	Data	Data to be passed to/from the JTAG/PCI debug interface to the core GPRs.

17.15: PCORE Cobra Core Boot Address

The PCORE Cobra Core Boot Address provides Cobra Core its boot time address.

Length	32 bits
Type	Read/Write
DCR Address	XXXX 4018
Power on Reset value	X'FF F0 01 00'
Restrictions	None

This is the PCORE register. It is used to provide the address that is used to fetch the first instruction.



Bit(s)	Name	Description
31-2	Boot Address	This is the address the processor will use at boot time.
1	Disable EVPR/IP override HID/MSR on Boot	Override MSR settings on System Reset Exception to use all of the PCORE Cobra Core Boot Address.
0	Reserved	Reserved.

17.16: PCORE Cobra Core Access Priority Control Register

The PCORE Cobra Core Access Priority Control Register provides Cobra Core defined status information about PCORE. It is used to control the order and priority of access to the various memory subsystems that the Cobra Core has access to. It is to be set up at initialization time an not dynamically changed.

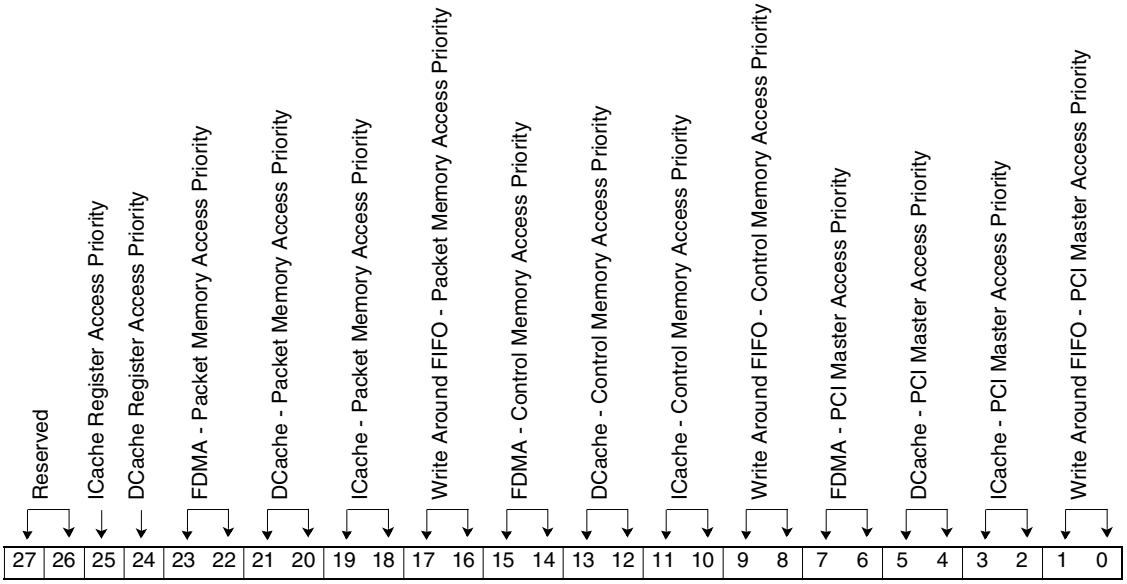
Length 28 bits

Type Read/Write

DCR Address XXXX 401C

Power on Reset value X'01 E4 E4 E4'

Restrictions For each controller accessed, the values cannot be duplicated. That is, all the priorities to a given target subsystem must be unique.



Bit(s)	Name	Description
27-26	Reserved	Reserved
25	ICache Register Access Priority	This sets the priority into the IBM3206K0424 register port. Zero is low priority; one is high priority.
24	DCache Register Access Priority	This sets the priority into the IBM3206K0424 register port. Zero is low priority; one is high priority.
23-22	FDMA - Packet Memory Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
21-20	DCache - Packet Memory Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
19-18	ICache - Packet Memory Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
17-16	Write Around FIFO - Packet Memory Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
15-14	FDMA - Control Memory Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.

Bit(s)	Name	Description
13-12	DCache - Control Memory Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
11-10	ICache - Control Memory Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
9-8	Write Around FIFO - Control Memory Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
7-6	FDMA - PCI Master Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
5-4	DCache - PCI Master Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
3-2	ICache - PCI Master Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority.
1-0	Write Around FIFO - PCI Master Access Priority	This sets the priority of access to Packet Memory. Zero is the lowest priority; three is the highest priority

17.17: PCORE Transaction Dead Man Timer Value Registers

These registers are used to load timers that count to zero from the value loaded in this register. The maximum wait for an I/O transaction is about 2ms when this is set to X'FFFF'. The value of this register is written into the corresponding timer after the transaction is initiated with the target. It continues to then count down until the target responds or zero is reached in the timer. When the timer reaches zero, a status bit is set and action can be taken from there.

Length	16 bits	
Type	Read/Write	
Address	PCI Master Dead Man Timer Value	XXXX 4020
	IBM3206K0424 Register Dead Man Timer Value	XXXX 4024
	Control Memory Dead Man Timer Value	XXXX 4028
	Packet Memory Dead Man Timer Value	XXXX 402C
	DCR Primitive Access Dead Man Timer Value	XXXX 40F8
Power on Reset value	X'FFFF'	
Restrictions	None	

17.18: PCORE High Priority Access Timer Value Registers

These registers are used to load timers that count to zero from the value loaded in this register. The maximum wait for an I/O transaction is about 2ms when this is set to X'FFFF'. The value of this register is written into the corresponding timer after the transaction is initiated with the target. It will continue to then count down until the target responds or zero is reached in the timer. When the timer reaches zero, a status bit is set and action can be taken from there.

Length	16 bits	
Type	Read/Write	
Address	Control Memory Dead Man Timer Value	TBD
	Packet Memory Dead Man Timer Value	TBS
Power on Reset value	X'FFFF'	
Restrictions	None	

17.19: PCORE Transaction Dead Man Timer Register

These timers are used to time transactions that are valid but the target does not respond right away. The timer counts on a 7.5 ns time base. The maximum wait for an I/O transaction is about 2ms when the timer counts down from X'FFFF'. This timer counts down to zero from the values set in the value register. When zero is reached, the transaction is considered broken and the request will be acknowledged back to the requestor.

Length	16 bits	
Type	Read	
Address	PCI Master Dead Man Timer Value	XXXX 40E0
	IBM3206K0424 Register Dead Man Timer Value	XXXX 40E4
	Control Memory Dead Man Timer Value	XXXX 40E8
	Packet Memory Dead Man Timer Value	XXXX 40EC
	DCR Primitive Access Dead Man Timer Value	XXXX 40F0
Power on Reset value	X'FFFF'	
Restrictions	None	

17.20: PCORE IBM3206K0424 Shadow Status Register

This register is used to shadow the INTST Interrupt Source. The purpose of this register is to allow polling for IBM3206K0424 interrupts without having to use the PCI bus.

Length	32 bits
Type	Read
DCR Address	X'208'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.21: PCORE IBM3206K0424 Packet Last Write with Error Address

This register is used to shadow the INTST Interrupt Source. The purpose of this register is to store the address associated with the previous virtual write error.

Length	32 bits
Type	Read
Address	X'260'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.22: PCORE IBM3206K0424 RXQUE Master Status Register

This register is used to shadow the RXQUE Master Status Register. The purpose of this register is to allow fast access to RXQUE's Master Status Register without having to use the regular register interface.

Length	32 bits
Type	Read
DCR Address	X'20F'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.23: PCORE IBM3206K0424 RXQUE Enabled Status Register 1

This register is used to shadow the RXQUE Enabled Status Register 1. The purpose of this register is to allow fast read access of the RXQUE Enabled Status Register 1 without having to use the normal IBM3206K0424 register interface.

Length	32 bits
Type	Read
DCR Address	X'250'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.24: PCORE IBM3206K0424 RXQUE Enabled Status Register 2

This register is used to shadow the RXQUE Enabled Status Register 1. The purpose of this register is to allow fast read access of the RXQUE Enabled Status Register 2 without having to use the normal IBM3206K0424 register interface.

Length	32 bits
Type	Read
Address	X'251'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.25: PCORE IBM3206K0424 RXQUE Upper Queues Status Register

This register is used to shadow the RXQUE Upper Queues Status Register. The purpose of this register is to allow fast read access of the RXQUE Upper Queues Status Register without having to use the normal IBM3206K0424 register interface.

Length	32 bits
Type	Read
DCR Address	X'252'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.26: PCORE IBM3206K0424 RXQUE Lower Queues Status Register

This register is used to shadow the RXQUE Lower Queues Status Register. The purpose of this register is to allow fast read access of the RXQUE Lower Queues Status Register without having to use the normal IBM3206K0424 register interface.

Length	32 bits
Type	Read
DCR Address	X'253'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.27: PCORE DMAQS Master Status Register

This register is used to shadow the DMAQS Master Status Register. The purpose of this register is to allow fast read access of the DMAQS Master Status Register without having to use the normal IBM3206K0424 register interface.

Length	32 bits
Type	Read
Address	X'257'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.28: PCORE DMAQS Enabled Status Register

This register is used to shadow the DMAQS Master Status Register. The purpose of this register is to allow fast read access of the DMAQS Master Status Register without having to use the normal IBM3206K0424 register interface.

Length	32 bits
Type	Read
Address	X'25B'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.29: PCORE RXQUE Queue Length Registers

The PCORE RXQUE Queue Length Registers provide event enqueue queue lengths to the Cobra Core. Reads from this address will return event queue lengths from RXQUE.

Length	32 bits
Type	Read/Write
Address	X'220' - X'22F'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.30: PCORE DMAQS Queue Length Registers

The PCORE DMAQS Queue Length Registers provide DMAQS queue lengths to the Cobra Core. Reads from this address will return DMAQS queue lengths from DMAQS.

Length	32 bits
Type	Read/Write
Address	X'254' - X'256'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.31: PCORE Interrupt Enable Register

This register is used to enable bits from the PCORE Status Register and potentially generate interrupts to the control processor. When both a bit in this register and the corresponding bit(s) in the PCORE Status Register are set, the PCORE interrupt to PCINT will be enabled. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See *PCORE Status Register* on page 467 for the bit descriptions.

Length	32 bits
Type	Clear/Set
Address	XXXX 4010 and 014
Power on Reset value	X'00 00 80 00'
Restrictions	None

17.32: PCORE User Interrupt Enable

This register is used to enable an interrupt based on bits from the corresponding PCORE User Status Register and potentially generate interrupts to the control processor. When both a bit in this register and the corresponding bit(s) in the [TBD] register are set, the PCORE status bit(s) will be set in the corresponding PCORE User Status Register. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See *PCORE User Status Register* on page 468 for the bit descriptions.

Length	32 bits
Type	Clear/Set
Address	X'204 and 205'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.33: PCORE Cobra Core Interrupt Enable Register

This register is used to enable bits from the PCORE Cobra Core External Status Register and generate interrupts to the Cobra Core processor. When both a bit in this register and the corresponding bit(s) in the PCORE Cobra Core External Status Register are set, the Cobra Core interrupt to the Cobra Core core will be enabled. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See *PCORE Status Register* on page 467 for the bit descriptions.

Length	32 bits
Type	Clear/Set
Address	X'206' - X'207'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.34: PCORE Cobra Core External Machine Check Enable Register

This register is used to enable bits from the PCORE Cobra Core External Machine Check Status Register and generate machine checks to the Cobra Core processor. When both a bit in this register and the corresponding bit(s) in the PCORE Cobra Core External Machine Check Status Register are set, the requisite Cobra Core Machine Check to the Cobra Core core will be enabled. See *Note on Set/Clear Type Registers on page 93* for more details on addressing. See *PCORE Status Register* on page 467 for the bit descriptions.

Length	32 bits
Type	Clear/Set
Address	X'25E' - X'25F'
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.35: PCORE Error Lock Enable Register

The PCORE Error Lock Enable Register provides the ability to halt PCORE when the corresponding status bit in the status register is set and locking is enabled. When a bit in this register corresponds to a bit that is set in the status register, the state machines in PCORE will be held in idle state until the lock is disabled.

See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	32 bits
Type	Clear/Set
Address	XXXX 4030 and 034
Power on Reset value	X'00 00 FE 9F'
Restrictions	None

17.36: PCORE User Error Lock Enable Register

The PCORE User Error Lock Enable Register provides the ability to halt PCORE when the corresponding status bit in the User Status Register is set and locking is enabled. When a bit in this register corresponds to a bit that is set in the Status Register, the state machines in PCORE will be held in idle state until the lock is disabled.

See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length 32 bits

Type Clear/Set

Address XXXX 4038 and 03C

Power on Reset value X’FF FF FF FF’

Restrictions None

17.37: PCORE RXQUE Event Interface Enqueue Register

The PCORE RXQUE Event Interface Enqueue Register provides event enqueue interface for the Cobra Core. Writes to this address will enqueue an event to an RXQUE queue.

Length 32 bits

Type Read/Write

Address X’230 - X’23F’

Power on Reset value X'00 00 00 00'

Restrictions None

Bit(s)	Name	Description
31-7	Event Data	User Defined Event Data
6-0	Event Signature	TBD Processor Event Signature

17.38: PCORE DMAQS DMA Enqueue Register

The PCORE DMAQS DMA Enqueue Register provides DMAQS enqueue interface for the Cobra Core. Writes to this address will enqueue an event to an RXQUE queue.

Length 32 bits

Type Read/Write

Address X’258 - X’25A’

Power on Reset value X'00 00 00 00'

Restrictions None

17.39: PCORE RXQUE Event Interface Deque Register

The PCORE RXQUE Event Interface Deque Register provides event deque interface for the Cobra Core. Reads from this address return an event.

Length	32 bits
Type	Read
Address	X’240 - X’24F’
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.40: PCORE Cobra SPR Read Data Access Register

The PCORE Cobra SPR Read Data Access Register stores the data from the requested Cobra facility on a read. These are message passing facilities. They are used for inter-device communication.

These facilities, with their control register bits, allow for either interrupt or polling-based message passing from the Cobra Core to a PCI bus device.

Length	32 bits
Type	Read Only
Address	XXXX 4040
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.41: PCORE Cobra SPR Write Data Access Register

This register stores the data from the PCIrequested Cobra facility on a read. These are message passing facilities. They are used for inter-device communication.

These facilities with their control register bits, allow for either interrupt or polling-based message passing from the Cobra Core to a PCI bus device.

Length	32 bits
Type	Write Only
Address	XXXX 40F4
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.42: PCORE Cobra SPR Access Address Register

This is the PCORE SPR Access Address Register. It is used to access the internal facilities in Cobra. This includes SPR/DCR and Debug facilities. The address is for a facility and represents a four-byte access.

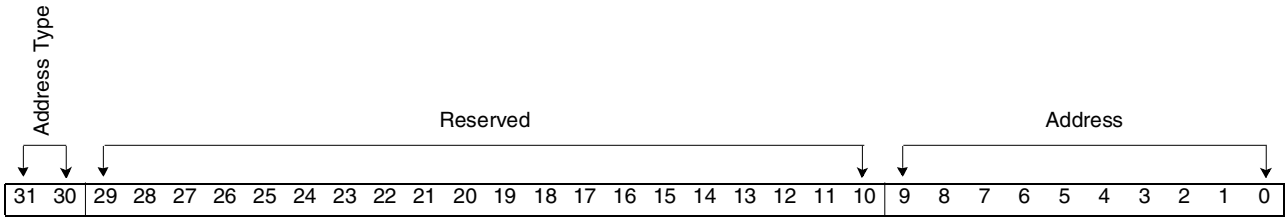
Length 32 bits

Type Read/Write

Address XXXX 4044

Power on Reset value X'80 00 00 00'

Restrictions None



Bit(s)	Name	Description
31-30	Address Type	These bits describe the address type. '00' Reserved '01' SPR Access '10' DCR Access '11' Register/Debug Access
29-10	Reserved	Reserved
9-0	Address	Address of Target Register.

17.43: PCORE Address Translation Offset Address Facilities

The PCORE Address Translation Offset Address Facilities provides the offset that is added to the Cobra Real Address to create the target subsystem address.

When an address is issued from the Cobra Core core it is accompanied by four target translation bits. The translation bits indicate which translation facility is to be used to translate the processor "real" physical address into a target system actual address. This grouping provides for the offset addresses for each target memory system. The offset is added to the Cobra Real Address to create the target system address. The following is a list of targets, each with their own translation facilities.

’0000’	OCM (Translation provided for in the MMUs)
’0001’	Packet Memory View 0
’0010’	Packet Memory View 1
’0011’	Packet Memory View 2
’0100’	IBM3206K0424 Registers
’0101’	Control Memory View 0
’0110’	Control Memory View 1
’0111’	Control Memory View 2
’1000’	PCI Master Access (Non IBM3206K0424) View 0
’1001’	PCI Master Access (Non IBM3206K0424) View 1
’1010’	PCI Master Access (Non IBM3206K0424) View 2
’1011’	PCI Master Access (Non IBM3206K0424) View 3
’1100’	Control/Packet View 0
’1101’	Control/Packet View 1
’1110’	Control/Packet View 2
’1111’	Control/Packet View 3

PCORE Address Translation Offset Address Facilities:

Length	32 bits	
Type	Read/Write	
Address	Packet Memory Offset View 0	XXXX 4048
	Packet Memory Offset View 1	XXXX 404C
	Packet Memory Offset View 2	XXXX 4050
	IBM3206K0424 Registers Offset View 0	XXXX 4054
	Control Memory Offset View 0	XXXX 4058
	Control Memory Offset View 1	XXXX 405C
	Control Memory Offset View 2	XXXX 4060
	PCI Master Offset View 0	XXXX 4064
	PCI Master Offset View 1	XXXX 4068

	PCI Master Offset View 2	XXXX 406C
	®pratbAa./Pci Master Offset View 3	
	®pratbBa./Control/Packet Memory Offset View 0	
	®pratbCa./Control/Packet Memory Offset View 1	
	®pratbDa./Control/Packet Memory Offset View 2	
	®pratbEa./Control/Packet Memory Offset View 3	
Power on Reset value	X'00000000'	
Restrictions	None	

17.44: PCORE PCI 64 Bit Address Translation Facilities

The PCORE PCI 64 Bit Address Translation Facilities provide the upper thirty-two bits of address in 64-bit addressing mode. When an access is issued to the PCI Master Interface in 64-bit addressing mode, these registers are used to create the upper 32 bits of the 64-bit address.

Length	32 bits	
Type	Read/Write	
Address/Storage Unit	Upper 32 Address Bits PCI Master View 0	XXXX 4084
	Upper 32 Address Bits PCI Master View 1	XXXX 4088
	Upper 32 Address Bits PCI Master View 2	XXXX 408C
	Upper 32 Address Bits PCI Master View 3	XXXX 4090
Power on Reset value	X'00000000'	
Restrictions	None	

17.45: PCORE PCI Master Target Tag Controls

The PCORE PCI Master Target Tag Controls contains the control for each PCI Tag/View. This register contains bits for each of the four PCI Master Views.

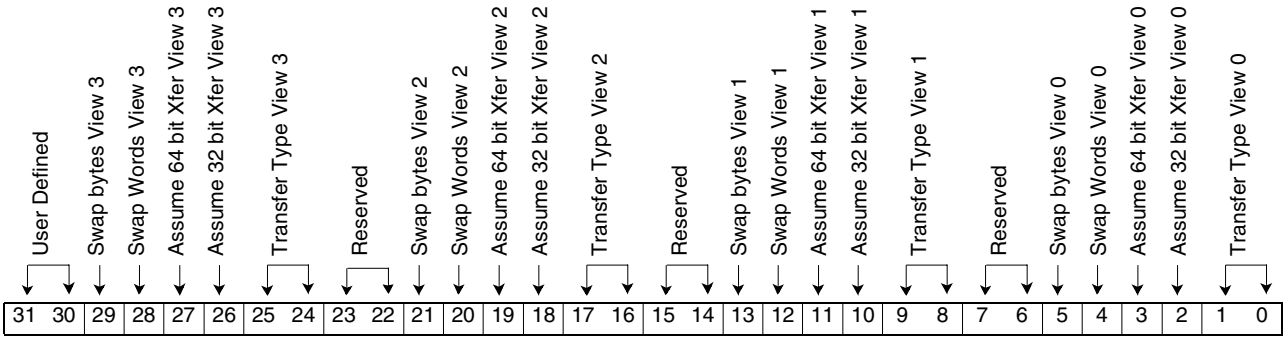
Length 32 bits

Type Read/Write

DCR Address XXXX 40FC

Power on Reset value X'06 06 06 06'

Restrictions None



Bit(s)	Name	Description
31-30	Reserved	Reserved
29	Swap bytes View 3	When set, this bit tells the PCI Master Logic to do byte swapping.
28	Swap Words View 3	When set, this bit tells the PCI Master to do word swapping.
27	Assume 64 bit Xfer View 3	When set, this bit tells the PCI Master Logic to assume a 64-bit data access.
26	Assume 32 bit Xfer View 3	When set, this bit tells the PCI Master Logic to assume a 32-bit data access.
25-24	Transfer Type View 3	These bits indicate the transaction type. '00' Config Cycle '01' I/O Cycle '1-' Memory Cycle
23-22	Reserved	Reserved
21	Swap bytes View 2	When set, this bit tells the PCI Master Logic to do byte swapping.
20	Swap Words View 2	When set, this bit tells the PCI Master to do word swapping.
19	Assume 64 bit Xfer View 2	When set, this bit tells the PCI Master Logic to assume a 64-bit data access.
18	Assume 32 bit Xfer View 2	When set, this bit tells the PCI Master Logic to assume a 32-bit data access.
17-16	Transfer Type View 2	These bits indicate the transaction type. '00' Config Cycle '01' I/O Cycle '1-' Memory Cycle
15-14	Reserved	Reserved
13	Swap bytes View 1	When set, this bit tells the PCI Master Logic to do byte swapping.
12	Swap Words View 1	When set, this bit tells the PCI Master to do word swapping.
11	Assume 64 bit Xfer View 1	When set, this bit tells the PCI Master Logic to assume a 64 bit data access.

Bit(s)	Name	Description
10	Assume 32 bit Xfer View 1	When set, this bit tells the PCI Master Logic to assume a 32 bit data access
9-8	Transfer Type View 1	These bits indicate the transaction type. '00' Config Cycle '01' I/O Cycle '1-' Memory Cycle
7-6	Reserved	Reserved
5	Swap bytes View 0	When set, this bit tells the PCI Master Logic to do byte swapping.
4	Swap Words View 0	When set, this bit tells the PCI Master to do word swapping.
3	Assume 64 bit Xfer View 0	When set, this bit tells the PCI Master Logic to assume a 64 bit data access.
2	Assume 32 bit Xfer View 0	When set, this bit tells the PCI Master Logic to assume a 32 bit data access.
1-0	Transfer Type View 0	These bits indicate the transaction type '00' Config Cycle '01' I/O Cycle '1-' Memory Cycle

17.46: PCORE Last Packet Address Register

The PCORE Last Packet Address Register is the last address to the Packet Memory bus at the time of the hang condition. When the system locks up, this register holds the last Packet Memory address that was or is currently being presented to the Packet Memory subsystem.

Length	32 bits
Type	Read
Address	XXXX 4094
Power on Reset value	X'FF FF FF FC'
Restrictions	None

17.47: PCORE Last Control Address Register

The PCORE Last Control Address Register is the last address to the Control Memory bus at the time of the hang condition. When the system locks up, this register holds the last Control Memory address that was or is currently being presented to the Control Memory subsystem.

Length	32 bits
Type	Read
Address	XXXX 4098
Power on Reset value	X'FF FF FF FC'
Restrictions	None

17.48: PCORE Last PCI Lower Address Register

The PCORE Last PCI Lower Address Register is the last address to the PCI bus at the time of the hang condition. When the system locks up, this register holds the last PCI bus address that was or is currently being presented to the Control Memory subsystem.

Length	32 bits
Type	Read
Address	XXXX 409C
Power on Reset value	X'FF FF FF FC'
Restrictions	None

17.49: PCORE Last Register Address Register

The PCORE Last Register Address Register is the last address to the PCI bus at the time of the Hang Condition. When the system locks up, this register holds the last PCI bus address that was or is currently being presented to the Control Memory subsystem.

Length	32 bits
Type	Read
Address	XXXX 40A0
Power on Reset value	X'FF FF FF FC'
Restrictions	None

17.50: PCORE SRAM Base Address

The SRAM Base Address register is used to select the base address of the 4K byte window to access the SRAM.

Length	32 bits (17:12) Active
Type	Read/Write
Address	XXXX 40A4
Power on Reset value	X'00 00 00 00'
Restrictions	None

17.51: PCORE Read Data Transfer Buffers

The PCORE Read Data Transfer Buffers hold the read data that is being transferred from one of the target subsystems and the Cobra Core. Eight bytes are buffered on the interfaces except for the IBM3206K0424 register interface which buffers four bytes.

Length	32 bits	
Type	Read	
Address	PCI Upper Read Data Transfer Buffer	XXXX 40A8
	PCI Lower Read Data Transfer Buffer	XXXX 40AC
	Packet Upper Read Data Transfer Buffer	XXXX 40B0
	Packet Lower Read Data Transfer Buffer	XXXX 40B4
	Control Upper Read Data Transfer Buffer	XXXX 40B8
	Control Lower Read Data Transfer Buffer	XXXX 40BC
	IBM3206K0424 Register Space Read Data Transfer Buffer	XXXX 40C0
Power on Reset value	X'00000000'	
Restrictions	None	

17.52: PCORE Write Data Transfer Buffers

The PCORE Write Data Transfer Buffers hold the data that is being transferred between the Cobra Core and one of the target subsystems. Eight bytes can be stored for each target subsystem, with the exception of the IBM3206K0424 Register Target which holds just four bytes.

Length	32 bits	
Type	Read	
Address	PCI Upper Write Data Transfer Buffer	XXXX 40C4
	PCI Lower Write Data Transfer Buffer	XXXX 40C8
	Packet Upper Write Data Transfer Buffer	XXXX 40CC
	Packet Lower Write Data Transfer Buffer	XXXX 40D0
	Control Upper Write Data Transfer Buffer	XXXX 40D4
	Control Lower Write Data Transfer Buffer	XXXX 40D8
	IBM3206K0424 Register Space Write Data Transfer Buffer	XXXX 40DC
Power on Reset value	X'00000000'	
Restrictions	None	

17.53: PCORE Polling Register

The PCORE Polling Register provides status information to PCORE about IBM3206K0424 operations. It allows PCORE to poll specific IBM3206K0424 status without using PCI bus bandwidth.

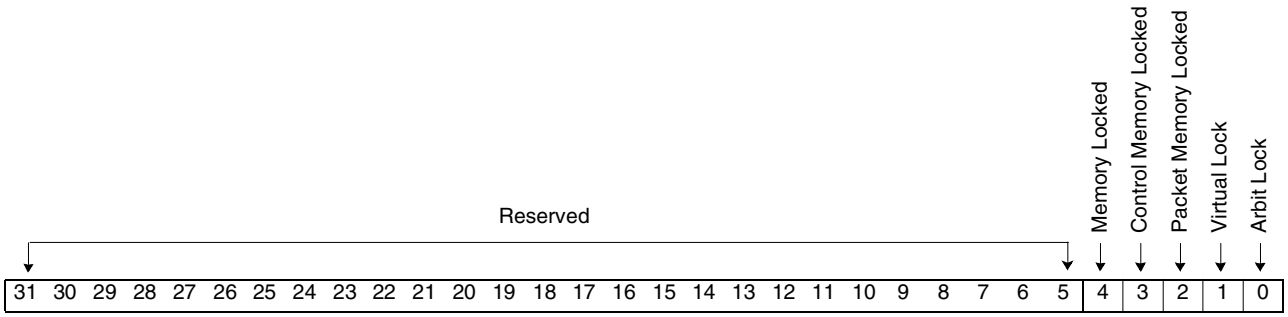
Length 32 bits

Type Read

DCR Address X'20E'

Power on Reset value X'00 00 00 00'

Restrictions During normal operations, if a status bit is cleared, it will be reset if the condition that is causing it is still present.



Bit(s)	Name	Description
31-5	Reserved	Reserved.
4	Memory Locked	Memory is locked.
3	Control Memory Locked	Control Memory is locked.
2	Packet Memory Locked	Packet Memory is locked.
1	Virtual Lock	VIMEM is the locker of memory.
0	Arbit Lock	Arbit is the locker of memory.

17.54: PCORE Integer Input Rate Conversion Register

This register is the integer input port for the rate conversion logic. An integer rate is placed in this register. The on board logic converts it to an ABR rate format.

Length 32 bits

Type Read/Write

DCR Address X'20B'

Power on Reset value X'00 00 00 00'

Restrictions None

17.55: PCORE ABR Output Rate Register

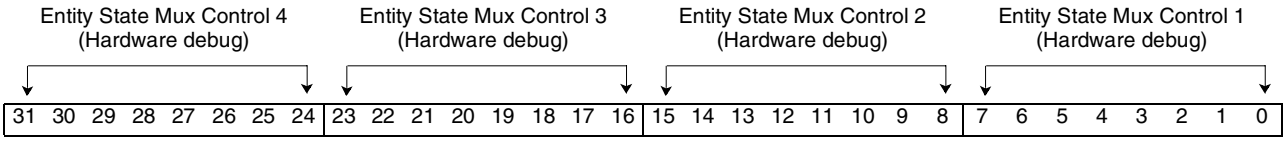
This register is the output port of the rate conversion logic. An integer rate was placed in the Integer Input Register. The logic converts it to an ABR rate and places the result in this register.

Length 16 bits
Type Read
DCR Address X'20C'
Power on Reset value X'00 00'
Restrictions None

17.56: PCORE Debug States Control

This register serves as the PCORE control for external debug states. The INTST Debug states control for the address range desired must be set to select these PCORE state bits. If that is done, then this register acts to select the four ranges. See bit descriptions below.

Length 32 bits
Type Read/Write
Address XXXX 431C
Power on Reset value X'0000 0000'
Restrictions None



Bit(s)	Name	Description
31-24	Entity State Mux Control 4 (Hardware debug)	Select of these bits allows internal state machines, counters, etc., to show up on chip outputs ENSTATE(63 down to 48). Selection encoding is the same as mux 1 control.
23-16	Entity State Mux Control 3 (Hardware debug)	Select of these bits allow internal state machines, counters, etc., to show up on chip outputs ENSTATE(47 down to 32). Selection encoding is the same as mux 1 control.
15-8	Entity State Mux Control 2 (Hardware debug)	Select of these bits allow internal state machines, counters, etc., to show up on chip outputs ENSTATE(31 down to 16). Selection encoding is the same as mux 1 control.
7-0	Entity State Mux Control 1 (Hardware debug)	Select of these bits allow internal state machines, counters, etc., to show up on chip outputs ENSTATE(15 down to 0). X'00' Disabled (no transition on outputs) X'01' Select 15-0 states X'40'-X'FF' Reserved

17.57: PCORE Debug States Config

This register serves as the PCORE configuration for external debug states. The INTST Debug states control for the address range desired must be set to select these PCORE state bits. If that is done, then this register acts to select the characteristics according to the bit descriptions below.

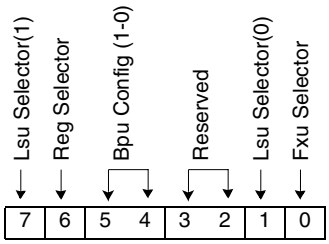
Length 8 bits

Type Read/Write

DCR Address XXXX 4320

Power on Reset value X'00'

Restrictions None



Bit(s)	Name	Description
7	LSU selector(1)	TBD
6	Reg selector	TBD
5-4	Bpu Config (1-0)	TBD
3-2	Reserved	Reserved
1	Lsu selector(0)	TBD
0	Fxu selector	TBD

Entity 18: PowerPC On-Chip Memory (PPOCM) Entity

The PPOCM entity is comprised of several SRAM arrays that provide a xxxK memory that may be used by the internal processor or by the IBM3206K0424. Also included in PPOCM is a DMA controller that the processor may use to do bulk data moves between the SRAM arrays and Control Memory, Packet Memory, or memory on an external PCI device. The PPOCM arrays will subsequently referred to as on-chip memory and the three external memories (control, packet, PCI) just mentioned will subsequently be referred to collectively as off-chip memory.

DMA Controller

The DMA controller moves data in eight byte aligned, eight byte portions. In real addressing mode, up to 64K bytes may be transferred at once. In virtual addressing mode, there are more restrictions. The DMA must remain within the virtual 4K page for both the PPOCM array address and the off-chip memory address.

18.1: PPOCM Control Register

This register contains information which controls the functions of the entity. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

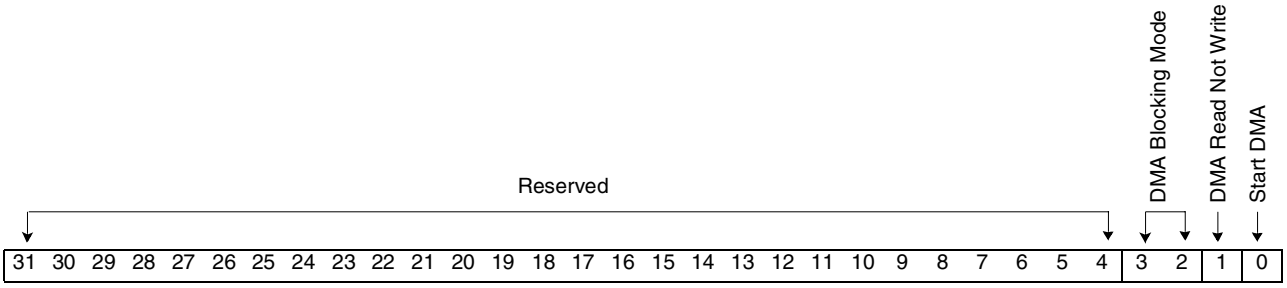
Length 32 bits

Type Read/Write

DCR Address 100 AND 01

Power on Reset value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-4	Reserved	Rserved
3-2	DMA Blocking Mode	These bits control how non-DMA accesses to PPOCM are handled while a DMA is in progress. They are encoded as follows: '00' Non-DMA PPOCM accesses are held off until the DMA is complete. '01' Non-DMA PPOCM accesses are held off only if the requester is attempting to use an array that will be involved in the DMA. '10' Reserved '11' Reserved
1	DMA Read Not Write	Setting this bit will indicate the DMA being set up is to transfer data from off-chip memory into PPOCM. Clearing this bit indicates the DMA should transfer data from PPOCM to off-chip memory.

Bit(s)	Name	Description
0	Start DMA	Setting this bit initiates the DMA operation. This bit will automatically clear when the DMA is completed.

18.2: PPOCM Status Register

This register contains status information that can be used to generate interrupts. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

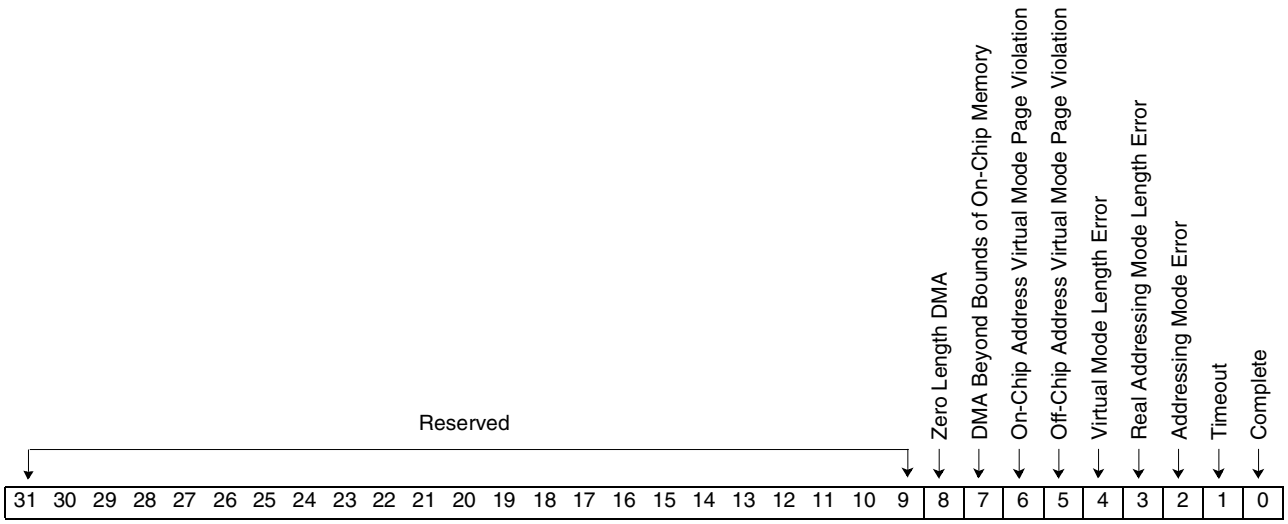
Length 32 bits

Type Read/Write

DCR Address 102 AND 03

Power on Reset value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-9	Reserved	Reserved
8	Zero Length DMA	This bit, set to '1', indicates that a DMA with a length of zero was attempted.
7	DMA Beyond Bounds of On-Chip Memory	This bit, set to '1', indicates that the on-chip address plus the DMA length yields a value that exceeds the address pace of the on-chip memory.
6	On-Chip Address Virtual Mode Page Violation	This bit, set to '1', indicates that the on-chip virtual address provided to PPOCM in combination with the DMA length results in a virtual page cross.
5	Off-Chip Address Virtual Mode Page Violation	This bit, set to '1', indicates that the off-chip virtual address provided to PPOCM in combination with the DMA length results in a virtual page cross.
4	Virtual Mode Length Error	This bit, set to '1', indicates that a virtual address mode DMA was started and the value in the DMA length register is greater than 4K.
3	Real Addressing Mode Length Error	This bit, set to a '1,' indicates that a real address mode DMA was started and the value in the DMA length register is greater than 64K.
2	Addressing Mode Error	This bit, set to a '1,' indicates that the off-chip memory and on-chip address written to the DMA address registers must be either both real or both virtual.

Bit(s)	Name	Description
1	Timeout	This bit, set to a '1,' indicates that the DMA timer expired.
0	Complete	This bit, set to a '1,' indicates that the DMA completed. The other bits in this register being a '0' will indicate a good completion.

18.3: PPOCM Interrupt Enable Register

This register enables the bits of the Status Register to generate an interrupt. The bits of this register correspond to the bits of the status register. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length 32 bits

Type Read/Write

DCR Address 104 AND 05

Power on Reset value X'00000000'

Restrictions None

18.4: PPOCM DMA Off-Chip Effective Address Register

This register provides the DMA controller the effective address of the off-chip portion of the DMA.

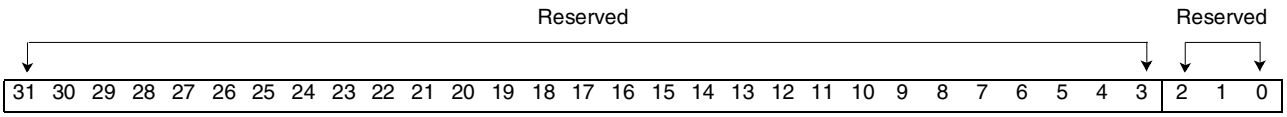
Length 32 bits

Type Read/Write

DCR Address 106

Power on Reset value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-3	Reserved	Reserved
2-0	Reserved	Reserved

18.5: PPOCM DMA On-Chip Effective Address Register

This register provides the DMA controller the effective address of the on-chip portion of the DMA.

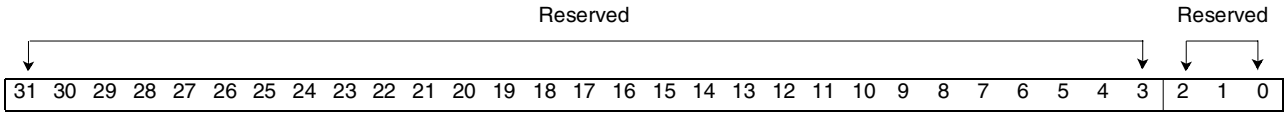
Length 32 bits

Type Read/Write

DCR Address 107

Power on Value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-3	Reserved	Reserved
2-0	Reserved	Reserved

18.6: PPOCM DMA Length Register

This register provides the DMA controller the length of the DMA. The maximum DMA length in real addressing mode is X'00010000' (64K). The maximum DMA length in virtual addressing mode is X'00001000' (4K).

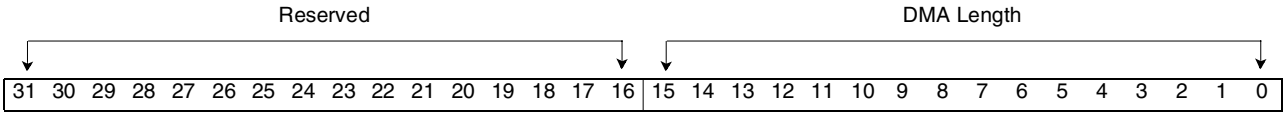
Length 32 bits

Type Read/Write

DCR Address 108

Power on Value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-16	Reserved	Reserved
16-0	DMA Length	Only bits 16-3 are writable as DMAs are done only in eight-byte segments.

18.7: PPOCM DMA Timeout Timer Register

This register is compared to a timer that begins running when bit 0 of the control register is set to '1'. When the timer reaches the value in this register, the DMA is terminated and a status bit is set. The default value of X'FFFFFF' results in a timeout value of 125 ms.

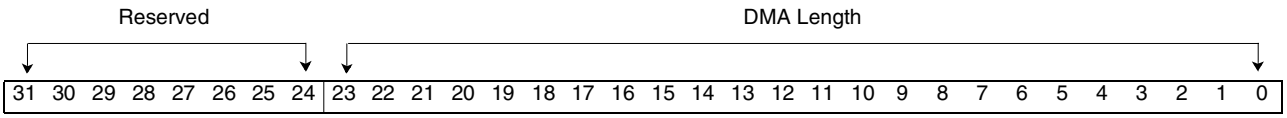
Length 32 bits

Type Read/Write

DCR Address 109

Power on Value X'00FFFFFF'

Restrictions None



Bit(s)	Name	Description
31-24	Reserved	Reserved
23-0	DMA Length	DMA timeout value

Entity 19: RS-232 Interface Logic (RS-232)

The RS232 entity provides a means by which an external debugger and the processor core can communicate. The RS-232 operates a one-or four-byte wide basis.

RS-232 Interface Logic Registers

19.1: RS-232 Control Register

This register controls the operation of the RS-232 logic. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

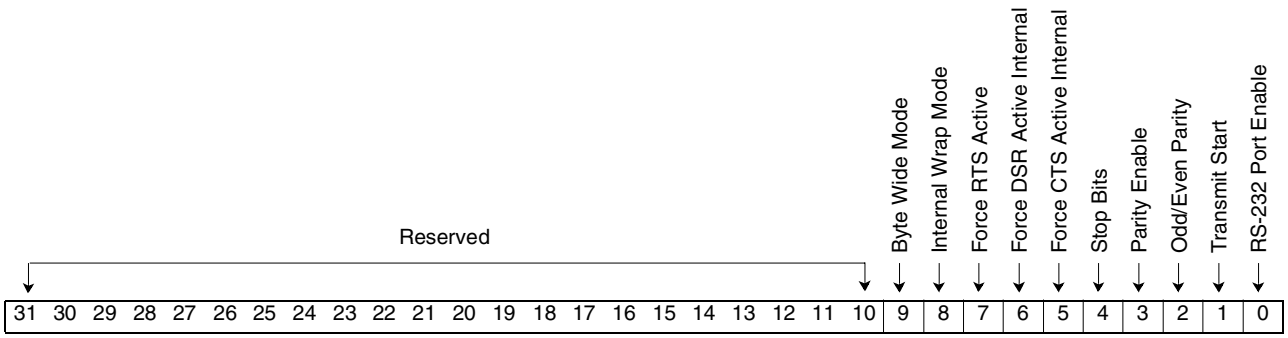
Length 32 bits

Type Read/Write

DCR Address X'210' AND 211'

Power On Value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-10	Reserved	Reserved
9	Byte Wide Mode	This bit set to '1' makes the port operate in byte-wide mode. When a transmit is started, only bits 7-0 of the transmit buffer are sent. A receive interrupt is generated whenever a byte is received.
8	Internal Wrap Mode	This bit set to '1' connects the transmit data stream to the receive data stream.
7	Force RTS Active	This bit set to '1' forces RTS to be driven active, regardless of what the transmit state machine is doing.
6	Force DSR Active Internal	This bit set to '1' forces DSR internal to RS-232 to appear active, regardless of the DSR input's state.
5	Force CTS Active Internal	This bit set to '1' forces CTS internal to RS-232 to appear active, regardless of the CTS input's state.
4	Stop Bits	This bit set to '1' indicates the port should use two stop bits. This bit set to '0' indicates the port should use one stop bit.
3	Parity Enable	This bit set to '1' enables parity.
2	Odd/Even Parity	If parity is enabled, this bit, set to '1', sets the parity type to odd. This bit set to '0' sets the parity type to even.

Bit(s)	Name	Description
1	Transmit Start	This bit set to '1' initiates the transmission of what is in the transmit buffer. This bit is cleared by hardware when the transmission is complete.
0	RS-232 Port Enable	This bit set to '1' enables the RS-232 port.

19.2: RS-232 Status Register

This register controls the operation of the RS-232 logic. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

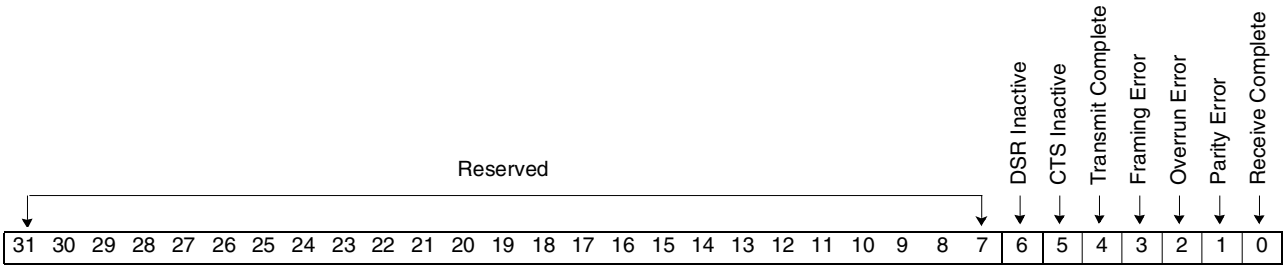
Length 32 bits

Type Read/Write

DCR Address X'212 AND 213'

Power On Value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-7	Reserved	Reserved
6	DSR Inactive	This bit set to '1' indicates DSR has gone inactive.
5	CTS Inactive	This bit set to '1' indicates CTS has gone inactive.
4	Transmit Complete	This bit set to '1' indicates the current transmission has completed.
3	Framing Error	This bit set to '1' indicates a framing error has been detected.
2	Overrun Error	This bit set to '0' indicates four bytes were received before the previous (one-byte mode) or four bytes (four-byte mode) had been read from the receive buffer.
1	Parity Error	This bit set to '1' indicates a parity error has been detected.
0	Receive Complete	This bit set to '1' indicates data from a clean reception is in the receive buffer.

19.3: RS-232 Interrupt Enable Register

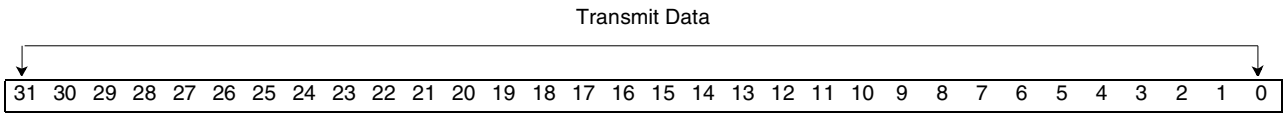
This register contains bits corresponding to the bits in the RS-232 status register. If a bit in this register is set and the corresponding bit is set in the RS-232 status register, an interrupt is generated. Bits six through four generate a transmit interrupt and bits three through zero generate a receive interrupt. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length	7 bits
Type	Read/Write
DCR Address	X'214 AND 215'
Power On Value:	X'00000000'
Restrictions	None

19.4: RS-232 Transmit Buffer

This register contains the data to be sent over the RS-232 connection.

Length	32 bits
Type	Read/Write
DCR Address	X'216'
Power On Value	X'00000000'
Restrictions	None



Bit(s)	Name	Description
31-0	Transmit Data	Data to be sent over link. Only bits 7-0 are sent in byte mode. The other bits are shifted, so the user can write all four bytes at once and just set the transmit bit four times.

19.5: RS-232 Receive Buffer

This register contains the data received over the RS-232 connection. Once this buffer is full, software has four byte receive times minimum to read it before an overrun condition can occur.

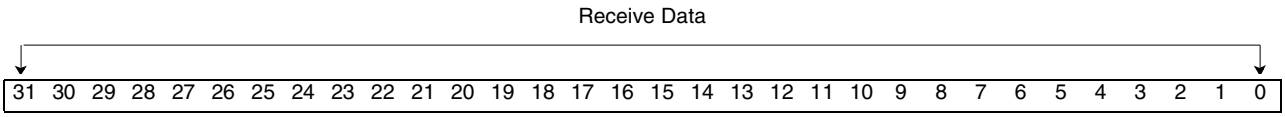
Length 32 bits

Type Read/Write

DCR Address X'217'

Power On Value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-0	Receive Data	Data received over the link. Only bits 7-0 are valid in byte mode.

19.6: RS-232 Baud Rate Register

This register contains the value used to determine the baud rate. The value to place in this register can be determined by this formula: BaudRate = 133MHz/(8*(Baud Rate Register + 1)).

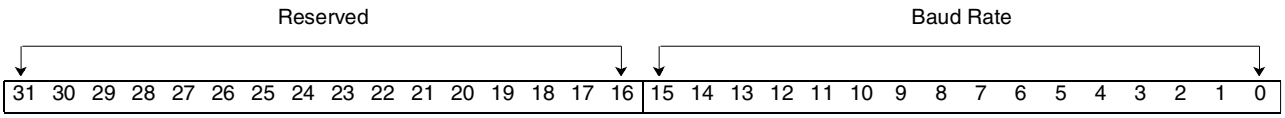
Length 16 bits

Type Read/Write

DCR Address X'218'

Power On Value X'00000000'

Restrictions None

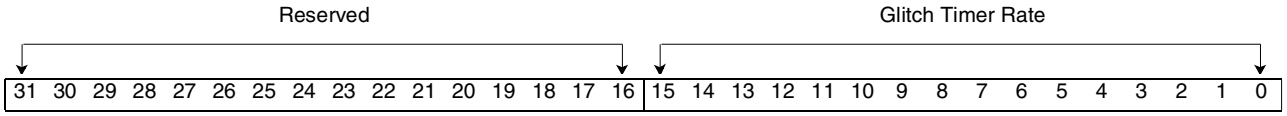


Bit(s)	Name	Description
31-16	Reserved	Reserved
15-0	Baud Rate	Suggested values: X'120' - 56600 X'1B0' - 38400 X'240' - 28800 X'360' - 19200

19.7: RS-232 CTS/DSR Glitch Timer Rate

This register contains the number of (baud rate/8) clocks CTS/DSR must be active/inactive before the state of CTS/DSR is considered valid. Transitions of shorter duration are assumed to be glitches.

Length 32 bits
Type Read/Write
DCR Address X'219'
Power On Value X'00000020'
Restrictions None

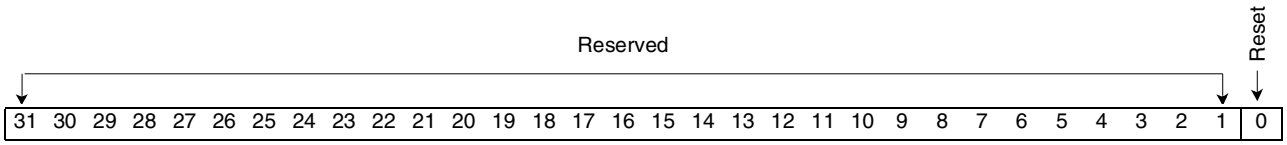


Bit(s)	Name	Description
31-16	Reserved	Reserved
15-0	Glitch Timer Rate	

19.8: RS-232 Reset Register

This register resets the port. See *Note on Set/Clear Type Registers on page 93* for more details on addressing.

Length 32 bits
Type Read/Write
DCR Address X'21A and 21B'
Power On Value X'00000000'
Restrictions None



Bit(s)	Name	Description
31-1	Reserved	Reserved
0	Reset	This bit set to '1' resets the port. The port will remain reset until this bit is cleared.

19.9: RS-232 Error Forcing Register

This register can be used by diagnostics in a wrap environment (external or internal) to force frame and parity errors. Overrun errors can be generated by sending four bytes, not reading the receive buffer, and sending four more bytes.

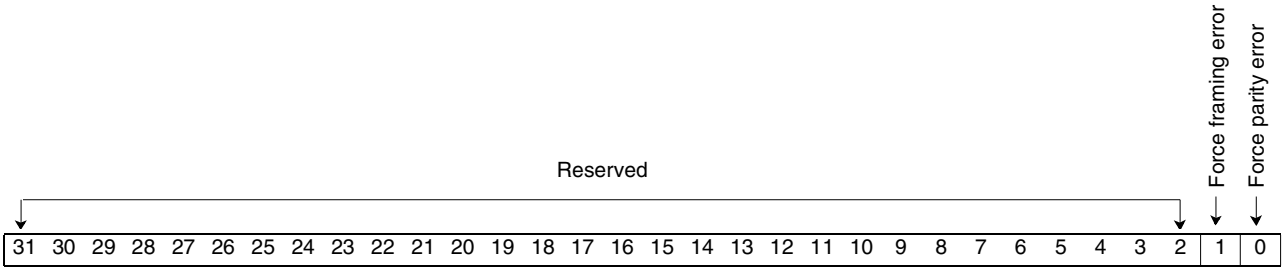
Length 32 bits

Type Read/Write

DCR Address X'21C'

Power On Value X'00000000'

Restrictions None



Bit(s)	Name	Description
31-2	Reserved	Reserved
1	Force framing error	Setting this bit to a '1' forces the first frame bit to a B'0' on all transmits.
0	Force parity error	Setting this bit to a '0' forces the receive logic to check for the opposite parity the transmit logic is using.

Entity 20: Reset and Power-on Logic (CRSET)

This entity performs BIST and flush operations. Chip software resets can be controlled by this entity, as well as the chip clock control.

Reset and Power-on Logic Registers

20.1: Reset Status Register

This register is used to reflect the last type of reset was. A hardware reset will clear software reset status bits, but a software reset will not have an affect on the hardware status bits.

Length 8 bits

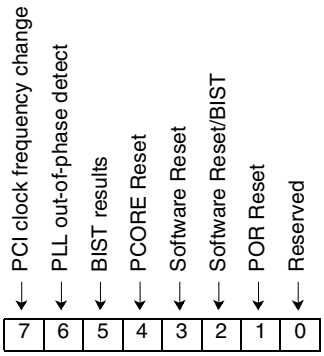
Type Read/Write

Address XXXX 0500

POR Value 'DB00001' or 'DB00010', where B is the state of the BIST results, and D is the PLL phase detection.

Software Reset Value 'DB001QQ' or 'DB010QQ', where Q is the state of this bit before the software reset and B is the state of the BIST results.

Restrictions None



Bit(s)	Name	Description
7	PCI clock frequency change	A value of '1' means that the real time PCI frequency calculator has detected a major change in frequency and has calculated new range bits for the PLL.
6	PLL out-of-phase detect	A value of '1' means that the out-of-phase detector circuit has triggered. This is just an indicator and is normal operation.
5	BIST results	A value of '1' means that a failure occurred within the BIST checking logic.
4	PCORE Reset	The PCORE entity has been reset via a software reset request (bit 3 of Software Reset Register).
3	Software Reset	A Software Reset has occurred; the chip was flushed.
2	Software Reset/BIST	A Software Reset has occurred; and BIST/flush was run.
1	POR Reset	A POR Hardware Reset that flushed the chip has occurred.
0		Reserved

20.2: Software Reset Enable Register

This register protects the Software Reset Register. If this register is not set, then a reset will not occur. Write a X'B4' to this register to enable software reset. A software reset will clear this register.

Length 8 bits

Type Read/Write

Address XXXX 0504

POR Value X'0'

Restrictions None

20.3: Software Reset Register

This register generates a scan path flush reset of the chip, or software initiated run of BIST, with the exception of the registers in the reset entity.

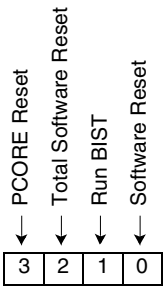
Length 4 bit

Type Write Only

Address XXXX 0508

POR Value B'0'

Restrictions Writing to this register without first setting the Software Reset Enable Register will have no affect. The register will not be set, thus the order of writing the enable and the software reset is important; the enable must be written first. Additionally, all current operations being performed by the IBM3206K0424 must be terminated before doing a reset operation. A minimum number of enable bits to turn off would be bits four, five, and six in INTST Control Register and bit 2 in PCINT Config Word 1.



Bit(s)	Name	Description
3	PCORE Processor Reset	Writing this bit to a '1' causes the internal processor core to reset.
2	Total Software Reset	Writing this bit to a '1' causes software reset and will be cleared after the software reset has occurred. The config registers in PCINT will also be put to their reset state.
1	Run BIST	Writing this bit to a '1' causes BIST to run and will be cleared after the software reset has occurred. This function is primarily for pre-loading the BIST registers to get more test coverage.
0	Software Reset	Writing this bit to a '1' causes software reset and will be cleared after the software reset has occurred. The config registers in PCINT will not be affected by this reset.

20.4: Memory Type Register

This register indicates the type of memory used for control and Packet Memory so that reset hardware will know how to properly preserve it during a reset.

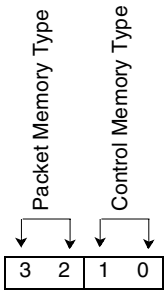
Length 4 bits

Type Read/Write

Address XXXX 050C

POR Value X'0'

Restrictions None



Bit(s)	Name	Description
3-2	Packet Memory Type	Decodes the same as bits nine through eight of *COMET/PAKIT Control Register* on page 186.
1-0	Control Memory Type	Decodes the same as bits nine through eight of *COMET/PAKIT Control Register* on page 186.

20.5: CRSET PLL Range Debug

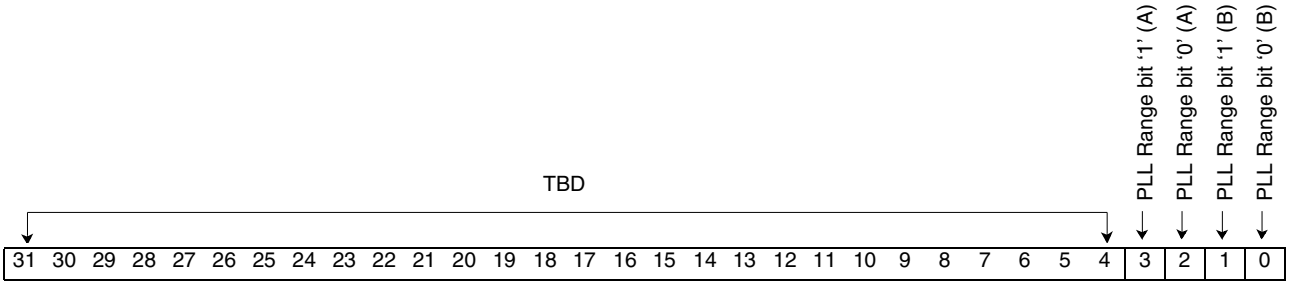
Used to debug the PPL operation.

Length 32 bits

Type Read Only

Address XXXX 0518

POR Value X'xxxxxxxx'



Bit(s)	Description
31-4	TBD
3	PLL Range bit '1' (A)
2	PLL Range bit '0' (A)
1	PLL Range bit '1' (B)
0	PLL Range bit '0' (B)

20.6: CRSET Control Register

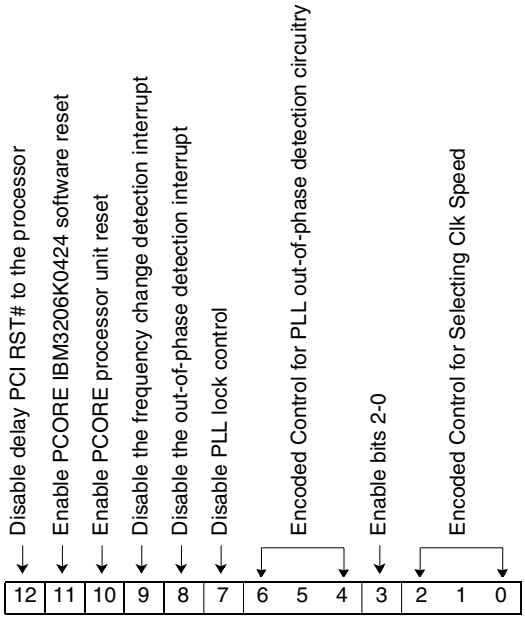
Used to control PCI frequency detection logic.

Length 13 bits

Type Read/Write

Address XXXX 0510

POR Value X'0330'



Bit(s)	Name	Description
12	Disable delay PCI RST# to the IBM3206K0424	Setting this bit to a '1' will disable the function that delays a PCI bus reset to the IBM3206K0424 if a serial EPROM is attached and still busy accessing data from the prior reset. By disabling this function, any PCI requirement to tri-state the I/O drivers would be met, but EPROM initialization information would be lost.
11	Enable PCORE IBM3206K0424 software reset	Setting this bit to a '1' will enable the PCORE entity to issue an IBM3206K0424 software reset.
10	Enable PCORE processor unit reset	Setting this bit to a '1' will enable the PCORE entity to issue a processor unit reset.
9	Disable the frequency change detection interrupt	Setting this bit to a '1' will disable using the frequency change detection bit as an interrupt source to INTST.
8	Disable the out-of-phase detection interrupt	Setting this bit to a '1' will disable using the out-of-phase detection bit as an interrupt source to INTST.
7	Disable PLL lock control	Setting this bit to a '1' will disable using the PLL lock output to make state transitions in the out-of-phase detection logic.
6-4	Encoded Control for PLL out-of-phase detection circuitry	These bits determine how much time buffering is allowed before an out-of-phase condition is detected. For a value of '000', a value of about 150 ps buffering is used. For each encoded increment value, an additional 150 ps is added. For example, the default value of three is about 600 ps of buffering.
3	Enable bits 2-0	Setting this bit to a '1' will enable bits 2-0 to override the pffcfg(2-0) bits that are strapped at the card level.

Bit(s)	Name	Description
2-0	Encoded Control for Selecting Clk Speed	These three bits have the same encoding as the chip I/O pffcfg(2-0) bits.

20.7: Clock Control Register (Nibble Aligned)

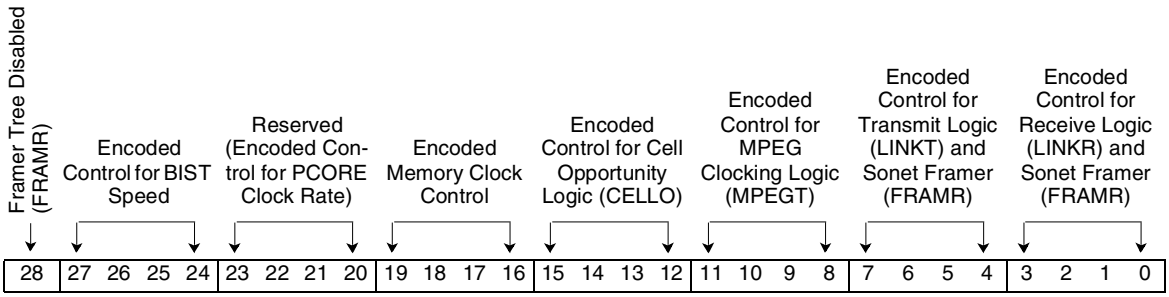
Used to disable clocks for power conservation and provide the Select A Clock function for MPEG and front end support. To change a nibble field in this register, always set it to '0' first, and then to the new value.

Length 29 bits

Type Read/Write

Address XXXX 0520

POR Value X'000E5532'



Bit(s)	Name	Description
28	Framer Tree Disabled (FRAMR)	When set, this bit will disable the clock tree to the Sonet Framer Logic.
27-24	Reserved (for Encoded Control for BIST Clock Rate)	Reserved
23-20	Reserved (for Encoded Control for PCORE Clock Rate)	Reserved
19-16	Memory Clock Control	Encoding of bits: X'D' Use an early version of the clock X'E' Use a nominal version of the clock X'F' Use a late version of the clock
15-12	Encoded Control for Cell Opportunity Logic (CELLO)	Same as bits 3-0.
11-8	Encoded Control for MPEG Clocking Logic (MPEGT)	Same as bits 3-0.
7-4	Encoded Control for Transmit Logic (LINKT) and Sonet Framer (FRAMR)	Same as bits 3-0.

Bit(s)	Name	Description
3-0	Encoded Control for Receive Logic (LINKR) and Sonet Framer (FRAMR)	Below is the encoded value of the bits that select a given clock. Always refer to *"Select A Clock" Selection Matrix* below for inputs supported for each clock out type. X'0' Turn this clock off. X'1' Use the external MPEG oscillator. X'2' Use the external RX clock. X'3' Use the external TX clock. X'4' Reserved X'5' Use the internal 15 ns clock. Assumes 33 or 66MHz PCI clock. X'6' Use the internal 30 ns clock. Assumes 33 or 66MHz PCI clock. X'7' Use the internal 60 ns clock. Assumes 33 or 66MHz PCI clock. X'8' Use the internal 120 ns clock. Assumes 33 or 66MHz PCI clock. X'9' Use the internal 240 ns clock. Assumes 33 or 66MHz PCI clock. X'A' Use the internal 480 ns clock. Assumes 33 or 66MHz PCI clock. X'B' Use the differential Receiver clock divided by eight. X'C' Use the differential Transmit clock divided by eight. X'D' Use the differential Receiver clock (chopped). X'E' Use the differential high speed receiver clock (divided by 8 and 50% duty cycle).

"Select A Clock" Selection Matrix

Clock Frequency Base	CELLO BCO CCO	MPEGT BMT CMT	LINKT(TX) BTX CTX RTX	LINKR(RX) BRX CRX RRX	Nibble Code
HS RX Rec Diff Osc			X		X'E'
RX Rec Diff Osc				X	X'D'
TX/8 Diff Osc			X		X'C'
RX/8 Diff Osc				X	X'B'
480 ns	X	X	X	X	X'A'
240 ns	X	X	X	X	X'9'
120 ns	X	X	X	X	X'8'
60 ns	X	X	X	X	X'7'
30 ns	X	X	X	X	X'6'
15 ns	X	X	X	X	X'5'
Reserved					X'4'
TX Osc	X	X	X	X	X'3'
RX Osc	X	X	X	X	X'2'
MPEG OSC	X	X	X	X	X'1'
OFF	X	X	X	X	X'0'
Control Bits	15-12	11-8	7-4	3-0	

20.8: CBIST PRPG Results

This is the PRPG results register, updated after BIST has run. It is used by the BIST function for chip test.

Length	32 bits
Type	Read/Write
Address	XXXX 05B0
POR Value	X'FFFFFFFF'

20.9: CBIST MISR Results

This is the MISR results register, updated after BIST has run. It is used by the BIST function for chip test.

Length	32 bits
Type	Read/Write
Address	XXXX 05B4
POR Value	X'00000000'

20.10: CBIST BIST Rate

This register holds a counter value that separates the time between when the A clock and the B clock are launched during BIST. This allows finer tuning to how much power BIST uses versus how much testing gets done within the time allowed. It is used by the BIST function for chip test.

Length	3 bits
Type	Read/Write
Address	XXXX 05B8
POR Value	X'0'

20.11: CBIST PRPG Expected Signature

This is the PRPG signature register, which should be written by CRISCO code with the expected value of signature, based on the value in CBIST CYCT Load Value and the clock selected for BIST to run from. It is used by the BIST function for chip test.

Length	32 bits
Type	Read/Write
Address	XXXX 05C0
POR Value	X'FFFFFFFF'

20.12: CBIST MISR Expected Signature

This is the MISR Signature Register, which should be written by CRISCO code with the expected value of signature, based on the value in CBIST CYCT Load Value and the clock selected for BIST to run from. It is used by the BIST function for chip test.

Length	32 bits
Type	Read/Write
Address	XXXX 05C4
POR Value	X'00000000'

20.13: CBIST CYCT Load Value

This register is the loaded value for the CBIST BIST Rate Register. The time for BIST to run can be computed by the following equation: (shift count) * (c30 clock*2) * (cycle time). It is used by the BIST function for chip test.

Length	18 bits
Type	Read/Write
Address	XXXX 05C8
POR Value	X'00005800'

Entity 21: JTAG Interface Logic (CJTAG)

The CJTAG entity contains logic to support a test access port (TAP) controller compliant with the IEEE 1149.1-1993 standard. The TAP controller is accessed via the following five pins:

TCK	Test Clock. All activity of the JTAG interface is clocked via TCK. Events occur on the rising or falling edge of TCK. TCK should have a maximum frequency of 20MHz.
TMS	Test Mode Select. Test Mode Select is used to control state transitions in the TAP controller. These transitions occur on the rising edge of TCK. The BTR selected for TMS should be one with an internal pullup.
TDI	Test Data In. Serial data input to the JTAG logic. The BTR selected for TDI should be one with an internal pullup.
TDO	Test Data Out. Serial data output to the JTAG logic.
TRST	Test Reset. Asynchronous, minus active reset to the TAP controller. Assertion of this input causes the TAP controller to reset and the JTAG instruction register to load the IDCODE instruction. It is preferable to have TRST be independent of any chip reset. With an independent reset, the JTAG logic can be reset, allowing the chip's state to be examined without having to reset the core logic. The BTR selected for TRST should be one with an internal pullup.

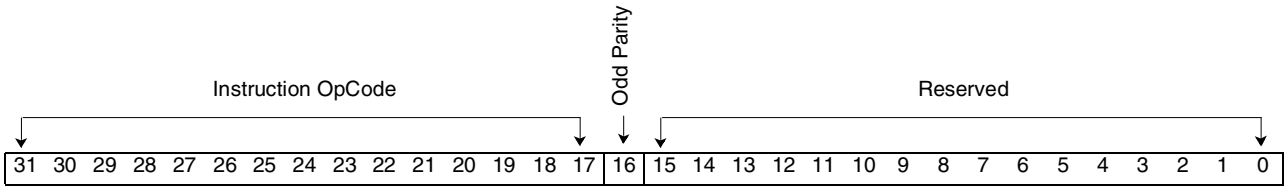
The proper operation of these signals and the TAP controller is defined in the IEEE 1149.1-1993 standard.

Scanning

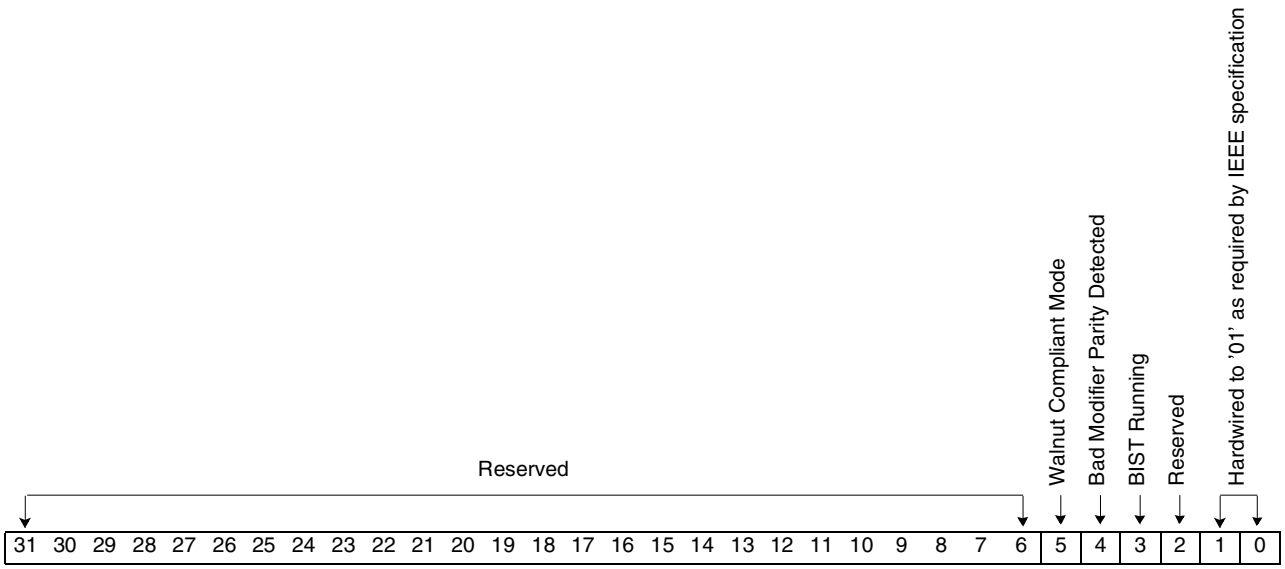
The TAP controller supports two types of scans: instruction scans and data scans. Instruction scans control the type of operation and select which (if any) scan chains are involved in the operation. Data scans generally clock the data on TDI into the selected scan chain.

21.1: Instruction Format

The processor's JTAG logic supports 32-bit instructions in one of two formats: the first format uses opcodes compliant with the IEEE standard; the other supports opcodes as defined by the Walnut chip that are compatible but not compliant with the IEEE standard. As an instruction is scanned in, status for the previous instruction is presented on TDO.



Bit(s)	Description
31-17	Instruction OpCode
16	In compatible mode, this is odd parity over bits 15-0
15-0	Reserved



Bit(s)	Description
31-6	Reserved
5	Walnut Compliant Mode.
4	Bad Modifier Parity Detected.
3	BIST Running.
2	Reserved
1-0	Hardwired to '01' as required by IEEE specification.

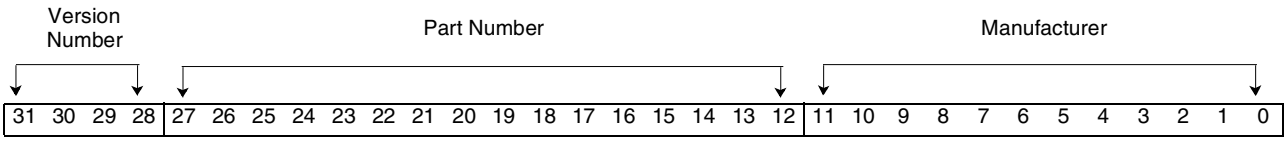
Instructions

The following instructions are supported:

21.2: IDCODE

Returns a 32-bit identification code when a data scan is performed. The IDCODE has the following structure:

Opcode X'0300XXXX'



Bit(s)	Name	Description
31-28	Version Number	This is set to X'4' for IBM3206K0424.
27-12	Part Number	This is set to X'1D00' for IBM3206K0424.
11-0	Manufacturer	This is set to X'049' for IBM.

21.3: SAMPLE/PRELOAD

Captures the state of the boundary scan I/O. As the values captured are scanned out, new values can be loaded into the boundary scan latches. This operation will not affect functional operation.

Opcode X'0402XXXX'

21.4: EXTEST

Drives the values in the boundary scan latches onto their respective I/O. This function can be used in conjunction with SAMPLE/PRELOAD to perform card wire tests.

Opcode (Compliant) X'00000000'

Opcode (Compatible) X'0600XXXX'

21.5: BYPASS

Selects the single bit bypass register for data scans.

Opcode (Compliant) X'FFFFFFFF'

Opcode (Compatible) X'FFFFXXXX' or X'0000XXXX'

21.6: RUNBIST

Causes built in self test (BIST) to execute.

Opcode X'0770XXXX'

21.7: BIST_RESULTS

Returns a 64-bit value when a data scan is performed. Bits 63-32 are the PRPG and bits 31-0 are the MISR from the BIST logic.

Opcode X'1F02XXXX'

21.8: WALNUT_MODE

This command enables Walnut compatible mode.

Opcode X'3000'

21.9: COMPLIANT_MODE

This command enables JTAG compliant mode.

Opcode X'33010000'

21.10: STOP

This command halts the functional clocks of IBM3206K0424 in anticipation of a scan. After the STOP command is scanned in, a data scan that takes the TAP controller through the Capture-DR, Exit1-DR, and Update-DR states should be performed. This will capture the state of the I/O so that they can be held in a known state if a scan command is issued.

Opcode X'2002'

21.11: SCAN

This command causes TDI to be clocked into the scan chain during a subsequent data scan. The scan out of the scan chain is placed on TDO. This command will not work unless a STOP command is sent down immediately before the SCAN command is issued.

Opcode X'0802'

21.12: SCAN_IN

This command causes TDI to be clocked into the scan chain during a subsequent data scan. TDO is forced to ’0’. This command will not work unless a STOP command is sent down immediately before the SCAN_IN command is issued.

Opcode X’0900’

21.13: SCAN_OUT

This command causes the scan out of the scan chain to be placed on TDO. Data is recirculated through the scan chains. TDI is ignored. This command will not work unless a STOP command is sent down immediately before the SCAN_OUT command is issued.

Opcode X’0A00’

21.14: Private_RW1

This command is used by RISCWATCH.

Opcode X'0500'

21.15: Private_RW2

This command is used by RISCWATCH.

Opcode X'0582'

21.16: Private_RW3

This command is used by RISCWATCH.

Opcode X’05C0’

Sonet Framer Core (FRAMR Chiplet Address Mapping)

FRAMR Chiplet Address Mapping

Chiplet Name	Short Name	Chiplet Base Address	Chiplet Address Range	Number of Bytes
Reserved		X'000'	X'000 - 0FF'	256
ACH_Tx	HT	X'100'	X'100 - 1FF'	256
ACH_Rx	HR	X'200'	X'200 - 2FF'	256
Reserved		X'300'	X'300 - 3FF'	256
OFP_Tx	OT	X'400'	X'400 - 7FF'	1024
OFP_Rx	OR	X'800'	X'800 - BFF'	1024
GPPINT	GP	X'C00'	X'C00 - CFF'	256
Reserved		X'D00'	X'D00 - FFF'	768

GPPINT Architecture

Overview

The General Purpose Processor INTerface (GPPINT) provides direct access to registers located in the GPPINT module; it provides delayed access to registers and counters located in the GppHandler modules of the various chiplets of the SONET core. GPPINT controls the handshaking with the external microprocessor as well as the handshaking with the GppHandlers at the asynchronous chiplet interfaces. Address decoding is done to the chiplet level in GPPINT. In addition, addresses are decoded to the register level for the local GPPINT registers.

Reset Register

Each chiplet is controlled by one reset bit. At power-on, all reset bits are active and the chiplets are disabled. They can be released by the General Purpose Processor (GPP) only after all global configuration parameters have been set and the clocks to the chiplets have been established. In addition, there are reset bits for the chiplets that do not have their own GppHandler.

Interrupt Registers

The interrupt register is used as a pointer to the chiplet interrupt registers with pending requests: the clock status error register, and the handshaking error register. An active bit of the interrupt register is reset by removing the cause for the request in the corresponding chiplet or by masking the active IRQ bit(s) in the chiplet; therefore, the interrupt registers (including the pointer) are read only. All interrupt and pointer registers have a corresponding MASK register (R/W). Every unmasked, active interrupt bit causes an active pointer bit. Every unmasked, active pointer bit causes activation of the interrupt signal to the microprocessor.

Handshaking Error Registers

Each bit of the handshaking error registers indicates a locked interface to one of the chiplet GppHandlers. Two additional bits indicate various timeout events. To reset an individual bit of the handshaking error register, the cause for the request must be removed and a one must be written into the bit location of the register (R/W). Reading the register will reset the whole (eight-bit) register if the corresponding "clear-register" option is set in the configuration register. The handshaking error indication register has a corresponding MASK register (R/W). Every unmasked, active handshaking error bit causes activation of the pointer bit in the GPPINT interrupt register.

Clock Monitor Status Registers

The clock monitor status register bits indicate the loss of a specific chiplet's clock. They are set whenever a difference between the clock test signal and the individual chiplet clock acknowledge signal occurs after one clock monitor test period. To reset an individual bit of the clock monitor status registers, the clock of the corresponding chiplet must be restored and a one must be written into the bit location of the register (R/W). Reading one of the registers will reset the whole (eight-bit) register if the corresponding "clear-register" option is set in the configuration register. The clock monitor status register has a corresponding MASK register (R/W). Every unmasked, active clock monitor status bit causes activation of the pointer bit in the GPPINT register.

Local Gppint Configuration Registers

There are registers (R/W) for the Clock Monitor Test Period, the Watchdog Timer Period and the "clear-register" option. A read-only register provides the Vital Product Data (VPD).

Global Static Configuration Registers

These are configuration parameters that are shared by many chiplets or that are needed by chiplets that have no GppHandler. The initial values can be modified by the microprocessor after power-on, but should not be changed later on. All global static configuration registers are R/W.

Status Registers

These registers provide status information from chiplets that have no GppHandler and are read only. Presently, there is only one status register for the SIM chiplet (PLL lock status).

GPPINT Chiplet Address Mapping Overview: Base Address = x'C00'

Register Name	Description[1]	Address Offset	Type	Initial Value
RESGP1	Reset register	X'00'	R/W	B'11111111'
...	Reserved	X'01 - 0F'		
IRQGP1	Chiplet interrupt request register #1	X'10'	R	B'00000000'
...	Reserved	X'11 - 17'		
IRMGP1	Chiplet interrupt mask register #1	X'18'	R/W	B'00000000'
...	Reserved	X'19 - 1F'		
HShake1	Handshaking error register #1	X'20'	R/W	B'00000000'
...	Reserved	X'21 - 27'		
HSMask1	Handshaking error mask register #1	X'28'	R/W	B'00000000'
...	Reserved	X'29 - 2F'		
ClkStat1	Clock status register #1	X'30'	R/W	B'00000000'
...	Reserved	X'31 - 37'		
ClkMask1	Clock status mask register #1	X'38'	R/W	B'00000000'
...	Reserved	X'39 - 47'		
CMonGP1	Clock monitor test period	X'48'	R/W	B'00000000'
WDTGP1	Watchdog Timer Period	X'49'	R/W	B'11111111'
ConfGP1	"Clear-register" option register	X'4A'	R/W	B'11111111'
...	Reserved	X'4B - 4F'		
VMD	Vital Macro Data register	X'50'	R	B'10000001'
...	Reserved	X'51 - 57'		
GATMCS	Common ATM/CS static configuration register	X'58'	R/W	B'00000000'
GCasc	Common Cascading static configuration register	X'59'	R/W	B'10101010'
GLoopTx	Transmit Loopback static configuration register	X'5A'	R/W	B'00000000'
GLoopRx	Receive Loopback static configuration register	X'5B'	R/W	B'00000000'
GExtRes	External clock recovery circuit reset register	X'5C'	R/W	B'00000000'
...	Reserved	X'5D - 67'		
OFPTXGP	OFP_Tx static configuration register	X'68'	R/W	B'00000000'
OFPRXGP1	OFP_Rx static configuration register #1	X'69'	R/W	B'00000000'
OFPRXGP2	OFP_Rx static configuration register #2	X'6A'	R/W	B'00000000'
...	Reserved	X'6B - 71'		
PIMRConf2	PIM_Rx static configuration register #2	X'73'	R/W	B'00000000'
...	Reserved	X'74 - 7E'		
SIMStat	SIM status register	X'7F'	R	N.A.
...	Reserved	X'80 - FF'		

1. All registers are of eight-bit width.

22: GPPINT Register Description

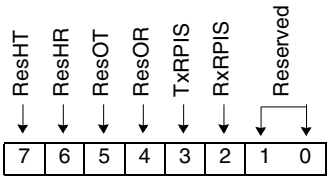
22.1: Chiplet Reset Register (RESGP)

The bits of the chiplet reset register control the resetting (enabling/disabling) of complete chiplets.

For each bit position:

0 = Reset inactive for this chiplet
1 = Reset active (chiplet is disabled; DEFAULT).

Length 8 bits
Type Read/Write
Address C00
Power On Value X'FF'



Bit(s)	Name	Description
7	ResHT	Reset to chiplet ACH_Tx
6	ResHR	Reset to chiplet ACH_Rx
5	ResOT	Reset to chiplet OFP_Tx
4	ResOR	Reset to chiplet OFP_Rx
3	TxRPIS	Reset to chiplet PIS_Tx
2	RxRPIS	Reset to chiplet PIS_Rx
1-0	Reserved	Reserved

22.2: Chiplet Interrupt and Mask Registers (IRQGP1 (IRMGP1))

The chiplet interrupt request register indicates pending interrupt requests from individual chiplets. An active bit of this register is reset by removing the cause for the request in the corresponding chiplet or by masking the active IRQ bit(s) in the chiplet; therefore, this register is read only.

For each bit position:

0 = No chiplet interrupt request pending.

1 = Chiplet has pending interrupt request(s). The chiplet interrupt request mask register bits control the propagation of a chiplet interrupt request to the Sonet Macro Interrupt output pin. The mask registers allow read and write access.

For each bit position:

0 = The corresponding interrupt request bit is masked (DEFAULT).

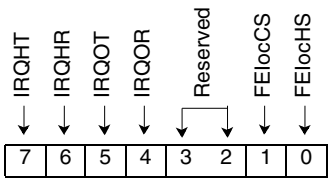
1 = The corresponding interrupt request bit is active (for IRMGP1, the corresponding interrupt request bit activates the Sonet Macro Interrupt).

Length 8 bits

Type Read

Address C10

Power On Value X'00'



Bit(s)	Name	Description
7	IRQHT	IRQ from ACH_Tx
6	IRQHR	IRQ from ACH_Rx
5	IRQOT	IRQ from OFP_Tx
4	IRQOR	IRQ from OFP_Rx
3-2	Reserved	Reserved
1	FElocCS	Pending clock status error active
0	FElocHS	Pending handshaking error active

22.3: Handshaking Error Indication and Mask Registers (HShake1)

The local handshaking error indication register indicates pending handshaking error requests from the GPPINT chiplets.

For each bit position:

0 = Normal operation of the corresponding chiplet.

1 = The corresponding chiplet did not deassert its DTACK signal.

Exception: The signals TOError and IntError (HShake2(1-0)) have the following meaning: Normal operation GPP deasserts Strobes without waiting for DTACK assertion Watchdog Timeout in REST state Watchdog Timeout in REQ state. An active bit of the handshaking error indication register is reset by removing the cause for the malfunctioning of the chiplet and by writing a one into the corresponding bit position. Reading one register will reset all bits of this register if the "clear-register" option is set in ConfGP1(2). The handshaking error indication mask register bits control the propagation of the GPPINT handshaking error requestof the register HShake1. HSMask1 controls propagation to the signal FElocHS (bit 0 of IRQGP1 register). The mask registers allow read and write access.

For each bit position:

0 = The corresponding handshaking error indication bit is masked (DEFAULT).

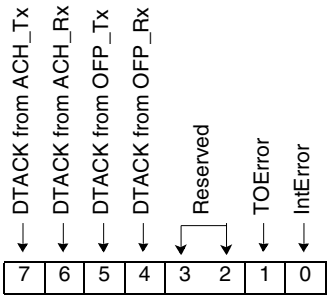
1 = The corresponding request bit is active (for HSMask1, the corresponding request bit activates signal FElocHS (bit 0 of IRQGP1 register). "Clear-register" option set in ConfGP1(2).

Length 8 bits

Type Read/Write

Address C20

Power On Value X'00'



Bit(s)	Name	Description
7		DTACK from ACH_Tx stuck at ONE
6		DTACK from ACH_Rx stuck at ONE
5		DTACK from OFP_Tx stuck at ONE
4		DTACK from OFP_Rx stuck at ONE
3-2	Reserved	Reserved
1	TOError	Time Out Error of the GPP interface (see above)
0	IntError	GPP interface error (see above)

22.4: Clock Monitor Status and Mask Registers (ClkStat1 (ClkMask1))

The clock monitor status register bits indicate the loss of a specific island's clock. They are set whenever a difference between the clock test signal and the individual island's clock acknowledge signal occurs after the clock monitor test period.

For each bit position:

0 = Normal operation of the corresponding clock island

1 = The corresponding island clock is lost. An active bit of this register is reset by restoring the clock of the corresponding clock island and by writing a one into the corresponding bit position. Reading one register will reset all bits of this register if the "clear-register" option is set in bit ConfGP1(3). The clock monitor mask register ClkMask1 controls the propagation of active clock monitor status signals. ClkMask1 controls propagation to the signal FElocCS (bit 1 of IRQGP1 register). The mask registers allow read and write access.

For each bit position:

0 = The corresponding clock status bit is masked (DEFAULT).

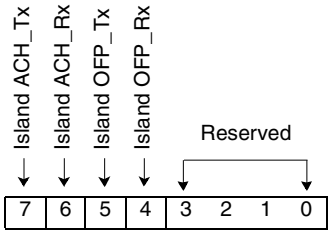
1 = The corresponding clock status bit is active (for ClkMask1, the corresponding bit activates the signal FElocCS (bit 1 of IRQGP1 register).

Length 8 bits

Type Read/Write

Address C30

Power On Value X'00'



Bit(s)	Description
7	Island ACH_Tx lost clock
6	Island ACH_Rx lost clock
5	Island OFP_Tx lost clock
4	Island OFP_Rx lost clock
3-0	Reserved

22.5: Clock Monitor Test Period Register (CMonGP1)

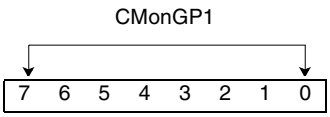
Divider ratio to derive the clock monitor test period from the GPPCLK clock. Clock monitoring is disabled if equal x'00' (DEFAULT).

Length 8 bits

Type Read/Write

Address C48

Power On Value X'00'



Bit(s)	Name	Description
7-0	CMonGP1(7-0)	Number of GPPCLK cycles/test period

22.6: Watchdog Timer Period Register (WDTGP1)

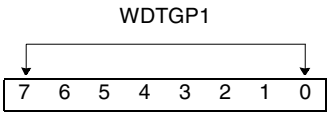
Divider ratio to derive the interface timeout period from the GPPCLK clock. This register is reset to x'FF' whenever a timeout occurs; it has to be reconfigured by a GPP write access.

Length 8 bits

Type Read/Write

Address C49

Power On Value X'FF'



Bit(s)	Name	Description
7-0	WDTGP1(7-0)	Number of GPPCLK clock cycles per timeout period

22.7: GPPINT Local Configuration Registers (ConfGP1)

The bits of this local configuration register control the resetting of complete registers upon read access ("clear register" option).

For each bit position:

0 = No action upon read access.

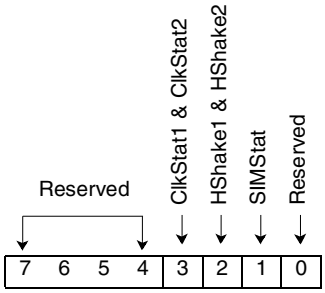
1 = The corresponding register is reset upon read access (DEFAULT).

Length 8 bits

Type Read/Write

Address C4A

Power On Value X'FF'



Bit(s)	Description
7-4	Reserved
3	Clear-bit for registers ClkStat1 & ClkStat2
2	Clear-bit for registers HShake1 & HShake2
1	Clear-bit for register SIMStat
0	Reserved

22.8: Vital Macro Data Register (VPD)

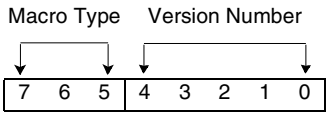
This read-only register displays the macro identification.

Length 8 bits

Type Read

Address C50

Power On Value X'01'



Bit(s)	Description
7-5	Macro type (000)
4-0	Version number

22.9: Static Configuration Register (GATMCS)

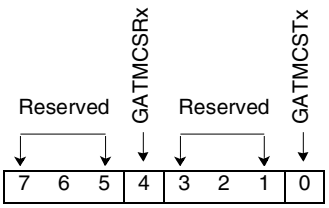
Common static configuration data, providing control signals that are distributed to multiple chiplets. Set once by the GPP before the individual chiplets get enabled and not changing during normal operation.

Length 8 bits

Type Read/Write

Address C58

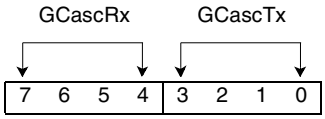
Power On Value X'00'



Bit(s)	Name	Description
7-5	Reserved	Reserved
4	GATMCSRx	ATM cell or CS mode for SDH macro in receive direction: 0 = SDH macro in ATM mode 1 = SDH macro in CS mode
3-1	Reserved	Reserved
0	GATMCSTx	ATM cell or CS mode for SDH macro in transmit direction: 0 = SDH macro in ATM mode, 1 = SDH macro in CS mode

22.10: GCasc

Length 8 bits
Type Read/Write
Address C59
Power On Value X'88'



Bit(s)	Name	Description
7-4	GCascRx(7-4)	Defines SDH macros in receive direction: 0001 STS3c 1000 STM1 others reserved
3-0	GCascTx(7-4)	Defines SDH macros in transmit direction: 0001 STS3c 1000 STM1 others reserved

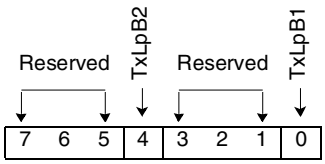
22.11: GLoopTx

Transmit loopback control.

For each bit position:

0 = ACH Loopback disabled (DEFAULT).
1 = ACH Loopback enabled.

Length 8 bits
Type Read/Write
Address C5A
Power On Value X'00'



Bit(s)	Name	Description
7-5	Reserved	Reserved
4	TxLpB2	Loopback #2 control, Tx macro
3-1	Reserved	Reserved
0	TxLpB1	Loopback #1 control, Tx macro

22.12: GLoopRx

Receive loopback control.

For each bit position:

0 = ACH Loopback disabled (DEFAULT).

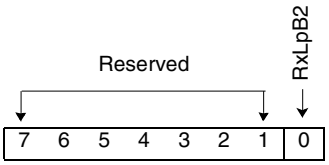
1 = ACH Loopback enabled.

Length 8 bits

Type Read/Write

Address C5B

Power On Value X'00'



Bit(s)	Name	Description
7-1	Reserved	Reserved
0	RxLpB2	Loopback #2 control, Rx macro

22.13: GExtRes

External clock recovery circuit reset signal. Delivered to external circuit (deserializer) via device pins. The active level depends on the external circuit used. Default value at power-on-reset is LOW.

Length 8 bits

Type Read/Write

Address C5C

Power On Value X'00'

Bit(s)	Name	Description
7-1	Reserved	Reserved
0	RSTCRec	External recovery reset

22.14: OFPTXGP

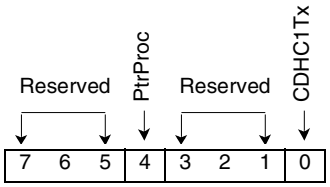
Static configuration data, providing control signals for chiplet OFP_Tx. Set once by the GPP before the individual chiplets are enabled and not changing during normal operation.

Length 8 bits

Type Read/Write

Address C68

Power On Value X'00'



Bit(s)	Name	Description
7-5	Reserved	Reserved
4	PtrProc	0 = AU pointer processing disabled in ATM mode 1 = AU pointer processing enabled in ATM mode
3-1	Reserved	Reserved
0	SDHC1Tx	0 = C1 byte replaced by section trace J1 byte (ITU-T standard) 1 = Old numbering scheme is used. OFPRXGP1 & 2: Static configuration data, providing control signals for chiplets OFP_Rx. Set once by the GPP before the individual chiplets are enabled and not changing during normal operation.

22.15: OFPRXGP1

Length 8 bits

Type Read/Write

Address C69

Power On Value X'00'

Bit(s)	Name	Description
7-1	Reserved	Reserved
0	SDHC1Rx	0 = The new (ITU-T standard) numbering scheme is used 1 = Old numbering scheme is used

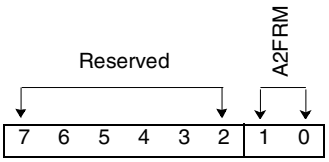
22.16: OFPRXGP2

Length 8 bits

Type Read/Write

Address C6A

Power On Value X'00'



Bit(s)	Name	Description
7-2	Reserved	Reserved
1-0	A2Frm	OFP_Rx RxSoFrm assertion controls: 00 RxSoFrm asserted during 3rd A2 byte 01 RxSoFrm asserted during 1st A2 byte 10 RxSoFrm asserted during 2nd A2 byte 11 RxSoFrm asserted during 3rd A2 byte

22.17: PIMRConf2

Static configuration data, providing control signals for chiplets PIM_Tx/PIM_Rx. Set once by the GPP before the individual chiplets are enabled and not changing during normal operation.

Length 8 bits

Type Read/Write

Address C73

Power On Value X'00'

Bit(s)	Name	Description
7-2	Reserved	Reserved
1-0	Algo1(7-6)	Selects frame pattern recognition algorithm: 00 All bits checked; maximum four bad frames 01 12 bits checked; only maximum four bad frames 10 All bits checked; maximum five bad frames 11 12 bits checked only; maximum five bad frames

22.18: SIMStat

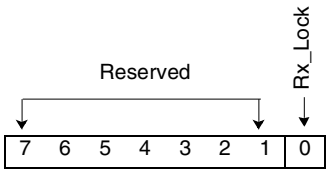
Status register, providing the GPP with information from the SIM chiplet via PIM. Either SIM-internal or external PLL lock status. "Clear-register" option set in ConfGP1(1).

Length 8 bits

Type Read

Address C7F

Power On Value N/A



Bit(s)	Name	Description
7-1	Reserved	Reserved
0	Rx_Lock	0 Rx PLL is still in phase aquisition process 1 Rx PLL is enabled and has locked to the incoming data stream incoming data stream

23: GPPHandler Architecture

Overview

All GPP handlers for the various chiplets have the following general register structure.

GPPHandler Architecture

Address Range	Register Function
X'0 - 1'	Read on the Fly registers
X'2 - 3'	Counter enable registers
X'4 - 2F'	Counters and counter threshold registers
X'30'	Reset register
X'31 - 32'	Command registers
X'33 - 37'	Event latch registers (was called status)
X'38 - 47'	Interrupt registers (addr=int reg, addr-1=int mask reg)
X'48 - 57'	Configuration registers

Counter Registers

Every counter has an enable bit in the counter enable register (addr 2 or 3), and optionally up to two programmable thresholds. Each counter has an interrupt bit for overflow and up to two interrupt bits for threshold crossing in the counter interrupt registers. For all counters in one handler there is one common 'read-on-the-fly register' that is used to store the higher order bytes to obtain a correct readback value for counters larger than eight bits. Counters are read-only registers; the count enable registers are read/write. Note: COUNTER reading is independent of the counter length, given that a counter has address n as base, reading address n or address n-1 both yield the least significant byte of the counter. Reading address n has no influence on the counter, but reading address n-1 will reset the counter after the read. Reading address n or n-1 will always latch the higher order bytes into the read on the fly register (before the optional automatic reset). Counters can only be read and not written to. For a 16-bit counter, the most significant byte should be read from ROFmid (address 0). For a 24-bit counter, the most significant byte is read from ROFhi (address 1), the next byte from ROFmid (address 0). To completely read a 24-bit counter: first read least significant byte from counter address n or n-1, then read ROFmid and ROFhi (address 0; address 1).

Reset Registers

Each handler has a two-bit reset register. Bit 0 is the chiplet reset control. This bit is active high after power on reset, causing the chiplet to be disabled. Bit 1 is the chiplet halt signal, which for selected chiplets freezes the state machines for diagnostic purposes. This is a read/write register.

Command Registers

The optional command register(s) will generate events to the chiplet. When a bit is written high by the microprocessor, it will remain high for one chiplet clock cycle. Therefore, reading back a command register will always read back zeroes. This is a read/write register.

Event Latch Registers

The optional event latch register(s) remember one ore more occurances of events that happen in a chiplet. This may be considered as a one-bit saturating counter. Each bit in the register corresponds to an event in the chiplet. Such bits remain high after the event happened until the microprocessor implicitly or explicitly resets the bit. This is configurable: implicit reset is done by writing a high value to the bit that is to be reset. Explicit will reset all bits of one register when the register is read. This is a read/write register.

Interrupt Registers

When there are counters, user interrupts, or fatal bits in a chiplet, a MAIN INTERRUPT register will be present. Bit 0 always is the fatal interrupt bit, which is set as soon as any of the fatal interrupt events occur. The other bits refer to counters or user interrupt registers to allow easy determination of the interrupt cause. Each Interrupt register has an interrupt MASK register to enable or disable interrupt. After power on Reset, interrupts are disabled. The interrupt registers are the same as the event latch registers, with the addition that when an interrupt register bit is set, and the corresponding mask register bit is set, the interrupt signal to the GPPINT chiplet is activated. The same mechanism to reset the interrupt register bits is used as for the event latch registers. The interrupt MASK registers are only changed by the microprocessor. The interrupt and interrupt mask registers are read/write.

Configuration Registers

These registers are programmed by the microprocessor with setup information, and are read/write. The first configuration register reserves bit 1 and seven to configure explicit or implicit reset of the event latch registers and interrupt registers respectively (when such registers are present).

Register Types

F Read-On-The-Fly register (auto-generated)

N Counter register

R Reset register

I Interrupt register (auto-generated)

C Configuration register

X Control or mask register (auto-generated)

S Status (event latch) register

O Command register

ATM Cell Handler Architecture : Transmit Direction

ACH_Tx GPP Handler Address Mapping Base Address = x'100'

Register Name	Description	Address Offset	Type Width	Initial Value
ROFmid	Read-on-the-fly register	X'0'	F 8	'00000000'
ROFhi	Read-on-the-fly register (MSByte)	X'1'	F 8	'00000000'
CntEn1	COUNT ENABLE register	X'2'	X 3	'000'
ACBC	Cell counter (read from external FIFO), no threshold[2]	X'4/5'[2]	N 24	'x'000000''
IUC	Idle/unassigned cell counter, no threshold[2]	X'6/7'[2]	N 24	'x'000000''
ACBE	Corrupted cell error counter[2]	X'8/9'[2]	N 8	'00000000'
ACBETh11	Threshold register for counter ACBE	X'A'	X 8	'10000000'
RESET	Default RESET register	X'30'	R 2	'01'
STAT1	Status register #1	X'33'	S 8	
IUCSTAT1	Status register #2	X'34'	S 2	
MainIRQ	MAIN INTerrupt register	X'38'	I 2	
M_MainIRQ	INT MASK register (for MainIRQ)	X'39'	X 2	'00'
CntrIRQ1	COUNTER INTerrupt register	X'3A'	I 4	
M_CntrIRQ1	INT MASK register (for CntrIRQ1)	X'3B'	X 4	'0000'
CELLTENABLE	Chiplet cofiguration register	X'48'	C 6	'001111'
ACBTXTHRPAE	Programmable almost empty threshold	X'49'	C 7	'0001110'
HEADERBYTE1	IU-cell header byte 1[1]	X'4A'	C 8	'00000000'
HEADERBYTE2	IU-cell header byte 2[1]	X'4B'	C 8	'00000000'
HEADERBYTE3	IU-cell header byte 3[1]	X'4C'	C 8	'00000000'
HEADERBYTE4	IU-cell header byte 4[1]	X'4D'	C 8	'00000001'
HEADERBYTE5	IU-cell header byte 5[1]	X'4E'	C 8	'01010010'
PAYLOADBYTE	IU-cell payload byte	X'4F'	C 8	'01101010'
HECENCTRL	HEC processing control	X'50'	C 7	'0001100'
HECOFFSET	HEC offset pattern register	X'51'	C 8	'01010101'
HECMASKAND	HEC error corruption mask (AND)	X'52'	C 8	'11111111'
HECMASKOR	HEC error corruption mask (OR)	X'53'	C 8	'00000000'
SDBTXTHRPAF	Programmable almost full threshold	X'54'	C 6	'110000'

1. Defaults according ITU I.432
2. Independent of the counter width, given that a counter has chiplet address N as a base. Reading address N or address N-1 both yield the least significant byte of the counter. Reading address N has no affect on the counter, but reading address N-1 resets the counter after read operation.

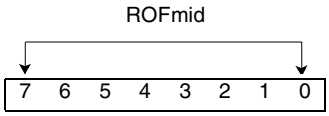
24: ACH Tx Register Description

Counter Registers

24.1: ROFmid

Read-on-the-fly register, middle significant byte.

Length 8 bits
Type Read
Address 100
Power On Value X'00'

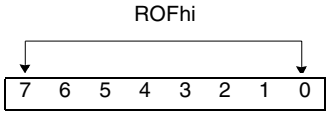


Bit(s)	Name	Description
7-0	ROFmid(7-0)	Read-on-the-fly register, middle significant byte

24.2: ROFhi

Read-on-the-fly register, most significant byte.

Length 8 bits
Type Read
Address 101
Power On Value X'00'



Bit(s)	Name	Description
7-0	ROFhi(7-0)	Read-on-the-fly register, most significant byte

24.3: ACBC

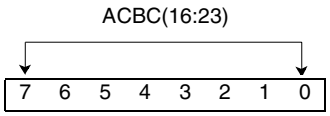
Number of cells read from external FIFO (24-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 104/105

Power On Value X'00'



Bit(s)	Name	Description
7-0	ACBC(16:23)	External FIFO cell counter, least significant byte

24.4: IUC

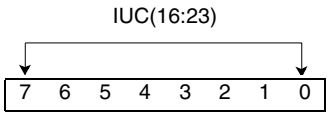
Number of transmitted idle and unassigned cells (24-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 106/107

Power On Value X'00'



Bit(s)	Name	Description
7-0	IUC(16:23)	Idle/unassigned cell counter, least significant byte

24.5: ACBE

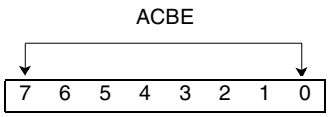
Number of errors (corrupted cell read from external FIFO). Eight-bit counter overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 108/109

Power On Value X'00'



Bit(s)	Name	Description
7-0	ACBE(7-0)	External FIFO error counter

24.6: ACBETh11

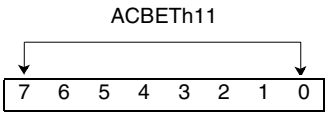
Threshold for number of errors. Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 10A

Power On Value X'80'



Bit(s)	Name	Description
7-0	ACBETh11(7-0)	Threshold for error counter

24.7: CntEn1

Counter On/Off control register for ACH_Tx.

For each bit position:

0 = Counter is disabled.

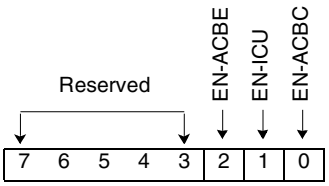
1 = Counter is enabled.

Length 8 bits

Type Read/Write

Address 102

Power On Value X'00'



Bit(s)	Name	Description
7-3	Reserved	Reserved
2	EN-ACBE	Error counter enable
1	EN-IUC	Idle/unassigned cell counter enable
0	EN-ACBC	Cell counter enable

24.8: Reset Register (RESET)

Reset/Halt chiplet control register. This register is automatically preset to the default value by the reset signal ResHT from the GPPINT.

For each bit position:

0 = Reset/Halt not active.

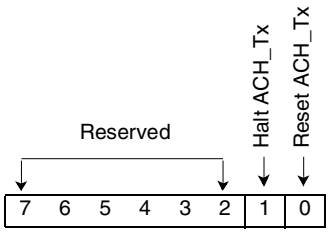
1 = Reset/Halt active.

Length 8 bits

Type Read/Write

Address 130

Power On Value X'01'



Bit(s)	Description
7-2	Reserved
1	Halt (freeze) ACH_Tx chiplet
0	Reset (disable) ACH_Tx chiplet

Status Registers

24.9: STAT1

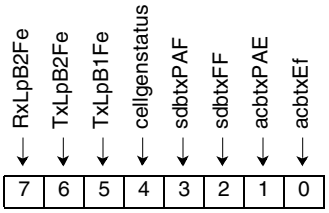
Status register #1 of this chiplet. This is an event latch register.

Length 8 bits

Type Read/Write

Address 133

Power On Value -



Bit(s)	Name	Description
7	RxLpB2Fe	Rx Loopback #2 configuration mismatch
6	TxLpB2Fe	Tx Loopback #2 configuration mismatch
5	TxLpB1Fe	Tx Loopback #1 configuration mismatch
4	cellgenstatus	0 = Idle/unassigned cell is transmitted 1 = Cell from external FIFO is transmitted
3	sdbtxPAF	Programmable almost full flag from SDB_Tx
2	sdbtxFF	FIFO full flag from SDB_Tx
1	acbtxPAE	Programmable almost empty flag from External Transmit FIFO
0	acbtxEf	FIFO empty flag from external Transmit FIFO

24.10: IUCSTAT1

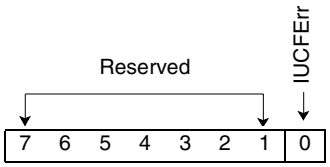
Status register #2 of this chiplet. This is an event latch register.

Length 8 bits

Type Read/Write

Address 134

Power On Value -



Bit(s)	Name	Description
7-1	Reserved	Reserved
0	IUCFErr	Unexpected state transition in FSM

Interrupt Request and Mask Registers

24.11: MainIRQ

Register to indicate fatal interrupt events and to point to user IRQ registers with active requests.

For each bit position;

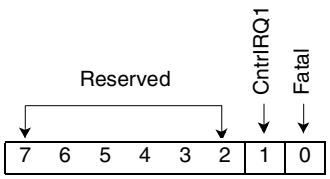
0 = No interrupt request pending.
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 138

Power On Value -



Bit(s)	Name	Description
7-2	Reserved	Reserved
1	CntrIRQ1	Active request in CntrIRQ1 register
0	Fatal	Fatal event occured

24.12: M_MainIRQ

Register to mask pending interrupt requests. A masked request will not generate an outgoing IRQ to the GPPINT.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

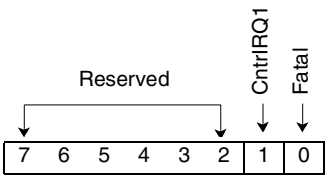
1 = The corresponding pending request bit activates signal IRQHT1 to GPPINT.

Length 8 bits

Type Read/Write

Address 139

Power On Value X'00'



Bit(s)	Name	Description
7-2	Reserved	Reserved
1	CntrIRQ1	Active request in CntrIRQ1 register
0	Fatal	Fatal event occured

24.13: CntrIRQ1

Register to indicate active counter interrupt requests of this chiplet.

For each bit position:

0 = No interrupt request pending.

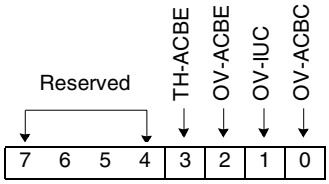
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 13A

Power On Value -



Bit(s)	Name	Description
7-4	Reserved	Reserved
3	TH-ACBE	Threshold overstep error counter
2	OV-ACBE	Overflow error counter
1	OV-IUC	Overflow idle/unassigned cell counter
0	OV-ACBC	Overflow cell counter

24.14: M_CntrIRQ1

Register to mask pending counter interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

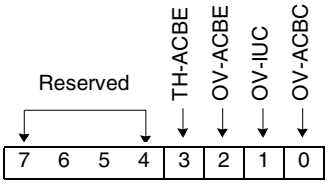
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 13B

Power On Value X'00'



Bit(s)	Name	Description
7-4	Reserved	Reserved
3	TH-ACBE	Threshold overstep error counter
2	OV-ACBE	Overflow error counter
1	OV-IUC	Overflow idle/unassigned cell counter
0	OV-ACBC	Overflow cell counter

Configuration Registers

24.15: CELLTENABLE

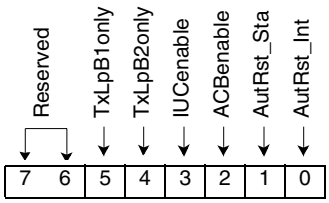
Register to control various modes of operation of this chiplet.

Length 8 bits

Type Read/Write

Address 148

Power On Value X'0F'



Bit(s)	Name	Description
7-6	Reserved	Reserved
5	TxLpB1only	0 = On the fly monitoring (LpB #1) 1 = Loopback #1 only
4	TxLpB2only	0 = On the fly monitoring (LpB #2) 1 = Loopback #2 only
3	IUCenable	0 = Generation of IUC disabled 1 = Generation of IUC enabled
2	ACBenable	0 = External FIFO read disabled 1 = External FIFO read enabled
1	AutRst_Sta	0 = No action on read access 1 = Auto-reset status registers upon read access
0	AutRst_Int	0 = No action on read access 1 = Auto-reset interrupt request registers upon read access

24.16: ACBTXTHRPAE

Threshold for Programmable Almost Empty flag of external FIFO in transmit direction.

Length 8 bits

Type Read/Write

Address 149

Power On Value X'0E'

Bit(s)	Name	Description
7	Reserved	Reserved
6-0	ACBTXTHRPAE(7-1)	Threshold for PAE flag

24.17: SDBTXTHRPAF

Threshold for Programmable Almost Full flag (SDB_Tx).

Length 8 bits

Type Read/Write

Address 154

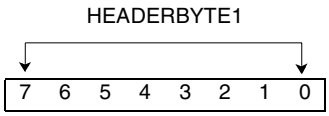
Power On Value X'30'

Bit(s)	Name	Description
7-6	Reserved	Reserved
5-0	SDBTXTHRPAF(7-2)	Threshold for PAF flag

24.18: HEADERBYTE1

Idle/Unassigned cell header byte #1. Default pattern according to ITU I.432.

Length 8 bits
Type Read/Write
Address 14A
Power On Value X'00'

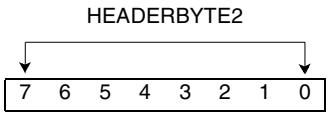


Bit(s)	Name	Description
7-0	HEADERBYTE1(7-0)	IU-cell header byte #1

24.19: HEADERBYTE2

Idle/Unassigned cell header byte #2. Default pattern according to ITU I.432.

Length 8 bits
Type Read/Write
Address 14B
Power On Value X'00'



Bit(s)	Name	Description
7-0	HEADERBYTE2(7-0)	IU-cell header byte #2

24.20: HEADERBYTE3

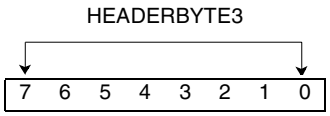
Idle/Unassigned cell header byte #3. Default pattern according to ITU I.432.

Length 8 bits

Type Read/Write

Address 14C

Power On Value X'00'



Bit(s)	Name	Description
7-0	HEADERBYTE3(7-0)	IU-cell header byte #3

24.21: HEADERBYTE4

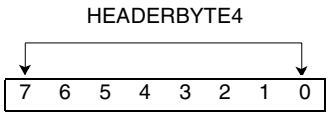
Idle/Unassigned cell header byte #4. Default pattern according to ITU I.432.

Length 8 bits

Type Read/Write

Address 14D

Power On Value X'01'

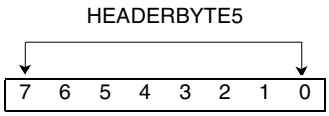


Bit(s)	Name	Description
7-0	HEADERBYTE4(7-0)	IU-cell header byte #4

24.22: HEADERBYTE5

Idle/Unassigned cell header byte #5. Default pattern according to ITU I.432.

Length 8 bits
Type Read/Write
Address 14E
Power On Value X'52'

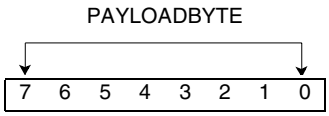


Bit(s)	Name	Description
7-0	HEADERBYTE5(7-0)	IU-cell header byte #5

24.23: PAYLOADBYTE

Idle/Unassigned cell payload byte.

Length 8 bits
Type Read/Write
Address 14F
Power On Value X'6A'



Bit(s)	Name	Description
7-0	PAYLOADBYTE(7-0)	IU-cell payload byte

24.24: HECENCTRL

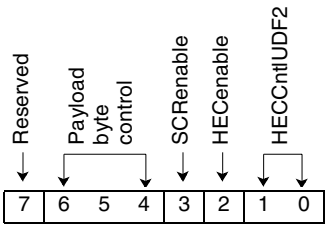
HEC processing control configuration register.

Length 8 bits

Type Read/Write

Address 150

Power On Value X'0C'



Bit(s)	Name	Description
7		Reserved
6-4	Payload byte control	000 Each payload byte is the same (default) 001 Increment payload byte for each ATM cell, start with default after reset 010 Increment each payload byte of a cell; start each cell with default byte 011 Increment each PL byte of a cell; cross cell boundaries; start first cell after reset with default byte 1xx Each payload byte is the same
3	SCRenable	0 = ATM cell payload scrambling disabled 1 = ATM cell payload scrambling enabled
2	HECenable	0 = HEC calculation/manipulation disabled 1 = HEC calculation/manipulation enabled
1-0	HECCntlUDF2	Mode of final HEC manipulation by UDF1 byte after HECOffset, HECMaskAND, HECMaskOR operations: 00 No manipulation 01 HEC XOR UDF1 10 HEC AND UDF1 11 HEC OR UDF1

24.25: HECOFFSET

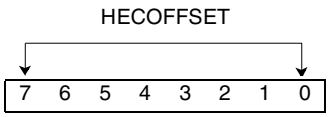
HEC offset pattern register for the byte pattern used in the ATM cell header HEC calculation as base offset according to ITU I.432.

Length 8 bits

Type Read/Write

Address 151

Power On Value X'55'



Bit(s)	Name	Description
7-0	HECOFFSET(7-0)	HEC offset pattern

24.26: HECMASKAND

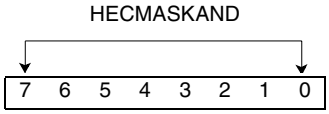
HEC mask pattern register for the byte pattern used in the ATM cell header HEC calculation as dedicated (ANDing) HEC error corruption mask.

Length 8 bits

Type Read/Write

Address 152

Power On Value X'FF'



Bit(s)	Name	Description
7-0	HECMASKAND(7-0)	HEC error corruption mask (AND)

24.27: HECMASKOR

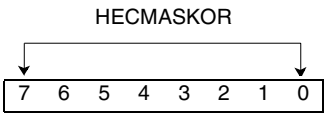
HEC mask pattern register for the byte pattern used in the ATM cell header HEC calculation as dedicated (ORing) HEC error corruption mask.

Length 8 bits

Type Read/Write

Address 153

Power On Value X'00'



Bit(s)	Name	Description
7-0	HECMASKOR(7-0)	HEC error corruption mask (OR)

ATM Cell Handler Architecture: Receive Direction

ACH_Rx GPP Handler Address Mapping Base Address = x'200'

Register Name	Description	Address Offset	Type Width	Initial Value
ROFmid	Read-on-the-fly register	X'0'	F 8	'00000000'
ROFhi	Read-on-the-fly register (MSByte)	X'1'	F 8	'00000000'
CntEn1	COUNT ENABLE register	X'2'	X 4	'0000'
FHR	Counter, ATM cells written into external FIFO, no threshold	X'4/5'[1]	N 24	'x'000000"
IHR	Counter, received Idle cells from OFP, no threshold	X'6/7'[1]	N 24	'x'000000"
EHR1	Counter, detected HEC errors with threshold	X'8/9'[1]	N 16	'x'0000"
EHR1Th12	Threshold reg Byte2 (LSByte) for counter EHR1	X'A'	X 8	'00000001'
EHR1Th11	Threshold reg Byte1 for counter EHR1	X'B'	X 8	'10000000'
BHR	Counter, FIFO full discarded cells: (DiscPAF1=1) AND (TxLpB11=0) with threshold.[1]	X'C/D'[1]	N 16	'x'0000"
BHRTh12	Threshold reg Byte2 (LSByte) for counter BHR	X'E'	X 8	'00000001'
BHRTh11	Threshold register Byte1 for counter BHR	X'F'	X 8	'10000000'
RESET	Default RESET register	X'30'	R 2	'01'
CMD1	Command register (FIFO reset)	X'31'	O 2	'00'
STAT1	Status register	X'33'	S 6	
MainIRQ	MAIN INTerrupt register	X'38'	I 2	
M_MainIRQ	INTerrupt MASK register (for MainIRQ)	X'39'	X 2	'00'
CntrIRQ1	COUNTER INTerrupt register	X'3A'	I 6	
M_CntrIRQ1	INTerrupt MASK register (for CntrIRQ1)	X'3B'	X 6	'000000'
CONF5	Chiplet configuration register	X'48'	C 8	'00000011'
CONF6	Chiplet configuration register (Alpha/Delta)	X'49'	C 8	'01100101'
H1CONF	Confirmation bytes to identify idle or unassigned cells.	X'4A'	C 8	'00000000'
H2CONF	Confirmation bytes to identify idle or unassigned cells.	X'4B'	C 8	'00000000'
H3CONF	Confirmation bytes to identify idle or unassigned cells.	X'4C'	C 8	'00000000'
H4CONF	Confirmation bytes to identify idle or unassigned cells.	X'4D'	C 8	'00000001'
H5CONF	Dummy byte to align Payload in external FIFO	X'4E'	C 8	'11010000'
CONFC	External FIFO buffer Almost Full threshold	X'4F'	C 7	'1100000'

1. Independent of the counter width, given that a counter has chiplet address N as a base. Reading address N or address N-1 both yield the least significant byte of the counter. Reading address N has no affect on the counter, but reading address N-1 resets the counter after read operation.

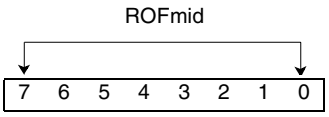
ACH_Rx Register Description

Counter Registers

24.28: ROFmid

Read-on-the-fly registers, middle significant byte.

Length 8 bits
Type Read
Address 200
Power On Value X'00'

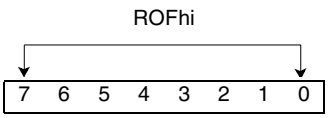


Bit(s)	Name	Description
7-0	ROFmid(7-0)	Read-on-the-fly register, middle significant byte

24.29: ROFhi

Read-on-the-fly registers, most significant byte.

Length 8 bits
Type Read
Address 201
Power On Value X'00'

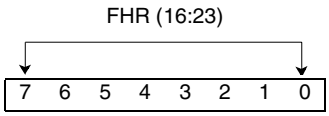


Bit(s)	Name	Description
7-0	ROFhi(7-0)	Read-on-the-fly register, most significant byte

24.30: FHR

Number of ATM cells written into external FIFO (24-bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 204/205
Power On Value X'00'

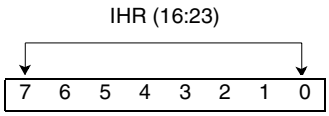


Bit(s)	Name	Description
7-0	FHR(16:23)	ATM cell counter, least significant byte

24.31: IHR

Number of idle cells received from OFP_Rx (24-bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 206/207
Power On Value X'00'



Bit(s)	Name	Description
7-0	IHR(16:23)	Idle/unassigned cell counter, least significant byte

24.32: EHR1

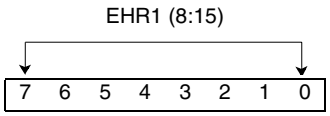
Number of detected HEC errors (16-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 208/209

Power On Value X'00'



Bit(s)	Name	Description
7-0	EHR1(8:15)	HEC error counter, least significant byte

24.33: EHR1Th11

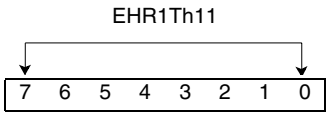
Threshold for number of HEC cells, most significant byte.

Length 8 bits

Type Read/Write

Address 20B

Power On Value X'80'

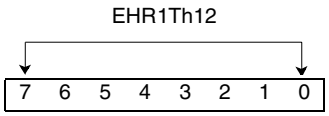


Bit(s)	Name	Description
7-0	EHR1Th11(7-0)	Threshold for HEC error counter, most significant byte

24.34: EHT1Th12

Threshold for number of HEC errors (least significant byte). Threshold overstep leads to an interrupt request.

Length 8 bits
Type Read/Write
Address 207
Power On Value X'01'

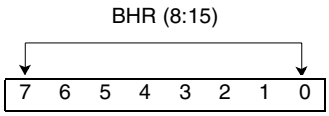


Bit(s)	Name	Description
7-0	EHR1Th12(7-0)	Threshold for HEC error counter, least significant byte

24.35: BHR

Number of discarded cells because of FIFO full condition (16 bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 20C/20D
Power On Value X'00'



Bit(s)	Name	Description
7-0	BHR(8:15)	Discarded cell counter, least significant byte

24.36: BHRTh11

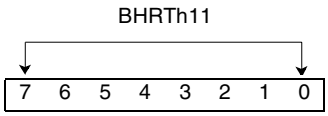
Threshold for number of discarded cells, most significant byte.

Length 8 bits

Type Read/Write

Address 20F

Power On Value X'80'



Bit(s)	Name	Description
7-0	BHRTh11(7-0)	Threshold for discarded cell counter, most significant byte

24.37: BHRTh12

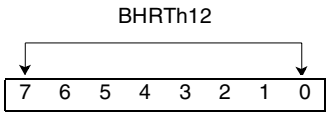
Threshold for number of discarded cells (least significant byte). Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 20E

Power On Value X'01'



Bit(s)	Name	Description
7-0	BHRTh12(7-0)	Threshold for discarded cell counter, least significant byte

24.38: CntEn1

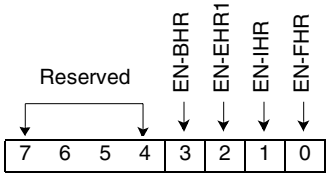
Counter On/Off control register for ACH_Rx.

For each bit position:

0 = Counter is disabled.

1 = Counter is enabled.

Length	8 bits
Type	Read/Write
Address	202
Power On Value	X'00'



Bit(s)	Name	Description
7-4	Reserved	Reserved
3	EN-BHR	Discarded cell counter enable
2	EN-EHR1	HEC error counter enable
1	EN-IHR	Idle cell counter enable
0	EN-FHR	ATM cell counter enable

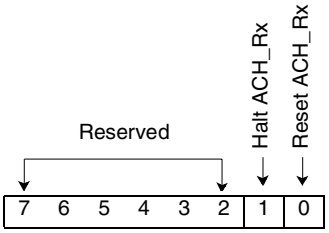
24.39: Reset Register (RESET)

Reset/Halt chiplet control register. This register is automatically preset to the default value by the reset signal ResHR from the GPPINT.

For each bit position:

0 = Reset/Halt not active.
1 = Reset/Halt active.

Length 8 bits
Type Read/Write
Address 230
Power On Value X'01'

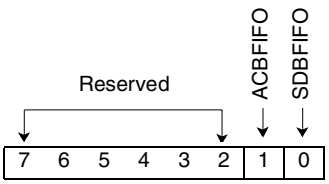


Bit(s)	Description
7-2	Reserved
1	Halt (freeze) ACH_Rx chiplet
0	Reset (disable) ACH_Rx chiplet

24.40: Command Register (CMD1)

Command register for this chiplet. Single-cycle active if '1' is written into bit position.

Length 8 bits
Type Read/Write
Address 231
Power On Value X'00'



Bit(s)	Name	Description
7-2	Reserved	Reserved
1	ACBFIFO	Reset External FIFO
0	SDBFIFO	Reset SDB_Rx FIFO

24.41: Status Register (STAT1)

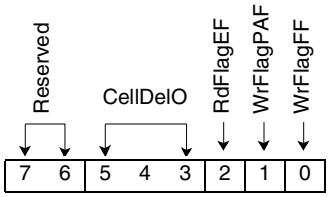
Status register of this chiplet. This is an event latch register.

Length 8 bits

Type Read/Write

Address 233

Power On Value -



Bit(s)	Name	Description
7-6	Reserved	Reserved
5-3	CellDelO	State of the cell delineation process : 000 Reset state 001 Hunt state 010 Presync state 100 Sync state
2	RdFlagEF	SDB FIFO: Read FIFO Empty Flag
1	WrFlagPAF	External FIFO: Write FIFO programmable almost full flag
0	WrFlagFF	External FIFO: Write FIFO Full flag

Interrupt Request and Mask Registers

24.42: MainIRQ

Register to indicate fatal interrupt events and to point to user IRQ registers with active requests.

For each bit position:

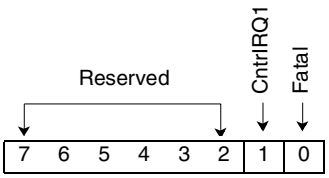
0 = No interrupt request pending.
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 238

Power On Value -



Bit(s)	Name	Description
7-2	Reserved	Reserved
1	CntrIRQ1	Active request in CntrIRQ1 register
0	Fatal	Fatal event occured

24.43: M_MainIRQ

Register to mask pending interrupt requests. A masked request will not generate an outgoing IRQ to the GPPINT.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

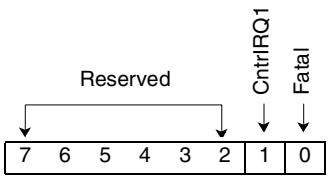
1 = The corresponding pending request bit activates signal IRQHR1 to GPPINT.

Length 8 bits

Type Read/Write

Address 239

Power On Value X'00'



Bit(s)	Name	Description
7-2	Reserved	Reserved
1	CntrIRQ1	Active request in CntrIRQ1 register
0	Fatal	Fatal event occured

24.44: CntrIRQ1

Register to indicate active counter interrupt requests of this chiplet.

For each bit position:

0 = No interrupt request pending.

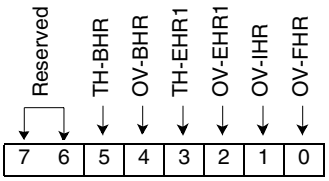
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 240

Power On Value -



Bit(s)	Name	Description
7-6	Reserved	Reserved
5	TH-BHR	Threshold overstep discarded cell counter
4	OV-BHR	Overflow discarded cell counter
3	TH-EHR1	Threshold overstep HEC error counter
2	OV-EHR1	Overflow HEC error counter
1	OV-IHR	Overflow idle cell counter
0	OV-FHR	Overflow ATM cell counter

24.45: M_CntrIRQ1

Register to mask pending counter interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

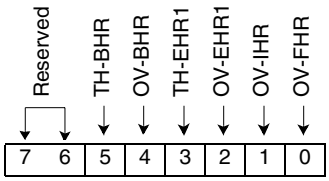
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 241

Power On Value X'00'



Bit(s)	Name	Description
7-6	Reserved	Reserved
5	TH-BHR	Threshold overstep discarded cell counter
4	OV-BHR	Overflow discarded cell counter
3	TH-EHR1	Threshold overstep HEC error counter
2	OV-EHR1	Overflow HEC error counter
1	OV-IHR	Overflow idle cell counter
0	OV-FHR	Overflow ATM cell counter

Configuration Registers

24.46: CONF5

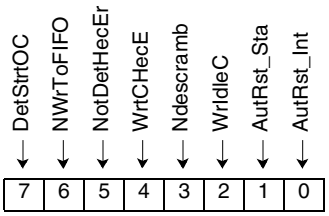
Register to control various modes of operation of this chiplet.

Length 8 bits

Type Read/Write

Address 248

Power On Value X'03'

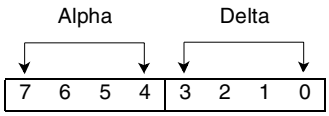


Bit(s)	Name	Description
7	DetStrtOC	0 Do not detect start of cell 1 Detect start of cell
6	NWrToFIFO	0 Write into ACB FIFO 1 Do not write into ACB FIFO; all received cells are discarded
5	NotDetHecEr	0 Detect ATM cell with HEC errors 1 Do not detect ATM cell with HEC errors
4	WrtCHecE	0 Do not write ATM cell with HEC errors 1 Write ATM cell with HEC errors
3	Ndescramb	0 Descramble ATM cell payload 1 Do not descramble ATM cell payload
2	WrIdleC	0 Do not write Idle cell into external FIFO 1 Write Idle cell into ACB FIFO
1	AutRst_Sta	0 No action on read access 1 Auto-reset status register upon read access
0	AutRst_Int	0 No action on read access 1 Auto-reset interrupt request registers upon read access

24.47: CONF6

Register to control ATM cell synchronization in this chiplet.

Length 8 bits
Type Read/Write
Address 249
Power On Value X'65'

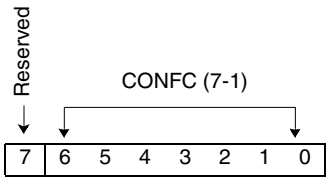


Bit(s)	Name	Description
7-4	Alpha(7-4)	Required number of consecutive false HEC detected to return from SYNC to HUNT state
3-0	Delta(7-4)	Required number of consecutive good HEC detected to jump from PRESYNC to SYNC state

24.48: CONFC

Threshold for Programmable Almost Full flag of the external FIFO.

Length 8 bits
Type Read/Write
Address 24F
Power On Value X'60'



Bit(s)	Name	Description
7	Reserved	Reserved
6-0	CONFC(7-1)	Threshold for PAF flag HEADERBYTE1/2/3/4/5: Idle/Unassigned cell header bytes, default pattern according to ITU I.432.

24.49: H1CONF

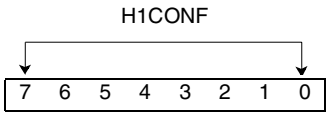
Header pattern #1 to identify idle/unassigned cells.

Length 8 bits

Type Read/Write

Address 24A

Power On Value X'00'



Bit(s)	Name	Description
7-0	H1CONF(7-0)	Header byte #1

24.50: H2CONF

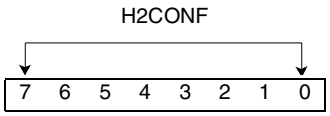
Header pattern #2 to identify idle/unassigned cells.

Length 8 bits

Type Read/Write

Address 24B

Power On Value X'00'

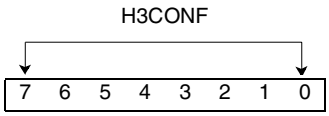


Bit(s)	Name	Description
7-0	H2CONF(7-0)	Header byte #2

24.51: H3CONF

Header pattern #3 to identify idle/unassigned cells.

Length 8 bits
Type Read/Write
Address 24C
Power On Value X'00'

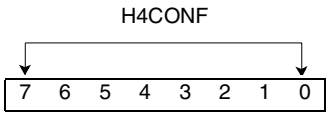


Bit(s)	Name	Description
7-0	H3CONF(7-0)	Header byte #3

24.52: H4CONF

Header pattern #4 to identify idle/unassigned cells.

Length 8 bits
Type Read/Write
Address 24D
Power On Value X'01'



Bit(s)	Name	Description
7-0	H4CONF(7-0)	Header byte #4

24.53: H5CONF

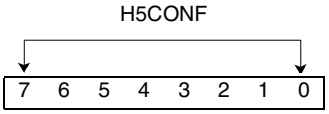
Dummy byte to align the Payload in the ACB_Rx buffer.

Length 8 bits

Type Read/Write

Address 24E

Power On Value X'D0'



Bit(s)	Name	Description
7-0	H5CONF(7-0)	Payload alignment byte

Overhead Frame Processor Architecture: Transmit Direction

OFP_Tx GPP Handler Address Mapping Base Address = x'400' (Page 1 of 3)

Register Name	Description	Address Offset	Type Width	Initial Value
CntEn1	COUNT ENABLE register	X'2'	X 4	'0000'
PTRINC	Pointer increment event counter[1]	X'4/5'[1]	N 8	'00000000'
	No threshold		N 8	
PTRDEC	Pointer decrement event counter[1]	X'6/7'[1]	N 8	'00000000'
	No threshold		N 8	
ND_EVCNT	New data event counter, no threshold[1]	X'8/9'[1]	N 8	'00000000'
JUSCNT	Justification error counter[1]	X'A/B'[1]	N 8	'00000000'
	With no threshold		N 8	
JUSCNTTh11	Threshold register for counter JUSCNT	X'C'	X 8	'10000000'
RESET	Default RESET register	X'30'	R 2	'01'
CMD1	Njus, Pjus, NDF	X'31'	O 3	'000'
STAT1	Init, hug, mode(7-5)	X'33'	S 6	
STAT2	Njus, Pjus, NDF	X'34'	S 3	
MainIRQ	MAIN INTerrupt register	X'38'	I 3	
M_MainIRQ	INTerrupt MASK register (for MainIRQ)	X'39'	X 3	'000'
CntrIRQ1	COUNTER INTerrupt register	X'3A'	I 5	
M_CntrIRQ1	INTerrupt MASK register (for CntrIRQ1)	X'3B'	X 5	'00000'
IRQ3	USER INTerrupt register	X'3C'	I 6	
M_IRQ3	INTerrupt MASK register (for IRQ3)	X'3D'	X 6	'000000'
CONF1	Configuration register #1 (general A)	X'48'	C 8	'00000011'
CONF2	Configuration register #2 (general B)	X'49'	C 3	'000'
CONF3	Configuration register #3	X'4A'	C 8	'11111110'
	(fscr reload pattern)		N 8	
CONF4	Configuration register #4 (errmask)	X'4B'	C 8	'00000000'
CONF5	Configuration register #5 (erraddress)	X'4C'	C 8	'00000000'
CONF6	Configuration register #6 (fscr control)	X'4D'	C 8	'00000001'
CONF7	Configuration register #7 (DCC control)	X'4E'	C 4	'0000'
CONF8	Configuration register #8 (ThrLoW)	X'4F'	C 6	'000011'
CONF9	Configuration register #9 (ThrNoW)	X'50'	C 6	'010001'
CONF10	Configuration register #10 (ThrHiW)	X'51'	C 6	'100000'
SOH-A11	First A1	X'100'	8	
SOH-A12	Second1 A1	X'101'	8	
SOH-A13	Third A1	X'102'	8	
SOH-A21	First A2	X'103'	8	
SOH-A22	Second A2	X'104'	8	

1. Independent of the counter width, given that a counter has chiplet address N as a base. Reading address N or address N-1 both yield the least significant byte of the counter. Reading address N has no affect on the counter, but reading address N-1 resets the counter after read operation
2. Address range 100-17F located in 128x8 GRA. Address range 180-1BF located in 64x8 GRA.
3. The 64-byte J1 path trace processing uses the 16-byte addresses of 16 byte J1 path trace to map a full 64 byte space

OFP_Tx GPP Handler Address Mapping Base Address = x'400' (Page 2 of 3)

Register Name	Description	Address Offset	Type Width	Initial Value
SOH-A23	Third A2	X'105'	8	
SOH-J0	J0	X'106'	8	
	Reserved for national use and not included in frame scrambling	X'107-8'	8	
SOH-B1	B1	X'109'	8	
	Media dependant bytes	X'10A-0B'	8	
SOH-E1	E1	X'10C'	8	
	Media dependant byte	X'10D'	8	
	Reserved for future standardization	X'10E'	8	
SOH-F1	F1	X'10F'	8	
	Reserved for national use	X'110-11'	8	
SOH-D1	D1	X'112'	8	
	Media dependant bytes	X'113-14'	8	
SOH-D2	D2	X'115'	8	
	Media dependant byte	X'116'	8	
	Reserved for future standardization	X'117'	8	
SOH-D3	D3	X'118'	8	
	Reserved for future standardization	X'119-1A'	8	
SOH-H1	H1	X'11B'	8	
SOH-J0		X'11C-1D'	8	b'1001SS11' with S unspecified
SOH-H2	H2	X'11E'	8	
SOH-1s	x'FF'	X'11F-20'	8	
SOH-H31	First H3	X'121'	8	
SOH-H32	Second H3	X'122'	8	
SOH-H23	Third H3	X'123'	8	
SOH-B21	First B2	X'124'	8	
SOH-B22	Second B2	X'125'	8	
SOH-B23	Third B2	X'126'	8	
SOH-K1	K1	X'127'	8	
	Reserved for future standardization	X'128-29'	8	
SOH-K2	K2	X'12A '	8	
	Reserved for future standardization	X'12B-2C'	8	
SOH-D4	D4	X'12D'	8	
	Reserved for future standardization	X'12E-2F'	8	
SOH-D5	D5	X'130'	8	
	Reserved for future standardization	X'131-32'	8	
SOH-D6	D6	X'133'	8	

1. Independent of the counter width, given that a counter has chiplet address N as a base. Reading address N or address N-1 both yield the least significant byte of the counter. Reading address N has no affect on the counter, but reading address N-1 resets the counter after read operation
2. Address range 100-17F located in 128x8 GRA. Address range 180-1BF located in 64x8 GRA.
3. The 64-byte J1 path trace processing uses the 16-byte addresses of 16 byte J1 path trace to map a full 64 byte space

OFP_Tx GPP Handler Address Mapping Base Address = x'400' (Page 3 of 3)

Register Name	Description	Address Offset	Type Width	Initial Value
	Reserved for future standardization	X'134-35'	8	
SOH-D7	D7	X'136'	8	
	Reserved for future standardization	X'137-38'	8	
SOH-D8	D8	X'139'	8	
	Reserved for future standardization	X'13A-3B'	8	
SOH-D9	D9	X'13C'	8	
	Reserved for future standardization	X'13D-3E'	8	
SOH-D10	D10	X'13F'	8	
	Reserved for future standardization	X'140-41'	8	
SOH-D11	D11	X'142'	8	
	Reserved for future standardization	X'143-44'	8	
SOH-D12	D12	X'145'	8	
	Reserved for future standardization	X'146-47'	8	
SOH-S1	S1	X'148'	8	
SOH-Z11	Reserved for future standardization	X'149-4C'	8	
SOH-M1	M1	X'14D'	8	
SOH-M1	E2	X'14E'	8	
	Reserved for future standardization	X'14F-50'	8	
	Justification stuff bytes	X'151-53'	8	
POH-J1	J1	X'154'	8	
POH-B3	B3	X'155'	8	
POH-C2	C2	X'156'	8	
POH-G1	G1	X'157'	8	
POH-F2	F2	X'158'	8	
POH-H4	H4	X'159'	8	
POH-F3	F3	X'15A'	8	
POH-K3	K3	X'15B'	8	
X'POH-N1'	N1	15C	8	
	Reserved	X'15D-5F'	8	
POH-J0-16	16 byte J0 section trace	X'160-6F'	8	
	Reserved	X'170-7F'	8	
POH-J1-16	16-byte J1 path trace[3]	X'180-8F'	8	
POH-J1-64	64-byte J1 path trace[3]	X'190-BF'	8	

1. Independent of the counter width, given that a counter has chiplet address N as a base. Reading address N or address N-1 both yield the least significant byte of the counter. Reading address N has no affect on the counter, but reading address N-1 resets the counter after read operation
2. Address range 100-17F located in 128x8 GRA. Address range 180-1BF located in 64x8 GRA.
3. The 64-byte J1 path trace processing uses the 16-byte addresses of 16 byte J1 path trace to map a full 64 byte space

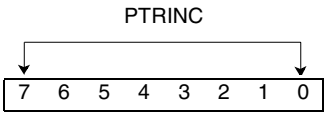
OFP_Tx Register Description

Counter Registers

24.54: PTRINC

Number of pointer increment events (eight-bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 404/405
Power On Value X'00'

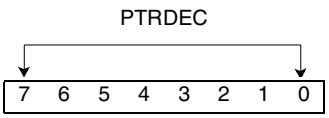


Bit(s)	Name	Description
7-0	PTRINC(7-0)	Pointer increment counter

24.55: PTRDEC

Number of pointer decrement events (eight-bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 406/407
Power On Value X'00'

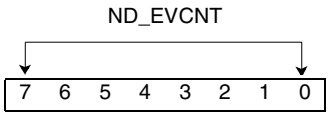


Bit(s)	Name	Description
7-0	PTRDEC(7-0)	Pointer decrement counter

24.56: ND_EVCNT

Number of new data events (eight-bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 408/409
Power On Value X'00'

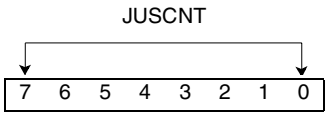


Bit(s)	Name	Description
7-0	ND_EVCNT(7-0)	New data event counter

24.57: JUSCNT

Number of justification errors detected (eight-bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 40A/40B
Power On Value X'00'



Bit(s)	Name	Description
7-0	JUSCNT(7-0)	Justification error counter

24.58: JUSCNTTh11

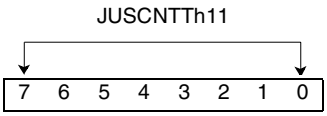
Threshold for number of justification errors. Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 40C

Power On Value X'80'



Bit(s)	Name	Description
7-0	JUSCNTTh11(7-0)	Threshold for justificication error counter

24.59: CntEn1

Counter On/Off control register for OFP_Tx.

For each bit position:

0 = Counter is disabled.

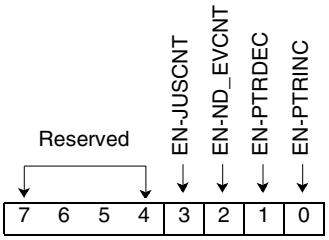
1 = Counter is enabled.

Length 8 bits

Type Read/Write

Address 402

Power On Value X'00'



Bit(s)	Name	Description
7-4	Reserved	Reserved
3	EN-JUSCNT	Justification error counter enable
2	EN-ND_EVCNT	New data event counter enable
1	EN-PTRDEC	Pointer decrement counter enable
0	EN-PTRINC	Pointer increment counter enable

24.60: Reset Register (RESET)

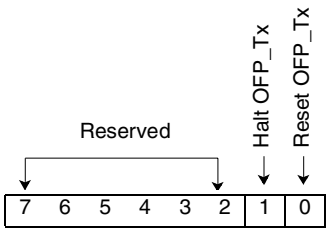
Reset/Halt chiplet control register. This register is automatically preset to the default value by the reset signal ResOT coming from GPPINT chiplet.

For each bit position:

0 = Reset/Halt not active.

1 = Reset/Halt active.

Length	8 bits
Type	Read/Write
Address	430
Power On Value	X'01'



Bit(s)	Description
7-2	Reserved
1	Halt (freeze) OFP_Tx chiplet
0	Reset (disable) OFP_Tx chiplet

24.61: Command Register (CMD1)

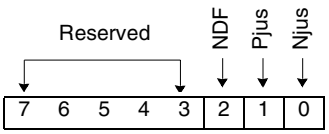
Command register for the chiplet. Single-cycle active if b'1' is written into bit position.

Length 8 bits

Type Read/Write

Address 431

Power On Value X'00'



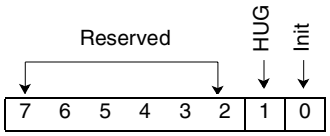
Bit(s)	Name	Description
7-3	Reserved	Reserved
2	NDF	Force a start-of-new-VC-4 event
1	Pjus	Perform a positive frequency justification
0	Njus	Perform a negative frequency justification

Status Registers

24.62: STAT1

Status register #1 of the chiplet. This is an event latch register.

Length 8 bits
Type Read/Write
Address 433
Power On Value X'00'

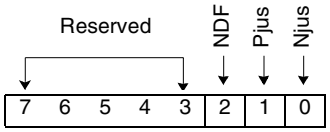


Bit(s)	Name	Description
7-2	Reserved	Reserved
1	HUG	0 Higher order unequipped generator inactive 1 Higher order unequipped generator active
0	Init	0 Default GRA initialization not completed 1 Default GRA initialization completed'

24.63: STAT2

Status register #2 of the chiplet. This is an event latch register.

Length 8 bits
Type Read/Write
Address 434
Power On Value -



Bit(s)	Name	Description
7-3	Reserved	Reserved
2	NDF	0 No NDF transmitted 1 NDF transmitted
1	Pjus	0 No positive frequency justification transmitted 1 Positive frequency justification transmitted
0	Njus	0 No negative frequency justification transmitted 1 Negative frequency justification transmitted

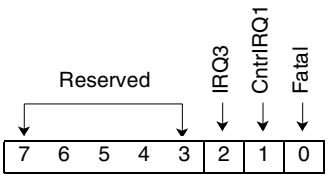
24.64: MainIRQ

Register to indicate fatal interrupt events and to point to user IRQ registers with active requests.

For each bit position:

0 = No interrupt request pending.
1 = Interrupt request pending.

Length 8 bits
Type Read/Write
Address 438
Power On Value -



Bit(s)	Name	Description
7-3	Reserved	Reserved
2	IRQ3	Active request in IRQ3 register
1	CntrIRQ1	Active request in CntrIRQ1 register
0	Fatal	Fatal event occured

24.65: M_MainIRQ

Register to mask pending interrupt requests. A masked request will not generate an outgoing IRQ to the GPPINT.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

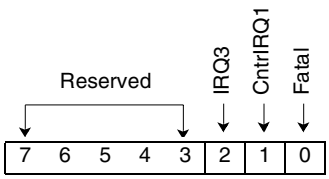
1 = The corresponding pending request bit activates signal IRQOT to GPPINT.

Length 8 bits

Type Read/Write

Address 439

Power On Value X'00'



Bit(s)	Name	Description
7-3	Reserved	Reserved
2	IRQ3	Active request in IRQ3 register
1	CntrIRQ1	Active request in CntrIRQ1 register
0	Fatal	Fatal event occured

24.66: CntrIRQ1

Register to indicate active counter interrupt requests of this chiplet.

For each bit position:

0 = No interrupt request pending.

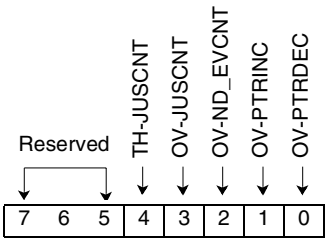
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 43A

Power On Value -



Bit(s)	Name	Description
7-5	Reserved	Reserved
4	TH-JUSCNT	Threshold overstep justification error counter
3	OV-JUSCNT	Overflow justification error counter
2	OV-ND_EVCNT	Overflow new data event counter
0	OV-PTRINC	Overflow pointer increment counter
1	OV-PTRDEC	Overflow pointer decrement counter

24.67: M_CntrIRQ1

Register to mask pending counter interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

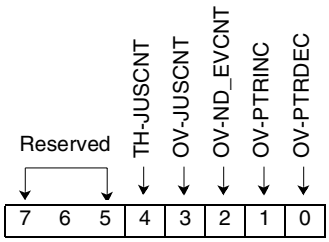
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 43B

Power On Value X'00'



Bit(s)	Name	Description
7-5	Reserved	Reserved
4	TH-JUSCNT	Threshold overstep justification error counter
3	OV-JUSCNT	Overflow justification error counter
2	OV-ND_EVCNT	Overflow new data event counter
0	OV-PTRINC	Overflow pointer increment counter
1	OV-PTRDEC	Overflow pointer decrement counter

24.68: IRQ3

Register to indicate active user interrupt requests of this chiplet.

For each bit position:

0 = No interrupt request pending.

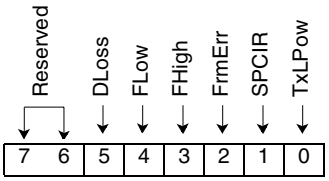
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 43C

Power On Value -



Bit(s)	Name	Description
7-6	Reserved	Reserved
5	DLoss	Data loss = Data FIFO empty
4	FLow	FIFO low threshold overflow
3	FHigh	FIFO high threshold overflow
2:	FrmErr	Framing error detected
1	SPCIR	SPC FSM interrupt request
0	TxLPow	Low Power indication from Optical/Electrical module

24.69: M_IRQ3

Register to mask pending user interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

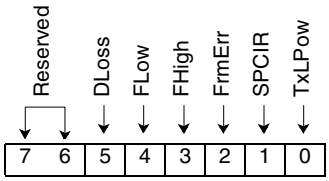
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 43D

Power On Value X'00'



Bit(s)	Name	Description
7-6	Reserved	Reserved
5	DLoss	Data loss = Data FIFO empty
4	FLow	FIFO low threshold overflow
3	FHigh	FIFO high threshold overflow
2	FrmErr	Framing error detected
1	SPCIR	SPC FSM interrupt request
0	TxLPow	Low Power indication from Optical/Electrical module

Configuration Registers

24.70: CONF1

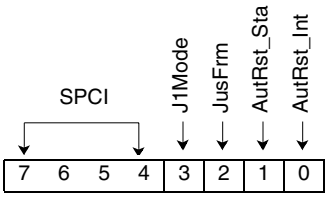
Configuration register #1. General OFP_Tx configuration signals A.

Length 8 bits

Type Read/Write

Address 448

Power On Value X'03'

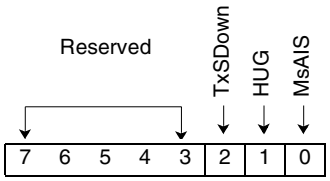


Bit(s)	Name	Description
7-4	SPCI(7-4)	Specifies STM-N row number in which an interrupt request will be issued
3	J1Mode	0 Transmit 16-byte J1 path trace 1 Transmit 64-byte J1 path trace
2	JusFrm	0 Allow pointer modification to be performed on frame-to-frame basis 1 Enforces three frames being interleaved between two pointer modification operations
1	AutRst_Sta	0 No action on read access 1 Auto-reset status register upon read access
0	AutRst_Int	0 No action on read access 1 Auto-reset interrupt request registers upon read access

24.71: CONF2

Configuration register #2. General OFP_Tx configuration signals B.

Length 8 bits
Type Read/Write
Address 449
Power On Value X'00'

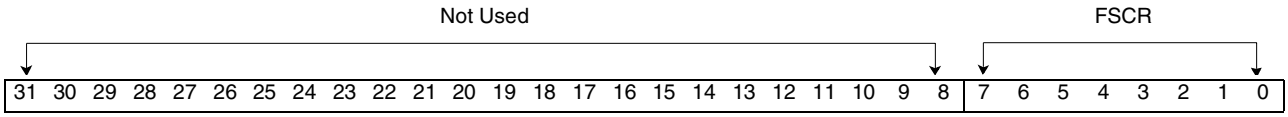


Bit(s)	Name	Description
7-3	Reserved	Reserved
2	TxSDown	Directly connected to output pin : 0 Optical/Electrical normal operation 1 Transmit shutdown for Optical/Electrical module
1	HUG	0 No unequipped STM-N signal 1 Enforce unequipped STM-N signal
0	MsAIS	0 No multiplex section AIS 1 Enforce multiplex section AIS

24.72: CONF3

Configuration register #3.

Length 8 bits
Type Read/Write
Address 44A
Power On Value X'FE'

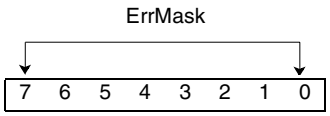


Bit(s)	Name	Description
31-0		Not used
7-0	FSCR(7-0)	Reload pattern for frame scrambler

24.73: CONF4

Configuration register #4.

Length 8 bits
Type Read/Write
Address 44B
Power On Value X'00'

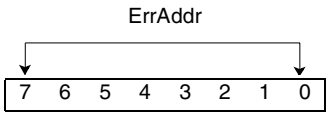


Bit(s)	Name	Description
7-0	ErrMask(7-0)	Mask register forcing bit error insertion. XORed with retrieved SOH/POH

24.74: CONF5

Configuration register #5.

Length 8 bits
Type Read/Write
Address 44C
Power On Value X'00'



Bit(s)	Name	Description
7-0	ErrAddr(7-0)	Error mask address register. Indicates address of SOH/POH byte to be corrupted

24.75: CONF6

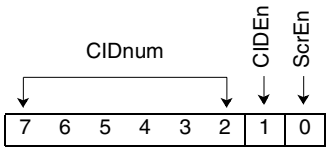
Configuration register #6. Frame scrambling control register.

Length 8 bits

Type Read/Write

Address 44D

Power On Value X'01'



Bit(s)	Name	Description
7-2	CIDnum(7-2)	Number of all - '1'/'0' bytes
1	CIDEn	CID Insertion Enable: 0 No CID insertion 1 Perform CID insertion
0	ScrEn	Scramble Enable: 0 No scrambling 1 Perform scrambling

24.76: CONF7

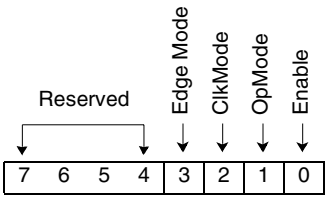
Configuration register #7. DCC control register.

Length 8 bits

Type Read/Write

Address 44E

Power On Value X'00'



Bit(s)	Name	Description
7-4	Reserved	Reserved
3	EdgeMode	0 Active falling edge 1 actIve rising edge
2	ClkMode	0 Continuous clock 1 Strobed clock
1	OpMode	0 DCC1 channel (D1 - D3) 1 CC2 channel (D4 - D12)
0	Enable	0 Disable DCC1 processing 1 Enable DCC1 processing

24.77: CONF8

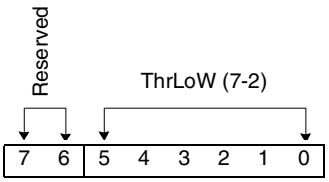
Configuration registers #8. Low water FIFO threshold register.

Length 8 bits

Type Read/Write

Address 44F

Power On Value X'03'



Bit(s)	Name	Description
7-6	Reserved	Reserved
5-0	ThrLoW(7-2)	Low Water FIFO threshold; default value is three

24.78: CONF9

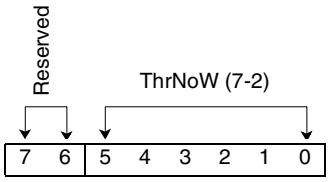
Configuration registers #9. Normal water FIFO threshold register.

Length 8 bits

Type Read/Write

Address 450

Power On Value X'11'



Bit(s)	Name	Description
7-6	Reserved	Reserved
5-0	ThrNoW(7-2)	Normal Water FIFO threshold; default value is 17

24.79: CONF10

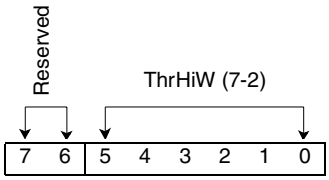
Configuration registers #10. High water FIFO threshold register.

Length 8 bits

Type Read/Write

Address 451

Power On Value X'20'



Bit(s)	Name	Description
7-6	Reserved	Reserved
5-0	ThrHiW(7-2)	High Water FIFO threshold; default value is 32

Overhead Frame Processor Architecture: Receive Direction

OFP_Rx GPP Handler Address Mapping Base Address = x'800' (Page 1 of 4)

Register Name	Description	Address Offset	Type Width	Initial Value
ROFmid	Read-on-the-fly register	X'0'	F 8	'00000000'
CntEn1	COUNT ENABLE register #1	X'2'	X 8	'00000000'
CntEn2	COUNT ENABLE register #2	X'3'	X 3	'000'
B1BITCNT	BIP-8 B1 bit error counter	X'4/5'[1]	N 16	'x'0000''
B1BITCNTTh12	Threshold register Byte2 (least significant byte) for B1BITCNT	X'6'	X 8	'00000000'
B1BITCNTTh11	Threshold register Byte1 for counter B1BITCNT	X'7'	X 8	'01111101'
B1BLKCNT	BIP-8 B1 block error counter[1])	X'8/9'[1]	N 16	'x'0000''
B1BLKCNTTh12	Threshold register Byte2 (least significant byte) for B1BLKCNT	X'A'	X 8	'00000000'
B1BLKCNTTh11	Threshold register Byte1 for counter B1BLKCNT	X'B'	X 8	'01111101'
B2BITCNT	BIP-24 B2 bit error counter, 2 thresholds[1]	X'C/D'[1]	N 16	'x'0000''
B2BITCNTTh12	Degradation threshold Byte2 (least significant byte) for B2BITCNT	X'E'	X 8	'00100000'
B2BITCNTTh11	Degradation threshold Byte1 for B2BITCNT	X'F'	X 8	'01001110'
B2BITCNTTh22	Failure threshold Byte2 (least significant byte) for B2BITCNT	X'10'	X 8	'00000000'
B2BITCNTTh21	Failure threshold Byte1 for B2BITCNT	X'11'	X 8	'01111101'
B2BLKCNT	BIP-24 B2 block error counter, 2 thresholds[1]	X'12/13'[1]	N 16	'x'0000''
B2BLKCNTTh12	Degration threshold Byte2 (least significant byte) for B2BLKCNT	X'14'	X 8	'00100000'
B2BLKCNTTh11	Degradation threshold Byte1 forB2BLKCNT	X'15'	X 8	'01001110'
B2BLKCNTTh22	Failure threshold Byte2 (least significant byte) for B2BLKCNT	X'16'	X 8	'00000000'
B2BLKCNTTh21	Failure threshold Byte1 for B2BLKCNT	X'17'	X 8	'01111101'
B3BITCNT	BIP-8 B3 bit error counter[1]	X'18/19'[1]	N 16	'x'0000''
B3BITCNTTh12	Threshold register Byte2 (least significant byte) for B3BITCNT	X'1A'	X 8	'00000000'
B3BITCNTTh11	Threshold register Byte1 for counter B3BITCNT	X'1B'	X 8	'01111101'
B3BLKCNT	BIP-8 B3 block error counter[1]	X'1C/1D'[1]	N 16	'x'0000''
B3BLKCNTTh12	Threshold register Byte2 (least significant byte) for B3BLKCNT	X'1E'	X 8	'00000000'
B3BLKCNTTh11	Threshold register Byte1 for counter B3BLKCNT	X'1F'	X 8	'01111101'
MSREICNT	Multiplex section remote error indication counter[1]	X'20/21'[1]	N 16	'x'0000''
MSREICNTTh12	Threshold register Byte2 (least significant byte) for MSREICNT	X'22'	X 8	'00000000'
MSREICNTTh11	Threshold register Byte1 for counter MSREICNT	X'23'	X 8	'01111101'
HPREICNT	Higher-order path remote error indication counter [1]	X'24/25'[1]	N 16	'x'0000''
HPREICNTTh12	Threshold register Byte2 (least significant byte) for HPREICNT	X'26'	X 8	'00000000'
HPREICNTTh11	Threshold register Byte1 for counter HPREICNT	X'27'	X 8	'01111101'
PJ_EVCNT	Positive justification counter, no threshold[1]	X'28/29'[1]	N 8	'00000000'
NJ_EVCNT	Negative justification counter, no threshold[1]	X'2A/2B'[1]	N 8	'00000000'

1. Independent of the counter width, given that a counter has chiplet address N as a base. Reading address N or address N-1 both yield the least significant byte of the counter. Reading address N has no affect on the counter, but reading address N-1 resets the counter after read operation.
2. Address range 100-17F located in 128x8. GRA Address range 180-1BF located in 64x8 GRA.
3. The 64-byte J1 path trace processing uses the 16 byte addresses of 16 byte J1 path trace to map a full 64 byte space.

OFP_Rx GPP Handler Address Mapping Base Address = x'800' (Page 2 of 4)

Register Name	Description	Address Offset	Type Width	Initial Value
ND_EVCNT	New data event counter, no threshold[1]	X'2C/2D'[1]	N 8	'00000000'
RESET	Default RESET register	X'30'	R 2	'01'
STAT1	Status register #1 (Mode)	X'33'	S 3	
STAT2	Status register #2 (AU pointer)	X'34'	S 6	
STAT3	Status register #3 (SOH)	X'35'	S 6	
STAT4	Status register #4 (POH)	X'36'	S 4	
MainIRQ	MAIN INTerrupt register	X'38'	I 7	
M_MainIRQ	INTerrupt MASK register for MainIRQ	X'39'	X 7	'0000000'
CntrIRQ1	COUNTER INTerrupt register	X'3A'	I 8	
M_CntrIRQ1	INTerrupt MASK register for CntrIRQ1	X'3B'	X 8	'00000000'
CntrIRQ2	COUNTER INTerrupt register	X'3C'	I 8	
M_CntrIRQ2	INTerrupt MASK register for CntrIRQ2	X'3D'	X 8	'00000000'
CntrIRQ3	COUNTER INTerrupt register	X'3E'	I 5	
M_CntrIRQ3	INTerrupt MASK register for CntrIRQ3	X'3F'	X 5	'00000'
IRQ6	USER INTerrupt register	X'40'	I 4	
M_IRQ6	INTerrupt MASK register for IRQ6	X'41'	X 4	'0000'
IRQ7	USER INTerrupt register	X'42'	I 8	
M_IRQ7	INTerrupt MASK register for IRQ7	X'43'	X 8	'00000000'
IRQ8	USER INTerrupt register	X'44'	I 8	
M_IRQ8	INTerrupt MASK register for IRQ8	X'45'	X 8	'00000000'
CONF1	Configuration register #1 (general)	X'48'	C 8	'00111111'
CONF2	Configuration register #2 (SOH processing)	X'49'	C 6	'0000'
CONF3	Configuration register #3 (POH processing)	X'4A'	C 4	'0000'
CONF4	Configuration register #4 (APS processing)	X'4B'	C 8	'00000000'
CONF7	Configuration register #7 (miscellaneous)	X'4E'	C 8	'00100000'
CONF8	Configuration register #8 (FSCR)	X'4F'	C 8	'11111110'
CONF9	Configuration register #9 (SL)	X'50'	C 8	'00010011'
SOH-A11	First A1	X'100'	8	
SOH-A12	Second1 A1	X'101'	8	
SOH-A13	Third A1	X'102'	8	
SOH-A21	First A2	X'103'	8	
SOH-A22	Second A2	X'104'	8	
SOH-A23	Third A2	X'105'	8	
SOH-J0	J0	X'106'	8	
	Reserved for national use and not included in frame scrambling (C1)	X'107-8'	8	

1. Independent of the counter width, given that a counter has chiplet address N as a base. Reading address N or address N-1 both yield the least significant byte of the counter. Reading address N has no affect on the counter, but reading address N-1 resets the counter after read operation.
2. Address range 100-17F located in 128x8. GRA Address range 180-1BF located in 64x8 GRA.
3. The 64-byte J1 path trace processing uses the 16 byte addresses of 16 byte J1 path trace to map a full 64 byte space.

OFP_Rx GPP Handler Address Mapping Base Address = x'800' (Page 3 of 4)

Register Name	Description	Address Offset	Type Width	Initial Value
SOH-B1	B1	X'109'	8	
	Media dependant bytes	X'10A-0B'	8	
SOH-E1	E1	X'10C'	8	
	Media dependant byte	X'10D'	8	
	Reserved for future standardization	X'10E'	8	
SOH-F1	F1	X'10F'	8	
	Reserved for national use	X'110-11'	8	
SOH-D1	D1	X'112'	8	
	Media dependant bytes	X'113-14'	8	
SOH-D2	D2	X'115'	8	
	Media dependant byte	X'116'	8	
	Reserved for future standardization	X'117'	8	
SOH-D3	D3	X'118'	8	
	Reserved for future standardization	X'119-1A'	8	
SOH-H1	H1	X'11B'	8	
SOH-J0	b'1001SS11' with S unspecified	X'11C-1D'	8	
SOH-H2	H2	X'11E'	8	
SOH-1s	x'FF'	X'11F-20'	8	
SOH-H31	First H3	X'121'	8	
SOH-H32	Second H3	X'122'	8	
SOH-H23	Third H3	X'123'	8	
SOH-B21	First B2	X'124'	8	
SOH-B22	Second B2	X'125'	8	
SOH-B23	Third B2	X'126'	8	
SOH-K1	K1	X'127'	8	
	Reserved for future standardization	X'128-29'	8	
SOH-K2	K2	X'12A'	8	
	Reserved for future standardization	X'12B-2C'	8	
SOH-D4	D4	X'12D'	8	
	Reserved for future standardization	X'12E-2F'	8	
SOH-D5	D5	X'130'	8	
	Reserved for future standardization	X'131-32'	8	
SOH-D6	D6	X'133'	8	
	Reserved for future standardization	X'134-35'	8	
SOH-D7	D7	X'136'	8	
	Reserved for future standardization	X'137-38'	8	
SOH-D8	D8	X'139'	8	
	Reserved for future standardization	X'13A-3B'	8	

1. Independent of the counter width, given that a counter has chiplet address N as a base. Reading address N or address N-1 both yield the least significant byte of the counter. Reading address N has no affect on the counter, but reading address N-1 resets the counter after read operation.
2. Address range 100-17F located in 128x8. GRA Address range 180-1BF located in 64x8 GRA.
3. The 64-byte J1 path trace processing uses the 16 byte addresses of 16 byte J1 path trace to map a full 64 byte space.

OFP_Rx GPP Handler Address Mapping Base Address = x'800' (Page 4 of 4)

Register Name	Description	Address Offset	Type Width	Initial Value
SOH-D9	D9	X'13C'	8	
	Reserved for future standardization	X'13D-3E'	8	
SOH-D10	D10	X'13F'	8	
	Reserved for future standardization	X'140-41'	8	
SOH-D11	D11	X'142'	8	
	Reserved for future standardization	X'143-44'	8	
SOH-D12	D12	X'145'	8	
	Reserved for future standardization	X'146-47'	8	
SOH-S1	S1	X'148'	8	
	Reserved for future standardization	X'149-9C'	8	
SOH-M1	M1	X'14D'	8	
SOH-M1	E2	X'14E'	8	
	Reserved for future standardization	X'14F-50'	8	
	Reserved	X'151-53'	8	
POH-J1	J1	X'154'	8	
POH-B3	B3	X'155'	8	
POH-C2	C2	X'156'	8	
POH-G1	G1	X'157'	8	
POH-F2	F2	X'158'	8	
POH-H4	H4	X'159'	8	
POH-F3	F3	X'15A'	8	
POH-K3	K3	X'15B'	8	
POH-N1	N1	X'15C'	8	
	Reserved	X'15D-5F'	8	
POH-J0-16-E	Expected 16-byte J0 section trace	X'160-6F'	8	
POH-J0-16-R	Received 16-byte J0 section trace	X'170-7F'	8	
POH-J1-16	Expected 16-byte J1 path trace[3]	X'180-8F'	8	
POH-J1-64	64-byte J1 path trace[3]	X'190-BF'	8	

1. Independent of the counter width, given that a counter has chiplet address N as a base. Reading address N or address N-1 both yield the least significant byte of the counter. Reading address N has no affect on the counter, but reading address N-1 resets the counter after read operation.
2. Address range 100-17F located in 128x8. GRA Address range 180-1BF located in 64x8 GRA.
3. The 64-byte J1 path trace processing uses the 16 byte addresses of 16 byte J1 path trace to map a full 64 byte space.

Counter Registers

24.80: ROFmid

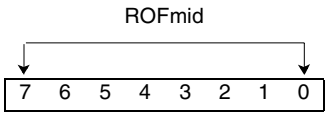
Read-on-the-fly registers.

Length 8 bits

Type Read

Address 800

Power On Value X'00'



Bit(s)	Name	Description
7-0	ROFmid(7-0)	Read-on-the-fly register, most significant byte

24.81: B1BITCNT

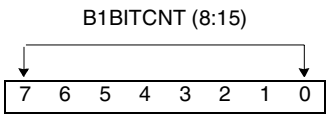
Number of BIP-8 B1 bit errors counted since last counter reset (16-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 804/805

Power On Value X'00'

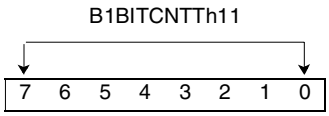


Bit(s)	Name	Description
7-0	B1BITCNT(8:15)	BIP-8 B1 bit error counter, least significant byte

24.82: B1BITCNTTh11

Threshold for number of BIP-8 B1 bit errors.

Length 8 bits
Type Read/Write
Address 807
Power On Value X'7D'

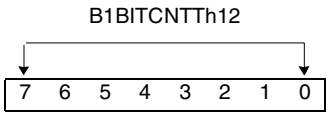


Bit(s)	Name	Description
7-0	B1BITCNTTh11(7-0)	Threshold for BIP-8 B1 bit error counter, most significant byte

24.83: B1BITCNTTh12

Threshold for number of BIP-8 B1 bit errors. Threshold overstep leads to an interrupt request.

Length 8 bits
Type Read/Write
Address 806
Power On Value X'00'



Bit(s)	Name	Description
7-0	B1BITCNTTh12(7-0)	Threshold for BIP-8 B1 bit error error counter, least significant byte

24.84: B1BLKCNT

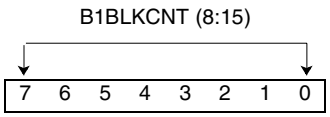
Number of BIP-8 B1 block errors counted since last counter reset (16-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 808/809

Power On Value X'00'



Bit(s)	Name	Description
7-0	B1BLKCNT(8:15)	BIP-8 B1 block error counter, least significant byte

24.85: B1BLKCNTTh11

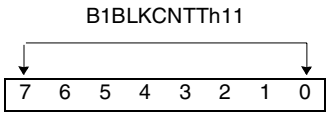
Threshold for number of BIP-8 B1 block errors.

Length 8 bits

Type Read/Write

Address 80B

Power On Value X'7D'



Bit(s)	Name	Description
7-0	B1BLKCNTTh11(7-0)	Threshold for BIP-8 B1 block error counter, most significant byte

24.86: B1BLKCNTTh12

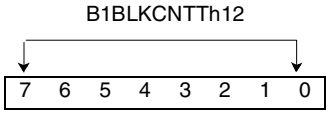
Threshold for number of BIP-8 B1 block errors. Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 80A

Power On Value X'00'



Bit(s)	Name	Description
7-0	B1BLKCNTTh12(7-0)	Threshold for BIP-8 B1 block error counter, least significant byte

24.87: B2BITCNT

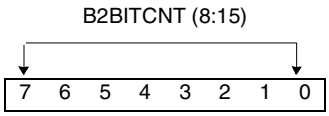
Number of BIP-24 B2 bit errors counted since last counter reset (16-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 80C/80D

Power On Value X'00'



Bit(s)	Name	Description
7-0	B2BITCNT(8:15)	BIP-24 B2 bit error counter, least significant byte

24.88: B2BITCNTTh11

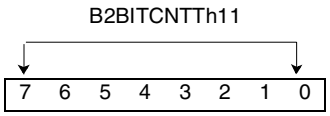
Degradation threshold for number of BIP-24 B2 bit errors.

Length 8 bits

Type Read/Write

Address 80F

Power On Value X'4E'



Bit(s)	Name	Description
7-0	B2BITCNTTh11(7-0)	Degradation threshold for BIP-24 B2 bit error counter, most significant byte

24.89: B2BITCNTTh12

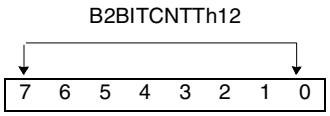
Degradation threshold for number of BIP-24 B2 bit errors. Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 80E

Power On Value X'20'

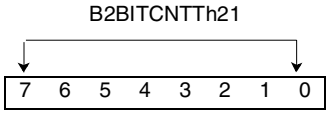


Bit(s)	Name	Description
7-0	B2BITCNTTh12(7-0)	Degradation threshold for BIP-24 B2 bit error counter, least significant byte

24.90: B2BITCNTTh21

Failure threshold for number of BIP-24 B2 bit errors.

Length 8 bits
Type Read/Write
Address 811
Power On Value X'7D'

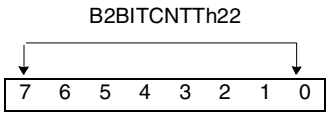


Bit(s)	Name	Description
7-0	B2BITCNTTh21(7-0)	Failure threshold for BIP-24 B2 bit error counter, most significant byte

24.91: B2BITCNTTh22

Failure threshold for number of BIP-24 B2 bit errors. Threshold overstep leads to an interrupt request.

Length 8 bits
Type Read/Write
Address 810
Power On Value X'00'



Bit(s)	Name	Description
7-0	B2BITCNTTh22(7-0)	Failure threshold for BIP-24 B2 bit error counter, least significant byte

24.92: B2BLKCNT

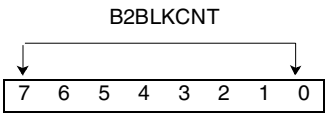
Number of BIP-24 B2 block errors counted since last counter reset (16-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 812/813

Power On Value X'00'



Bit(s)	Name	Description
7-0	B2BLKCNT(8:15)	BIP-24 B2 block error counter, least significant byte

24.93: B2BLKCNTTh11

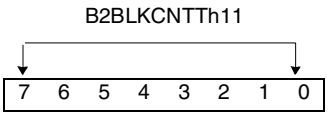
Degradation threshold for number of BIP-24 B2 block errors.

Length 8 bits

Type Read/Write

Address 815

Power On Value X'4E'



Bit(s)	Name	Description
7-0	B2BLKCNTTh11(7-0)	Degradation threshold for BIP-24 B2 block error counter, most significant byte

24.94: B2BLKCNTTh12

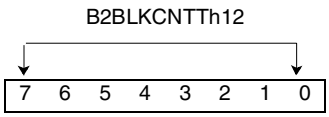
Degradation threshold for number of BIP-24 B2 block errors. Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 814

Power On Value X'20'



Bit(s)	Name	Description
7-0	B2BLKCNTTh12(7-0)	Degradation threshold for BIP-24 B2 block error counter, least significant byte

24.95: B2BLKCNTTh21

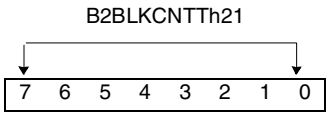
Failure threshold for number of BIP-24 B2 block errors.

Length 8 bits

Type Read/Write

Address 817

Power On Value X'7D'



Bit(s)	Name	Description
7-0	B2BLKCNTTh21(7-0)	Failure threshold for BIP-24 B2 block error counter, most significant byte

24.96: B2BLKCNTTh22

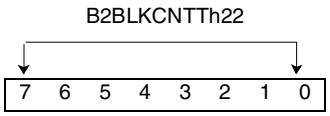
Failure threshold for number of BIP-24 B2 block errors. Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 816

Power On Value X'00'



Bit(s)	Name	Description
7-0	B2BLKCNTTh22(7-0)	Failure threshold for BIP-24 B2 block error counter, least significant byte

24.97: B3BITCNT

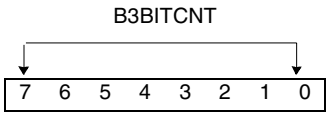
Number of BIP-8 B3 bit errors counted since last counter reset (16-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 818/819

Power On Value X'00'

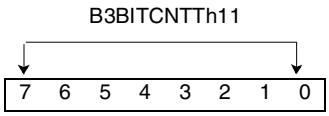


Bit(s)	Name	Description
7-0	B3BITCNT(8:15)	BIP-8 B3 bit error counter, least significant byte

24.98: B3BITCNTTh11

Threshold for number of BIP-8 B3 bit errors.

Length 8 bits
Type Read/Write
Address 81B
Power On Value X'7D'

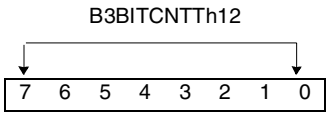


Bit(s)	Name	Description
7-0	B3BITCNTTh11(7-0)	Threshold for BIP-8 B3 bit error counter, most significant byte

24.99: B3BITCNTTh12

Threshold for number of BIP-8 B3 bit errors. Threshold overstep leads to an interrupt request.

Length 8 bits
Type Read/Write
Address 81A
Power On Value X'00'



Bit(s)	Name	Description
7-0	B3BITCNTTh12(7-0)	Threshold for BIP-8 B3 bit error counter, least significant byte

24.100: B3BLKCNT

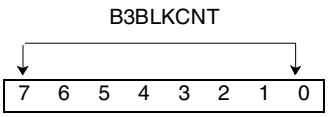
Number of BIP-8 B3 block errors counted since last counter reset (16-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 81C/81D

Power On Value X'00'



Bit(s)	Name	Description
7-0	B3BLKCNT(8:15)	BIP-8 B3 block error counter, least significant byte

24.101: B3BLKCNTTh11

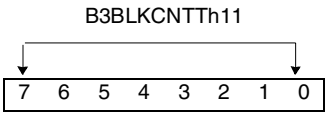
Threshold for number of BIP-8 B3 block errors.

Length 8 bits

Type Read/Write

Address 81F

Power On Value X'7D'



Bit(s)	Name	Description
7-0	B3BLKCNTTh11(7-0)	Threshold for BIP-8 B3 block error counter, most significant byte

24.102: B3BLKCNTTh12

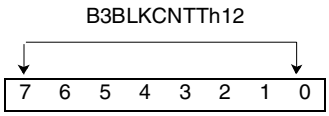
Threshold for number of BIP-8 B3 block errors. Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 81E

Power On Value X'00'



Bit(s)	Name	Description
7-0	B3BLKCNTTh12(7-0)	Threshold for BIP-8 B3 block error counter, least significant byte

24.103: MSREICNT

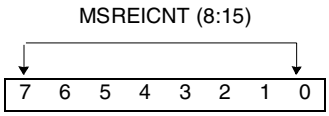
Multiplex Section Remote Error Indication counter (16-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 820/821

Power On Value X'00'



Bit(s)	Name	Description
7-0	MSREICNT(8:15)	Multiplex Section Remote Error Indication counter, least significant byte

24.104: MSREICNTTh11

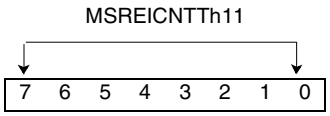
Threshold for number of Multiplex Section Remote Errors.

Length 8 bits

Type Read/Write

Address 823

Power On Value X'7D'



Bit(s)	Name	Description
7-0	MSREICNTTh11(7-0)	Threshold for Multiplex Indication counter Section Remote Error, most significant byte

24.105: MSREICNTTh12

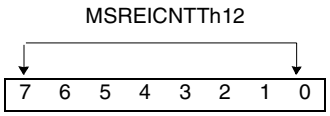
Threshold for number of Multiplex Section Remote Errors. Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 822

Power On Value X'00'

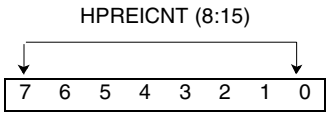


Bit(s)	Name	Description
7-0	MSREICNTTh12(7-0)	Threshold for Multiplex Indication counter Section Remote Error, least significant byte

24.106: HPREICNT

Higher-order Path Remote Error Indication counter (16-bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 824/825
Power On Value X'00'

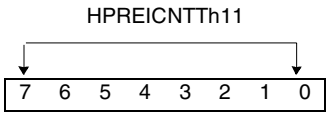


Bit(s)	Name	Description
7-0	HPREICNT(8:15)	Higher-order Path Remote Error Indication counter, least significant byte

24.107: HPREICNTTh11

Threshold for number of Higher-order Path Remote Errors.

Length 8 bits
Type Read/Write
Address 827
Power On Value X'7D'



Bit(s)	Name	Description
7-0	HPREICNTTh11(7-0)	Threshold for Higher-order Path Remote Error Indication counter, most significant byte

24.108: HPREICNTTh12

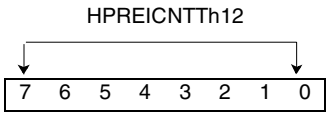
Threshold for number of Higher-order Path Remote Errors. Threshold overstep leads to an interrupt request.

Length 8 bits

Type Read/Write

Address 826

Power On Value X'00'



Bit(s)	Name	Description
7-0	HPREICNTTh12(7-0)	Threshold for Higher-order Path Remote Error Indication counter, least significant byte

24.109: PJ_EVCNT

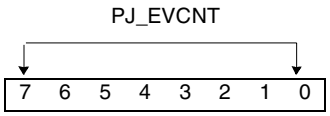
Positive Justification Event counter (eight-bit counter). Overflow leads to an interrupt request.

Length 8 bits

Type Read

Address 828/829

Power On Value X'00'

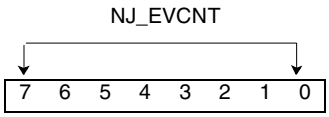


Bit(s)	Name	Description
7-0	PJ_EVCNT(7-0)	Positive Justification Event counter

24.110: NJ_EVCNT

Negative Justification Event counter (eight-bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 82A/82B
Power On Value X'00'

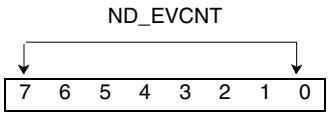


Bit(s)	Name	Description
7-0	NJ_EVCNT(7-0)	Negative Justification Event counter

24.111: ND_EVCNT

New Data Event counter (eight-bit counter). Overflow leads to an interrupt request.

Length 8 bits
Type Read
Address 82C/82D
Power On Value X'00'



Bit(s)	Name	Description
7-0	ND_EVCNT(7-0)	New Data Event counter

24.112: CntEn1

Counter On/Off control register #1 for OFP_Rx.

For each bit position:

0 = Counter is disabled.

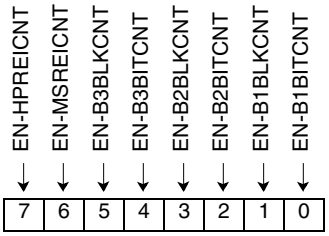
1 = Counter is enabled.

Length 8 bits

Type Read/Write

Address 802

Power On Value X'00'



Bit(s)	Name	Description
7	EN-HPREICNT	Higher-order Path Remote Error Indication counter enable
6	EN-MSREICNT	Multiplex Section Remote Error Indication counter enable
5	EN-B3BLKCNT	BIP-8 B3 block error counter enable
4	EN-B3BITCNT	BIP-8 B3 bit error counter enable
3	EN-B2BLKCNT	BIP-24 B2 block error counter enable
2	EN-B2BITCNT	BIP-24 B2 bit error counter enable
1	EN-B1BLKCNT	BIP-8 B1 block error counter enable
0	EN-B1BITCNT	BIP-8 B1 bit error counter enable

24.113: CntEn2

Counter On/Off control register #2 for OFP_Rx.

For each bit position:

0 = Counter is disabled.

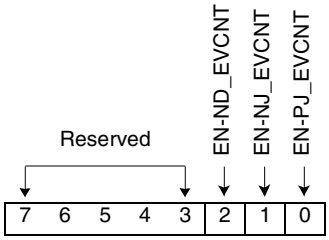
1 = Counter is enabled.

Length 8 bits

Type Read/Write

Address 803

Power On Value X'00'



Bit(s)	Name	Description
7-3	Reserved	Reserved
2	EN-ND_EVCNT	New Data Event counter enable
1	EN-NJ_EVCNT	Negative Justification Event counter enable
0	EN-PJ_EVCNT	Positive Justification Event counter enable

24.114: Reset Register (RESET)

Reset/Halt chiplet control register. This register is automatically preset to the default value by the reset signal ResOT coming from GPPINT chiplet.

For each bit position:

0 = Reset/Halt not active.

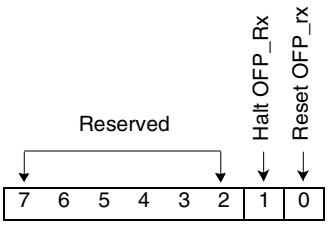
1 = Reset/Halt active.

Length 8 bits

Type Read/Write

Address 830

Power On Value X'01'



Bit(s)	Description
7-2	Reserved
1	Halt (freeze) OFP_Rx chiplet
0	Reset (disable) OFP_Rx chiplet

Status Registers

24.115: STAT1

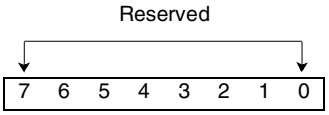
Status register #1 of the chiplet. OFP_Rx Mode status information. This is an event latch register.

Length 8 bits

Type -

Address 833

Power On Value -



Bit(s)	Description
7-0	Reserved

24.116: STAT2

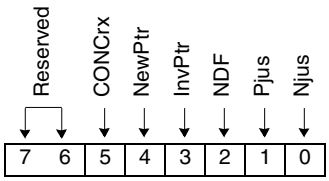
Status register #2 of the chiplet. AU pointer status information of OFP_Rx. This is an event latch register.

Length 8 bits

Type Read/Write

Address 834

Power On Value -



Bit(s)	Name	Description
7-6	Reserved	Reserved
5	CONCrx	Concatenation indication received
4	NewPtr	Valid New Pointer received
3	InvPtr	Invalid Pointer received
2	NDF	NDF received
1	Pjus	Positive frequency justification received
0	Njus	Negative frequency justification received

24.117: STAT3

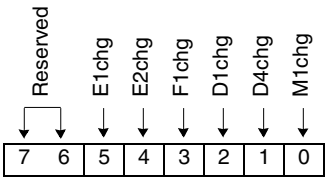
Status register #3 of the chiplet. Section Overhead (SOH) status of OFP_Rx. This is an event latch register.

Length 8 bits

Type Read/Write

Address 835

Power On Value -



Bit(s)	Name	Description
7-6	Reserved	Reserved
5	E1chg	Orderwire channel E1 content changed
4	E2chg	Orderwire channel E2 content changed
3	F1chg	User communication channel F1 content changed
2	D1chg	D1-D3 communication channel content changed
1	D4chg	D4-D12 communication channel content changed
0	M1chg	Number of bit blocks in error changed

24.118: STAT4

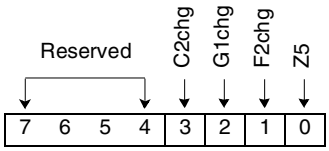
Status register #4 of the chiplet. Path Overhead (POH) status of OFP_Rx. This is an event latch register.

Length 8 bits

Type Read/Write

Address 836

Power On Value -



Bit(s)	Name	Description
7-4	Reserved	Reserved
3	C2chg	Payload composition indication changed
2	G1chg	Path status indication changed
1	F2chg	User communication channel F2 content changed
0	Z5	Z5 content changed

Interrupt and Mask Registers

24.119: MainIRQ

Register to indicate fatal interrupt events and to point to user IRQ registers with active requests.

For each bit position:

0 = No interrupt request pending.

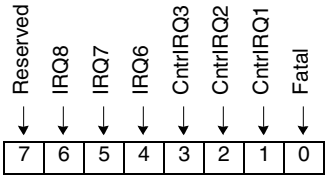
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 838

Power On Value -



Bit(s)	Name	Description
7	Reserved	Reserved
6	IRQ8	Active request in IRQ8 register
5	IRQ7	Active request in IRQ7 register
4	IRQ6	Active request in IRQ6 register
3	CntrIRQ3	Active request in CntrIRQ3 register
2	CntrIRQ2	Active request in CntrIRQ2 register
1	CntrIRQ1	Active request in CntrIRQ1 register
0	Fatal	Fatal event occured

24.120: M_MainIRQ

Register to mask pending interrupt requests. A masked request will not generate an outgoing IRQ to the GPPINT.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

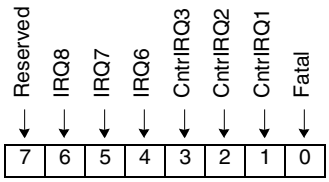
1 = The corresponding pending request bit activates signal IRQOR to GPPINT.

Length 8 bits

Type Read/Write

Address 839

Power On Value X'00'



Bit(s)	Name	Description
7	Reserved	Reserved
6	IRQ8	Active request in IRQ8 register
5	IRQ7	Active request in IRQ7 register
4	IRQ6	Active request in IRQ6 register
3	CntrIRQ3	Active request in CntrIRQ3 register
2	CntrIRQ2	Active request in CntrIRQ2 register
1	CntrIRQ1	Active request in CntrIRQ1 register
0	Fatal	Fatal event occured

24.121: CntrIRQ1

Register #1 to indicate active counter interrupt requests of this chiplet.

For each bit position:

0 = No interrupt request pending.

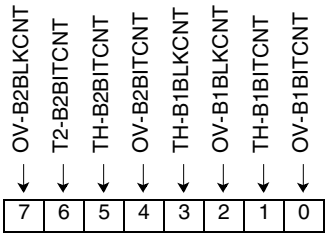
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 83A

Power On Value -



Bit(s)	Name	Description
7	OV-B2BLKCNT	Overflow BIP-24 B2 block error counter
6	T2-B2BITCNT	Failure threshold overstep BIP-24 B2 bit error counter
5	TH-B2BITCNT	Degradation threshold overstep BIP-24 B2 bit error counter
4	OV-B2BITCNT	Overflow BIP-24 B2 bit error counter
3	TH-B1BLKCNT	Threshold overstep BIP-8 B1 block error counter
2	OV-B1BLKCNT	Overflow BIP-8 B1 block error counter
1	TH-B1BITCNT	Threshold overstep BIP-8 B1 bit error counter
0	OV-B1BITCNT	Overflow BIP-8 B1 bit error counter

24.122: M_CntrIRQ1

Register to mask pending counter interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

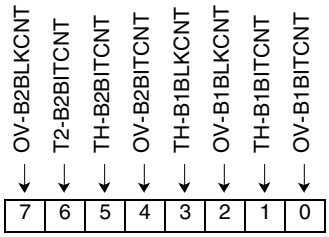
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 83B

Power On Value X'00'



Bit(s)	Name	Description
7	OV-B2BLKCNT	Overflow BIP-24 B2 block error counter
6	T2-B2BITCNT	Failure threshold overstep BIP-24 B2 bit error counter
5	TH-B2BITCNT	Degradation threshold overstep BIP-24 B2 bit error counter
4	OV-B2BITCNT	Overflow BIP-24 B2 bit error counter
3	TH-B1BLKCNT	Threshold overstep BIP-8 B1 block error counter
	2OV-B1BLKCNT	Overflow BIP-8 B1 block error counter
1	TH-B1BITCNT	Threshold overstep BIP-8 B1 bit error counter
0	OV-B1BITCNT	Overflow BIP-8 B1 bit error counter

24.123: CntrIRQ2

Register #2 to indicate active counter interrupt requests of this chiplet.

For each bit position:

0 = No interrupt request pending.

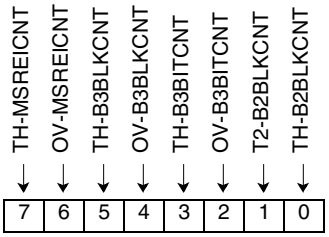
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 83C

Power On Value -



Bit(s)	Name	Description
7	TH-MSREICNT	Threshold overstep Multiplex Section Remote Error Indication counter
6	OV-MSREICNT	Overflow Multiplex Section Remote Error indication counter
5	TH-B3BLKCNT	Threshold overstep BIP-8 B3 block error counter
4	OV-B3BLKCNT	Overflow BIP-8 B3 block error counter
3	TH-B3BITCNT	Threshold overstep BIP-8 B3 bit error counter
2	OV-B3BITCNT	Overflow BIP-8 B3 bit error counter
1	T2-B2BLKCNT	Failure threshold overstep BIP-24 B2 block error counter
0	TH-B2BLKCNT	Degradation threshold overstep BIP-24 B2 block error counter

24.124: M_CntrIRQ2

Register to mask pending counter interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

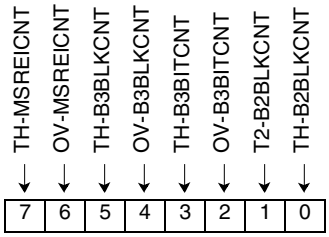
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 83D

Power On Value X'00'



Bit(s)	Name	Description
7	TH-MSREICNT	Threshold overstep Multiplex Section Remote Error Indication counter
6	OV-MSREICNT	Overflow Multiplex Section Remote Error indication counter
5	TH-B3BLKCNT	Threshold overstep BIP-8 B3 block error counter
4	OV-B3BLKCNT	Overflow BIP-8 B3 block error counter
3	TH-B3BITCNT	Threshold overstep BIP-8 B3 bit error counter
2	OV-B3BITCNT	Overflow BIP-8 B3 bit error counter
1	T2-B2BLKCNT	Failure threshold overstep BIP-24 B2 block error counter
0	TH-B2BLKCNT	Degradation threshold overstep BIP-24 B2 block error counter

24.125: CntrIRQ3

Register #3 to indicate active counter interrupt requests of this chiplet.

For each bit position:

0 = No interrupt request pending.

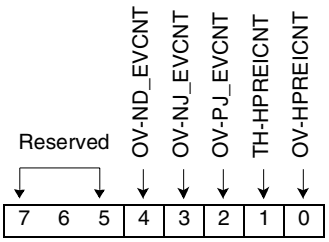
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 83E

Power On Value -



Bit(s)	Name	Description
7-5	Reserved	Reserved
4	OV-ND_EVCNT	Overflow New Data event counter
3	OV-NJ_EVCNT	Overflow Negative Justification event counter
2	OV-PJ_EVCNT	Overflow Positive Justification event counter
1	TH-HPREICNT	Threshold overstep Higher-order Path Remote Error indication counter
0	OV-HPREICNT	Overflow HPR error indication counter

24.126: M_CntrIRQ3

Register to mask pending counter interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

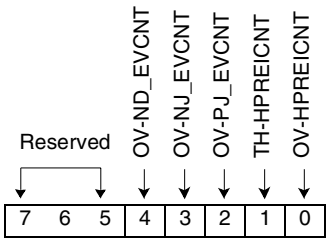
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 83F

Power On Value X'00'



Bit(s)	Name	Description
7-5	Reserved	Reserved
4	OV-ND_EVCNT	Overflow New Data event counter
3	OV-NJ_EVCNT	Overflow Negative Justification event counter
2	OV-PJ_EVCNT	Overflow Positive Justification event counter
1	TH-HPREICNT	Threshold overstep Higher-order Path Remote Error indication counter
0	OV-HPREICNT	Overflow HPR error indication counter

24.127: IRQ6

Register to indicate active user interrupt requests of this chiplet.

For each bit position:

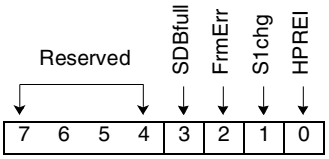
0 = No interrupt request pending.
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 840

Power On Value -



Bit(s)	Name	Description
7-4	Reserved	Reserved
3	SDBfull	SDB_Rx FIFO full
2	FrmErr	Interrupt from ORxAUG FSM
1	S1chg	Synchronization status changed
0	HPREI	Higher-order Path Remote Error Indication

24.128: M_IRQ6

Register to mask pending user interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

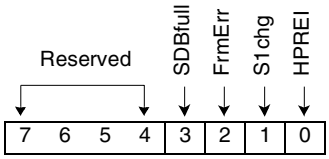
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 841

Power On Value X'00'



Bit(s)	Name	Description
7-4	Reserved	Reserved
3	SDBfull	SDB_Rx FIFO full
2	FrmErr	Interrupt from ORxAUG FSM
1	S1chg	Synchronization status changed
0	HPREI	Higher-order Path Remote Error Indication

24.129: IRQ7

Register to indicate active user interrupt requests of this chiplet.

For each bit position:

0 = No interrupt request pending.

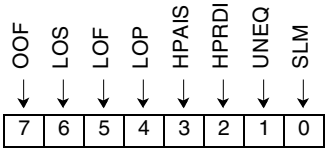
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 842

Power On Value -



Bit(s)	Name	Description
7	OOF	Out of frame alarm
6	LOS	Loss of signal alarm
5	LOF	Loss of frame alarm
4	LOP	Loss of pointer alarm
3	HPAIS	Higher-order path AIS
2	HPRDI	Higher-order path RDI
1	UNEQ	Unequipped signal
0	SLM	Signal label mismatch alarm

24.130: M_IRQ7

Register to mask pending user interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

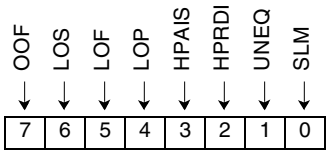
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 843

Power On Value X'00'



Bit(s)	Name	Description
7	OOF	Out of frame alarm
6	LOS	Loss of signal alarm
5	LOF	Loss of frame alarm
4	LOP	Loss of pointer alarm
3	HPAIS	Higher-order path AIS
2	HPRDI	Higher-order path RDI
1	UNEQ	Unequipped signal
0	SLM	Signal label mismatch alarm

24.131: IRQ8

Register to indicate active user interrupt requests of this chiplet.

For each bit position:

0 = No interrupt request pending.

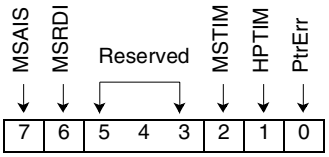
1 = Interrupt request pending.

Length 8 bits

Type Read/Write

Address 844

Power On Value -



Bit(s)	Name	Description
7	MSAIS	Multiplex Section AIS
6	MSRDI	Multiplex Section RDI
5-3	Reserved	Reserved
2	MSTIM	Multiplex Section trace identifier mismatch
1	HPTIM	Higher-order path trace identifier mismatch
0	PtrErr	Pointer processing error

24.132: M_IRQ8

Register to mask pending user interrupt requests.

For each bit position:

0 = The corresponding pending request bit is masked (DEFAULT).

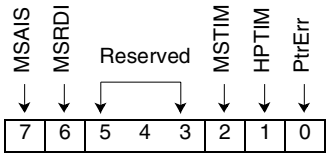
1 = The corresponding pending request bit activates the pointer bit in MainIRQ register.

Length 8 bits

Type Read/Write

Address 845

Power On Value X'00'



Bit(s)	Name	Description
7	MSAIS	Multiplex Section AIS
6	MSRDI	Multiplex Section RDI
5-3	Reserved	Reserved
2	MSTIM	Multiplex Section trace identifier mismatch
1	HPTIM	Higher-order path trace identifier mismatch
0	PtrErr	Pointer processing error

Configuration Registers

24.133: CONF1

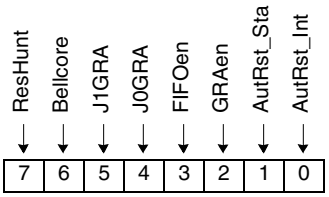
Configuration register #1. General OFP_Rx configuration signals.

Length 8 bits

Type Read/Write

Address 848

Power On Value X'3F'



Bit(s)	Name	Description
7	ResHunt	0 Hunt free running 1 Reset Hunt to PIM
6	Bellcore	0 Operate according to ITU standard 1 Operate according to Bellcore specification
5	J1GRA	0 Do not write J1 section trace to GRA 1 Write J1 section trace to GRA
4	J0GRA	0 Do not write J0 section trace to GRA 1 Write J0 section trace to GRA
3	FIFOen	0 Do not write C4 payload to FIFO 1 Write C4 payload to FIFO
2	GRAen	0 Do not write SOH/POH info to GRA 1 Write received SOH/POH info to GRA
1	AutRst_Sta	0 No action on read access 1 Auto-reset status register upon read access
0	AutRst_Int	0 No action on read access 1 Auto-reset interrupt request registers upon read access

24.134: CONF2

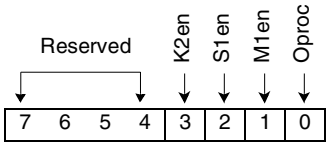
Configuration register #2. SOH processing configuration signals.

Length 8 bits

Type Read/Write

Address 849

Power On Value X'00'



Bit(s)	Name	Description
7-4	Reserved	Reserved
3	K2en	0 Disable K2 AIS processing 1 Enable K2 AIS processing
2	S1en	0 Disable S1 synchronization status processing 1 Enable S1 synchronization status processing
1	M1en	0 Disable M1 REI processing 1 Enable M1 REI processing
0	J0proc	0 Disable J0 section trace processing 1 Enable J0 section trace processing

24.135: CONF3

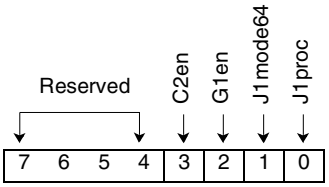
Configuration register #3. POH byte processing configuration signals.

Length 8 bits

Type Read/Write

Address 84A

Power On Value X'00'



Bit(s)	Name	Description
7-4		Reserved
3	C2en	0 Disable C2 signal label processing 1 Enable C2 signal label processing
2	G1en	0 Disable G1 path status processing 1 Enable G1 path status processing
1	J1mode64	0 16-byte J1 trace 1 64-byte J1 trace
0	J1proc	0 Disable J1 path trace processing 1 Enable J1 path trace processing

24.136: CONF4

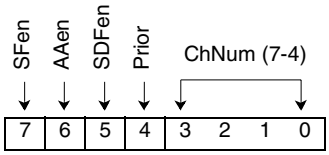
Configuration register #4. APS processing configuration signals.

Length 8 bits

Type Read/Write

Address 84B

Power On Value X'00'



Bit(s)	Name	Description
7	SFen	0 Disable SF K2 MS_RDI processing 1 Enable SF K2 MS_RDI processing
6	AAen	0 Disable automatic Alarm processing for K2 1 Enable automatic Alarm processing for K2
5	SDFen	0 Disable automatic SDF K1 processing 1 Enable automatic SDF K1 processing
4	Prior	Priority level
3-0	ChNum(7-4)	Channel Number

24.137: CONF7

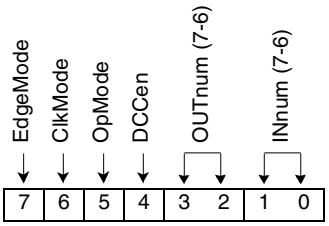
Configuration register #7. Miscellaneous OFP_Rx configuration signals.

Length 8 bits

Type Read/Write

Address 84E

Power On Value X'20'

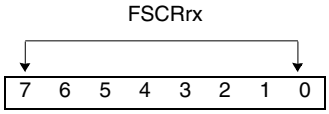


Bit(s)	Name	Description
7	EdgeMode	0 Active falling edge 1 Active rising edge
6	ClkMode	0 Continuous clock mode 1 Strobed clock mode
5	OpMode	0 DCC 1 channel selected 1 DCC 2 channel selected
4	DCCen	0 Disable DCC processing 1 Enable DCC processing
3-2	OUTnum(7-6)	Number of µs for in-frame to out-of-frame transition
1-0	INnum(7-6)	Number of µs for out-of-frame to in-frame transition

24.138: CONF8

Configuration register #8. Pattern register signals.

Length 8 bits
Type Read/Write
Address 84F
Power On Value X'FE'

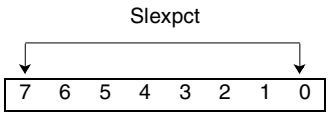


Bit(s)	Name	Description
7-0	FSCRrx(7-0)	Frame descrambling reload pattern

24.139: CONF9

Configuration register #9. Pattern register signals.

Length 8 bits
Type Read/Write
Address 850
Power On Value X'13'



Bit(s)	Name	Description
7-0	Slexpct(7-0)	Expected signal label

Memory Map for Registers and Arrays

Address	Entity	Elements Accessed
XXXX 0000 - FF	PCINT	Registers
XXXX 0100 - FF	GPDMA	Registers & Array
XXXX 0400 - FF	INTST	Registers
XXXX 0500 - FF	CRSET	Registers
XXXX 0600 - 8FF	DMAQS	Registers & Array
XXXX 0900 - FF	COMET/PAKIT	Registers
XXXX 0A00 - FF	CHKSM	Registers
XXXX 0B00 - FF	LINKC	Registers
XXXX 0D00 - FF	VIMEM	Registers
XXXX 0E00 - FF	ARBIT	Registers
XXXX 1000 - 1FF	BCACH	Registers & Array
XXXX 1200 - 3FF	CSKED	Registers & Array
XXXX 1400 - 5FF	SEGBF	Registers & Array
XXXX 1600 - 7FF	REASM	Registers & Array
XXXX 1800 - FFF	RXQUE	Registers & Arrays
XXXX 2000 - FFF	NPBUS/FRAMR	Registers & External EEPROM
XXXX 3000 - FFF	POOLS	Registers & Arrays
XXXX 4000 - FFF	PCORE	Registers & Arrays
XXXX 5000 - FFF	PPOCM	Arrays

Signal Pin Listing By Signal Name (Page 1 of 5)

Signal Name	Grid Position	Book	Signal Name	Grid Position	Book	Signal Name	Grid Position	Book
BIST0DI1	0W01	K	CMCLK(4)	0N01	B	CMDATA(38)	0E04	C
CM0CS(0)	AD17	C	CMCLKE	AD13	C	CMDATA(4)	0L03	C
CM0CS(1)	0Y15	C	CMDATA(0)	0M09	C	CMDATA(5)	0L09	C
CM0CS(2)	AE16	C	CMDATA(1)	0N03	C	CMDATA(6)	0M03	C
CM0CS(3)	AE15	C	CMDATA(10)	0L05	C	CMDATA(7)	0L04	C
CM0DQM(0)	0U14	C	CMDATA(11)	0L06	C	CMDATA(8)	0L08	C
CM0DQM(1)	AB15	C	CMDATA(12)	0J02	C	CMDATA(9)	0K03	C
CM0DQM(2)	AE14	C	CMDATA(13)	0L07	C	CMSYNCAS(0)	0U15	C
CM0DQM(3)	AC14	C	CMDATA(14)	0L10	C	CMSYNCAS(1)	AC17	C
CMADDR(0)	0W02	C	CMDATA(15)	0K05	C	CMSYNRAS(0)	AE18	C
CMADDR(1)	0U03	C	CMDATA(16)	0J05	C	CMSYNRAS(1)	AA17	C
CMADDR(10)	0R05	C	CMDATA(17)	0H01	C	CMWE(0)	0W13	C
CMADDR(11)	0P03	C	CMDATA(18)	0J03	C	CMWE(1)	AA13	C
CMADDR(12)	0R03	C	CMDATA(19)	0K09	C	CTS	0P07	T
CMADDR(13)	0R04	C	CMDATA(2)	0N02	C	DSR	0W03	O
CMADDR(14)	0R02	C	CMDATA(20)	0J04	C	DTR	AA05	O
CMADDR(15)	0N04	C	CMDATA(21)	0J07	C	ENSTATE(0)	0C24	A
CMADDR(16)	0N07	C	CMDATA(22)	0H03	C	ENSTATE(1)	0B25	A
CMADDR(17)	0N08	C	CMDATA(23)	0K07	C	ENSTATE(10)	AE24	A
CMADDR(18)	0M11	C	CMDATA(24)	0G02	C	ENSTATE(11)	AD21	A
CMADDR(19)	0N10	C	CMDATA(25)	0F01	C	ENSTATE(12)	AE04	A
CMADDR(2)	0R08	C	CMDATA(26)	0J09	C	ENSTATE(13)	AD05	A
CMADDR(20)	0N06	C	CMDATA(27)	0H05	C	ENSTATE(14)	AE01	A
CMADDR(3)	0U02	C	CMDATA(28)	0J06	C	ENSTATE(15)	AC02	A
CMADDR(4)	0T05	C	CMDATA(29)	0G04	C	ENSTATE(16)	AD01	A
CMADDR(5)	0U04	C	CMDATA(3)	0L02	C	ENSTATE(17)	AB01	A
CMADDR(6)	0R07	C	CMDATA(30)	0F03	C	ENSTATE(18)	AA02	A
CMADDR(7)	0T03	C	CMDATA(31)	0J08	C	ENSTATE(19)	0B01	A
CMADDR(8)	0R06	C	CMDATA(32)	0G05	C	ENSTATE(2)	0E24	A
CMADDR(9)	0N09	C	CMDATA(33)	0H07	C	ENSTATE(20)	0E02	A
CMCLK(0)	0J01	B	CMDATA(34)	0D03	C	ENSTATE(21)	0D01	A
CMCLK(1)	0K01	B	CMDATA(35)	0F05	C	ENSTATE(22)	0C02	A
CMCLK(2)	0L01	B	CMDATA(36)	0G06	C	ENSTATE(23)	0B03	A
CMCLK(3)	0M01	B	CMDATA(37)	0G07	C	ENSTATE(24)	0A02	A

Signal Pin Listing By Signal Name (Page 2 of 5)

Signal Name	Grid Position	Book	Signal Name	Grid Position	Book	Signal Name	Grid Position	Book
ENSTATE(25)	0A04	A	ENSTATE(56)	AE23	K	FYRDAT(12)	0M19	T
ENSTATE(26)	0B05	A	ENSTATE(57)	AC23	K	FYRDAT(13)	0C19	T
ENSTATE(27)	0A22	A	ENSTATE(58)	AC25	K	FYRDAT(14)	AA14	T
ENSTATE(28)	0B21	A	ENSTATE(59)	AA25	K	FYRDAT(15)	0W14	T
ENSTATE(29)	0A25	A	ENSTATE(6)	AC24	A	FYRDAT(2)	0W23	T
ENSTATE(3)	AA24	A	ENSTATE(60)	0C25	K	FYRDAT(3)	AC21	T
ENSTATE(30)	0B23	A	ENSTATE(61)	0C23	K	FYRDAT(4)	AA21	T
ENSTATE(31)	0A24	A	ENSTATE(62)	0A23	K	FYRDAT(5)	0P21	T
ENSTATE(32)	0N23	J	ENSTATE(63)	0A21	K	FYRDAT(6)	0M21	T
ENSTATE(33)	0R22	J	ENSTATE(7)	AD23	A	FYRDAT(7)	0E21	T
ENSTATE(34)	0R19	J	ENSTATE(8)	AE25	A	FYRDAT(8)	0C21	T
ENSTATE(35)	0U22	J	ENSTATE(9)	AE22	A	FYRDAT(9)	AE19	T
ENSTATE(36)	0T21	J	FCDP(0)	AA20	L	FYREOP	0L23	L
ENSTATE(37)	0U24	J	FCDP(1)	0W18	L	FYRMOD	0L24	L
ENSTATE(38)	0R18	J	FCGBUSY	AA19	L	FYRPAR(0)	0K23	N
ENSTATE(39)	0U23	J	FCPHASE(0)	AB21	L	FYRPAR(1)	0M17	N
ENSTATE(4)	AB25	A	FCPHASE(1)	0Y19	L	FYRRDB	0R23	N
ENSTATE(40)	0W24	J	FCSYNC	AC20	L	FYRSCLKN	0J25	DA
ENSTATE(41)	0V23	J	FY0EMP	0L17	N	FYRSCLKP	0J24	DA
ENSTATE(42)	0V25	J	FY0FUL	0J23	N	FYRSDATN	0M25	DA
ENSTATE(43)	0T19	J	FY0RENB	0H25	I	FYRSDATP	0M23	DA
ENSTATE(44)	0R17	J	FY0TENB	0J21	I	FYRSOC	0R24	L
ENSTATE(45)	0R16	J	FYDISCRD	0K21	L	FYTADR(0)	0N20	G
ENSTATE(46)	0U20	J	FYDTCT	0L18	N	FYTADR(1)	0N21	G
ENSTATE(47)	0U21	J	FYRADR(0)	0L19	G	FYTADR(2)	0T25	G
ENSTATE(48)	0V21	J	FYRADR(1)	0L20	G	FYTADR(3)	0N22	G
ENSTATE(49)	0W22	J	FYRADR(2)	0K25	G	FYTADR(4)	0N24	G
ENSTATE(5)	AD25	A	FYRADR(3)	0L25	G	FYTCA	0N25	L
ENSTATE(50)	AA01	K	FYRADR(4)	0L21	G	FYTDAT(0)	0F21	J
ENSTATE(51)	AC01	K	FYRCA	0L22	L	FYTDAT(1)	0K17	J
ENSTATE(52)	AC03	K	FYRDAT(0)	AA23	T	FYTDAT(10)	0H21	G
ENSTATE(53)	AE03	K	FYRDAT(1)	0W25	T	FYTDAT(11)	0M15	G
ENSTATE(54)	AE05	K	FYRDAT(10)	AC19	T	FYTDAT(12)	0K19	G
ENSTATE(55)	AE21	K	FYRDAT(11)	0P19	T	FYTDAT(13)	0H23	G

Signal Pin Listing By Signal Name (Page 3 of 5)

Signal Name	Grid Position	Book	Signal Name	Grid Position	Book	Signal Name	Grid Position	Book
FYTDAT(14)	0L16	G	MGNT	0C22	D	PAD(25)	0A18	D
FYTDAT(15)	0J22	G	MHALTPPC	0U19	N	PAD(26)	0G19	D
FYTDAT(2)	0G20	J	MINT2	0J16	D	PAD(27)	0E20	D
FYTDAT(3)	0G21	J	MINTA	0J17	D	PAD(28)	0F19	D
FYTDAT(4)	0E22	J	MIRDY	0H15	D	PAD(29)	0E19	D
FYTDAT(5)	0J18	J	MPCIRST	AA03	E	PAD(3)	0G13	D
FYTDAT(6)	0D23	J	MPEGCLK	0C07	T	PAD(30)	0C20	D
FYTDAT(7)	0G22	J	MPERR	0C16	D	PAD(31)	0A20	D
FYTDAT(8)	0J19	G	MPLLRESET	AA12	V	PAD(4)	0H13	D
FYTDAT(9)	0J20	G	MPMEVENT	0Y25	F	PAD(5)	0J13	D
FYTEOP	0R21	H	MREQ	0D21	D	PAD(6)	0K13	D
FYTMOD	0P17	H	MREQ64	0D11	D	PAD(7)	0D13	D
FYTPAR(0)	0P15	J	MSERR	0A15	D	PAD(8)	0B13	D
FYTPAR(1)	0N19	J	MSTOP	0D17	D	PAD(9)	0A13	D
FYTSCLKN	0R25	DA	MTRDY	0B17	D	PAD64(32)	0C04	D
FYTSCLKP	0T23	DA	NSELFT	AC05	T	PAD64(33)	0D05	D
FYTSDATN	0P25	DB	PAD(0)	0C13	D	PAD64(34)	0A06	D
FYTSDATP	0P23	DB	PAD(1)	0F13	D	PAD64(35)	0C06	D
FYTSOC	0U25	G	PAD(10)	0C15	D	PAD64(36)	0E06	D
FYTWRB	0R20	N	PAD(11)	0D15	D	PAD64(37)	0E07	D
IBDINH1	0E05	Q	PAD(12)	0A16	D	PAD64(38)	0B07	D
IBDINH2	0A03	R	PAD(13)	0C14	D	PAD64(39)	0F07	D
IBDRINH	0C01	U	PAD(14)	0A14	D	PAD64(40)	0D07	D
JTAG0RST	0G12	T	PAD(15)	0E15	D	PAD64(41)	0G09	D
JTAGTCK	0E12	T	PAD(16)	0J15	D	PAD64(42)	0F09	D
JTAGTDI	0E14	T	PAD(17)	0K15	D	PAD64(43)	0E08	D
JTAGTDO	0A07	K	PAD(18)	0G16	D	PAD64(44)	0J10	D
JTAGTMS	0G14	T	PAD(19)	0H17	D	PAD64(45)	0G08	D
JTCOMPLY	0C03	M	PAD(2)	0L14	D	PAD64(46)	0H09	D
LEAKTST	0A05	S	PAD(20)	0G18	D	PAD64(47)	0G10	D
MACK64	0C10	D	PAD(21)	0F17	D	PAD64(48)	0C08	D
MDEVSEL	0E16	D	PAD(22)	0E17	D	PAD64(49)	0K11	D
MEXTPMEVENT	0G15	N	PAD(23)	0E18	D	PAD64(50)	0D09	D
MFRAME	0C17	D	PAD(24)	0C18	D	PAD64(51)	0J11	D

Signal Pin Listing By Signal Name (Page 4 of 5)

Signal Name	Grid Position	Book	Signal Name	Grid Position	Book	Signal Name	Grid Position	Book
PAD64(52)	0C09	D	PCBE(1)	0A17	D	PMADDR(16)	0U06	C
PAD64(53)	0A08	D	PCBE(2)	0B19	D	PMADDR(17)	0P11	C
PAD64(54)	0E09	D	PCBE(3)	0D19	D	PMADDR(18)	0P09	C
PAD64(55)	0E10	D	PCBE64(4)	0A12	D	PMADDR(19)	0T07	C
PAD64(56)	0H11	D	PCBE64(5)	0C12	D	PMADDR(2)	0Y05	C
PAD64(57)	0G11	D	PCBE64(6)	0L12	D	PMADDR(20)	0V03	C
PAD64(58)	0A09	D	PCBE64(7)	0C11	D	PMADDR(3)	0T09	C
PAD64(59)	0B09	D	PCICLK	0G25	PLL	PMADDR(4)	0V07	C
PAD64(60)	0F11	D	PDBLCLK	0Y23	N	PMADDR(5)	0R10	C
PAD64(61)	0A10	D	PFFCFG(0)	0G01	T	PMADDR(6)	0W05	C
PAD64(62)	0A11	D	PFFCFG(1)	0G03	T	PMADDR(7)	0U08	C
PAD64(63)	0E11	D	PFFCFG(2)	0C05	T	PMADDR(8)	AB03	C
PB0EPRM	0V17	N	PFFOSC	0E03	T	PMADDR(9)	0R09	C
PB0PHY1	AC22	N	PIDSEL	0G17	D	PMCLK(0)	0P01	B
PB0PHY2	AB19	N	PINTCLK	0T17	N	PMCLK(1)	0R01	B
PBADDR16	0W17	N	PLLTUNE(0)	0G23	P	PMCLK(2)	0T01	B
PBADDR17	0Y17	N	PLLTUNE(1)	0E23	P	PMCLK(3)	0U01	B
PBALE1	AA18	N	PM0CS(0)	0R14	C	PMCLK(4)	0V01	B
PBALE2	0U16	N	PM0CS(1)	AC15	C	PMCLKE	AC11	C
PBDATA(0)	0W21	J	PM0CS(2)	AD15	C	PMDATA(0)	AB11	C
PBDATA(1)	0U18	J	PM0CS(3)	AE13	C	PMDATA(1)	AE10	C
PBDATA(2)	0W20	J	PM0DQM(0)	0Y13	C	PMDATA(10)	0W11	C
PBDATA(3)	0V19	J	PM0DQM(1)	0V13	C	PMDATA(11)	AB09	C
PBDATA(4)	AB23	J	PM0DQM(2)	0U13	C	PMDATA(12)	AA10	C
PBDATA(5)	0Y21	J	PM0DQM(3)	AD11	C	PMDATA(13)	AD09	C
PBDATA(6)	0W19	J	PM66EN	0A19	E	PMDATA(14)	0V11	C
PBDATA(7)	AA22	J	PMADDR(0)	AA04	C	PMDATA(15)	AC09	C
PBINTRA	AE20	N	PMADDR(1)	0W06	C	PMDATA(16)	AD07	C
PBPHYRST	AD19	N	PMADDR(10)	0Y03	C	PMDATA(17)	AC08	C
PBRDRDY	0W16	N	PMADDR(11)	0Y01	C	PMDATA(18)	AE08	C
PBRNWRT	AC18	N	PMADDR(12)	0U07	C	PMDATA(19)	0W10	C
PBSCLK	0T15	N	PMADDR(13)	0W04	C	PMDATA(2)	AE12	C
PBSDATA	AB17	J	PMADDR(14)	0V05	C	PMDATA(20)	0U11	C
PCBE(0)	0B15	D	PMADDR(15)	0U05	C	PMDATA(21)	0T11	C

Signal Pin Listing By Signal Name (Page 5 of 5)

Signal Name	Grid Position	Book
PMDATA(22)	0Y09	C
PMDATA(23)	AA09	C
PMDATA(24)	AA08	C
PMDATA(25)	AB07	C
PMDATA(26)	0W09	C
PMDATA(27)	AE06	C
PMDATA(28)	AC06	C
PMDATA(29)	0U10	C
PMDATA(3)	AC12	C
PMDATA(30)	0U09	C
PMDATA(31)	0Y07	C
PMDATA(32)	AA07	C
PMDATA(33)	0V09	C
PMDATA(34)	AA06	C
PMDATA(35)	0W08	C
PMDATA(36)	AB05	C
PMDATA(37)	0W07	C
PMDATA(38)	AC04	C
PMDATA(4)	AA11	C
PMDATA(5)	0U12	C
PMDATA(6)	AE09	C
PMDATA(7)	0Y11	C
PMDATA(8)	AE11	C
PMDATA(9)	AC10	C

Signal Name	Grid Position	Book
PMSYNCAS(0)	0V15	C
PMSYNCAS(1)	AA16	C
PMSYNRAS(0)	0W15	C
PMSYNRAS(1)	AE17	C
PMWE(0)	0R12	C
PMWE(1)	AC13	C
PPAR	0F15	D
PPAR64	0B11	D
PPLLOUT	0U17	J
PPLLTI	0W12	V
PVDDA	0E25	PLL
RTS	0M07	O
RXCLK	AE07	T
RXD	0M05	T
TESTM	0E01	W
TXCLK	AC07	T
TXD	0P05	O

Signal Name	Grid Position	Book
UNUSED_GND00	0D25	X
UNUSED_GND01	0F23	X
UNUSED_GND02	0H19	X
UNUSED_GND03	0N16	X
UNUSED_GND04	0N18	X
UNUSED_GND05	AA15	X
UNUSED_GND06	0T13	X
UNUSED_GND07	AD03	X
UNUSED_GND08	0J12	X
UNUSED_GND09	0J14	X
UNUSED_VDD2500	AC16	Y
UNUSED_VDD2501	AB13	Y
UNUSED_VDD2502	AE02	Y
UNUSED_VDD2503	0N05	Y
UNUSED_VDD3300	0F25	Z
UNUSED_VDD3301	0G24	Z
UNUSED_VDD3302	0N17	Z
UNUSED_VDD3303	0E13	Z

AC Timing Characteristics

PHY Timing

Description	Min	Max	Units
FYTWRB High to FYTDAT(15:0)	2	10	ns
FYTWRB High to FYTPAR(1:0)	2	10	ns
FYTWRB High to FYTSOC, FY0TENB, FYTMOD, FYTEOP, FYTADR(4:0)	2	10	ns
FYTCA to FYTWRB Setup	3	--	ns
FYTCA to FYTWRB Hold	1	--	ns
FY0FUL to FYTWRB Setup	3	--	ns
FY0FUL to FYTWRB Hold	1	--	ns
FYRRDB High to FYRENB, FYRADR(4:0)	2	10	ns
FYRDAT(15:0) to FYRRDB Setup	3	--	ns
FYRDAT(15:0) to FYRRDB Hold	1	--	ns
FYRPAR(1:0) to FYRRDB Setup	3	--	ns
FYRPAR(1:0) to FYRRDB Hold	1	--	ns
FYRSOC to FYRRDB Setup	3	--	ns
FYRSOC to FYRRDB Hold	1	--	ns
FY0FUL to FYRRDB Setup	3	--	ns
FY0FUL to FYRRDB Hold	1	--	ns
FY0EMP to FYRRDB Setup	3	--	ns
FY0EMP to FYRRDB Hold	1	--	ns
FY0DISCRD to FYRRDB Setup	3	--	ns
FY0DISCRD to FYRRDB Hold	1	--	ns
FY0DTCT to FYRRDB Setup	3	--	ns
FY0DTCT to FYRRDB Hold	1	--	ns
FYRCA to FYRRDB Setup	3	--	ns
FYRCA to FYRRDB Hold	1	--	ns
FYRMOD to FYRRDB Setup	3	--	ns
FYRMOD to FYRRDB Hold	1	--	ns
FYREOP to FYRRDB Setup	3	--	ns
FYREOP to FYRRDB Hold	1	--	ns

NPBUS Timing

Description	Min	Max	Units
PINTCLK High to PBDATA(7:0)	2	6	ns
PINTCLK High to PBDATAP	2	6	ns
PINTCLK High to PBRNWRT	2	6	ns
PINTCLK High to PBRDRDY	2	6	ns
PINTCLK High to PBADDR(15:0)	2	6	ns
PBDATA(7:0) to PINTCLK Setup	15	--	ns
PBDATA(7:0) to PINTCLK Hold	0	--	ns
PBDATAP to PINTCLK Setup	15	--	ns
PBDATAP to PINTCLK Hold	0	--	ns
PBRDRDY to PINTCLK Setup	15	--	ns
PBRDRDY to PINTCLK Hold	0	--	ns
PBRNWRT to PINTCLK Setup	15	--	ns
PBRNWRT to PINTCLK Hold	0	--	ns

I/O PCI Bus Timing (Page 1 of 2)

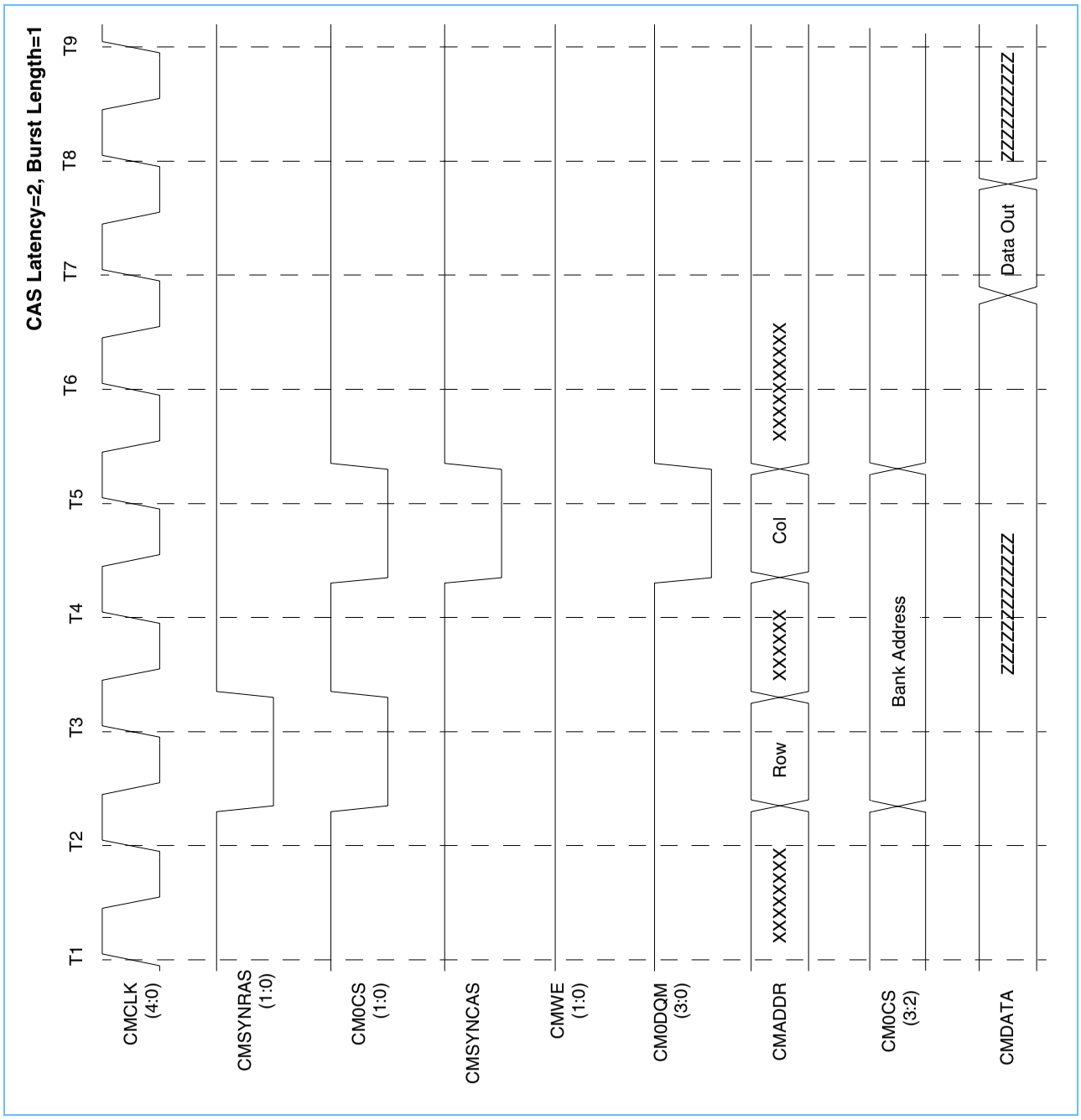
Description	Min	Max	Units
PCICLK High to PAD(31:0)	2	6	ns
PCICLK High to PPAR	2	6	ns
PCICLK High to PCBE(3:0)	2	6	ns
PCICLK High to MFRAME	2	6	ns
PCICLK High to MTRDY	2	6	ns
PCICLK High to MIRDY	2	6	ns
PCICLK High to MSTOP	2	6	ns
PCICLK High to MDEVSEL	2	6	ns
PCICLK High to MREQ	2	6	ns
PCICLK High to MPERR	2	6	ns
PCICLK High to MSERR	2	6	ns
PCICLK High to MINITA	2	6	ns
PCICLK High to MINT2	2	6	ns
PCICLK High to MREQ64	2	6	ns
PCICLK High to PPAR64	2	6	ns
PCICLK High to PAD64(63:32)	2	6	ns
PCICLK High to MACK64	2	6	ns
MFRAME to PCICLK Setup	3	--	ns
MFRAME to PCICLK Hold	0	--	ns
PCBE(3:0) to PCICLK Setup	3	--	ns

I/O PCI Bus Timing (Page 2 of 2)

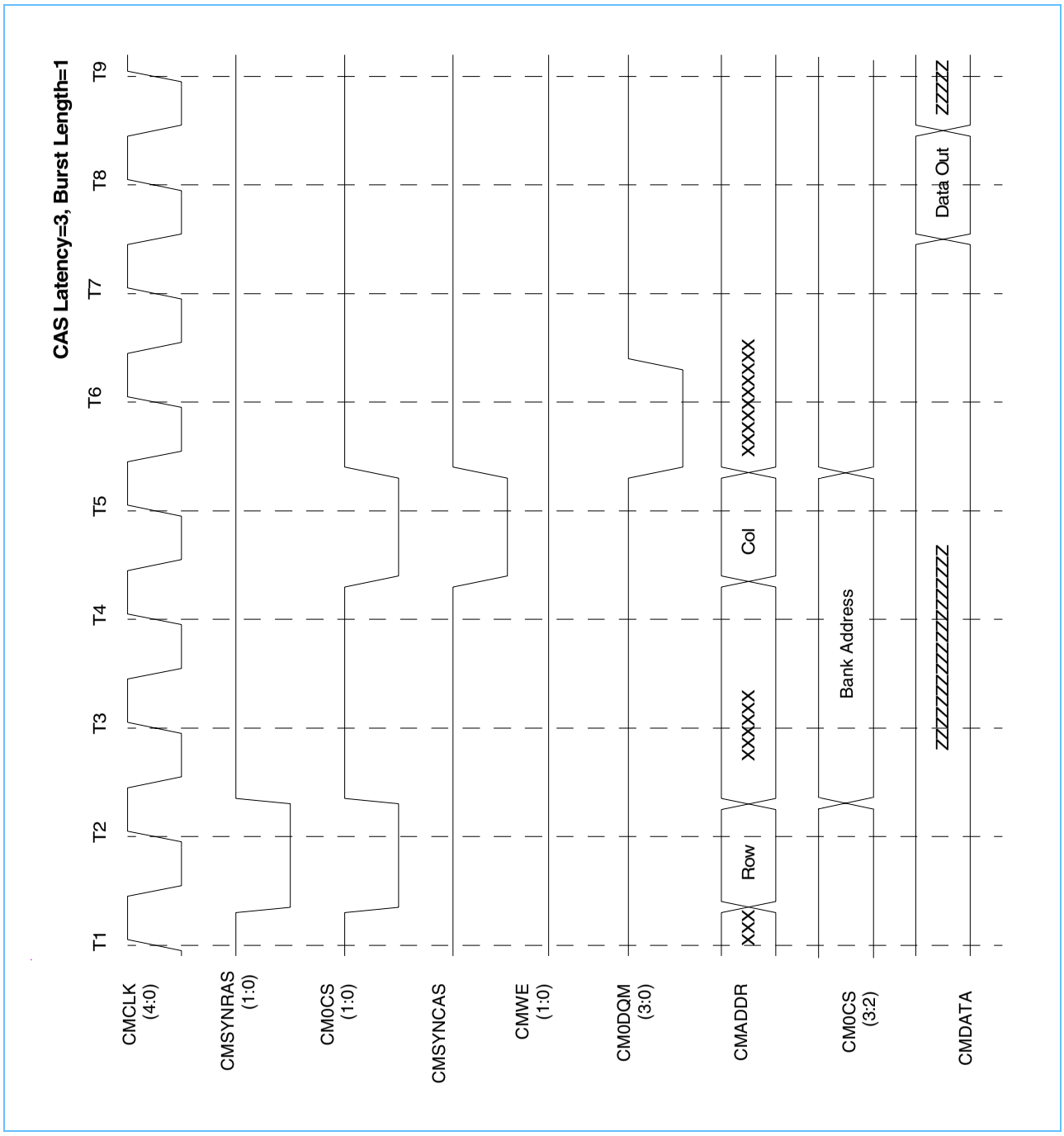
Description	Min	Max	Units
PCBE(3:0) to PCICLK Hold	0	--	ns
PAD(31:0) to PCICLK Setup	3	--	ns
PAD(31:0) to PCICLK Hold	0	--	ns
PPAR to PCICLK Setup	3	--	ns
PPAR to PCICLK Hold	0	--	ns
MPERR to PCICLK Setup	3	--	ns
MPERR to PCICLK Hold	0	--	ns
PIDSEL to PCICLK Setup	3	--	ns
PIDSEL to PCICLK Hold	0	--	ns
MDEVSEL to PCICLK Setup	3	--	ns
MDEVSEL to PCICLK Hold	0	--	ns
MTRDY to PCICLK Setup	3	--	ns
MTRDY to PCICLK Hold	0	--	ns
MIRDY to PCICLK Setup	3	--	ns
MIRDY to PCICLK Hold	0	--	ns
MSTOP to PCICLK Setup	3	--	ns
MSTOP to PCICLK Hold	0	--	ns
MGNT to PCICLK Setup	3	--	ns
MGNT to PCICLK Hold	0	--	ns
PCBE64(7:4) to PCICLK Setup	3	--	ns
PCBE64(7:4) to PCICLK Hold	0	--	ns

Synchronous DRAM Timing Diagrams

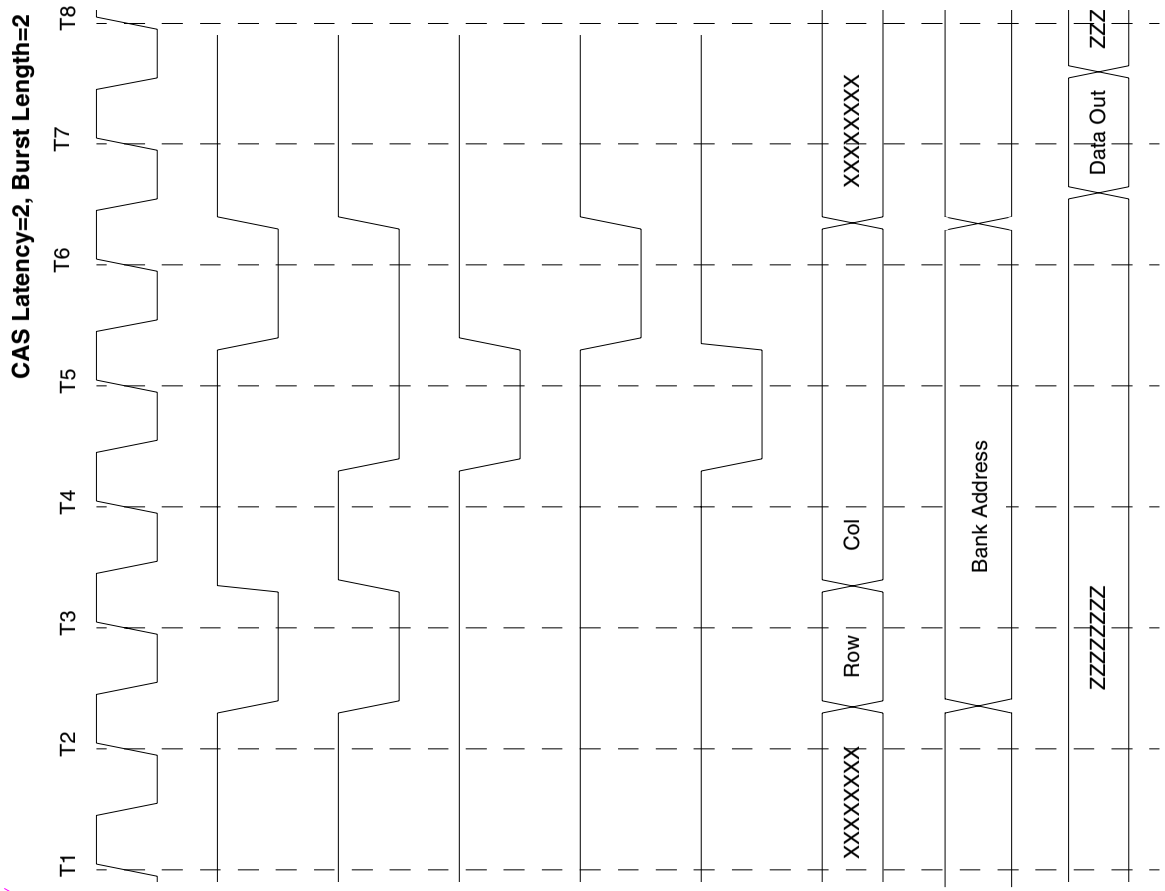
SDRAM Read Cycle (1 of 4)



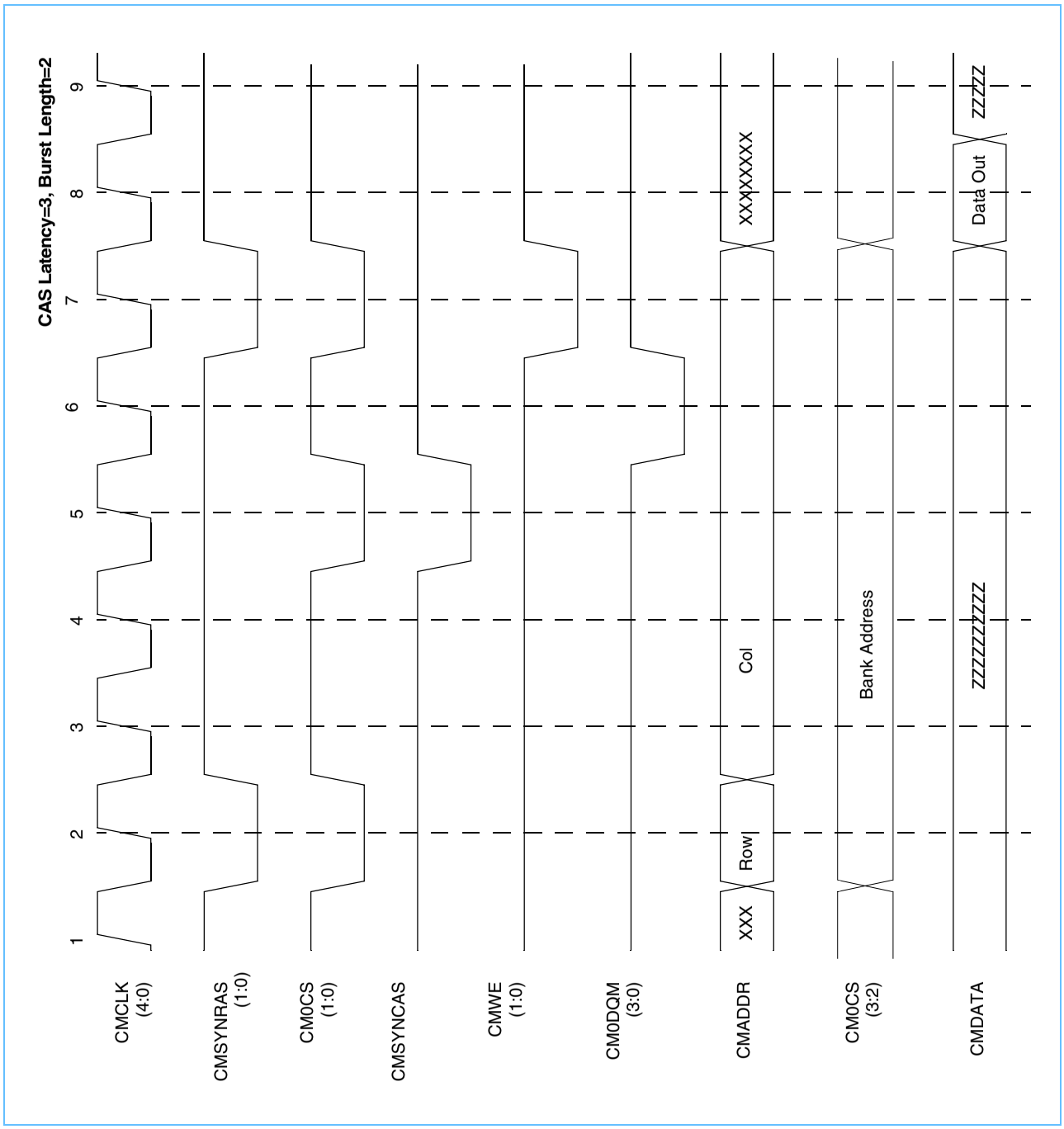
SDRAM Read Cycle (2 of 4)



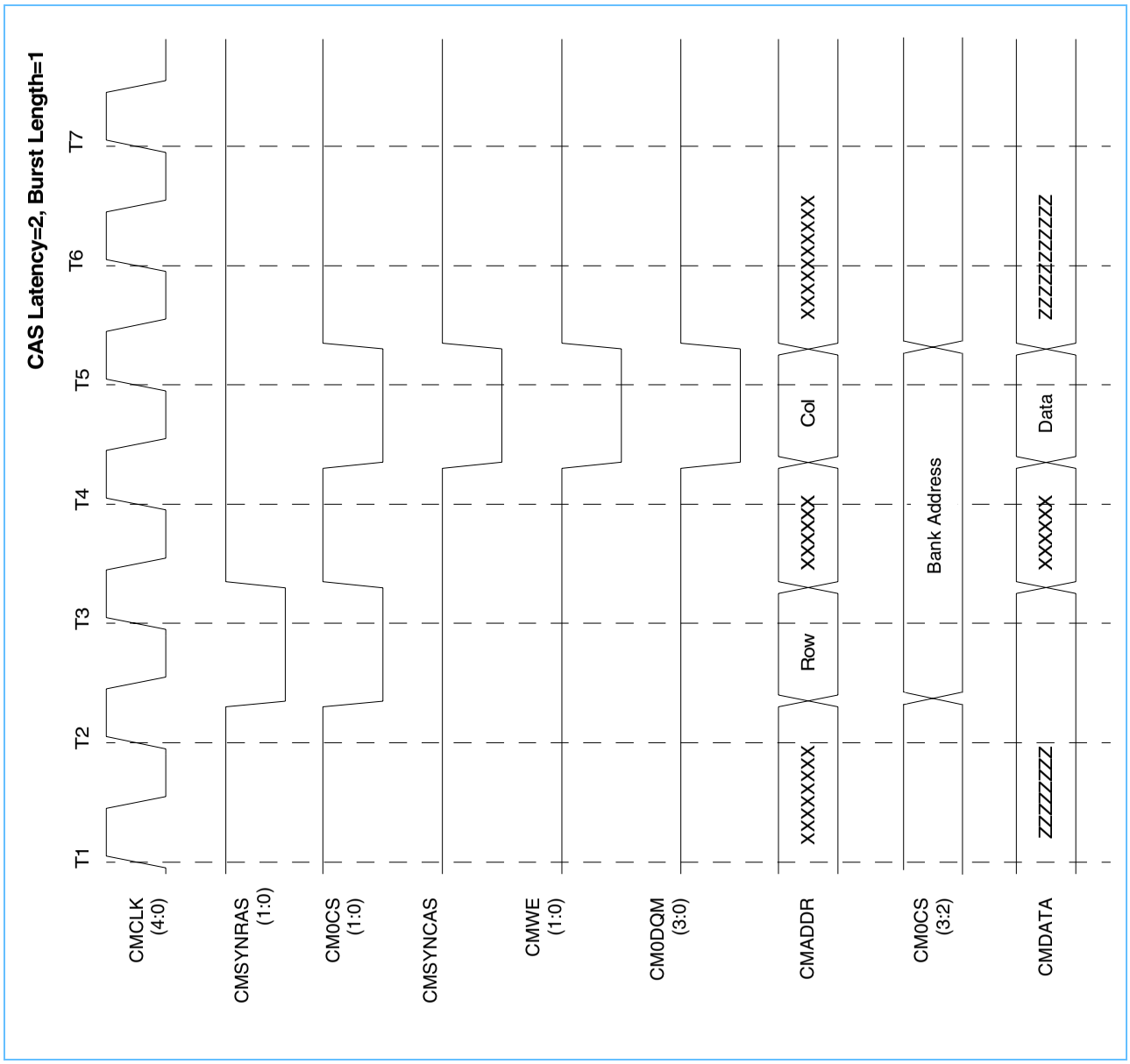
SDRAM Read Cycle (3 of 4)



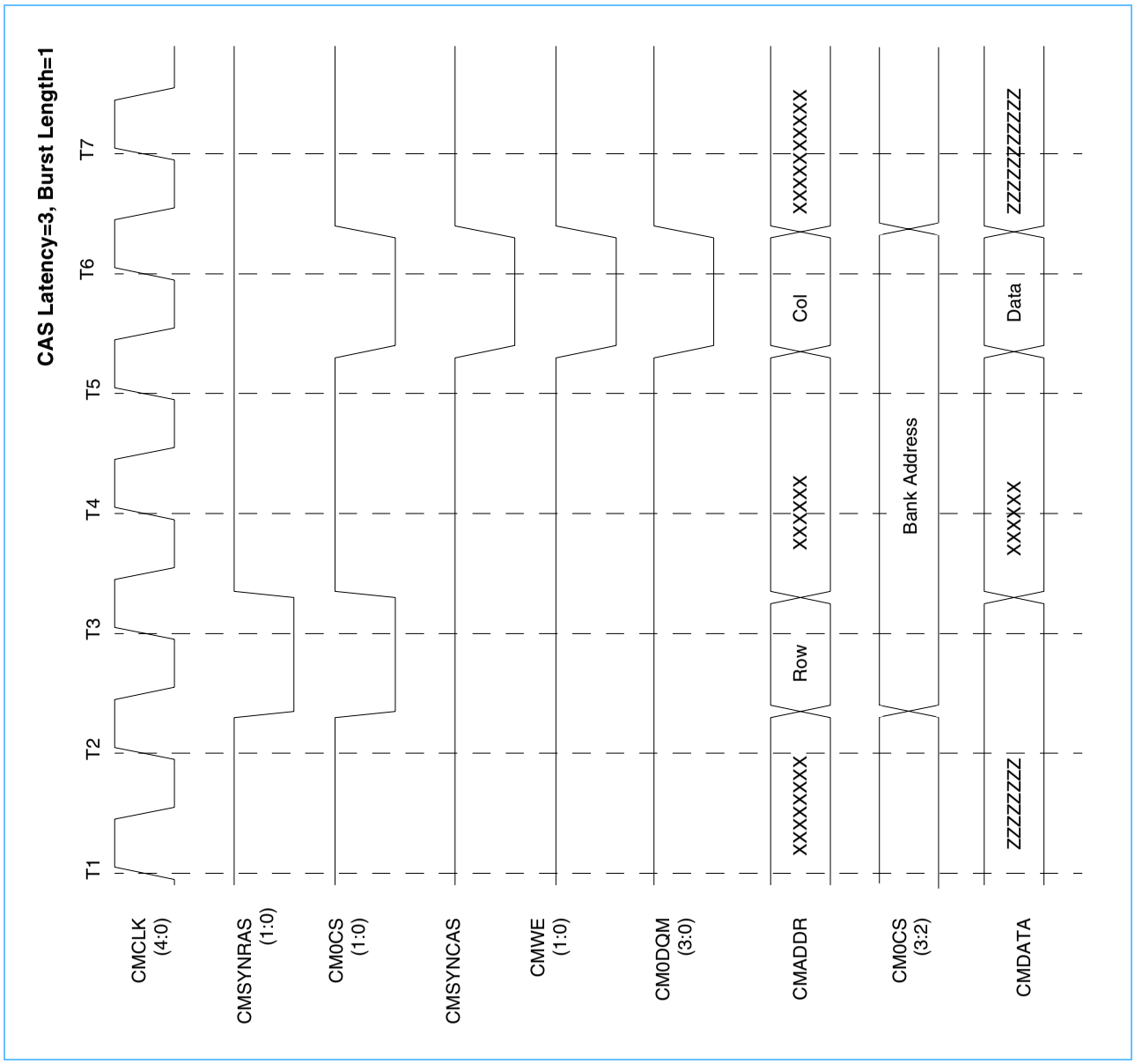
SDRAM Read Cycle (4 of 4)



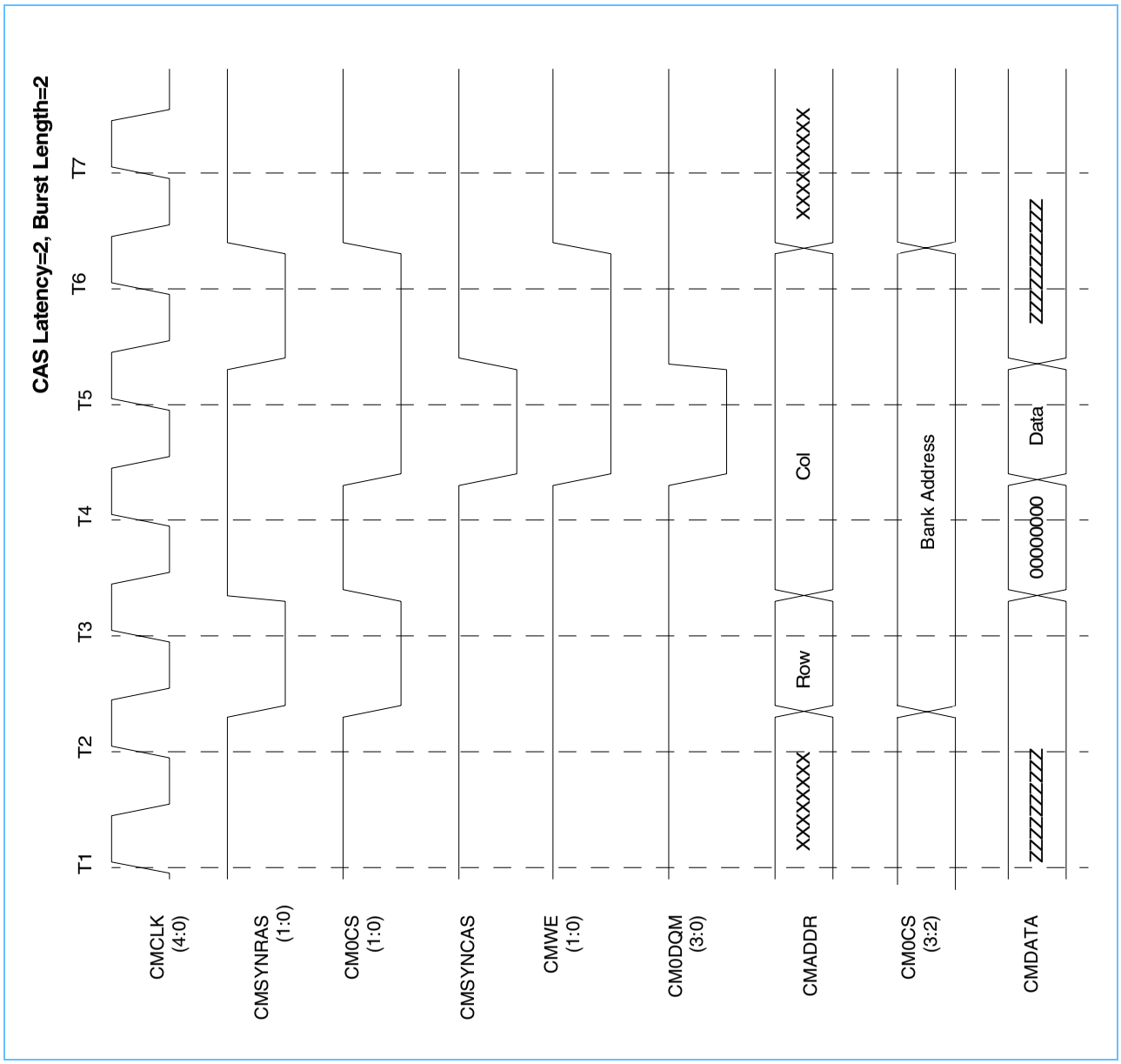
SDRAM Write Cycle (1 of 4)



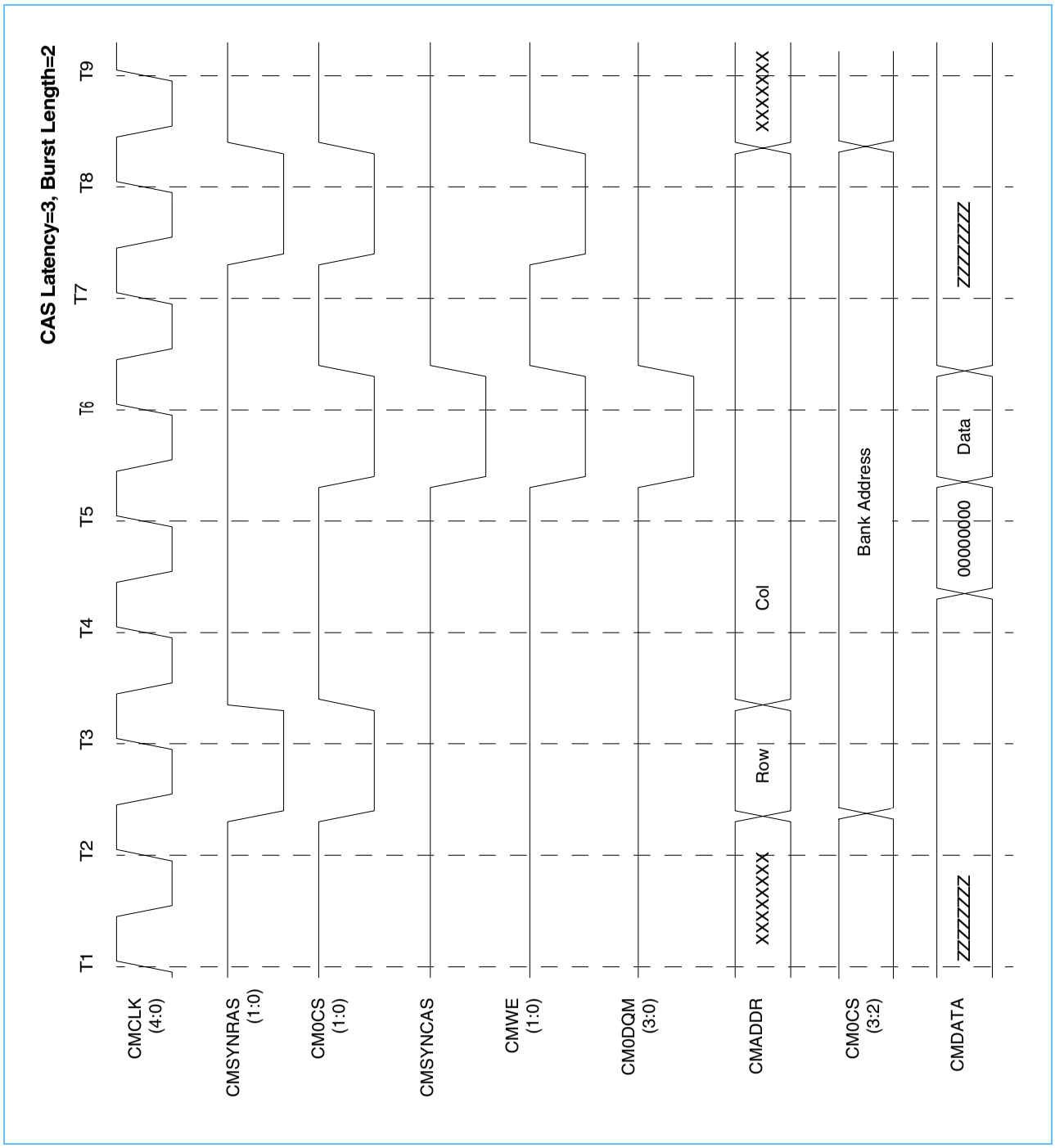
SDRAM Write Cycle (2 of 4)



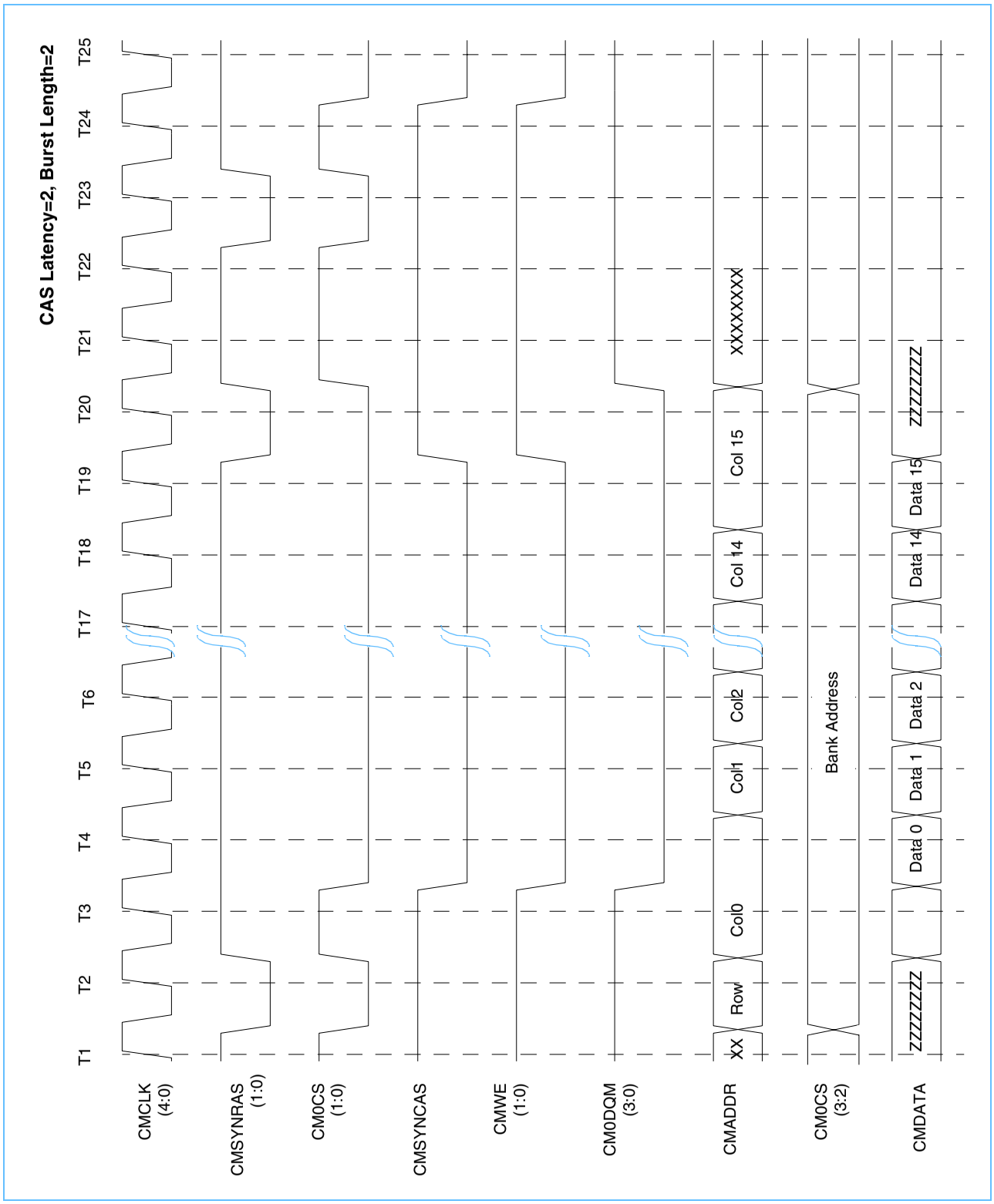
SDRAM Write Cycle (3 of 4)



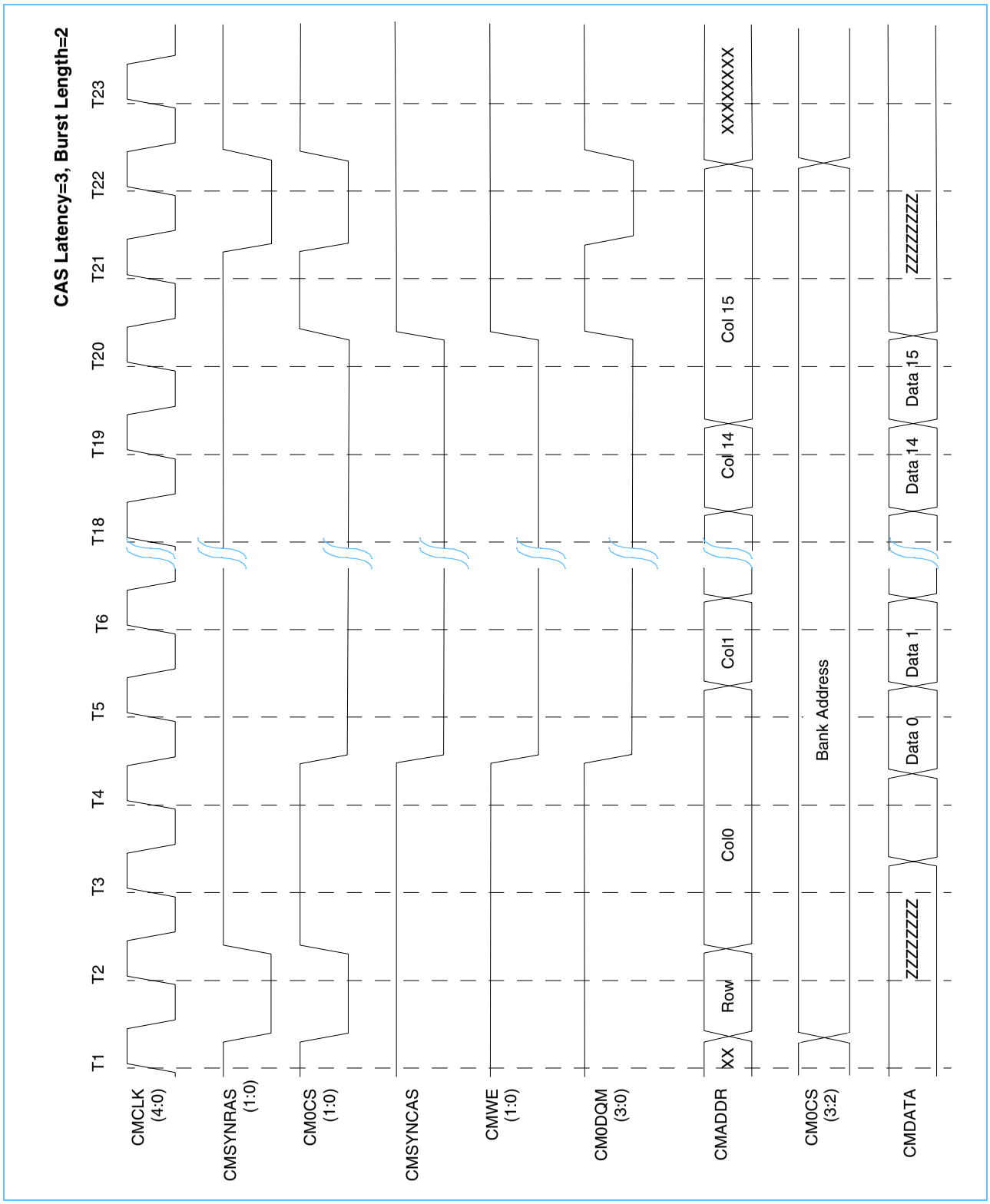
SDRAM Write Cycle (4 of 4)



SDRAM Write of 64-byte Burst with CAS Latency=2

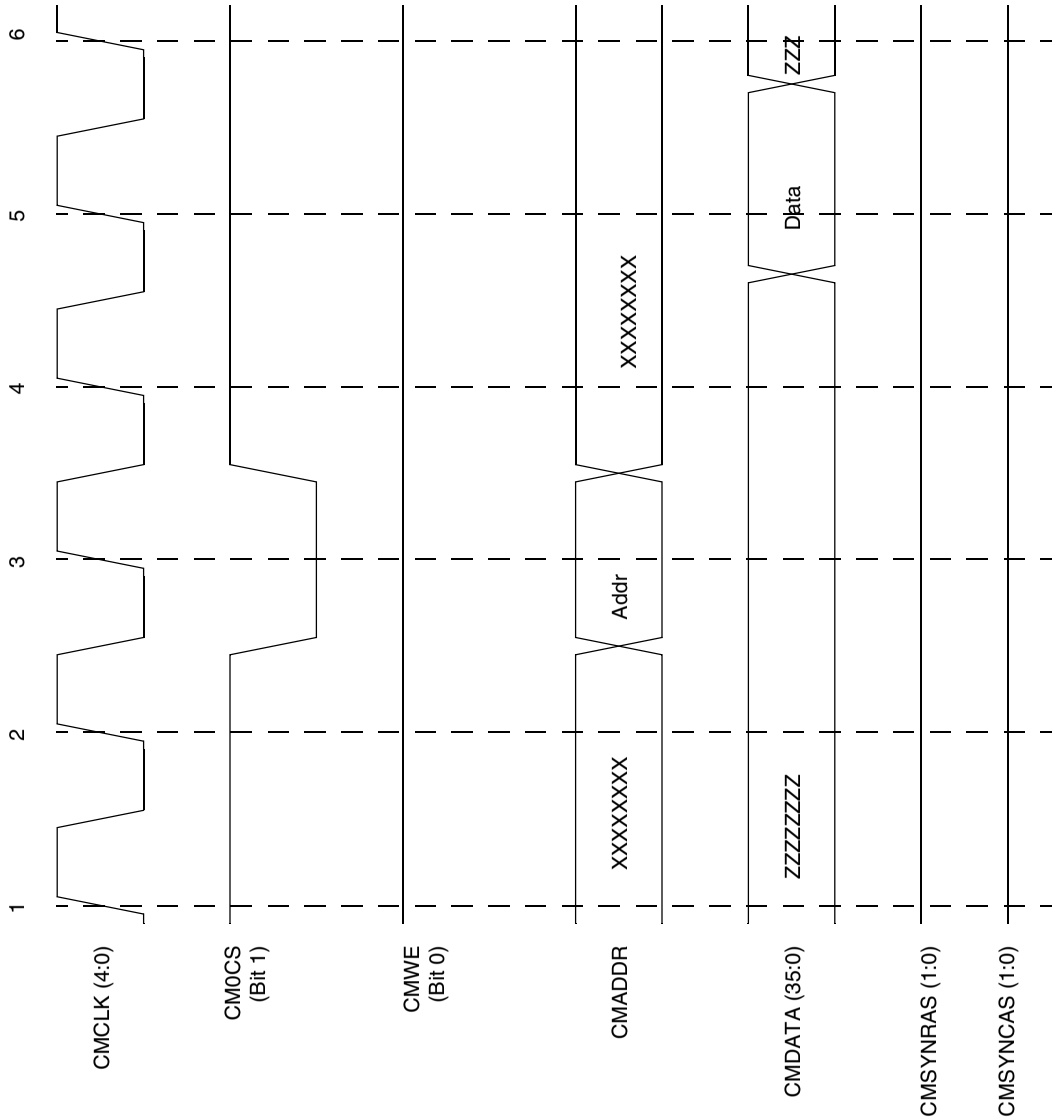


SDRAM Write of 64-byte Burst with CAS Latency=3

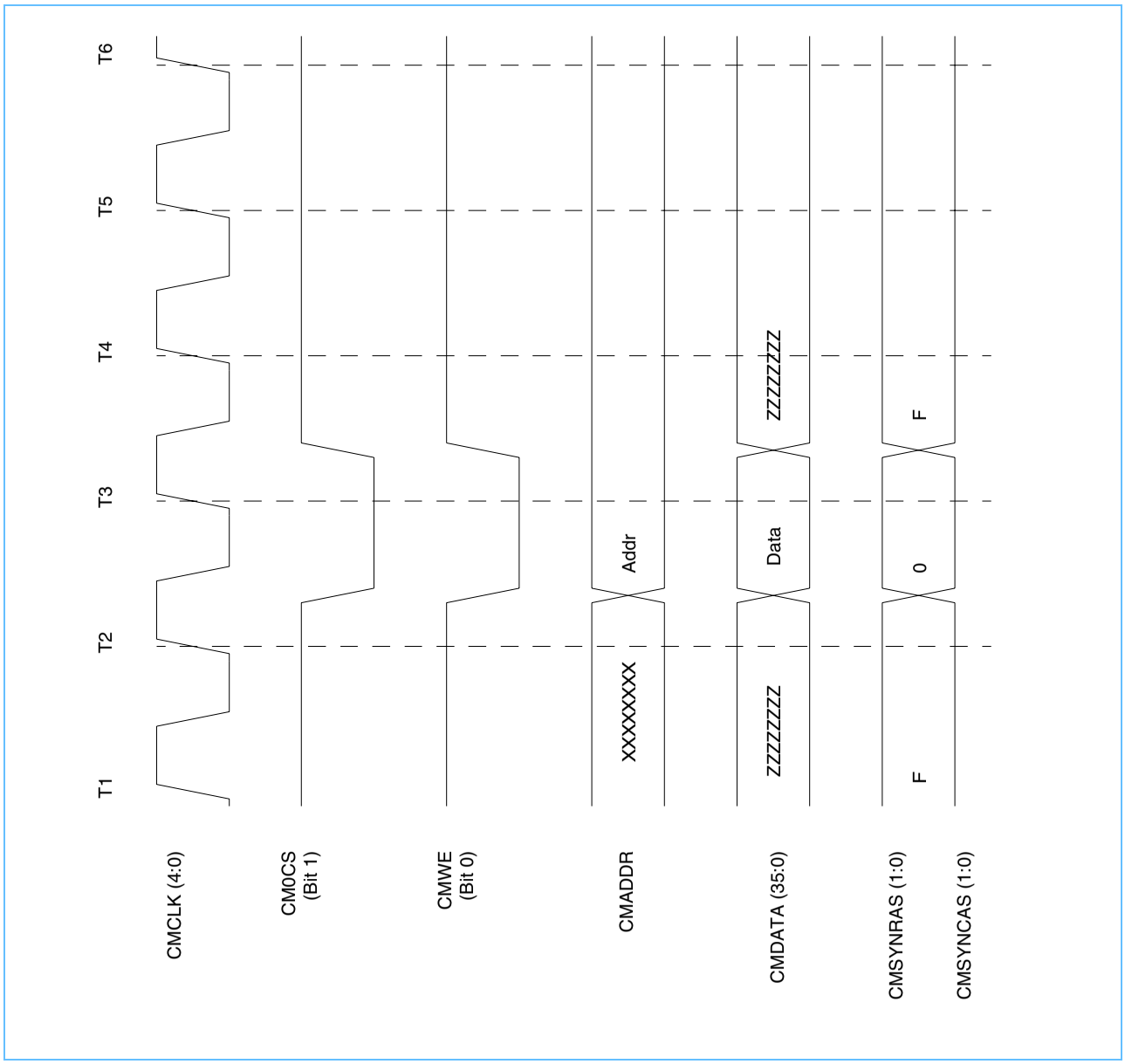


SRAM Timing Diagrams

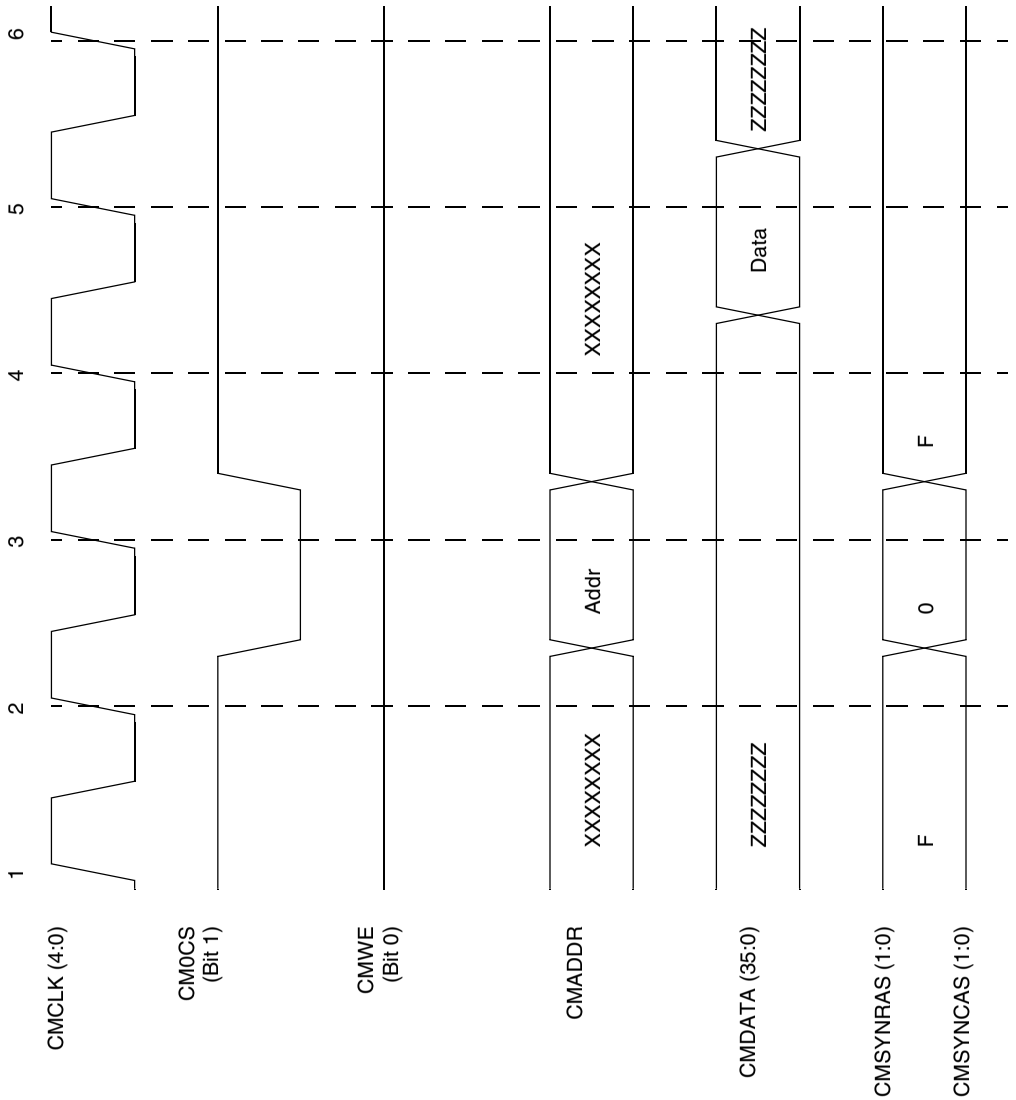
SRAM Read Cycle



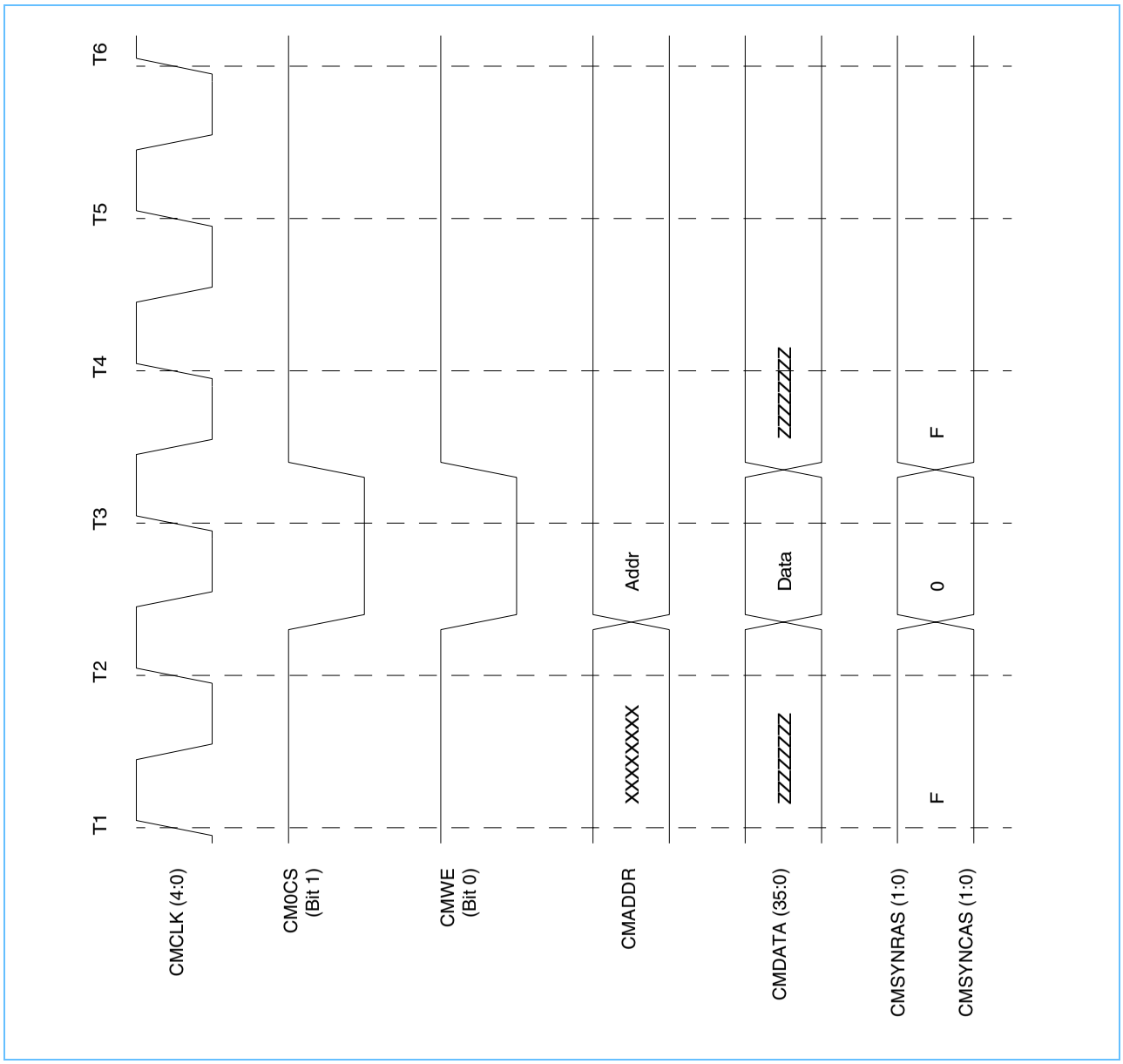
SRAM Write Cycle



SRAM Read Cycle with Byte Enables

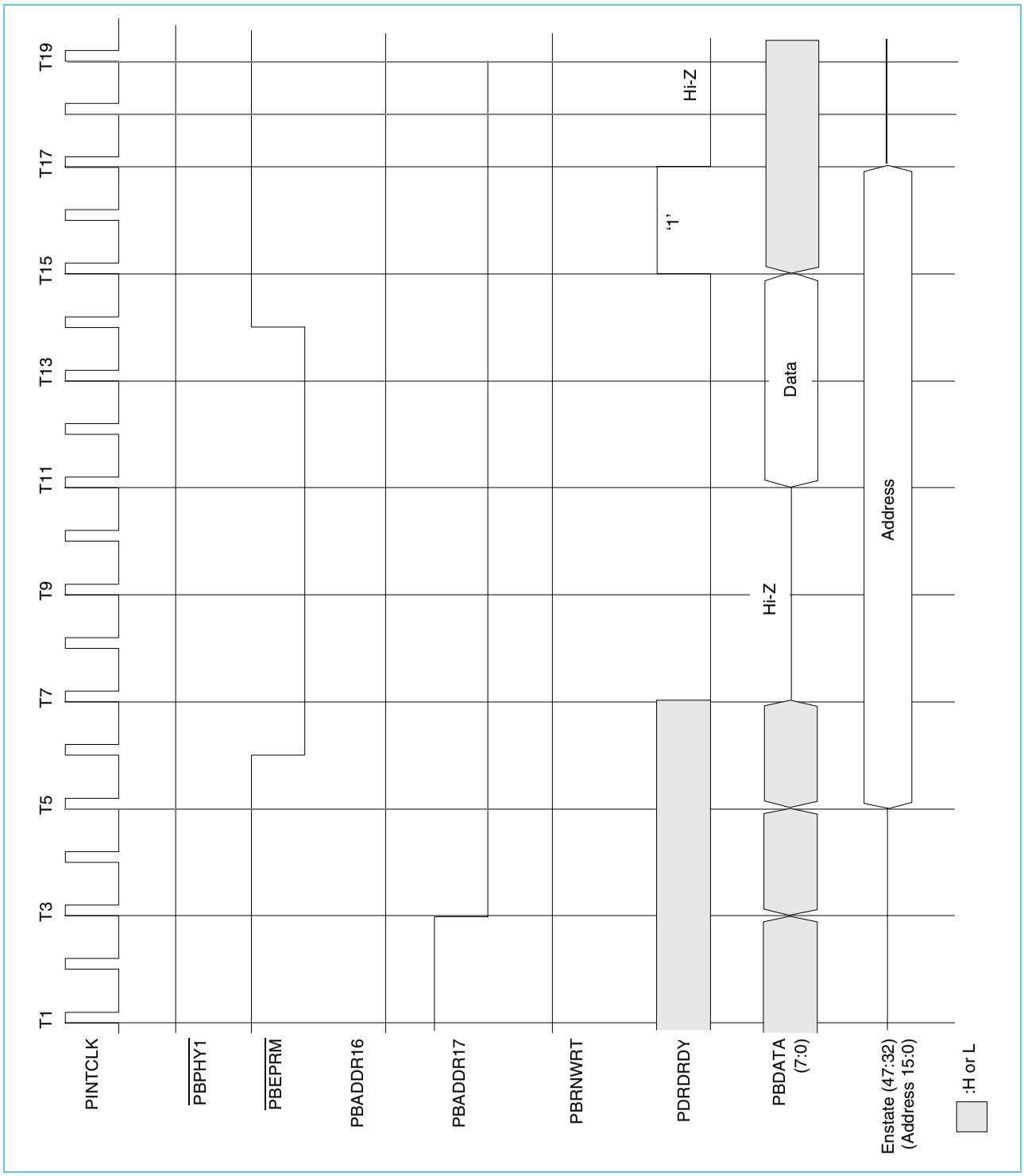


SRAM Write Cycle with Byte Enables

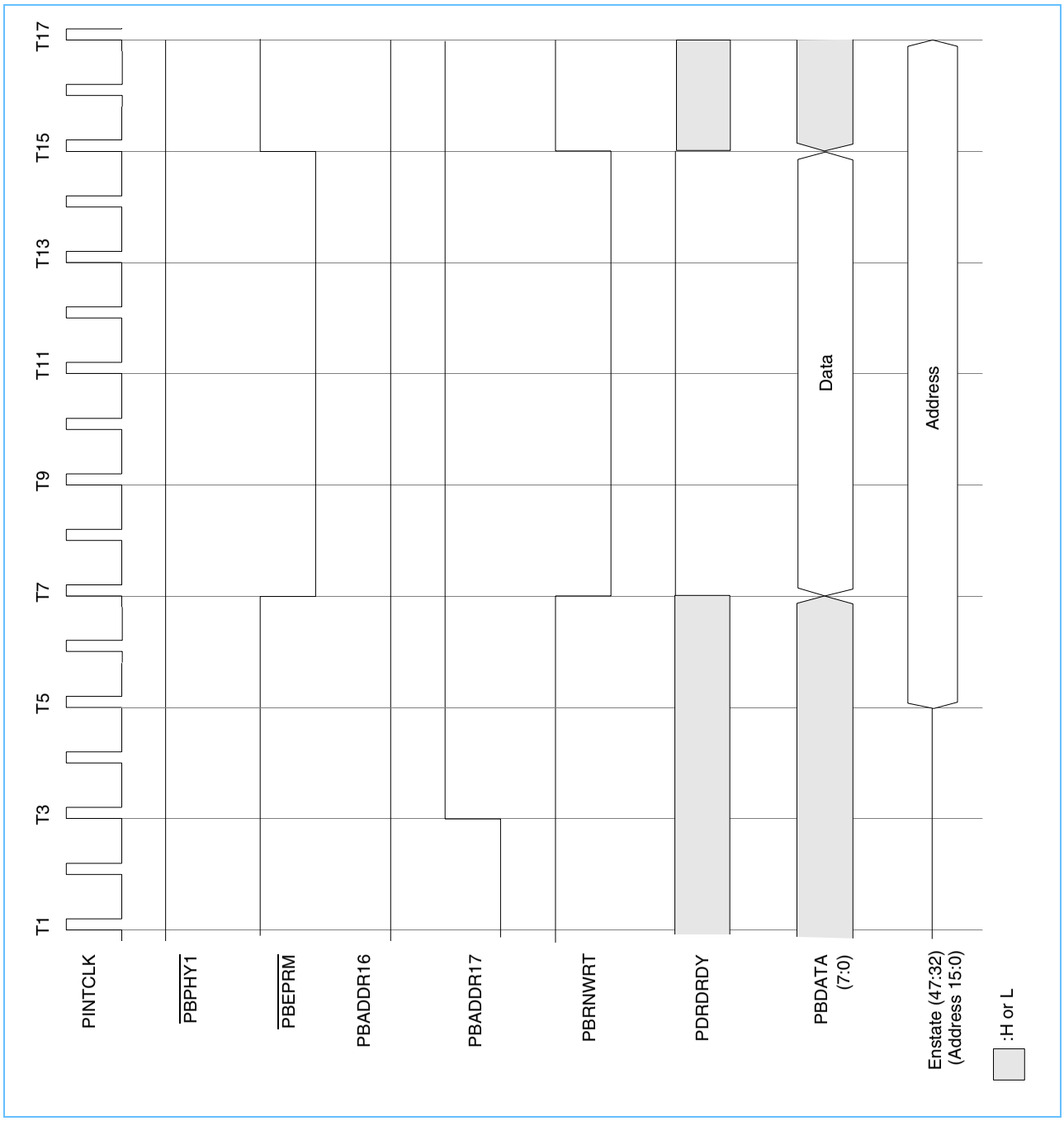


EPROM Timing Diagrams

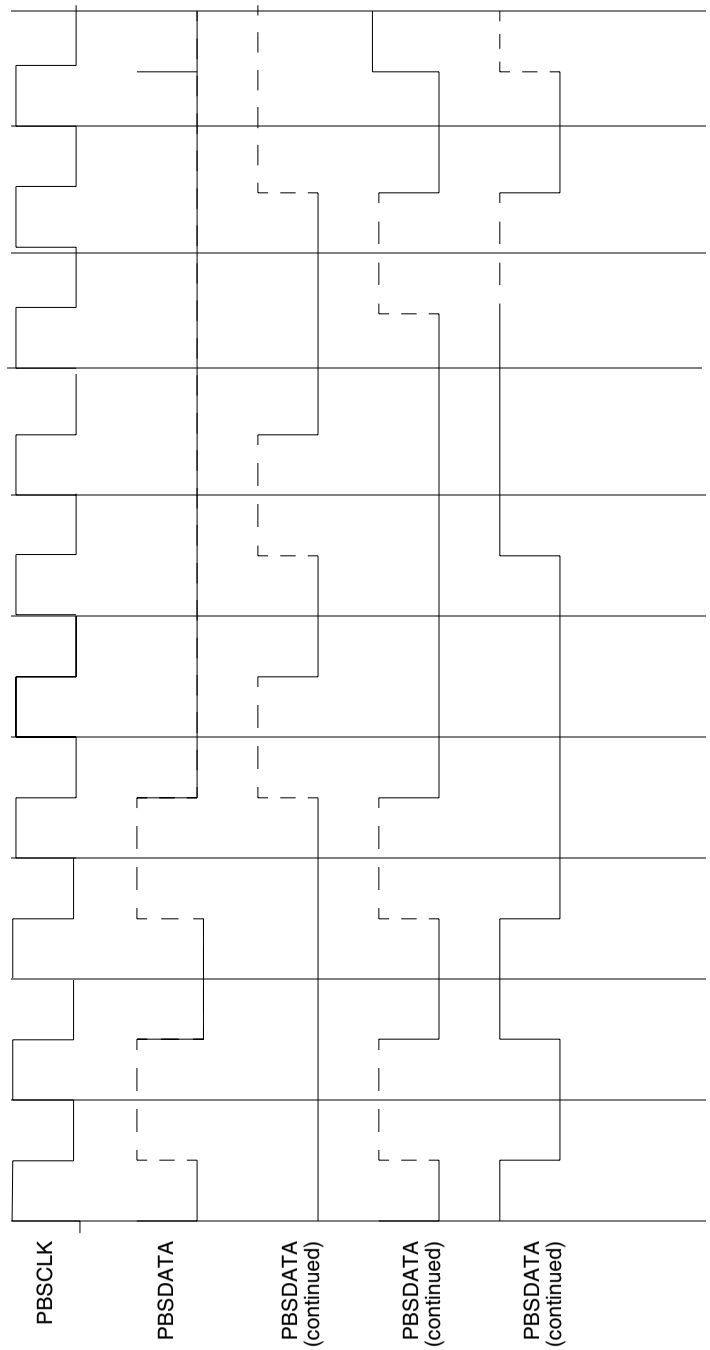
Parallel EPROM Read



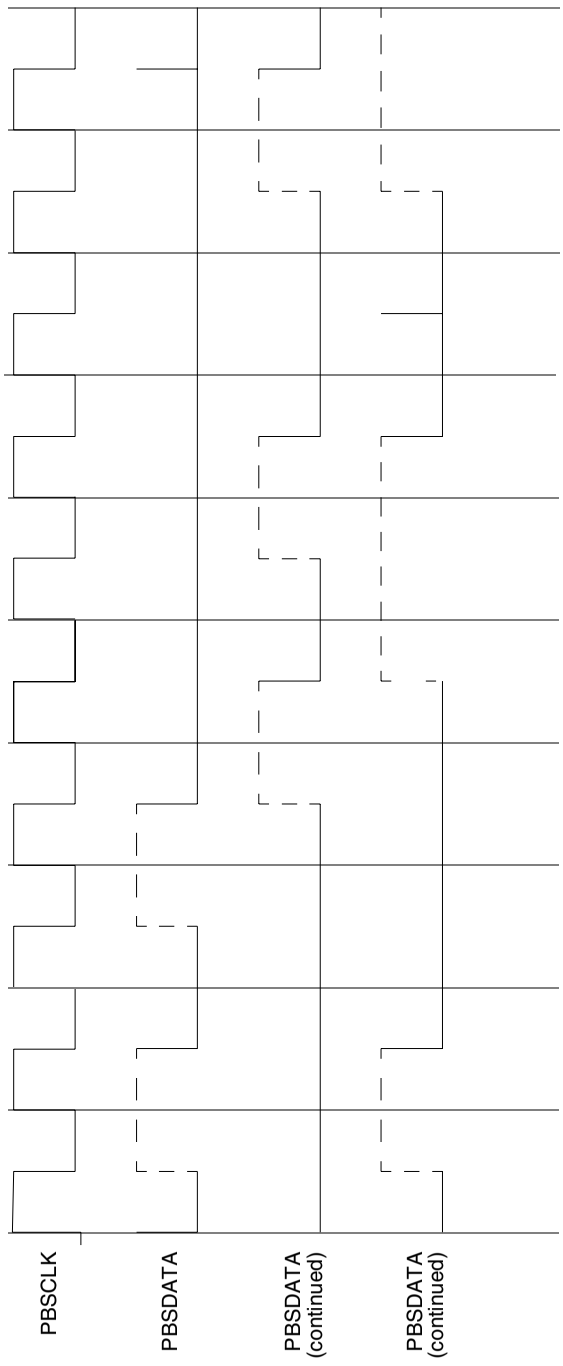
Parallel EPROM Write



Serial EPROM Read

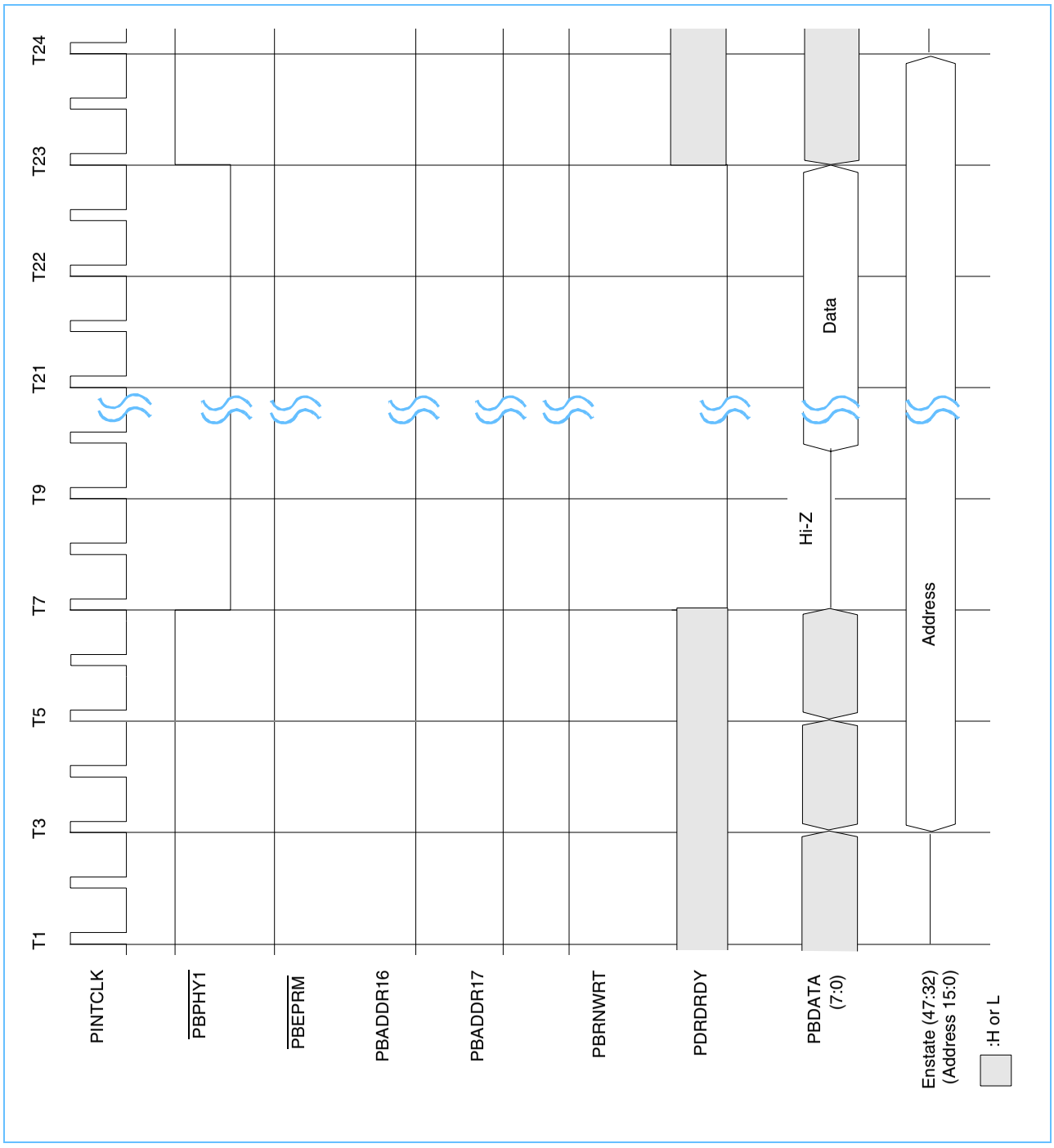


Serial EPROM Write

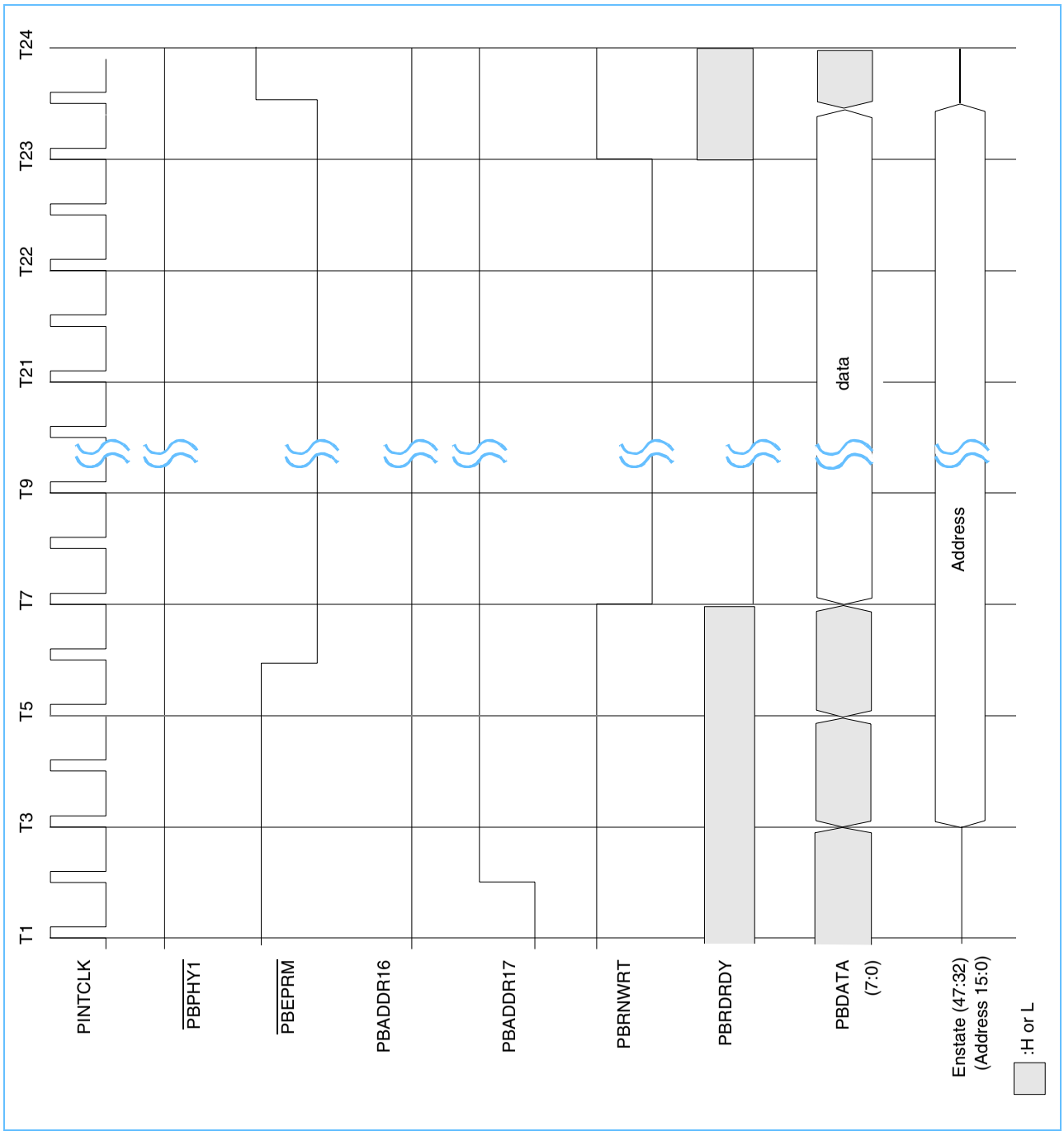


PHY Timing Diagrams

PHY Read



PHY Write


T1
T3
T5
T7
T9
T21
T22
T23
T24
PINTCLK
PBPHY1
PBEPRM
PBADDR16
PBADDR17
PBRNWRT
PBRDRDY
PBDATA (7:0)
Enstate (47:32) (Address 15:0)
data
Address
:H or L

Revision Log

Rev.	Description
8/31/99	Initial release (00).
8/14/00	First revision (01). In the Input/Output Definitions section, corrected the definition of RXCLK in *Clock, Configuration, and LSSD Pin Descriptions* on page 58. Changed page numbering of *Contents* from roman to numeric.