Preliminary

FUJITSU

MB86613S
IEEE1394 Open HCI Controller

Product Specification

Provisional

Revision 1.00, July 2, 2001

Fujitsu Limited
Fujitsu VLSI Ltd.

1. Introduction

Related Documents

This specification document was prepared based on the following documents:

1) IEEE1394- 1995 High Performance Serial Bus and P1394a draft2.0
2) 1394 Open Host Controller Interface (Open HCI) Specification Release1.1
3) PCI Local Bus Specification (Revision 2.2)
4) PCI Bus Power Management Specification (Version 1.1)

1.1. Overview

MB86613S is Fujitsu's IEEE1394- OHCI (Open Host Controller Interface) Controller LSI that is compliant with IEEE1394- 1995, P1394a and OHCI (revision 1.1, release) standard drafts. This LSI integrates both 1394 PHY and LINK layers including analog PLL, transceiver, and comparator circuits using Fujitsu's advanced full CMOS process for the cost- effective single- chip solution.

In addition to the 1394 block, the MB86613S contains various DMA engines called ContextProgram Controllers used for OHCI functions and PCI block. ContextProgram block consists of total 13 channels of independent DMA that are each dedicated to asynchronous and isochronous transmit and isochronous- asynchronous common receive operations. On- chip, 5V and 3.3V operable, PCI bus controller is compliant with PCI local bus standard (revision2.2) incorporating one 32- bit DMA controller and power management functions as specified in PCI bus power management specification (version 1.1).

For valuable host side design, this chip also incorporates serial Configuration ROM interface.

The device operates by +5V or +3.3V power supply for the PCI and DMA blocks and +3.3V for the whole 1394 block.

To provide with the cost- effective solution, the LSI is housed in a 100- pin plastic small QFP package.

1.2. Features

1) 1394 Serial Bus Controller Block:

   - Compliant with IEEE1394- 1995 and P1394a draft2.0
   - Integrates PHY and LINK layers into single- chip.
   - 1394 port number : 1 port
   - Transfer Data Rate : S100, S200, and S400
   - On- chip PLL : 400MHz for PHY and 50MHz for Link core.
   - Cycle- Master Function
   - On- chip Bus Management CSRs
   - 6- pin cable supported
   - On- chip transceiver and comparator
   - On- chip another comparator for detecting the cable power

2) ContextProgram Controller Block :

   - Compliant with Open HCI standard draft (revision 1.1)
   - Total 13 independent ContextProgram Controllers:
     a) Asynchronous Transmit DMA : 2 channels for response and request each
     b) Isochronous Transmit DMA : 4 channels
     c) Receive DMA : 7 channels for Asynchronous response and request each, 4 isochronous, and 1 self- ID receive

- On- chip 6KB FIFO :
  a) Asynchronous Transmit- FIFO : 1.5KB
  b) Isochronous Transmit- FIFO : 1.5KB
  c) Asynchronous/Isochronous Receive- FIFO : 3.0KB

- On- chip context program work memory : 128B x 3

3) PCI Bus Controller Block :

- Compliant with PCI local bus specification (revision 2.2)
- On- chip 32- bit DMA controller
- On- chip power management (PCI power management standard, revision 1.1, compliant)
- Alignment function
- Byte swap function
- 33MHz operation
- On- chip serial ROM interface
- On- chip universal type (5V/3.3V) PCI buffer.

4) Others:

-
- 100- pin plastic LQFP package
- Two power supply systems : +5V and +3.3V

1.3. Block Diagram

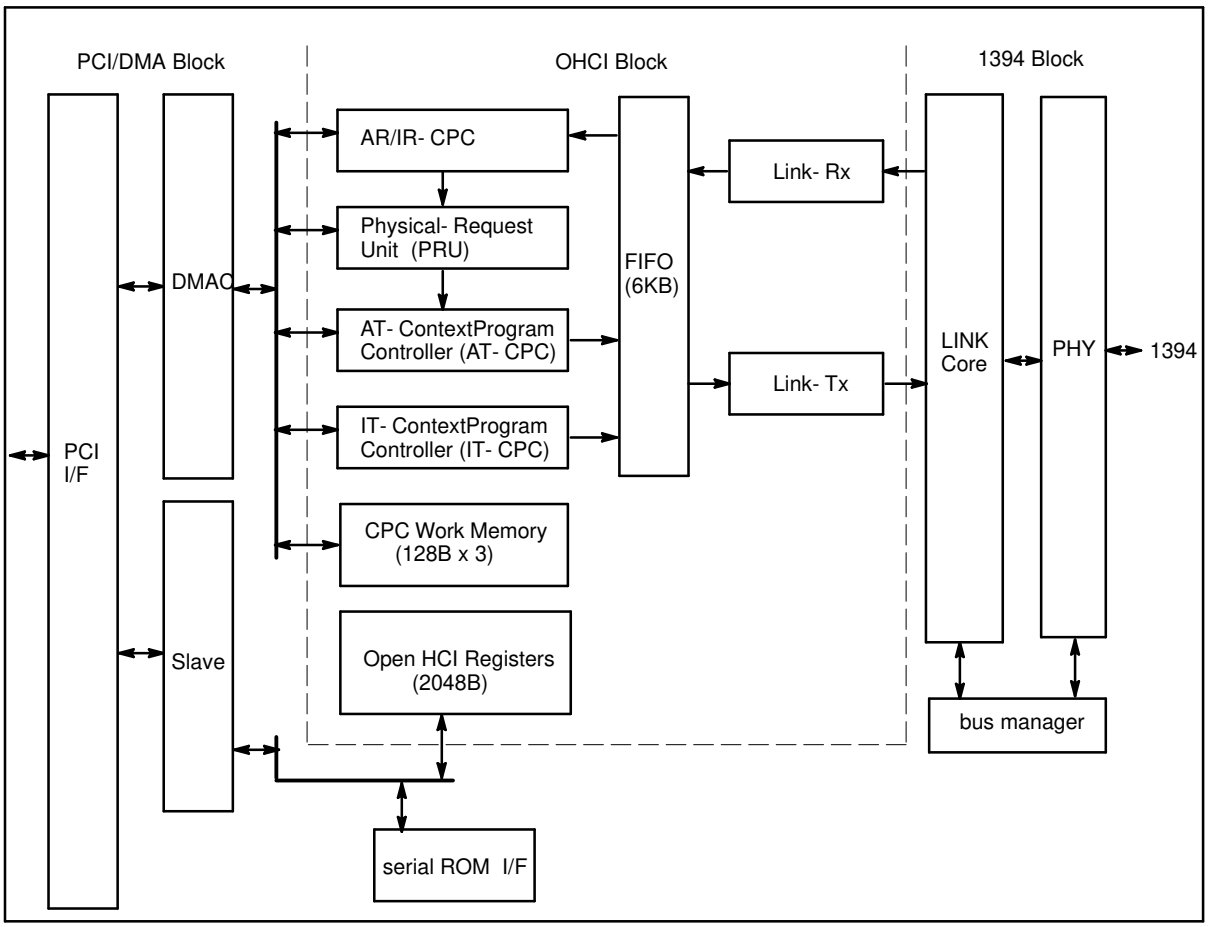


Fig. 1.1 Block Diagram of MB86613S

1.4. Block Description

1.5.1. PCI Block

PCI block consists of the following components :

(1) PCI Interface:
32- bit, 33MHz, PCI local bus interface compliant with PCI spec revision 2.2. This is the 5V/3.3V operable local bus interface with host side.

(2) DMAC
DMA bus master which is associated with the PCI interface.

(3) Slave:
DMA slave mode controller which is used for the register access in response to the DMA bus master request.

(4) Serial ROM Interface:
This interface connects with an EEPROM (PCI Configuration ROM) containing various information such as PCI subsystem ID, and subsystem vendor ID.

1.5.2. OHCI Block

OHCI block includes the powerful DMA engines for ContextProgram Control, work memory, FIFO, and LINK- Tx, - Rx sections:

(1) ContextProgramController (CPC) :
This controller is ContextProgram Processor Unit that analyzes the context programs stored in the system memory.

For Receive state, the on- chip R- CPC transfers the OHCI packet stored in the FIFO to the system memory via PCI interface. For Transmit state, the on- chip AT- , and IT- CPC transmit the packet stored in the system memory to the FIFO.

This section also includes a PHY Request Unit (PRU) which works for automatically decoding the physical request packet if received.

1.5.3. ContextProgram Work Memory (CP Work)

This block is work RAM area used for storing IT- , AT- , and, Receive- context program to be processed, and also storing header information on received physical request packet. The area is formed by 128B x 3 units of SRAM.

1.5.4. Open HCI Registers

This block contains total 2048- byte register set as defined in the Open HCI standard. For the register map, see Section 3 in this document.

1.5.5. FIFO

This is a 6Kbytes FIFO memory used for storing the packet received and to be transmitted. The internal area is divided, one for Asynchronous/Isochronous Transmit (AT- FIFO and IT- FIFO) and one for Asynchronous/ Isochronous Receive (R- FIFO).

1.5.6. LINK- Rx

This block is on OHCI- 1394 boundary that is placed between the internal FIFO and LINK core. LINK- Rx is responsible for adding the OHCI trailer data to the packet received by 1394 LINK core and storing into the FIFO.

1.5.7. LINK- Tx

Similarly with LINK- Rx, this Link- Transmit section converts the OHCI packet stored in the FIFO by CPC into the 1394 format packet for the LINK core.

1.5.8. LINK Core

This block generates the data CRC, header CRC, and also transmits self ID packets and controls packet transmit and receive.

1.5.9. PHY

This block controls the 1394 serial bus protocol, including data encoding and bus arbitration.

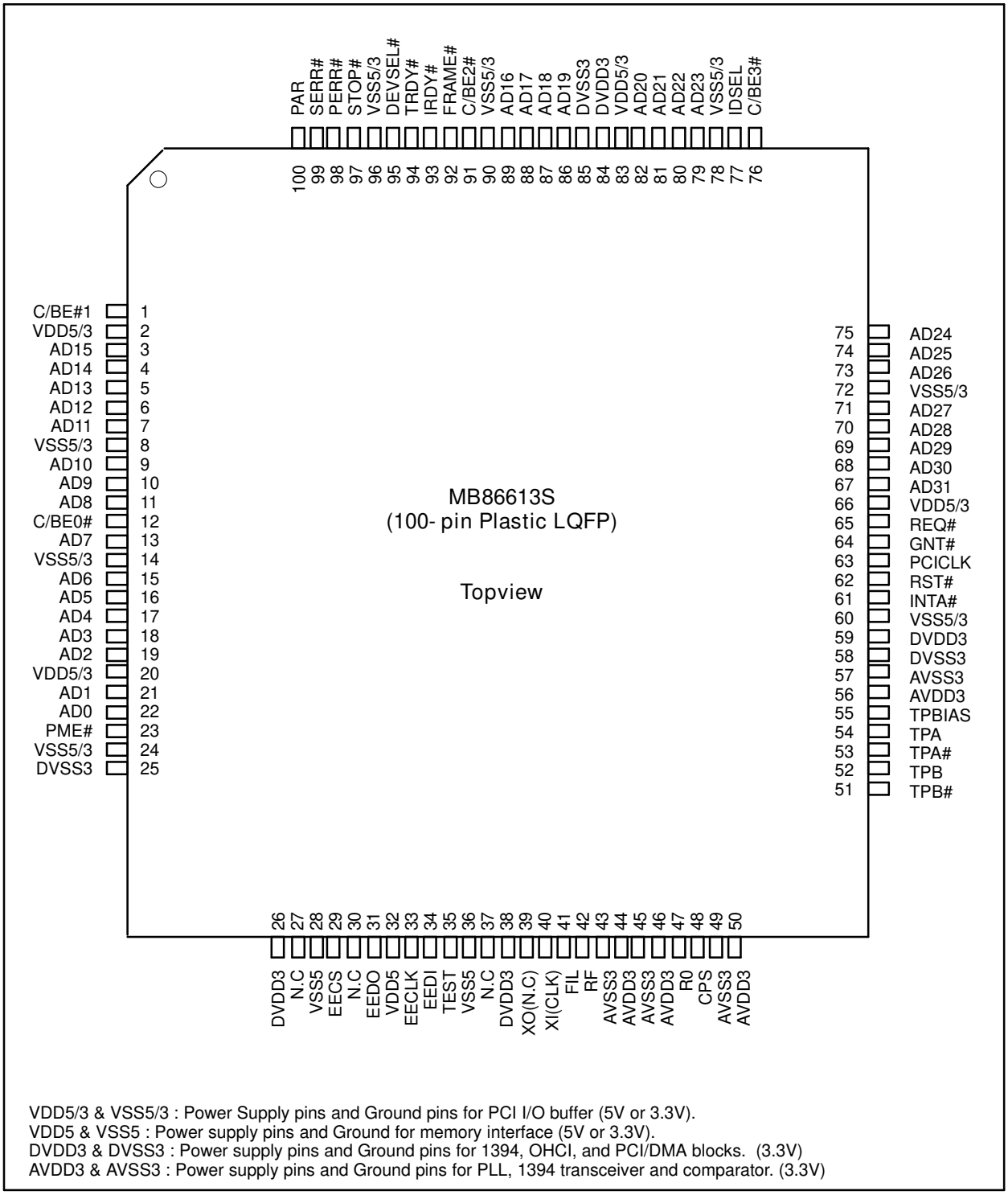
1.5.10. bus manager

This block manages the cycle timer, generates interrupts, and holds the bus management CSRs.

2. Pin Functions

2.2. Pin Assignment

Figure 2.1 shows the MB86613S pin assignment.



VDD5/3 & VSS5/3 : Power Supply pins and Ground pins for PCI I/O buffer (5V or 3.3V).
VDD5 & VSS5 : Power supply pins and Ground for memory interface (5V or 3.3V).
DVDD3 & DVSS3 : Power supply pins and Ground pins for 1394, OHCI, and PCI/DMA blocks. (3.3V)
AVDD3 & AVSS3 : Power supply pins and Ground pins for PLL, 1394 transceiver and comparator. (3.3V)

Fig. 2.1 Pin Assignment

2.3. Pin Function

2.3.1. PCI Bus Interface

Notes:

I/O denotes input/output pin.
O denotes output pin.
I denotes input pin.
OD denotes open- drain output pin.

Name of pin	I/O	Function
PCICLK	I	PCI bus clock input pin (Max. 33MHz)
RST#	I	System reset input pin.
AD31 : 0	I/O	32- bit PCI Address/Data multiplexed pins.
C/BE3# : 0#	I/O	PCI Bus Command / Byte Enable multiplexed pins.
PAR	I/O	Even Parity pin for AD31:0 and C/BE3#:0#. This pin state becomes valid after 1 PCICLK.
FRAME#	I/O	Frame signal pin that indicates the PCI bus is driven by the master.
IRDY#	I/O	Data Ready signal pin for bus master device.
TRDY#	I/O	Data Ready signal pin for target device.
STOP#	I/O	Stop signal pin for the data transfer from target to master.
IDSEL	I	Chip select pin to access the configuration register.
DEVSEL#	I/O	Device select pin. While the device is a target, this pin outputs the select signal that indicates the self device is selected. While the device is a master, this pin functions as an input pin to indicate that a device on the bus is selected.
REQ#	O	Request signal output pin to the bus arbiter to request for the PCI bus use.
GNT#	I	Grant signal input pin from the bus arbiter to receive the response to the REQ# signal.
PERR#	I/O	Data Parity Error input/output pin.
SERR#	OD	Address Parity Error output pin. (Open- drain type output pin.)
INTA#	OD	Interrupt output pin. (Open- drain type output pin.)
PME#	O	PCI power management enable

2.3.2. Memory Interface

Name of pin	I/O	Function
EEDI	O	Data output pin for EEPROM device.
EEDO	I	Data input pin for EEPROM device.
EECS	O	Chip Select pin to select the externally connected EEPROM device.
EECLK	O	Clock output pin for EEPROM device.

2.3.3. 1394 Interface

Name of pin	I/O	Function
TPA	I/O	TPA positive signals of 1394 cable port.
TPB	I/O	TPB positive signals of 1394 cable port.
TPA#	I/O	TPA negative signals of 1394 cable port.
TPB#	I/O	TPB negative signal of 1394 cable port.
TPBIAS	O	TP Bias voltage supply pin.
CPS	I	Cable Power input pin.
RO	O	Load resistance connection pin (Connect with GND via a 5.1kΩ. resistor)

2.3.4. Others

Name of pin	I/O	Function
X1(CLK), X0	I	Clock input pin for the on- chip PLL (24.576MHz) or Crystal oscillator pins
RF	O	Connect this pin with GND via a 5.1kΩ. resistor)
FIL	O	Filter circuit connection pin.
TEST	I	This pin is used for the test mode. Normally connect this pin with GND.
N.C.	-	Non- connection pins. Do not connect with these pins.

Table 2.1 Supported PCI Bus Command

C/BE	Command Type	Target	Master
0000	Interrupt Acknowledge	NO	NO
0001	Special Cycle	NO	NO
0010	I/O Read	YES	NO
0011	I/O Write	YES	NO
0100			
0101			
0110	Memory Read	YES	YES
0111	Memory Write	YES	YES
1000			
1001			
1010	Configuration Read	YES	NO
1011	Configuration Write	YES	NO
1100	Memory Read Multiple	-	YES
1101	Dual Address Cycle	NO	NO
1110	Memory Read Line	-	YES
1111	Memory Write & Invalidate	NO	YES

: denotes the reserved (unused) area.

2.4. Connection Example with External Components

2.4.1. 1394 Port

Figure 2.2 shows an example of connection diagram on 1394 port.

CPS pin inputs a cable power divided by a 510kΩ and a 91kΩ resistors. Insert an 1μF capacitor between TPBIAS pin and GND and also the TPBIAS pin is to be connected with each TPA pin through a 56Ω resistor. TPB and TPB# pins are connected with GND through 56Ω resistors also. (a CR circuit composed of a 250pF and 5.1kΩ is further required between the 56Ω resistor and GND.) RO pin is connected with GND through a 5.1kΩ resistor. TPA and TPB pins at the unused ports should be connected to GND, and TPBIAS pins to Open.

Those resistor and capacitor are reference values and does not guarantee stable operation on your application system.

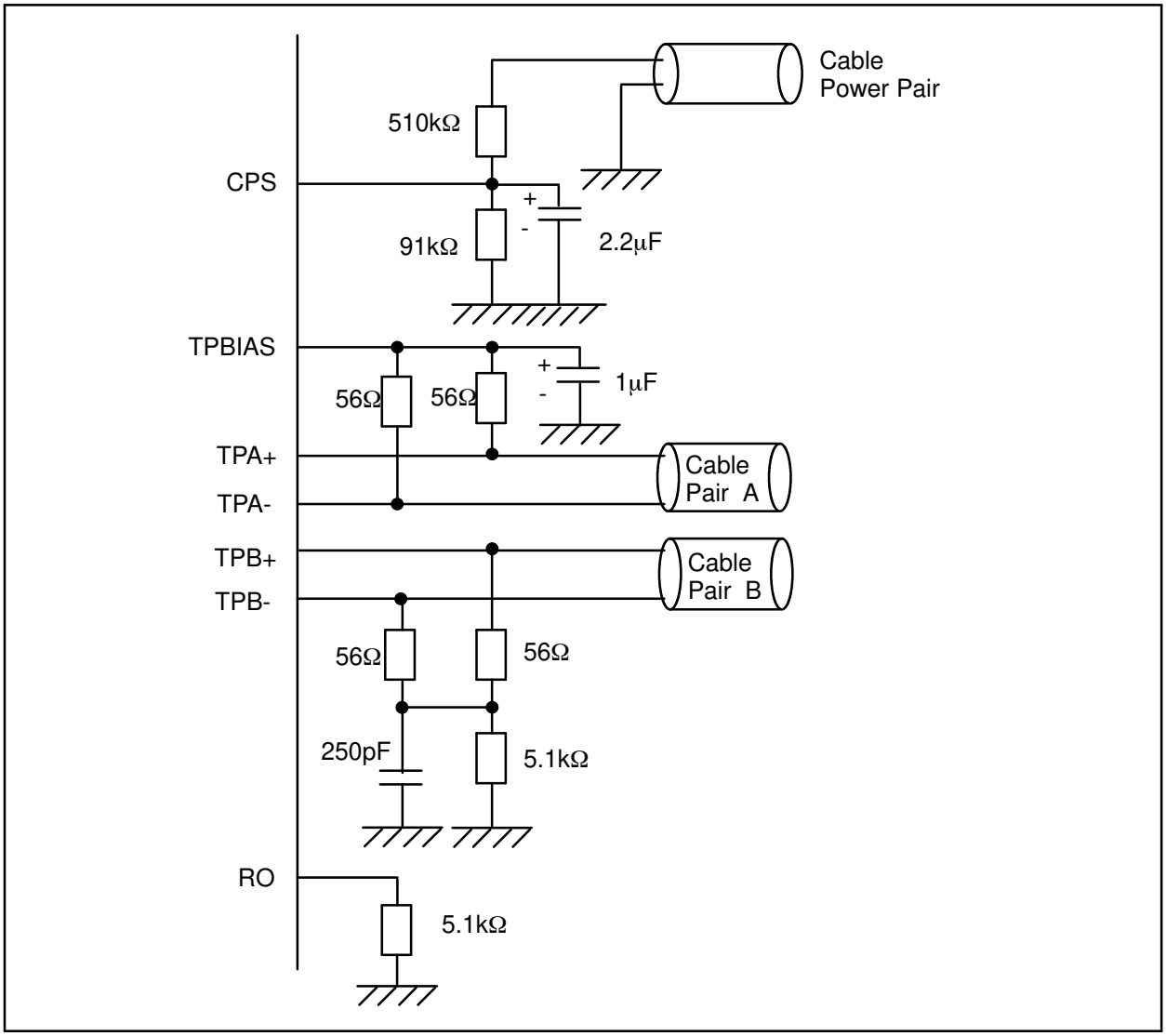


Fig. 2.2 1394 Cable Port Connection Example

2.4.2. Filter Circuit

Figure 2.3 and 2.4 shows an example of connection diagram on PLL filter circuit. A circuit where the 390Ω and a 3300pF are connected is required between the FIL pin and GND. RF pin is connected with GND through a 5.1KΩ resister. The CLK pin requires a 24.576MHz of clock module operating at +3.3V. Pin39 must be open when using a external clock module. When using the crystal oscillator, connect it and capacitors as Figure 2.4.

Those resistor and capacitor are reference values and does not guarantee stable operation on your application system.

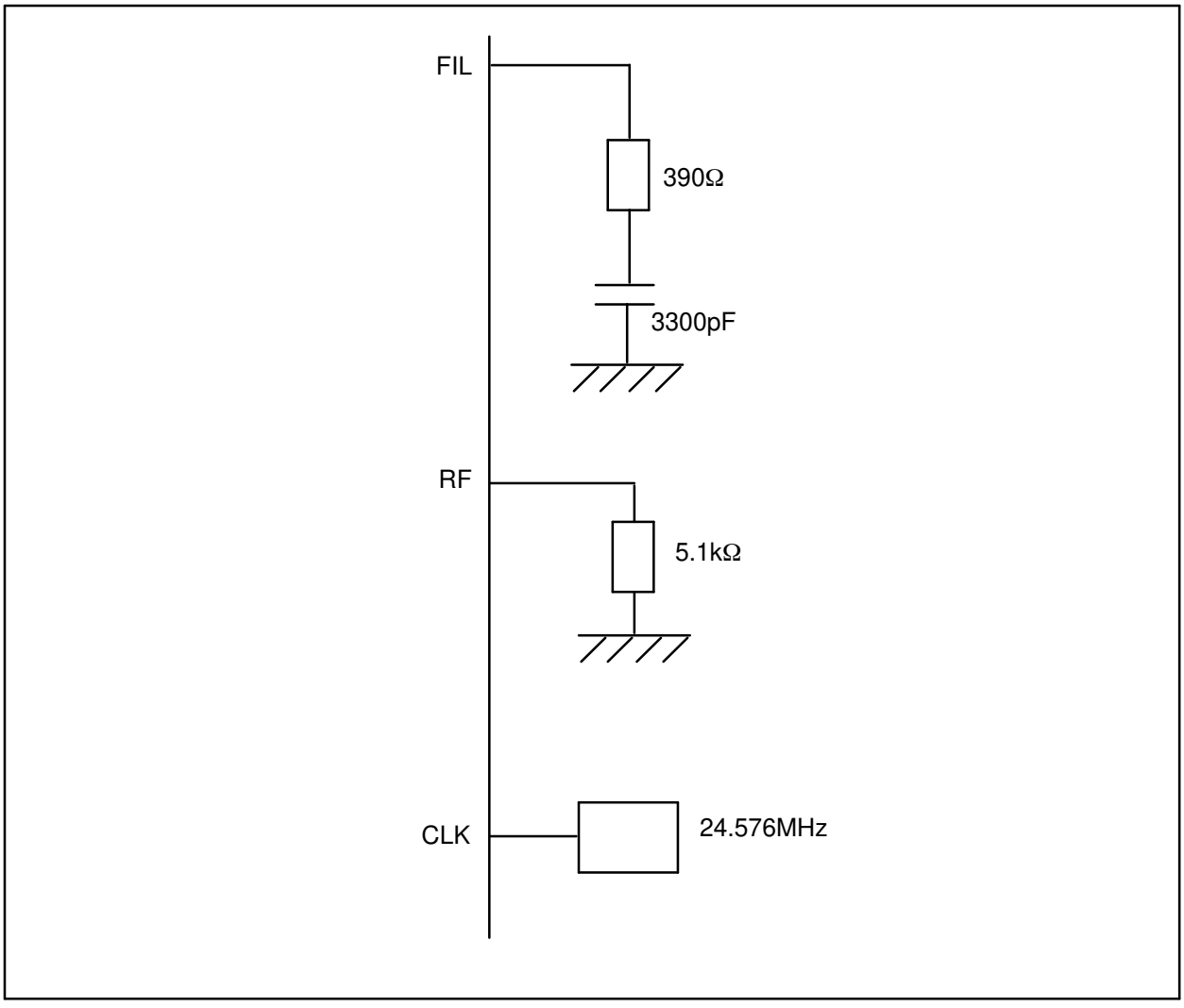


Fig. 2.3 PLL Filter Connection Example A

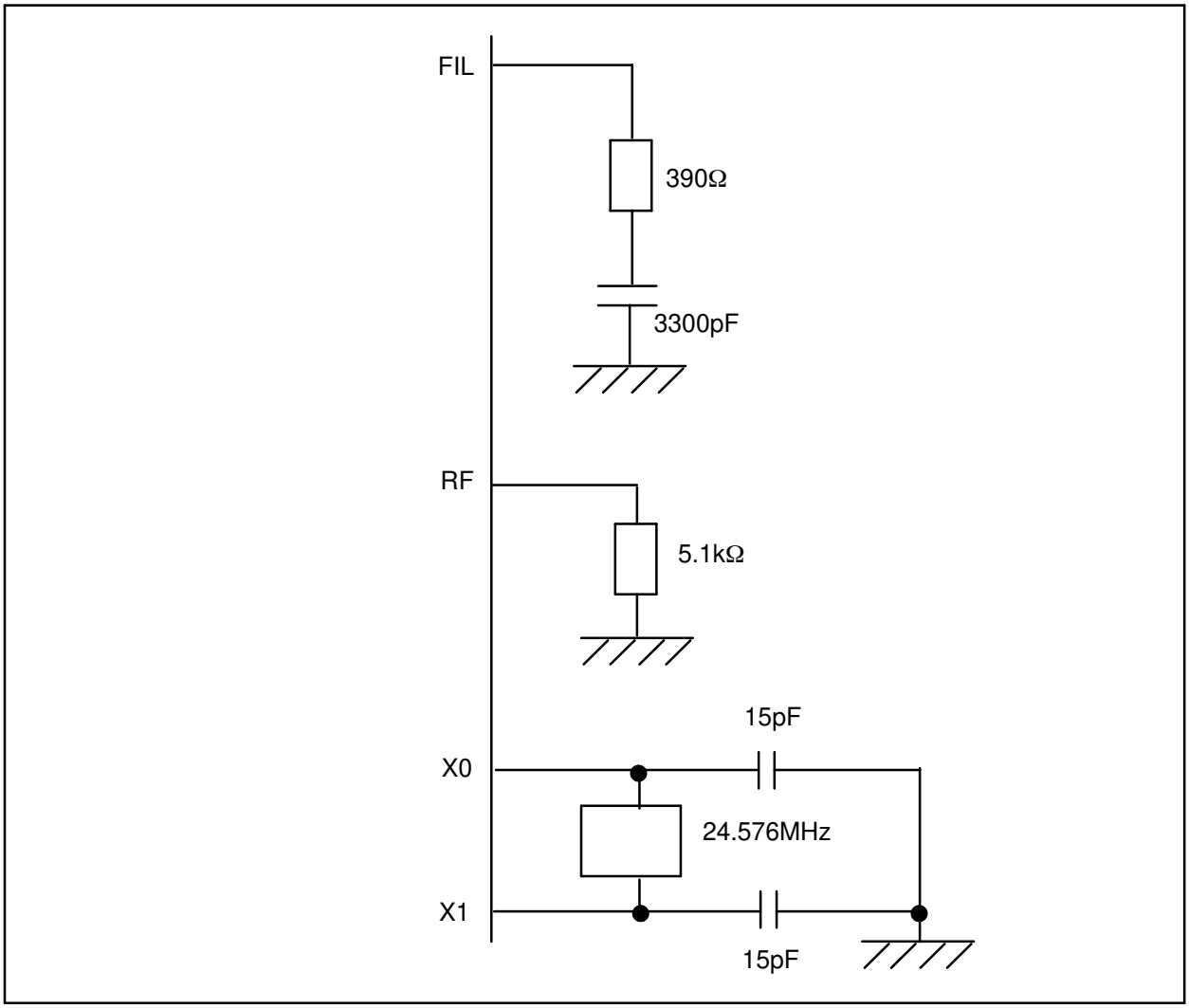


Fig. 2.4 PLL Filter Connection Example B

2.4.3. Memory Interface

Figure 2.5 shows connection diagram with external memory devices.

a) EEPROM:
..... An external EEPROM is a mandatory device. For the EEPROM address mapping, see section xxx.

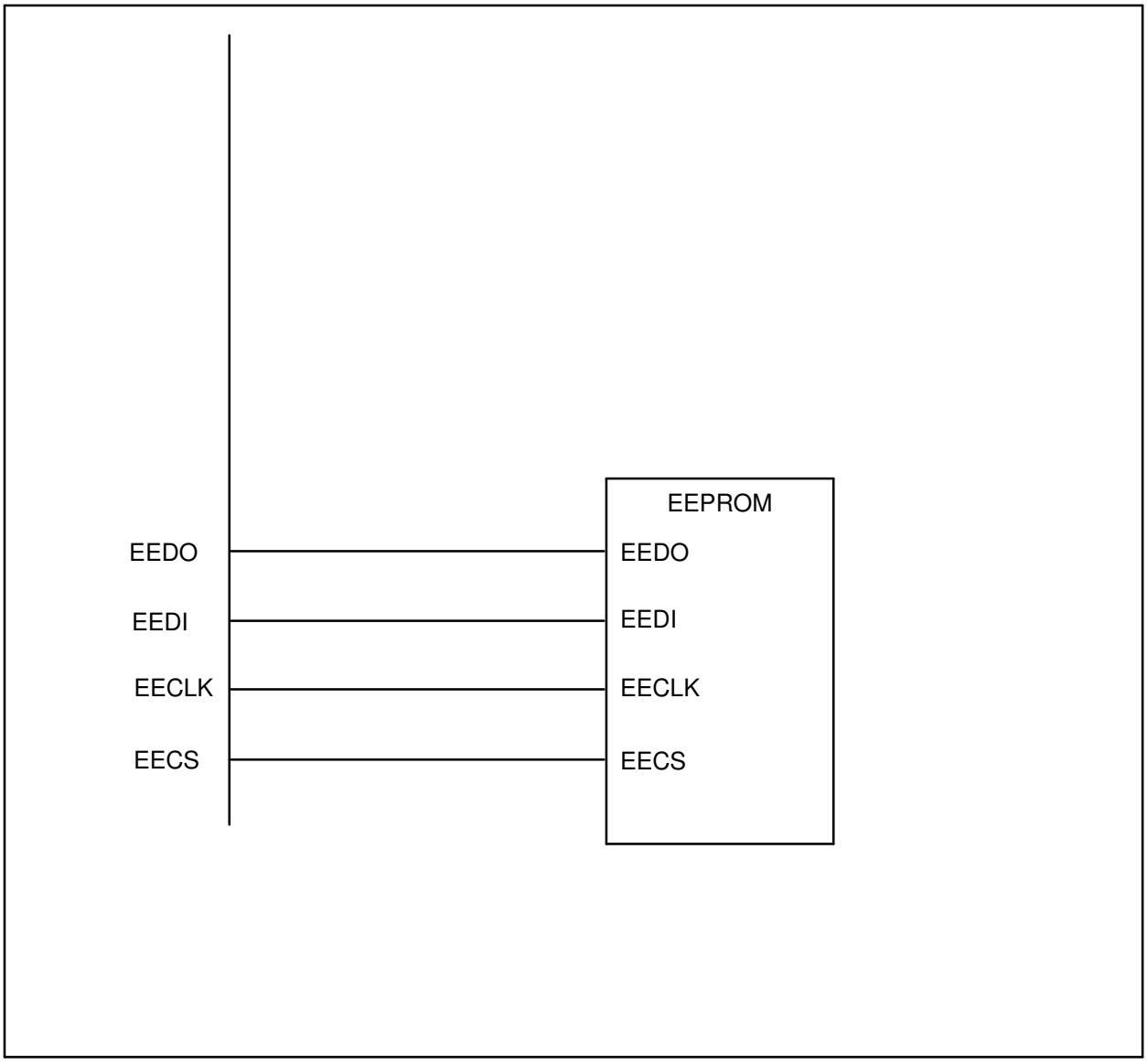


Fig. 2.5 Memory Interface Connection Diagram

2.5. System Configuration Example

Figure 2.6 shows an example of MB86613S system configuration. This example assumes a system where it provides the cable power and also receives the cable power to activate even while the device is powered off. The device power supply voltage (3.3V) should be provided from DC+12V PCI or the cable power regulated. 5VDC is connected with PCI ground through a 10KΩ register.

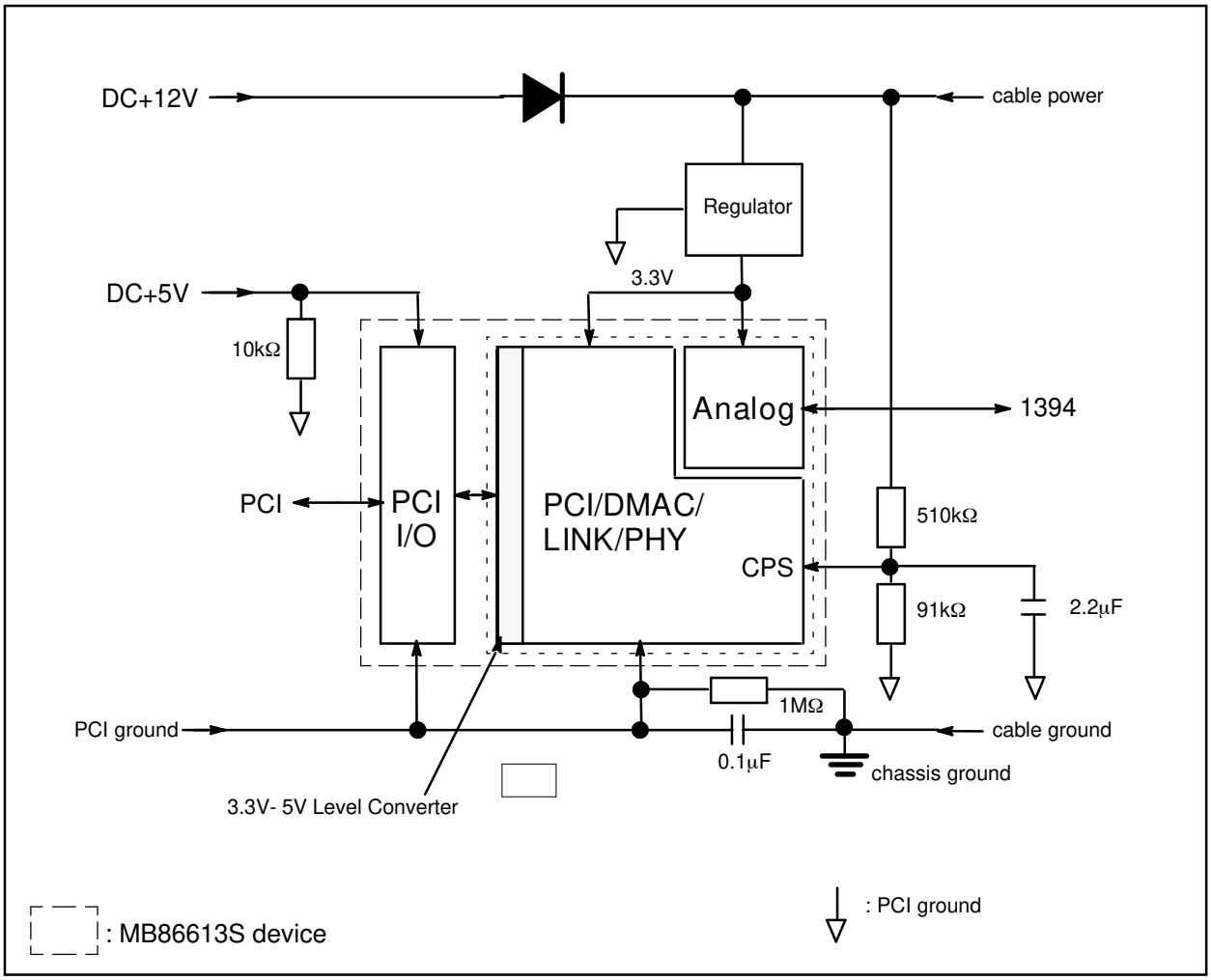
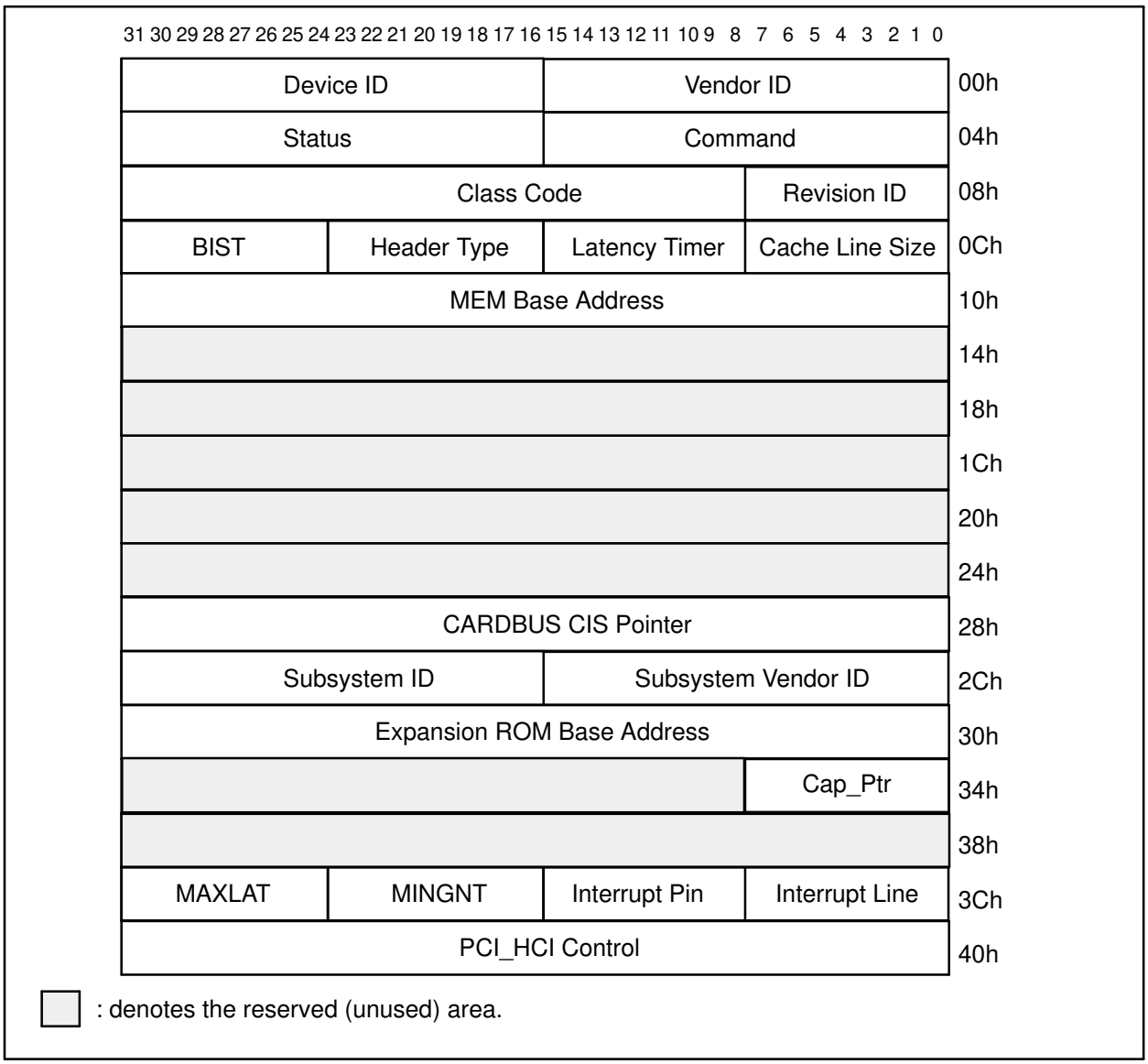


Fig. 2.6 System Configuration Example

3. PCI Configuration Register

This section describes the on- chip PCI configuration registers.

”r” denotes the register that can be read out.
”w” denotes the register that can be written.
”c” denotes the register whose value can be cleared to ”0”.
”u” denotes the register whose value is undefined depending on the MB86613S device status.



3.1. Register Description

3.1.1. Vendor ID

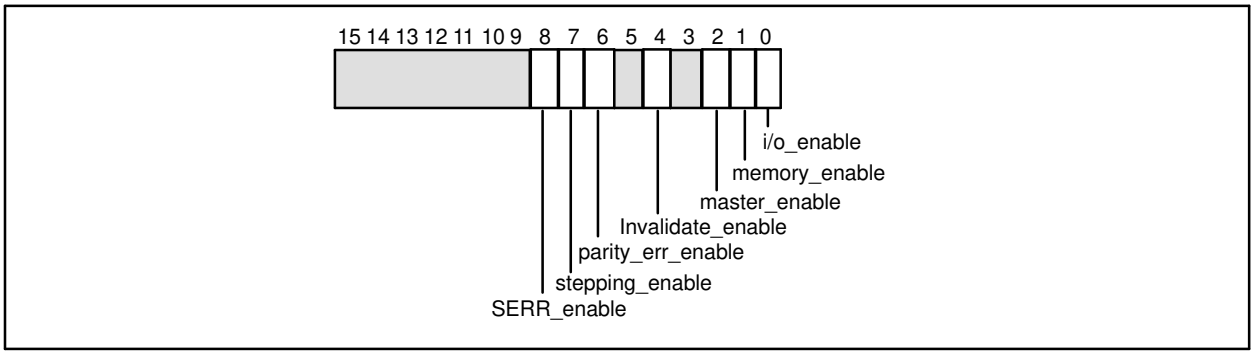
Bit	Field Name	rwcu	reset	description
15:0	Vendor ID	r	10CFh	Indicates the PCI vendor ID. (Vendor ID = ”10CF”h.)

3.1.2. Device ID

Bit	Field Name	rwcu	reset	description
31:16	Device ID	r	2010h	Device ID = ”2010”h.

3.1.3. Command

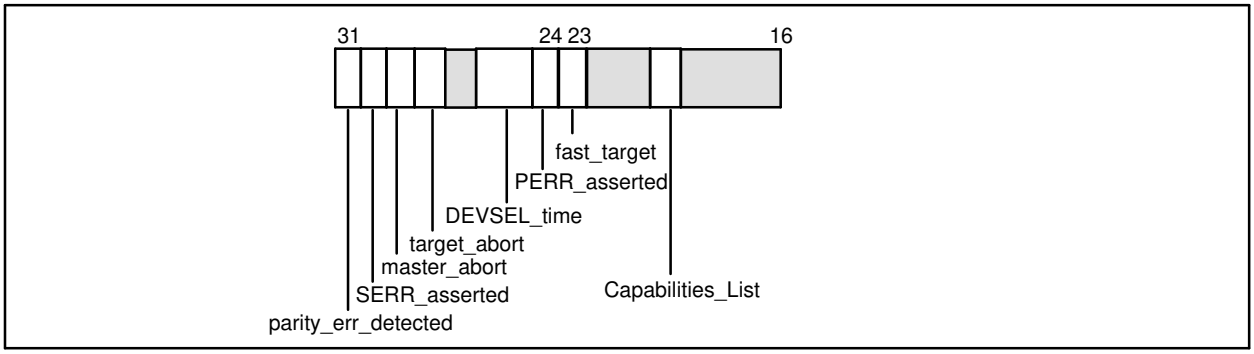
This register controls the MB86613S functions when generating and responding to the PCI cycle. Writing ”0000”h at this register separate the device from PCI interface except for the configuration access.



Bit	Field Name	rwcu	reset	description
8	SERR_enable	rw	0b	0 : Does not output SERR#. 1 : Outputs SERR# if a parity error is detected.
7	stepping_enable	r	0b	Since the device does not support the stepping function, this bit always indicates ”0”.
6	parity_err_enable	rw	0b	0 : Does not output PERR#. 1 : Outputs PERR# if a parity error is detected.
4	Invalidate_enable	rw	0b	0 : Uses memory write command. 1 : Uses memory write & invalidate command.
2	master_enable	rw	0b	0 : Prohibits the bus master operation. 1 : Allows the bus master operation.
1	memory_enable	rw	0b	0 : 0 : Prohibits the memory access. 1 : Allows the memory access.
0	i/o_enable	rw	0b	0 : Prohibits the I/O access. 1 : Allows the I/O access.

3.1.4. Status

This register holds event status on PCI interface. When any of events occurred, please clear the bit corresponding to the event by writing ”1” at that bit.



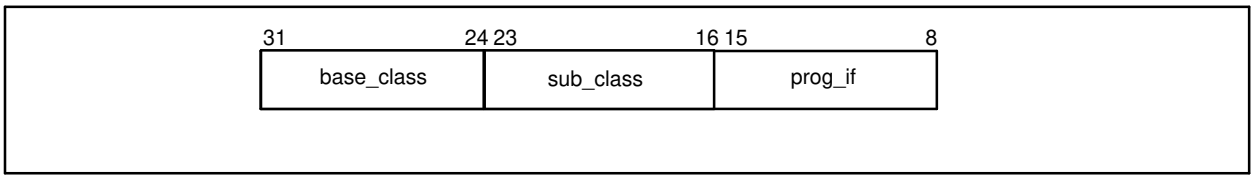
Bit	Field Name	rwcu	reset	description
31	parity_err_detected	rcu	0b	This bit indicates ”1” when a parity error occurs.
30	SERR_asserted	rcu	0b	This bit indicates ”1” when a system error occurs.
29	master_abort	rcu	0b	This bit indicates ”1” when detecting the master- abort and the transfer stops.
28	target_abort	rcu	0b	This bit indicates ”1” when detecting the target- abort and the transfer is stopped.
26:25	DEVSEL_time	r	01b	Read out value from this bit is ”01h”. The contents of the bits reacts within 2 PCICLK after FRAME# asserted.
24	PERR_asserted	rcu	0b	This bit indicates ”1” when a parity error is detected, that is, when ’Command.parity_enable’ bit is set and PERR# signal is output or PERR# is detected.
23	fast_target	r	1b	This bit indicates ”1” because the device supports the high- speed back- to- back transfer.
20	Capabilities_List	r	1b	This bit indicates ”1” because the device supports the PCI bus power management.

3.1.5. Revision ID

Bit	Field Name	rwcu	reset	description
7:0	revision	r	00h	This bit indicates the device revision ID. It is ”00h”.

3.1.6. Class Code

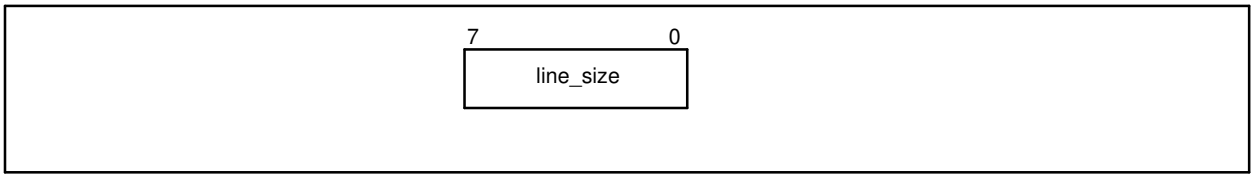
This register is used for identifying the MB86613S functions and programming interface. Each class code is read only field that displays the functional and programming interface information.



Bit	Field Name	rwcu	reset	description
31:24	base_class	r	0Ch	These bits indicate ”0Ch” for the serial bus controller.
23:16	sub_class	r	00h	These bits indicate ”00h” for the IEEE1394 compliant.
15:8	prog_if	r	10h	These bits indicate ”10h” for the Open HCI.

3.1.7. Cache Line Size

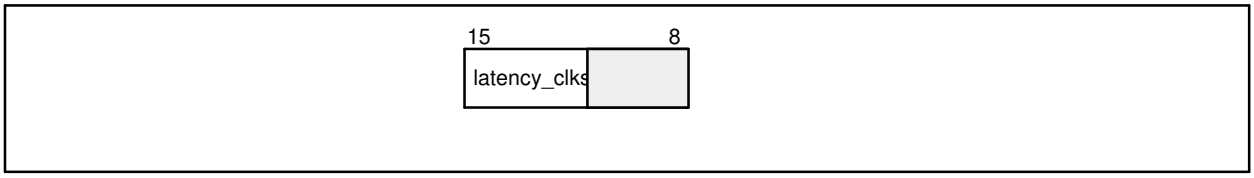
This register specifies the cache line size in 32- bit long- word that is guaranteed for the memory write and invalidate command.



Bit	Field Name	rwcu	reset	description
7:0	line_size	rw	00h	These bits specify the cache line size with the following setting: Value Size 00h Unused for memory write & invalidate command 01h - FFh 1- to 255- long word.

3.1.8. Latency Timer

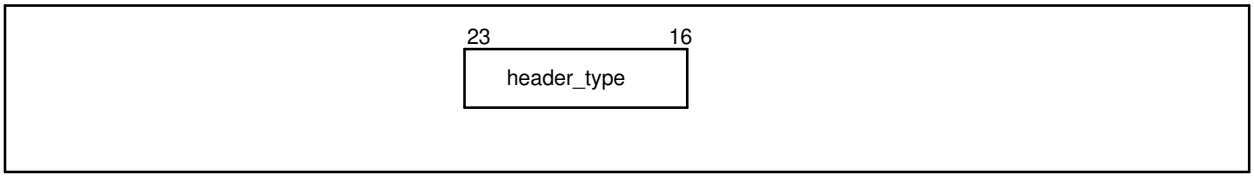
This register specifies the PCI latency timer value. The latency timer counts the time from the FRAME# asserted until the PCI bus occupied.



Bit	Field Name	rwcu	reset	description
15:11	latency_clks	rw	00h	These bits specify the latency timer. The unit is 8 PCI clocks.

3.1.9. Header Type

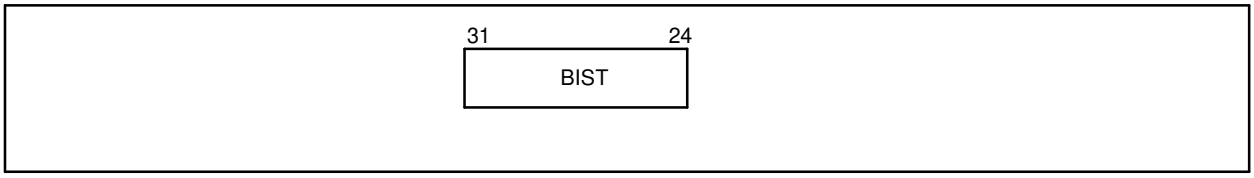
This register indicates the register configuration at addresses 10h to 3Fh in configuration space and the supported function(s). MB86613S supports the single- function so this register indicates ”00h”.



Bit	Field Name	rwcu	reset	description
23:16	header_type	r	00h	These bits indicate the header type. ”00h” is read out. (Single- function)

3.1.10. BIST

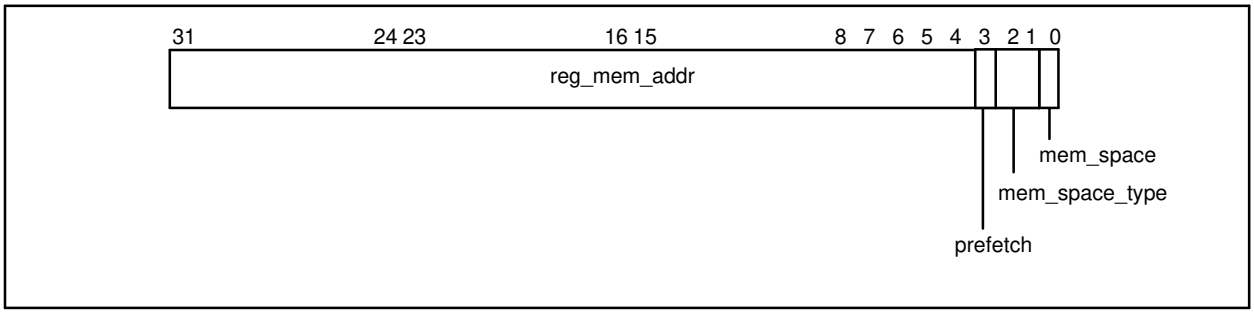
This is a BIST control register. But the MB86613S does not support the BIST and so this register indicates ”00h”.



Bit	Field Name	rwcu	reset	description
31:24	BIST	r	00h	”00h” is read out. (No BIST supported)

3.1.11. MEM Base Address

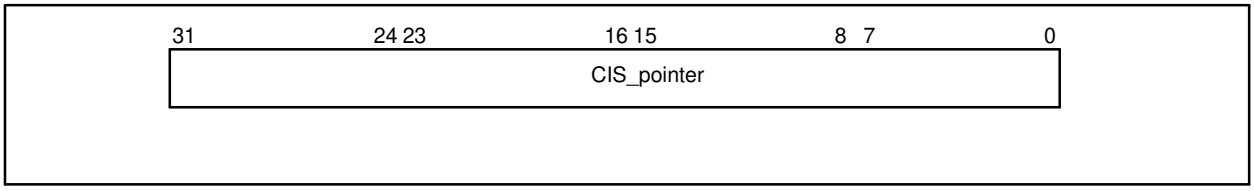
This register specifies the base address for the Open HCI and debug registers’ space used for memory command. When the debug register is enable the base address is defined in 4096- byte boundary. The first 4096- byte space is used for the Open HCI register and the second 2048- byte for the debug register.



Bit	Field Name	rwcu	reset	description
31:10	reg_mem_addr	rw	00_0000h	These bits set the base address for the OHCI and debug registers. (when debug- reg enabled: 4096- byte boundary. when debug- reg disabled: 2048- byte boundary)
11	reg_mem_addr	rw	0b	This bit is available only when the debug register is disable. 0 is indicated for the debug register enabled.
10:4	reg_mem_addr	r	00h	”00h” is indicated. (Read only field)
3	prefetch	r	0b	This bit indicates ”0”.
2:1	mem_space_type	r	00b	These bits indicate ”00”. (i,e, 32- bit address width)
0	mem_space	r	0b	This bit indicates ”0”. (i.e., use the memory address.)

3.1.12. CARDBUS CIS Pointer

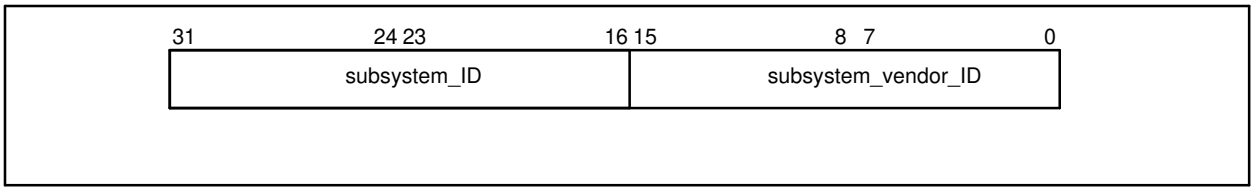
This register indicates the head address for the EPROM (BIOS ROM) that contains the card bus information. When the device is released from a hardware reset, it automatically loads the required data from the externally connected EEPROM.



Bit	Field Name	rwcu	reset	description
31:0	CIS_pointer	r	undefined	These bits indicate the CIS pointer.

3.1.13. Subsystem

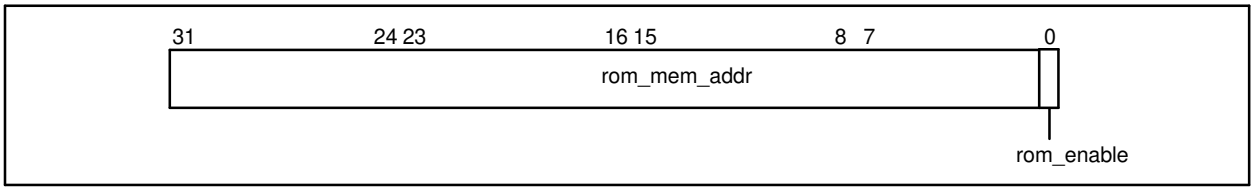
This register indicates the subsystem and subsystem- vendor ID (product ID) to identify the MB86613S device when it is connected together with another device on the same add- on board. The data are automatically loaded from the external EEPROM when the chip is released from the hardware reset.



Bit	Field Name	rwcu	reset	description
31:16	subsystem_ID	r	undefined	These bits indicate the subsystem ID.
15:0	subsystem_vender_ID	r	undefined	These bits indicate the subsystem vendor ID.

3.1.14. Expansion ROM Base Address

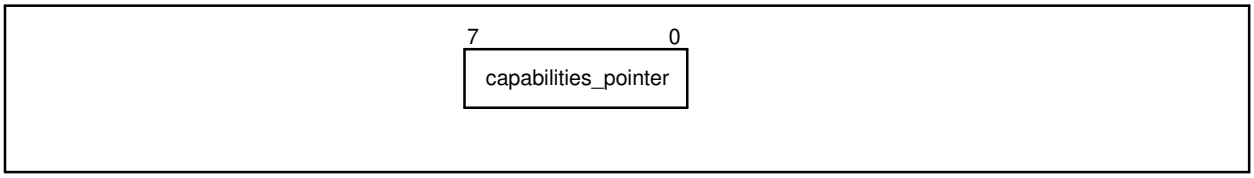
This register specifies the base address for the EPROM (BIOS ROM) device externally connected with the MB86613S device. The data are automatically loaded by the rom_enable bit from the external EEPROM when the chip is released from the hardware reset.



Bit	Field Name	rwcu	reset	description
31:16	rom_mem_addr	rw	0000h	These bits specify the base address for the BIOS ROM. (64K- byte boundary)
15:1	rom_mem_addr	r	0000h	These bits indicate ”0000h”.
0	rom_enable	rw	0b	The specified base address is enabled by writing 1 at this bit if the Command.memory_enable bit is also set.

3.1.15. CAP_PTR

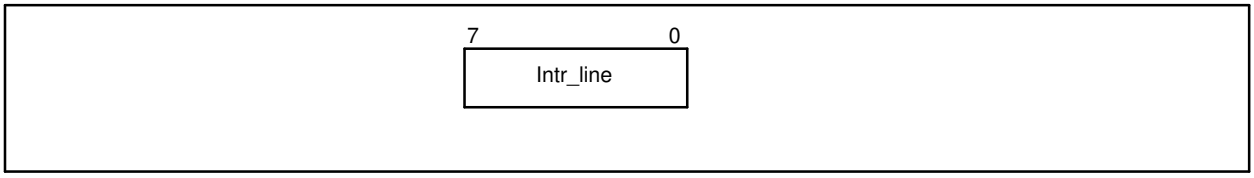
This register indicates the address for ”Capabilities Linked List” that exists in configuration space.



Bit	Field Name	rwcu	reset	description
7:0	capabilities_pointer	r	undefined	These bits indicate the CAP pointer.

3.1.16. Interrupt Line

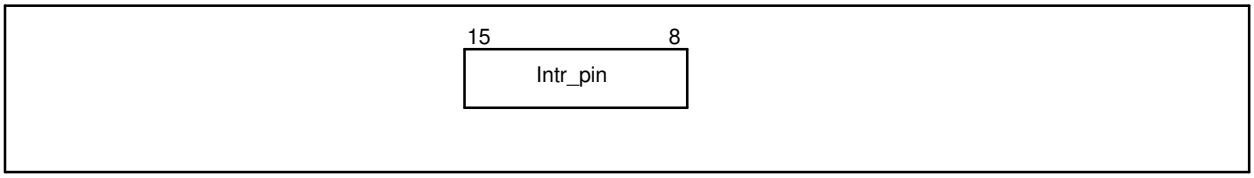
This register indicates the interrupt line status.



Bit	Field Name	rwcu	reset	description
7:0	Intr_line	rw	00h	These bits specify the connectivity between the device interrupt signal and host system interrupt signal with the following setting: Value — Connection 00h- 0Fh — connects with any of IRQ0- 15. 10h- FEh — reserve FFh — unknown or no connection

3.1.17. Interrupt Pin

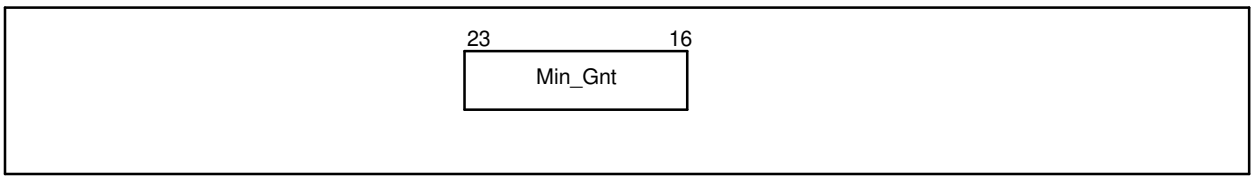
This register indicates the interrupt pin usage.



Bit	Field Name	rwcu	reset	description
15:8	Intr_pin	r	01h	These bits indicate ”01h” (use the INTA#.)

3.1.18. MINGNT

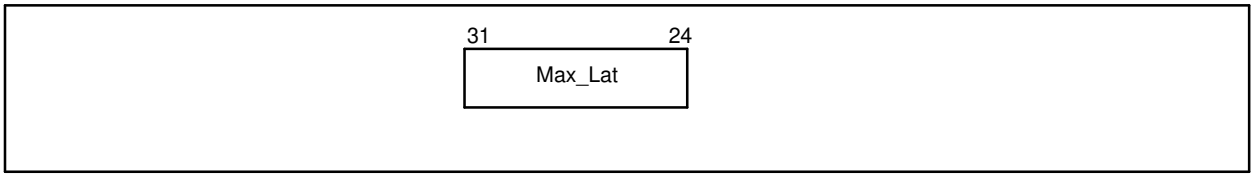
This read- only register indicates burst time required for the device. The time is 1/4 micro seconds unit.



Bit	Field Name	rwcu	reset	description
23:16	Min_Gnt	r	20h	These bits indicate ”20h” (8μs).

3.1.19. MAXLAT

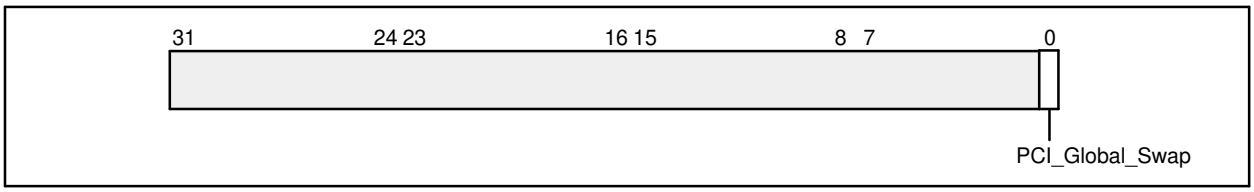
This read- only register indicates PCI bus access time required for the device. The time is 1/4 micro seconds unit.



Bit	Field Name	rwcu	reset	description
31:24	Max_Lat	r	50h	These bits indicate ”50h” (20μs).

3.1.20. PCI_HCI Control

This register specifies the byte- swap control defined in the Open HCI specification.



Bit	Field Name	rwcu	reset	description
0	PCI_Global_Swap	rw	0b	Writing 1 at this bit performs the byte- swap for the data accessed to/from PCI interface.

3.2. Open HCI Register

The addresses for the following listed Open HCI register set must be specified with the MEM Base Address register in the PCI configuration register.

r denotes the register can be read.
w denotes the register can be written.
s denotes the bit can be written (1b)
c denotes the bit can be cleared (0b)
u denotes the read value undefined depending on the MB86613S device status.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Version	000h
GUID_ROM	004h
ATRetries	008h
CSRData	00Ch
CSRCompare	010h
CSRControl	014h
ConfigROMhdr	018h
BusID	01Ch
BusOption	020h
GUIDHi	024h
GUIDLo	028h
	02Ch to 030h
ConfigROMMap	034h
PostedWriteAddressLo	038h
PostedWriteAddressHi	03Ch
Vendor ID	040h
	044h to 04Ch
HCControlSet	050h
HCControlClear	054h
	058h to 060h
SelfIDBuffer [Self ID]	064h
SelfIDCount [Self ID]	068h
	06Ch

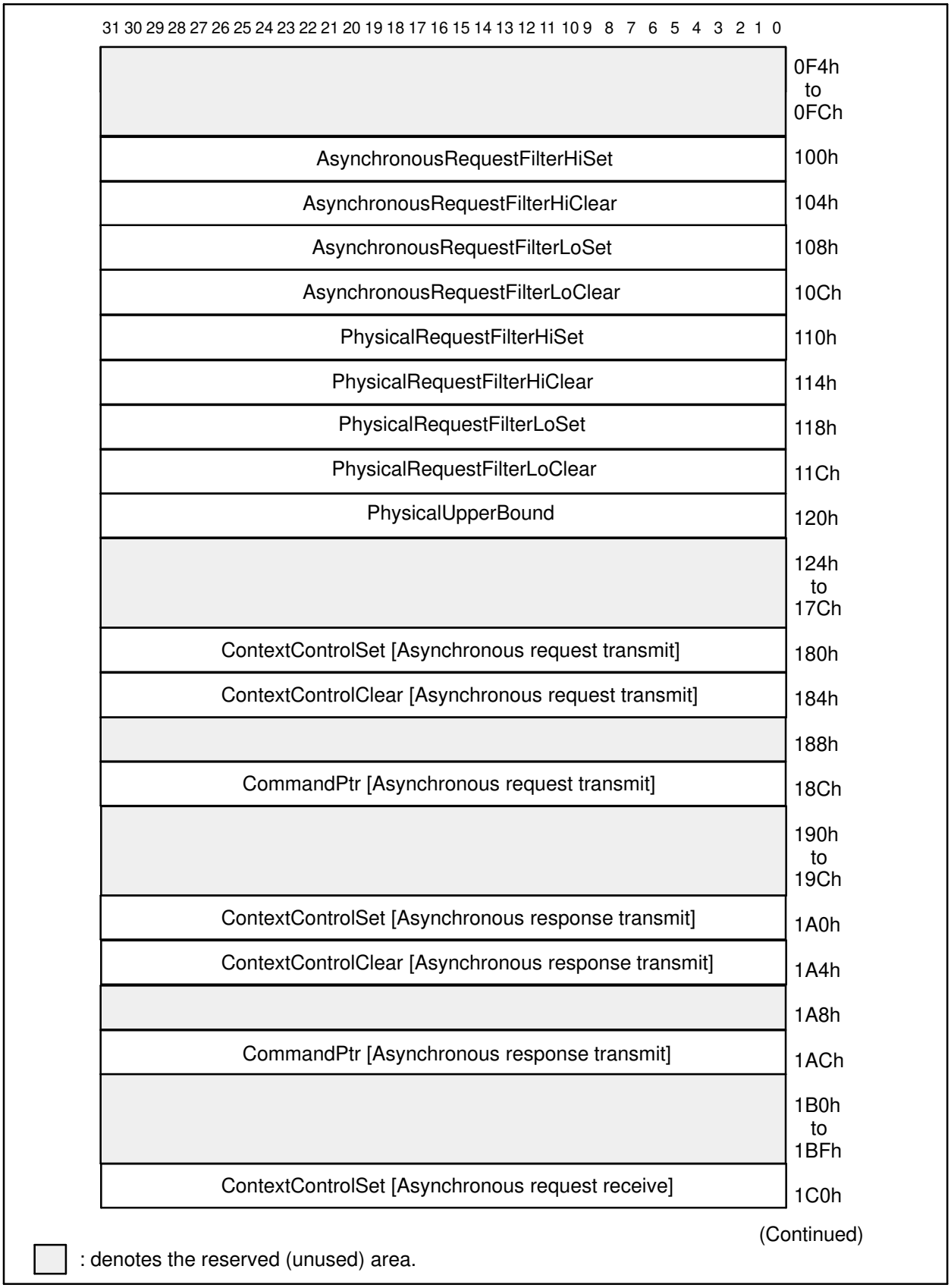
(Continued)

: denotes the reserved (unused) area.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
IRMultiChanMaskHiSet	070h
IRMultiChanMaskHiClear	074h
IRMultiChanMaskLoSet	078h
IRMultiChanMaskLoClear	07Ch
IntEventSet	080h
IntEventClear	084h
IntMaskSet	088h
IntMaskClear	08Ch
IsoXmitIntEventSet	090h
IsoXmitIntEventClear	094h
IsoXmitIntMaskSet	098h
IsoXmitIntMaskClear	09Ch
IsoRecvIntEventSet	0A0h
IsoRecvIntEventClear	0A4h
IsoRecvIntMaskSet	0A8h
IsoRecvIntMaskClear	0ACh
InitialBandwidthAvailable	0B0h
InitialChannelsAvailableHi	0B4h
InitialChannelsAvailableLo	0B8h
	0BCh to 0D8h
FairnessControl	0DCh
LinkControlSet	0E0h
LinkControlClear	0E4h
NodeID	0E8h
PhyControl	0ECh
IsochronousCycleTimer	0F0h

(Continued)

: denotes the reserved (unused) area.



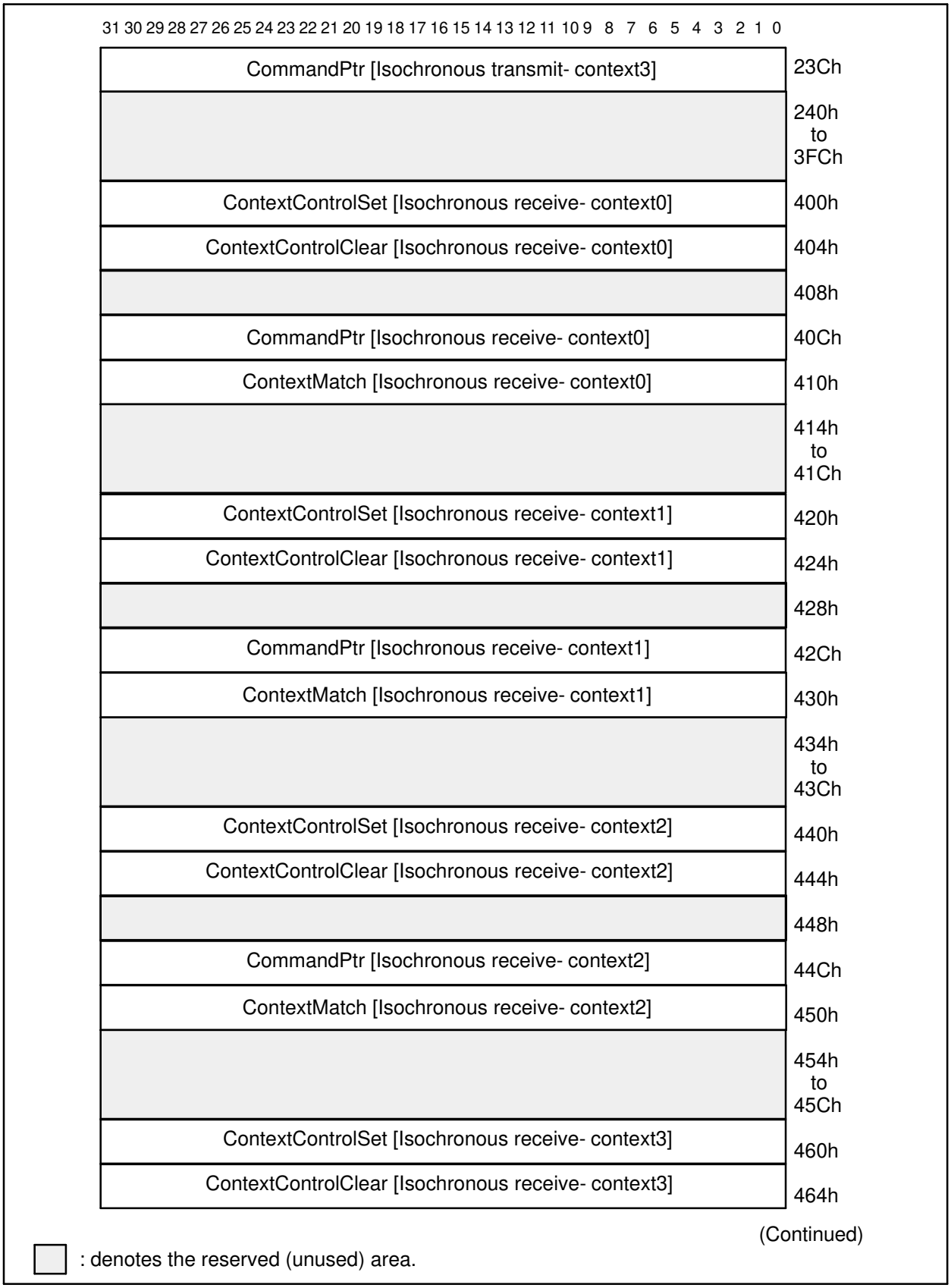
(Continued)

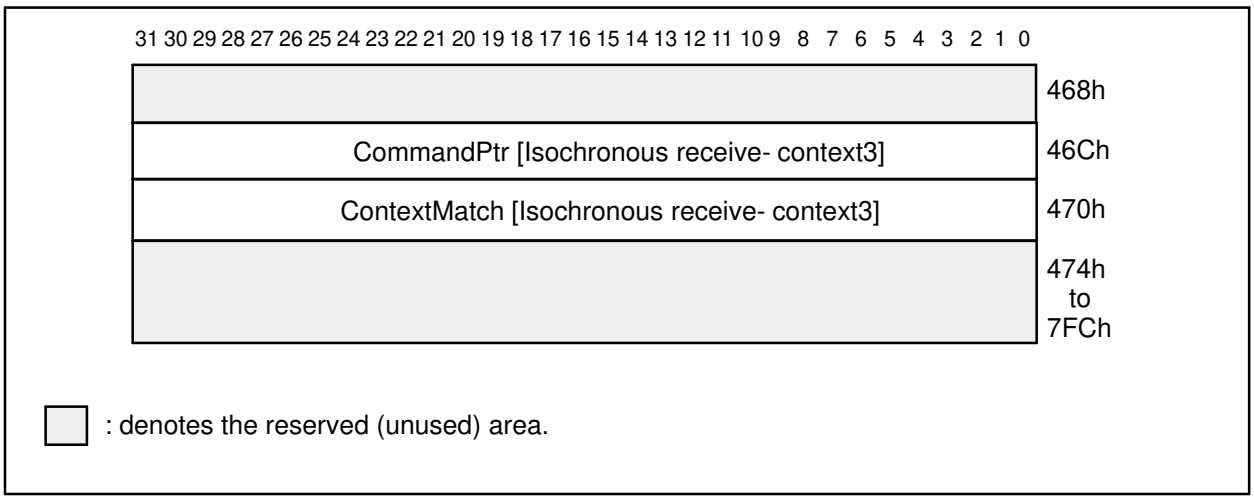
: denotes the reserved (unused) area.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
ContextControlClear [Asynchronous request receive]	1C4h
	1C8h
CommandPtr [Asynchronous request receive]	1CCh
	1D0h to 1DFh
ContextControlSet [Asynchronous response receive]	1E0h
ContextControlClear [Asynchronous response receive]	1E4h
	1E8h
CommandPtr [Asynchronous response receive]	1ECh
	1F0h to 1FCh
ContextControlSet [Isochronous transmit- context0]	200h
ContextControlClear [Isochronous transmit- context0]	204h
	208h
CommandPtr [Isochronous transmit- context0]	20Ch
ContextControlSet [Isochronous transmit- context1]	210h
ContextControlClear [Isochronous transmit- context1]	214h
	218h
CommandPtr [Isochronous transmit- context1]	21Ch
ContextControlSet [Isochronous transmit- context2]	220h
ContextControlClear [Isochronous transmit- context2]	224h
	228h
CommandPtr [Isochronous transmit- context2]	22Ch
ContextControlSet [Isochronous transmit- context3]	230h
ContextControlClear [Isochronous transmit- context3]	234h
	238h

(Continued)

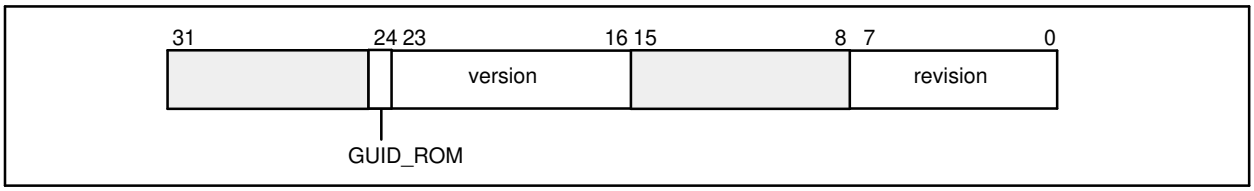
: denotes the reserved (unused) area.


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
CommandPtr [Isochronous transmit- context3]
23Ch
240h to 3FCh
ContextControlSet [Isochronous receive- context0]
400h
ContextControlClear [Isochronous receive- context0]
404h
408h
CommandPtr [Isochronous receive- context0]
40Ch
ContextMatch [Isochronous receive- context0]
410h
414h to 41Ch
ContextControlSet [Isochronous receive- context1]
420h
ContextControlClear [Isochronous receive- context1]
424h
428h
CommandPtr [Isochronous receive- context1]
42Ch
ContextMatch [Isochronous receive- context1]
430h
434h to 43Ch
ContextControlSet [Isochronous receive- context2]
440h
ContextControlClear [Isochronous receive- context2]
444h
448h
CommandPtr [Isochronous receive- context2]
44Ch
ContextMatch [Isochronous receive- context2]
450h
454h to 45Ch
ContextControlSet [Isochronous receive- context3]
460h
ContextControlClear [Isochronous receive- context3]
464h
(Continued)
: denotes the reserved (unused) area.



3.2.1. Version

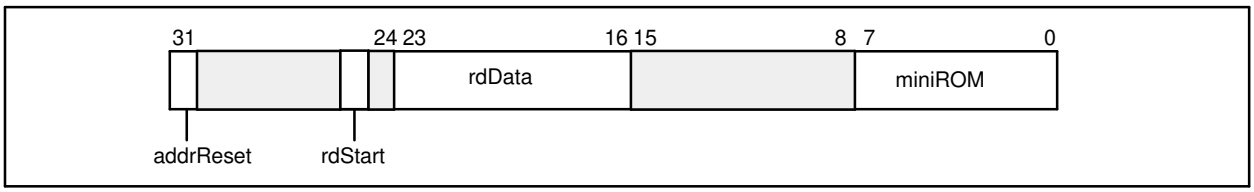
This register indicates the revised edition number of the Open HCI Specification. “01h” is indicated in the version field and “10h” in the revision field. Normally “1b” is indicated in the GUIDE_ROM bit because the data in the GUIDE ROM are loaded automatically with the Global Unique ID register.



Bit	Field Name	rwu	reset	description
24	GUID_ROM	r	1b	”1b” is indicated.
23:16	version	r	01h	These bits indicate ”01h” which means the 1394A draft 1.xx compliant.
7:0	revision	r	10h	These bits indicate ”00h” which means the 1394A draft x.10 compliant.

3.2.2. GUID_ROM

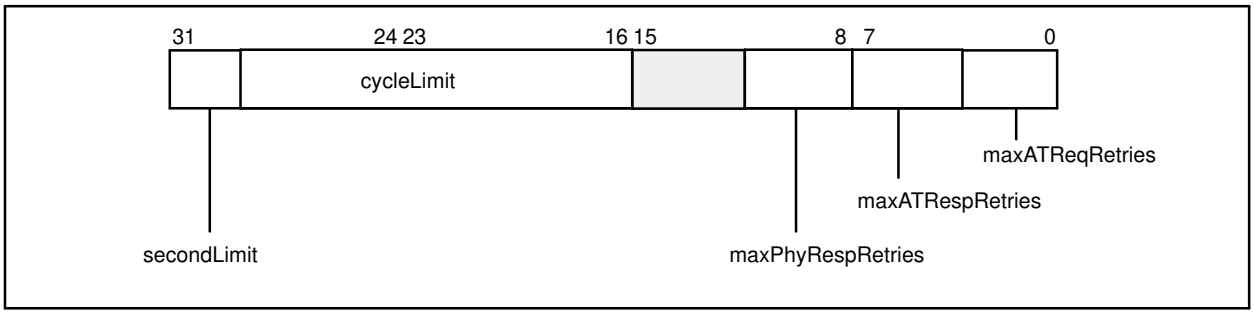
This is a port type register to read out the data that the GUID ROM contains. This register is not supported.



Bit	Field Name	rscu	reset	description
31	addrReset	r	0b	“0b” is indicated.
25	rdStart	r	0b	“0b” is indicated.
23:16	rdData	r	00h	“00h” is indicated.
7:0	miniROM	r	00h	“00h” is indicated

3.2.3. ATRetries

This register sets the retry count when transmitting the asynchronous packet or receiving the “ack_busy_#” or “ack_data_error” acknowledge. The register performs the retries of the count set here when the received acknowledge is “ack_busy_#” or “ack_data_error”. “0” are indicated on the secondLimit and the cycleLimit fields respectively because the MB86613S does not support the dual phase relay.



Bit	Field Name	rwu	reset	description
31:29	secondLimit	r	0h	The device supports only single phase retry function. So, these bits indicate ”0”.
28:16	cycleLimit	r	000h	
11:8	maxPhyResp Retries	rw	undefined	Set the retry count for the transmit phy response packet.
7:4	maxATResp Retries	rw	undefined	Set the retry count for the transmit asynchronous response packet.
3:0	maxATReq Retries	rw	undefined	Set the retry count for the transmit asynchronous request packet.

3.2.4. Bus Management CSR

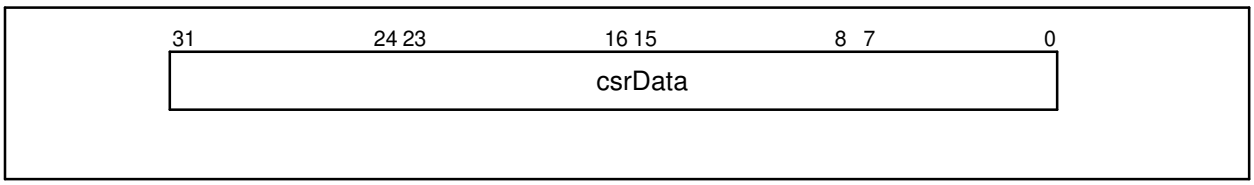
This register is used for making the device operate the automatic compare- swap for the on- chip bus management CSR while the MB86613S device is operating as the resource manager. This register is used as follows:

Store the data that you want to set in the bus management CSR in the csrData. Then, store the data to be compared with the bus management CSR in the csrCompare. After that, select any of register for the bus management CSR by csrSel bit. At that time, the csrDone bit is cleared.

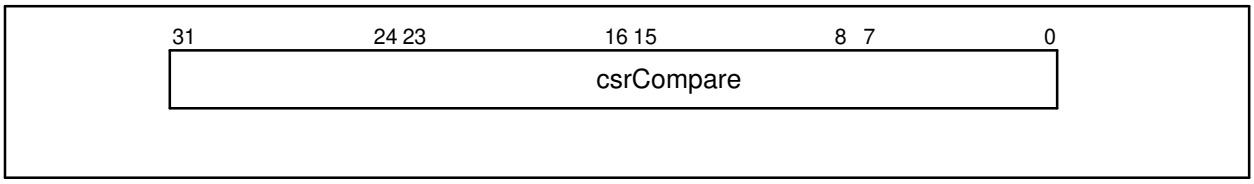
With these steps, if the compare- swap is completed, ”1” is indicated in the csrDone bit. At that time, the csrData indicate the data previously stored in the bus management CSR which has been selected by the csrSel bit. This means, if the compare- swap is succeeded, the same data are read out from the csrData and csrCompare. In case the different data are read out from these bits, please set the data read out from the csrData in the csrCompare and repeat the same steps as above.

Bus reset loads ’3Fh’ to bus_manager_ID register and the following registers value to other three Bus management CSR registers.

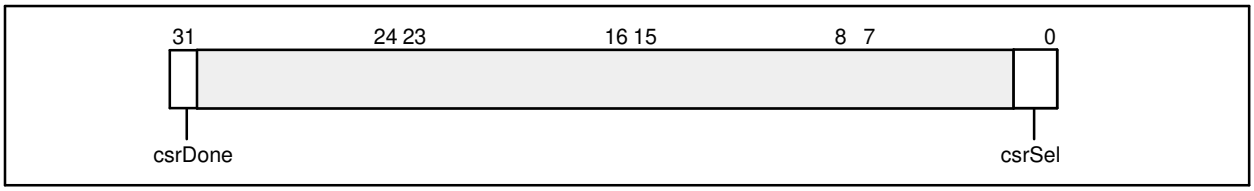
CSR Address	csrSel	by Bus reset	description
”FFFF_F000_021Ch”	’00b’	3Fh	bus_manager_ID
”FFFF_F000_0220h”	’01b’	InitialBandwidthAvailable	bandwidth_available
”FFFF_F000_0224h”	’10b’	InitialChannelsAvailableHi	channels_available_hi
”FFFF_F000_0228h”	’11b’	InitialChannelsAbailableLo	channels_available_lo



Bit	Field Name	rwu	reset	description
31:0	csrData	rwu	undefined	Store the data to be set in the bus management CSR.



Bit	Field Name	rwu	reset	description
31:0	csrCompare	rw	undefined	Store the data to be compared with the bus management CSR.

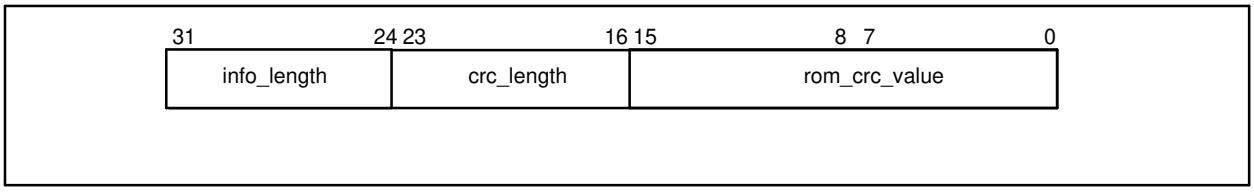


Bit	Field Name	rwu	reset	description
31	csrDone	ru	’1b’	This bit indicates ”1” when the compare is done.
1:0	csrSel	rw	undefined	These bits select the bus management CSR with the following setting:

Value	Bus Management CSR
’00b’	bus_manager_ID
’01b’	bandwidth_available
’10b’	channels_available_hi
’11b’	channels_available_lo

3.2.5. Config ROM Header

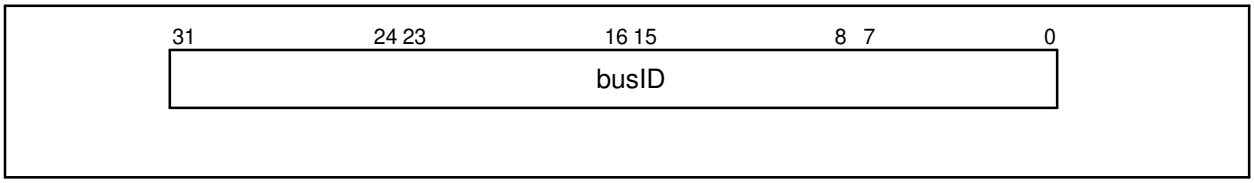
This register stores and indicates the first data of the 1934 configuration ROM (off- set “FFFF_F000_0400h”). For the software reset state (HCControl.softReset), the current stored data are indicated.



Bit	Field Name	rwu	reset	description
31:24	info_length	rw	00h	Set the quadlet count of the bus_info_block included in the 1394 configuration ROM.
23:16	crc_length	rw	00h	Set the 1394 configuration ROM byte count in quadlet unit.
15:0	rom_crc_value	rw	0000h	Set the 1394 configuration ROM CRC.

3.2.6. Bus Identification

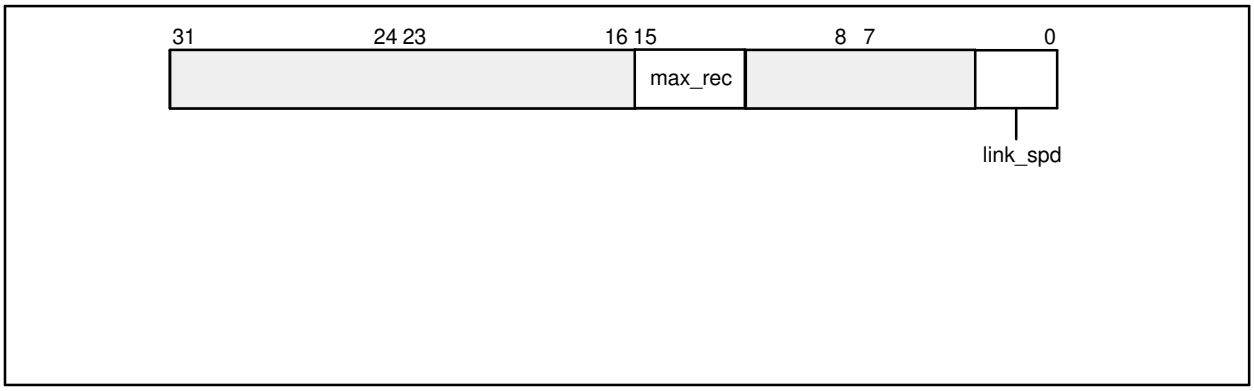
This register stores and indicates the first data of the bus_Info_block field in the 1934 configuration ROM (off-set “FFFF_F000_0404h”).



Bit	Field Name	rwu	reset	description
31:0	busID	r	3133 _ 3934h	These bits indicate ”3133_3934h” which means the ”1394”d in ASCII code.

3.2.7. Bus Options

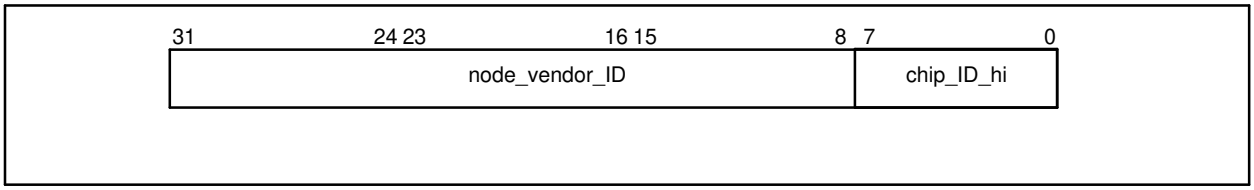
This register stores and indicates the second data of the bus_Info_block field in the 1934 configuration ROM (off- set “FFFF_F000_0404h”). The max_rec field is available only when the HCControl.linkEnable bit is set. If a packet, which has the transmission speed over the setting, is received, it is invalid.



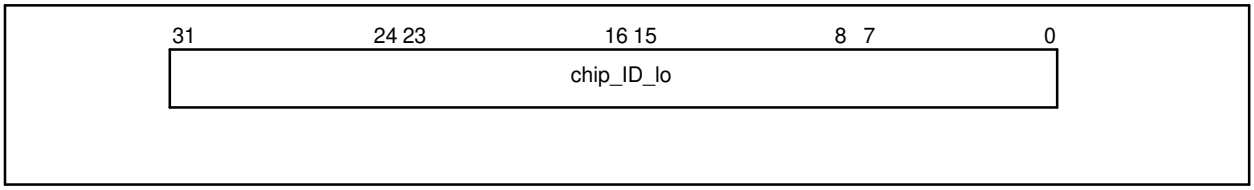
Bit	Field Name	rwu	reset	description
15:12	max_rec	rw	*	These bits are used to specify the maximum byte count that is possible for payloading at the block- write- request packet receipt. *: ”512” is indicated at the hardware reset state. For the software reset state, the current stored data are indicated.
2:0	link_spd	ru	010b	Specify the link transfer rate. As the maximum transfer rate is S400, Link_spd shows ”010b.”

3.2.8. Global Unique ID

This register contains and indicates a part of data in bus_info_block of 1394 configuration ROM with FFFF_F0000_040C, 0410h offset. The data for vendor ID and chip ID contained in the externally connected EEPROM are loaded into this register after releasing from the hardware reset state. Yet the software reset does not flush the data and the register still holds the previous data.



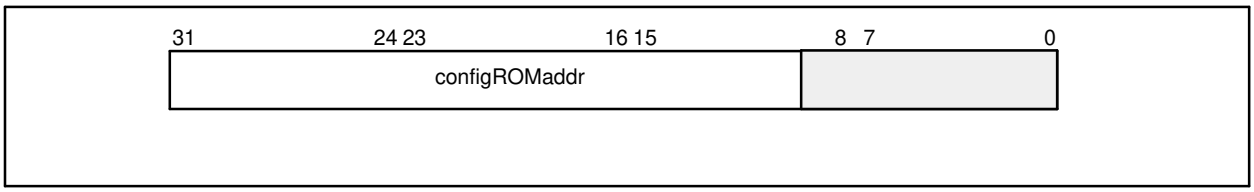
Bit	Field Name	rwu	reset	description
31:8	node_vendor_ID	rw	undefined	1394 OHCI vendor ID is loaded from the EEPROM.
7:0	chip_ID_hi	rw	undefined	The upper 8- bit code of chip ID included in EEPROM is loaded.



Bit	Field Name	rwu	reset	description
31:0	chip_ID_lo	rw	undefined	Load the lower 32- bit code of chip ID included in EEPROM.

3.2.9. Configuration ROM Mapping

This register specifies the base address for the 1394 configuration ROM space. However, since the first 20- byte (offset: FFFF_F000_0400- 0410h) in the ROM is built in the MB86613S, the specified value in this register has no meaning. The base address must be set to this register before setting the HCControl.linkEnable bit.



Bit	Field Name	rwu	reset	description
31:10	configROMaddr	rw	undefined	Specify the base address of 1394 configuration ROM in 1024- byte boundary.

3.2.10. PostedWriteAddress

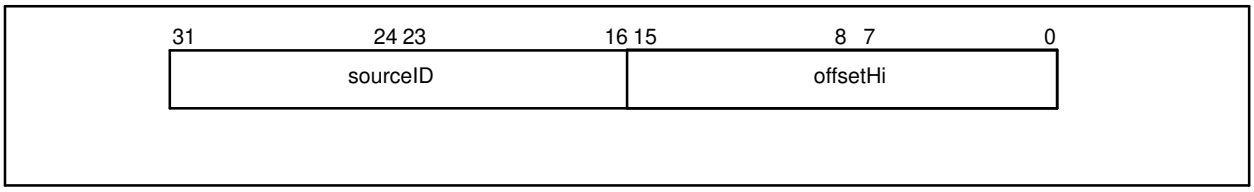
This register is used for storing and indicating the packet source ID and destinationOffset if a host bus error occurs while the payload data for received physical write request packet is storing into a host memory. This register is valid only when the HCControl.postedWriteEnable bit is set.

The structure of register is 4 FIFOs.

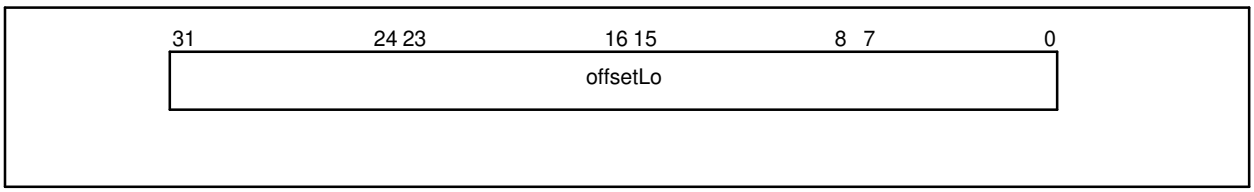
When a host bus error occurs, this register stores the packet source ID and destination offset. After that, IntEvent.PostedWriteErr bit is set. Software needs to take the following steps when the IntEvent.PostedWriteErr bit is set:

1) Read out the PostedWriteAddress_Hi register.
2) Read out the PostedWriteAddress_Lo register.
3) Clear the IntEvent.PostedWriteErr bit.

If the IntEvent.PostedWriteErr bit is still set, take the above steps repeatedly.



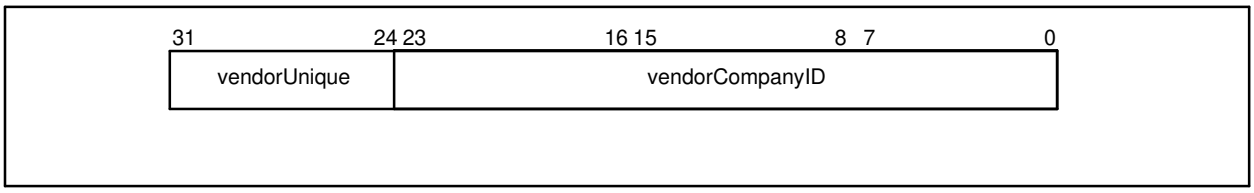
Bit	Field Name	rwu	reset	description
31:16	sourceID	ru	undefined	Stores the source ID written in the packet that caused an error during the physical write request packet receive.
15:0	offsetHi	ru	undefined	Stores the upper 16- bit of offset_address written in the packet that caused an error during the packet receive.



Bit	Field Name	rwu	reset	description
31:0	offsetLo	ru	undefined	Stores the lower 32- bit of offset_address written in the packet that caused an error during the packet receive.

3.2.11. Vendor ID

This register indicates the vender ID of the 1394 Open HCI acquired from the Registration Authority Committee (RAC). This bits indicate “000000”h because this MB86613S is complied with the OHCI.

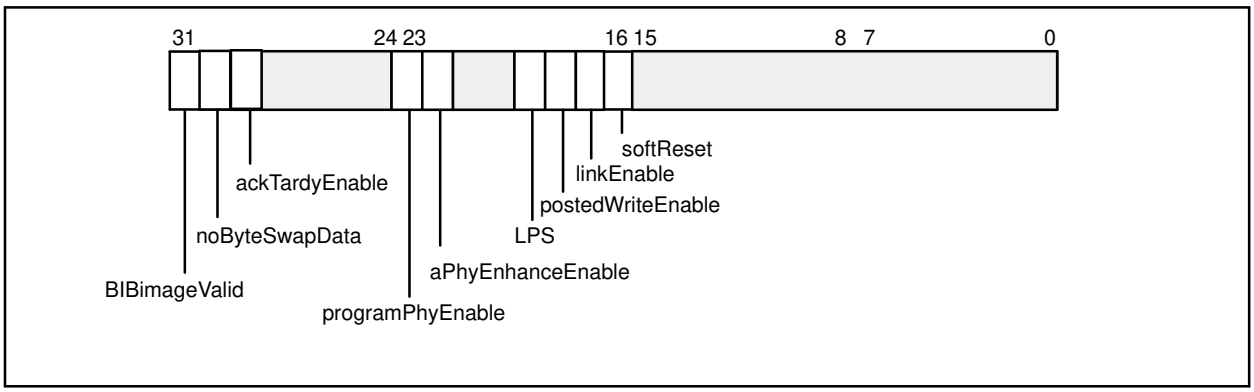


Bit	Field Name	rwu	reset	description
31:24	vendorUnique	r	00h	These bits indicate ”00h”.
23:0	vendorCompanyID	r	000000h	These bits indicate “000000h”.

3.2.12. HCControl

This register controls the host operation conditions such as byte- swap and link enable.

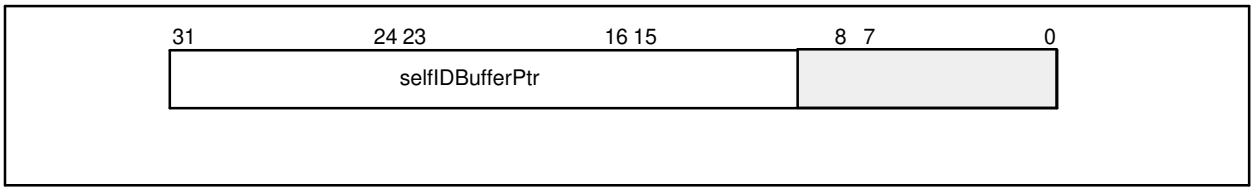
noByteSwapData and PostedWriteEnable bits must be set when the linkEnable is not set.
MB86613S contains PHY and LINK connected together inside of the chip. So, software can not control the performance between PHY and LINK. Therefore, programPhyEnable bit indicates ”0” and aPhyEnhanceEnable bit indicates ”1”. When a ”disconnect” is found in the 1394 network, the device automatically clears the linkEnable bit. Do not set linkEnable bit before Configuration ROM mapping register is set.
As D1 state of PCI power management doesn’t supported, ackTaedyEnable bit is reserved.



Bit	Field Name	rscu	reset	description
31	BIBimageValid	rsc	0b	When this bit is zero, return ack_type_error on block read requests. When this bit is set, enable the mechanism of automatically updating configuration ROM.
30	noByteSwapData	rsc	undefined	Big Endian: specify ”1”. Little Endian: specify ”0”.
29	ackTardyEnable	rsc	0b	reserved
23	programPhyEnable	r	0b	This bit always indicates ”0”.
22	aPhyEnhanceEnbl	r	1b	This bit always indicates ”1”.
19	LPS	rsc	0b	Specifying “1” at this bit sets the LINK to ON.
18	postedWriteEnable	rsc	undefined	Specifying ”1” at this bit supports the posted write function.
17	linkEnable	rsu	0b	Specifying ”1” at this bit makes the LINK enabled.
16	softReset	rsu	*	”1” at this bit performs the software reset. ”1” is indicated during the hardware or software reset.

3.2.13. Self ID Buffer Pointer

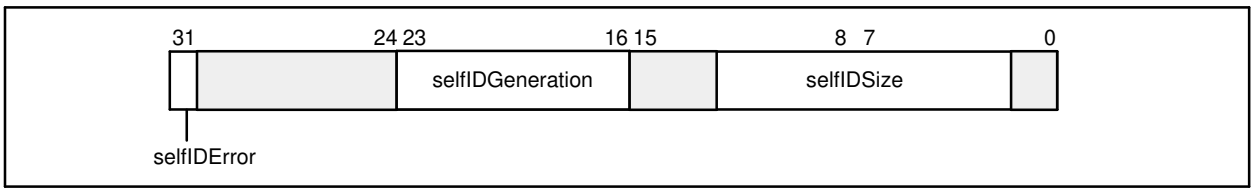
Specify the host memory base address where the received self ID packet is stored.



Bit	Field Name	rwu	reset	description
31:0	selfIDBufferPtr	rw	undefined	Specify the base address of the host memory where the received self ID packet is stored. (2048- byte boundary)

3.2.14. Self ID Count

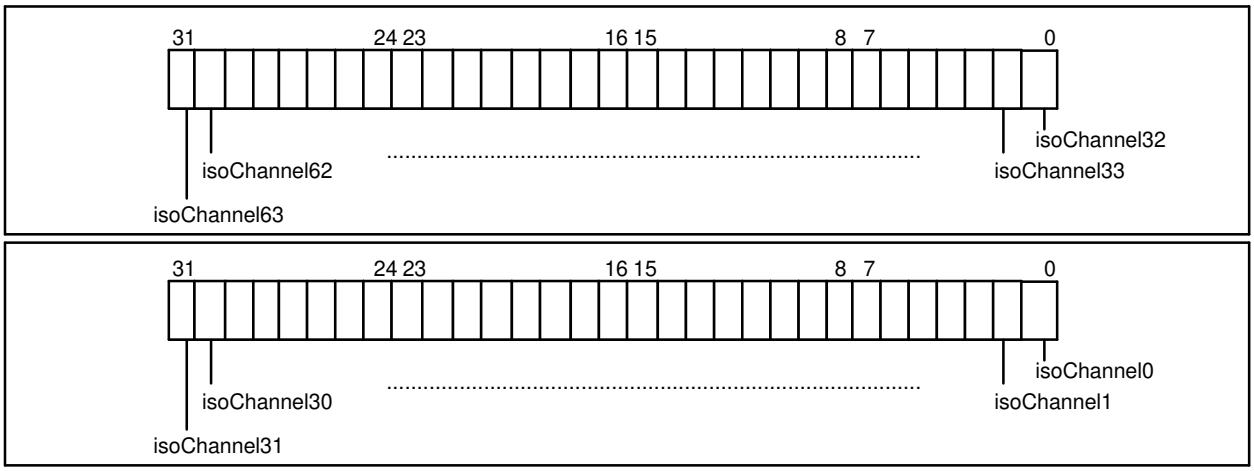
This register indicates the status information with regard to self ID packet at a bus reset. selfIDSize field is cleared by the bus reset.



Bit	Field Name	rwu	reset	description
31	selfIDError	ru	undefined	This bit indicates ”1” when an error occurs while the received self ID packet is stored in the host memory.
23:16	selfIDGeneration	ru	undefined	These are selfID generation counter that increments every bus reset detected.
12:2	selfIDSize	ru	undefined	These bits indicate the size of received self ID packet in quadlet unit.

3.2.15. IRMultiChanMask

This register is used for setting the channel of packet when isochronous packet with multiple channels is received. This register is valid only when IRContextControl.multiChanMode bit is set. Also, only one channel out of 4 channels of IR- CPC can be used. If multiple channels are set with their IRContextControl.multiChanMode bit, The priority is made to channel 0 in order. Using this register requires the IRContextControl.bufferFill and isochHeader bits to be set. Also, the IRContextMatch.channelNumber filed is invalid.

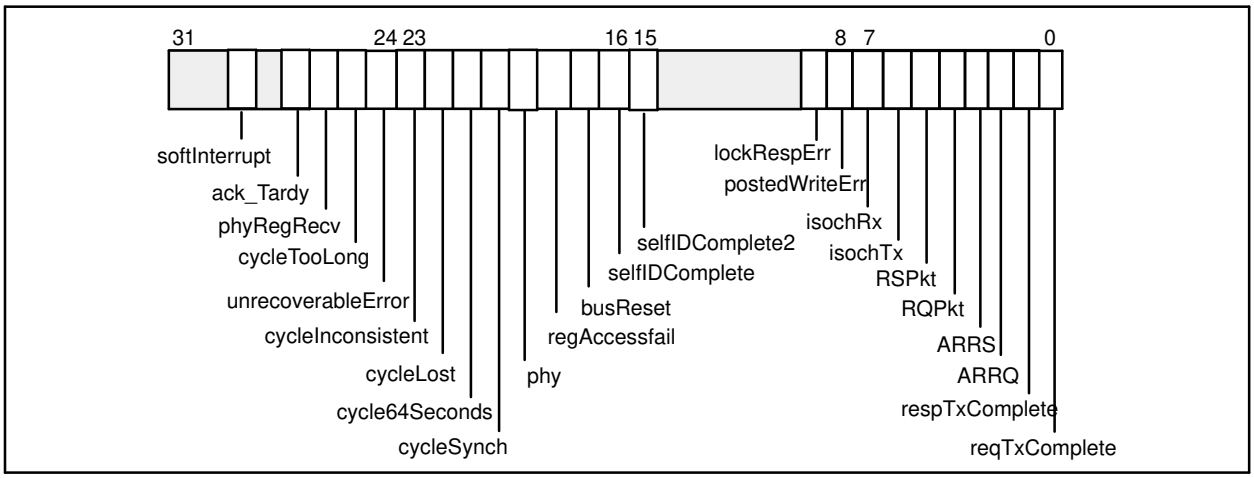


Bit	Field Name	rscu	reset	description
31:0	isoChannelN	rsc	undefined	N denotes the channel number. When IRContextControl.multiChanMode is set, the isochronous packet on the channel being specified with this register is received.

3.2.16. IntEvent

This register indicates the interrupt event for transfer error during the packet transfer or between the device and host memory. The read value from the IntEventClear register contains the AND operated value between IntEvent register and IntMask register. The device clears selfIDcomplete bit by setting the IntEvent.busReset bit at the bus reset. Software must not clear the busReset bit until the selfIDComplete bit contains ”1”.

Also, the interrupt event can be desirably generated by setting the desired bit.



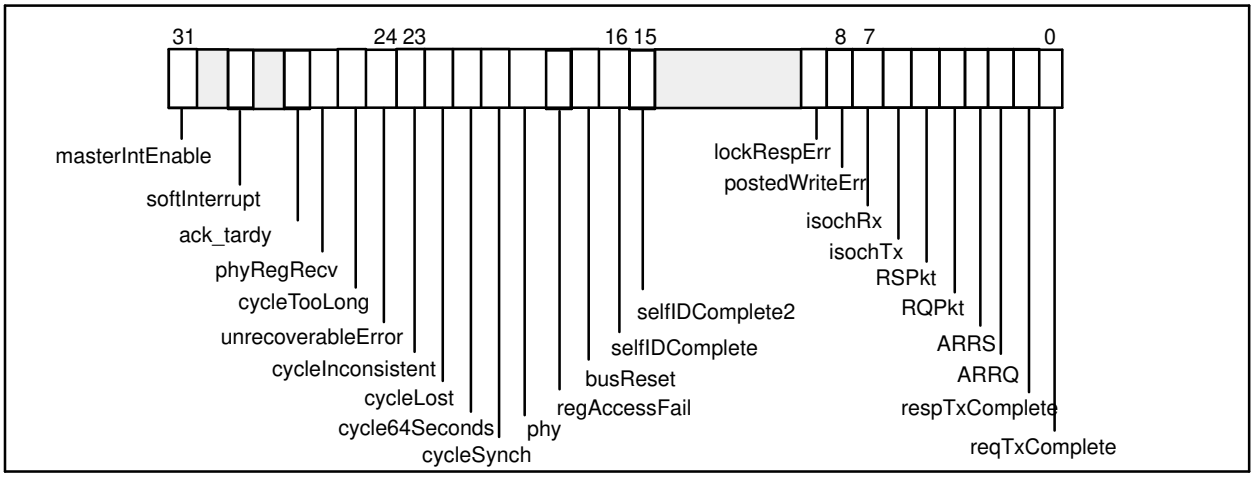
Bit	Field Name	rscu	reset	description
29	softInterrupt	rsc	undefined	Software Interrupt
27	ack_Tardy	rscu	undefined	reserved
26	phyRegRecv	rscu	undefined	This bit indicates ”1” when reading out the PHY register is completed.
25	cycleTooLong	rscu	undefined	This bit indicates ”1” when the isochronous cycle time exceeded beyond the allowable time in the cycle master operation.
24	unrecoverable-Error	rscu	undefined	This bit indicates ”1” when an unrecoverable error such as abort the PCI data transfer occurs.
23	cycle-Inconsistent	rscu	undefined	This bit indicates if the received cycle start packet contains the different data from those in the cycletimer.second and cycleTimer.count.
22	cycleLost	rscu	undefined	This bit indicates ”1” when no cycle start packet was received or transmitted between two cyclesynch periods.
21	cycle64-Seconds	rscu	undefined	This bit indicates ”1” every 64 seconds.
20	cycleSynch	rscu	undefined	This bit indicates ”1” every new isochronous cycle is detected.
19	phy	rscu	undefined	This bit indicates ”1” when the PHY detected the following interrupt: 1) loop detected interrupt 2) cable power failure detected interrupt 3) arbitration state machine timeout interrupt 4) port event interrupt Each interrupt register in the PHY register is cleared after this bit is cleared.

Bit	Field Name	rscu	reset	description
18	regAccessFail	rscu	undefined	This bit indicates”1” when OHCI register access failed by SCLK missing from PHY.
17	busReset	rscu	undefined	This bit indicates ”1” when a bus reset is detected.
16	selfIDComplete	rscu	undefined	This bit indicates ”1” when self ID packet was completely received.
15	selfIDcomplete2	rscu	undefined	This bit indicates”1” when next self ID packet was completely received during selfIDComplete is”1”.
9	lockRespError	rscu	undefined	This bit indicates ”1” when an acknowledge except ”ack_complete” is received after transmitting the lock response packet in response to the received lock request packet.
8	posted-WriteError	rscu	undefined	This bit indicates ”1” when a host bus error occurs while the received phy write request packet is stored in the host memory and ack_complete acknowledge has already been reported.
7	isochRx	ru	undefined	This bit indicates ”1” when the context program process for the isochronous- receive is done. For details, see section 3.2.19.
6	isochTx	ru	undefined	This bit indicates ”1” when the isochronous packet transmit is done. For details, see section 3.2.18.
5	RSPkt	rscu	undefined	This bit indicates ”1” when the received asynchronous response packet is completely stored in the host memory.
4	RQPkt	rscu	undefined	This bit indicates ”1” when the received asynchronous request packet is completely stored in the host memory.
3	ARRS	rscu	undefined	This bit indicates ”1” when the context program process for the asynchronous- receive response is done.
2	ARRQ	rscu	undefined	This bit indicates ”1” when the context program process for the asynchronous- receive request is done.
1	respTx-Complete	rscu	undefined	This bit indicates ”1” when the asynchronous response packet is completely transmitted.
0	reqTx-Complete	rscu	undefined	This bit indicates ”1” when the asynchronous request packet is completely transmitted.

3.2.17. IntMask

This register masks the interrupt event with INTA# signal when an error occurs during the packet receive or between the device and host memory. The register format is the same as one for the IntEvent register. The INTA# signal is active when the enabled mask bit and it’s intEvent bit are both ON.

When the IntMask.masterIntEnable bit is cleared, the interrupt mask setting with this register is invalid.

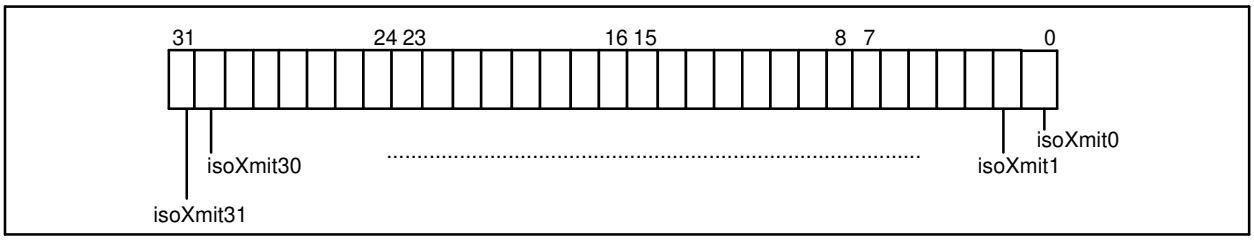


Bit	Field Name	rscu	reset	description
31	masterIntEnable	rsc	0b	When this bit state is ”1”, the INTA# signal is active at the interrupt generation. The effective interrupts must be set in the bits 26:0 in this register. When this bit state is ”0”, the INTA# signal is not active even if any interrupts occur.
29:0	interruptevents	rsc	undefined	These bits are assigned to the maskable interrupt source. For each interrupt source, see section 3.2.16.

3.2.18. IsoXmitIntEvent/Mask

This register indicates the result of each context program operation for IT- CPC. When the context program process completed and ” i ” field at context program is set to ”11b”, the applied channel indicates ”1”.

IntEvent.isochXmit bit indicates the OR operation output with each bit in this register.
Also, the register read value from the IsoXmitIntEventClear indicates AND operation output between the IsoXmitIntEvent register and IsoRecvIntMask register.

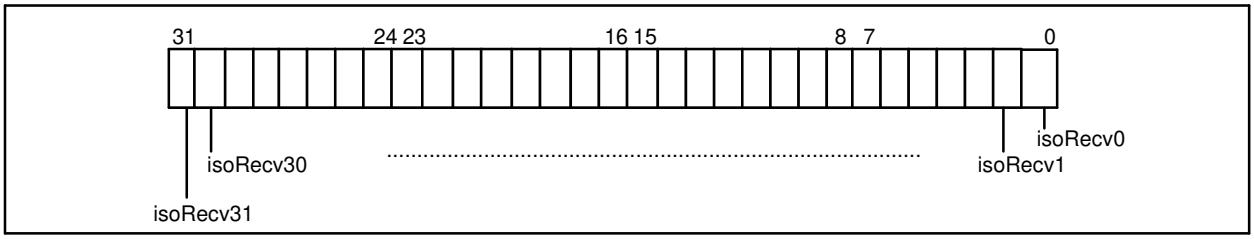


Bit	Field Name	rscu	reset	description
31:4	isoXmit31:4	r	000_0000h	Since the device contains only 4channels of context program controller, these bits are fixed with ”0”.
3:0	isoXmit3:0	rscu	undefined	Each controller processes the context program and ”1” is indicated in the corresponding bit when the isochronous packet is completely sent out.

3.2.19. IsoRecvIntEvent/Mask

This register indicates the result of each context program operation for IR- CPC. When the context program process completed and " i " field at context program is set to "11b", the applied channel indicates "1".

IntEvent.isochRx bit indicates the OR operation output with each bit in this register.
Also, the register read value from the IsoRecvIntEventClear indicates AND operation output between the IsoRecvIntEvent register and IsoRecvIntMask register.

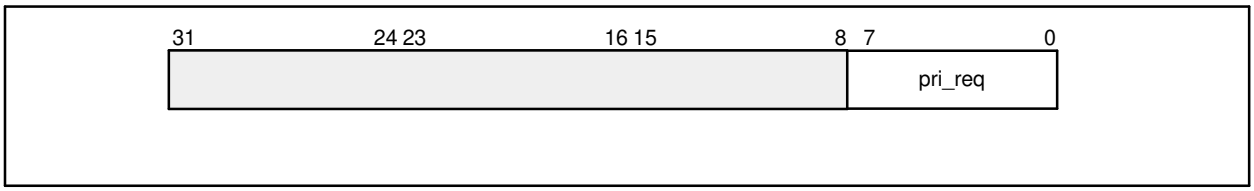


Bit	Field Name	rscu	reset	description
31:4	isoRecv31:4	r	000_0000h	Since the device contains only 4channels of context program controller, these bits are fixed with "0".
3:0	isoRecv3:0	rscu	undefined	Each controller processes the context program and "1" is indicated in the corresponding bit when the isochronous packet is completely received.

3.2.20. FairnessControl

This register contains and indicates the maximum execution count of priority request to transmit an asynchronous request packet. Software needs to support and implement the priority budget register defined in P1394a standard. The on- chip LINK layer loads the value specified in pri_req field in this register before activating the fairness interval and decrements the value every execution of priority request.

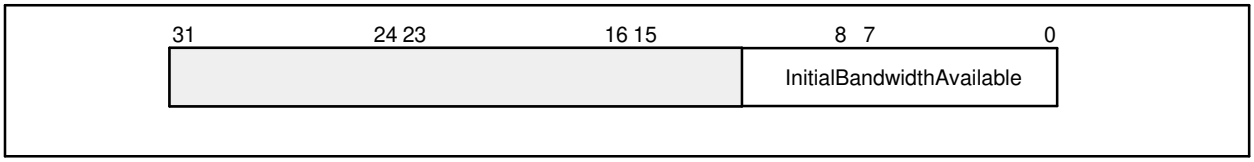
This register keeps the previous value with the software reset (HCControl.softReset).



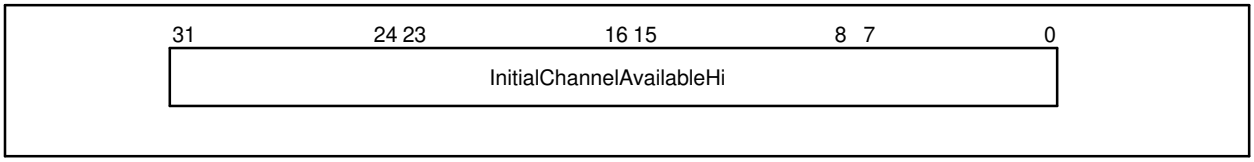
Bit	Field Name	rwu	reset	description
7:0	pri_req rw	undefined		These bits specify the maximum number of priority request

3.2.21. BUS Management CSR Initialization

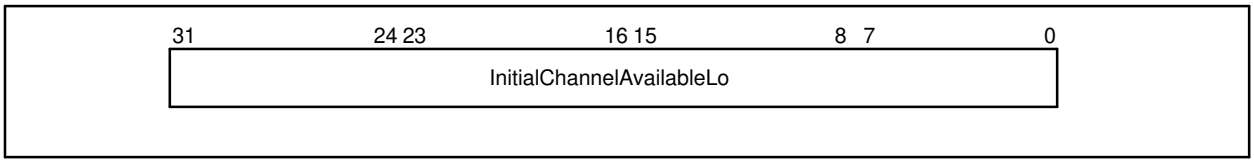
These registers keep initial value of bandwidth_available and channel_available_hi/lo registers.
BY hardware reset, software reset and Bus reset these registers values are loaded to band width_available and channel_available_hi/lo registers.



Bit	Field Name	rscu	reset	description
12:0	InitialBandwidthAvailable	rw	1333h	These bits specify the initial value of bandwidth_available register.



Bit	Field Name	rscu	reset	description
31:0	InitialChannelAvailableHi	rw	FFFFFFFFh	These bits specify the initial value of Channel_available_Hi register.

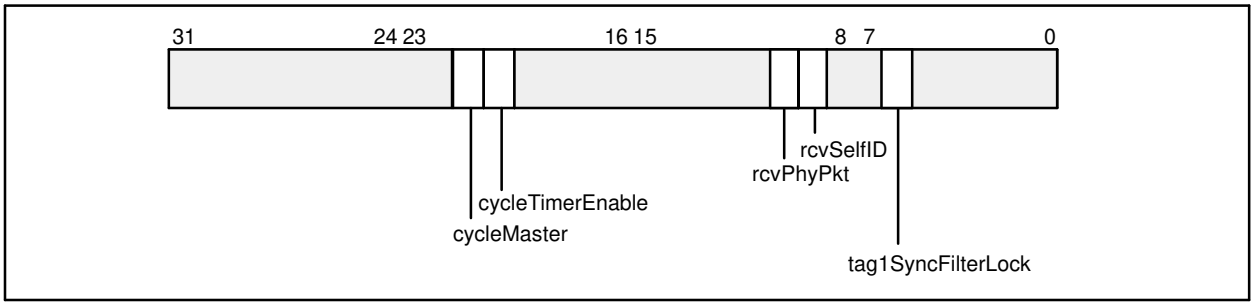


Bit	Field Name	rscu	reset	description
31:0	InitialChannelAvailableLo	rw	FFFFFFFFh	These bits specify the initial value of Channel_available_Lo register.

3.2.22. LinkControl

This register controls the on- chip LINK layer, such as the packet process and cycle timer.

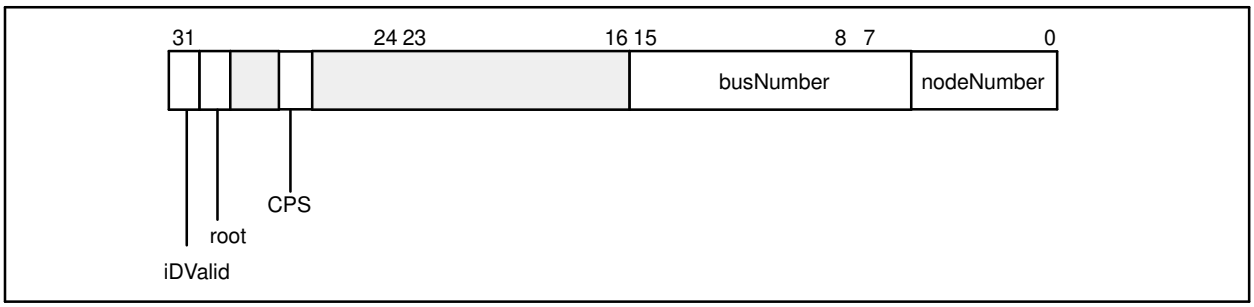
CycleSource bit is valid only when cycleMaster bit is enable. CycleMaster bit is cleared only when IntEvent cycleTooLong bit is set. Also, if the rcvPhyPkt bit and HCControl.linkEnable bit are set, PHY packets (received phy configuration, link- on, ping packets and so on) are stored in the FIFO.



Bit	Field Name	rscu	reset	description
21	cycleMaster	rscu	undefined	Cycle master enable bit. Writing ”1” at this bits makes the device cycle master function.
20	cycleTimerEnable	rsc	undefined	Specifying ”1” counts up the CycleTimer.cycleOffset by the 24.576MHz clock.
10	rcvPhyPkt	rsc	undefined	”1” at this bit receives the the PHY packet in the FIFO.
9	rcvSelfID	rsc	undefined	”1” at this bit receives the self ID packet in the FIFO.
6	tag1SyncFilterLock	rs	*	“1” at this bit set the IRContexMatch and tag1SyncFilter bit. *:this bit is cleared only by hardware reset.

3.2.23. Node Identification and Status

This register indicates a part of data stored in the PHY register. Do not set the Contextcontrol.run bit when the iDValid bit is cleared.

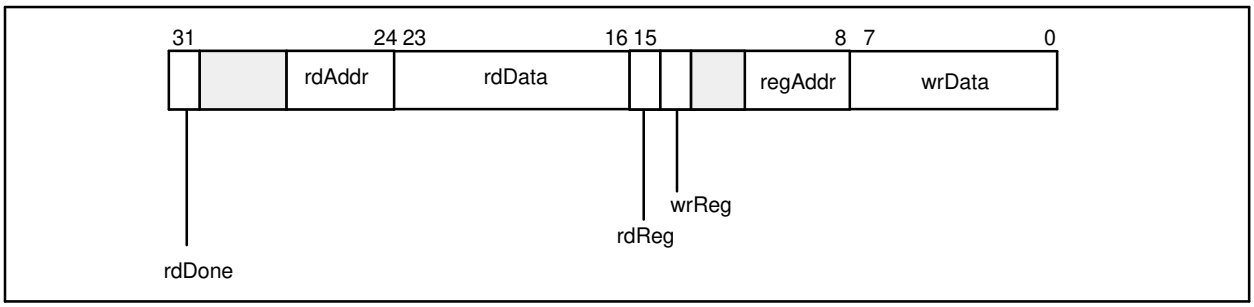


Bit	Field Name	rwu	reset	description
31	iDValid	ru	0b	This bit indicates ”1” when the nodeNumber filed has valid data.
30	root	ru	0b	This bit indicates ”1” when the MB86613S is the root device.
27	CPS	ru	0b	This bit indicates the cable power status.
15:6	busNumber	rwu	3FFh	These bits store the bus number inserted in the packet header.
5:0	nodeNumber	ru	undefined	These bits indicate the device’s node number.

3.2.24. PHY Control

PHY control register is used to allow software to access the PHY registers. When reading out the data from the PHY registers, set the required register address in the regAddr and set the rdReg. rdDone is cleared at this time. After that, the rdDone and IntEvent.PhyRegRecv bits are set if the data in the specified address is stored in the rdData and the PHY register address is stored in the rdAddr. the rdReg is cleared at this time. When writing the data to the PHY registers, set the required register address in the regAddr and write the data in the wrData, and set the wrReg. the rdDone is cleared at this time. the rdDone is then set if the required data is written to the wrData. The wrReg is cleared at this time.

Please do not set the rdReg and wrReg bits while the rdDone is cleared.



Bit	Field Name	rwu	reset	description
31	rdDone	ru	undefined	This bit indicates ”0” when rdReg or wrReg is set. After that, the bit indicates ”1” when the r/w operation to PHY registers is done.
27:24	rdAddr	ru	undefined	These bits indicate the PHY register address in which the data is specified in rdData bits
23:16	rdData	ru	undefined	These bits indicate the data in the PHY register specified in the rdAddr bits.
15	rdReg	rwu	0b	This bit is used for reading out the data from the register address specified in the regAddr bits.
14	wrReg	rwu	0b	This bit is used for writing the data stored in the wrData bits to the register address specified in the regAddr bits.
11:8	regAddr	rw	undefined	Specify the PHY register address.
7:0	wrData	rw	undefined	Specify the data written to the PHY register.

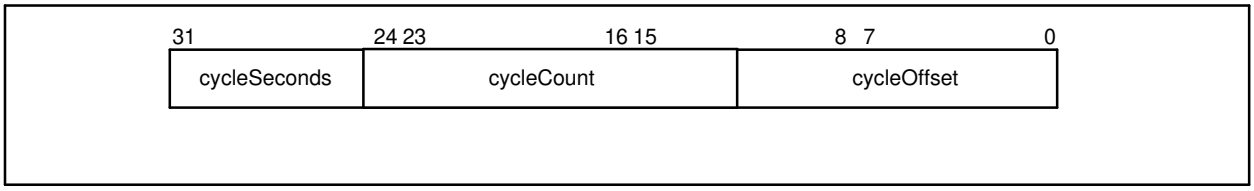
3.2.25. Isochronous Cycle Timer

This register sets and indicates the cycle timer value that bus manager owns.

When the MB86613S device is a cycle master, the transmit cycle start packet is generated with data set in this register. In this case, if the cycleCount field is counted by the external CLK (8kHz) using the LinkControl.cycleSource bit, cycleOffset field is also counted using the LinkControl.cycleTimerEnable bit, and external clock is input to the device, then the cycleOffset field is cleared. Also, if no external clock has been input until cycleOffset makes 3072 counts, the cycleCount field is not incremented and waits for the external clock input pausing the cycleOffset count.

When the device is not a cycle master, it stores the received cycle start packet data into this register. In case no cycle start packet is not coming, the counter keeps its function by continuing the cycleOffset count.

This register contains the previous data when the hardware or software reset(HCControl.softReset) is done.

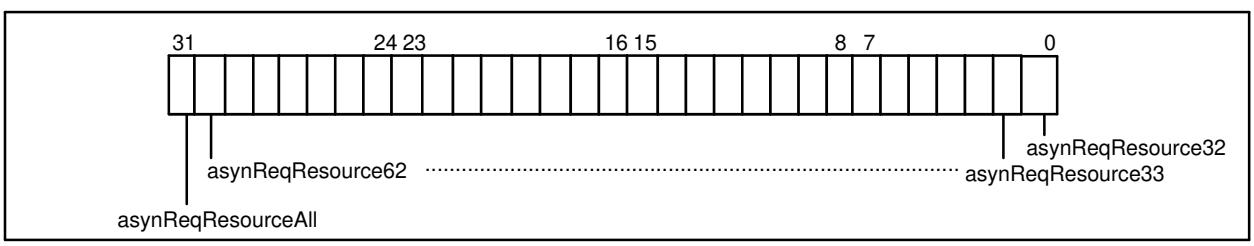


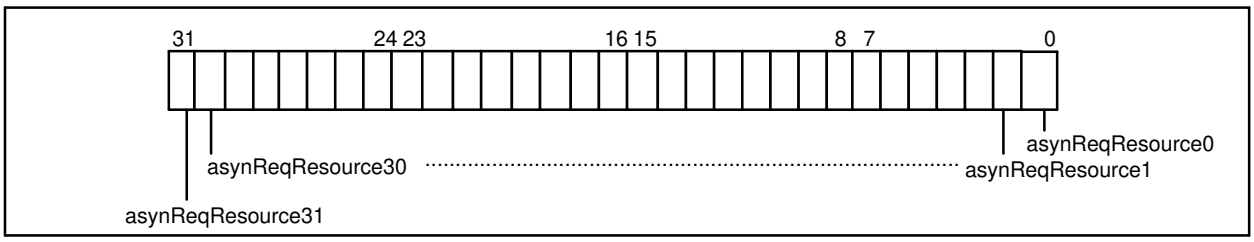
Bit	Field Name	rwu	reset	description
31:25	cycleSeconds	rwu	undefined	The value in these bits is increment every 8000 counts of the cycleCount.
24:12	cycleCount	rwu	undefined	When the LinkControl.cycleSource bit is set to ”1”, these counter bits are increment by the CSCLK (8kHz). When the LinkControl.cycleSource bit is set to ”0”. these counter bits are increment every 3072 counts of cycleOffset.
11:0	cycleOffset	rwu	undefined	These are the cycleOffset bits to manage the 125μs cycle by 3072 counts using the 24.576MHz clock.

3.2.26. AynchronousRequestFilter

This register functions as a packet filter by referring to the source ID of received asynchronous request packet. The device stores asynchronous request packet which has the source ID specified in this register. (Any other asynchronous request packets are not stored and no acknowledge is reported.) asynchronous response packets and request packets for 1394 configuration ROM and bus management CSR are not applied to the filtering conditions.

This register (except asynReqResourceAll bit) is cleared by a bus reset.



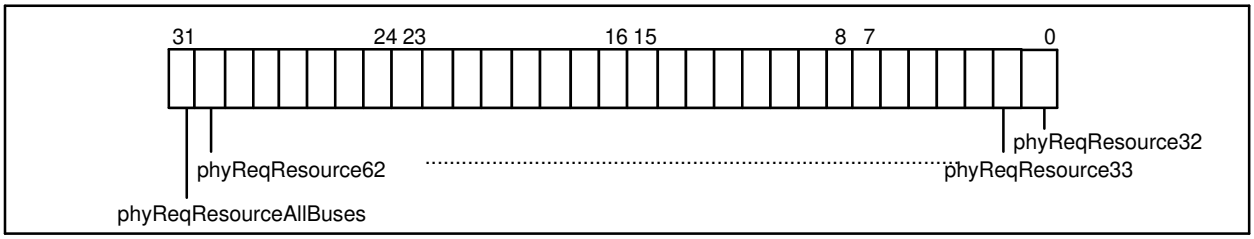


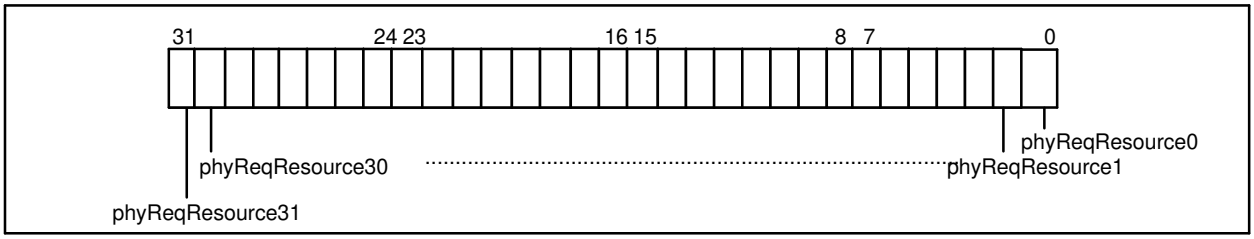
Bit	Field Name	rscu	reset	description
31	asynReqResource All	rsc	0b	”1” at this bit indicates that all the asynchronous request packets are received regardless of the source ID bus number, and node number in the packet header.
30:0	asynReqResourceN	rsc	0b	N denotes the node number. ”1” at these bits means that the device receives the asynchronous request packet on the specified node number N when the bus number in the source ID is ”3FFh” or ”NodeID.busNumber”.

3.2.27. PhysicalRequestFilter

This register filters the asynchronous request packets that have the source ID set in AsynchronousRequest Filter register to handle them as physical request packets. In the asynchronous request packets, the packet which has the offset address in range from 0 to the value specified in the PhysicalUpperBound register is filtered by the source ID. The filtered packet is handled as the physical request packet. The request packets for the 1394 configuration ROM and bus management CSR are not filtered and they are directly handled as the physical request packets without the conditions.

This register(except physReqResourceAllBuses bit) is cleared by the bus reset.



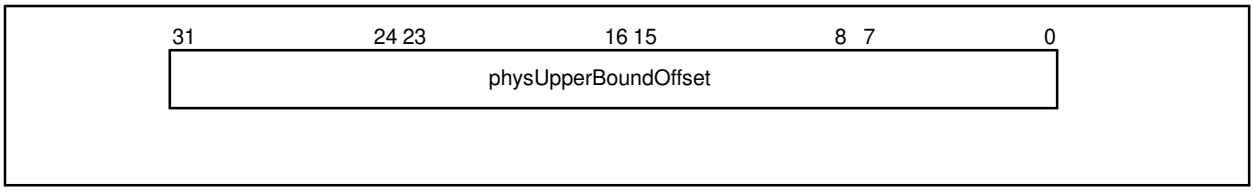


Bit	Field Name	rscu	reset	description
31	phyReqResource AllBuses	rsc	0b	”1” at this bit indicates that all the physical request packets are received regardless of the node number, if the bus number at source ID is not ”3FFh” and it is not set in NodeID busNumber field.
30:0	phyReqResourceN	rsc	b0	N denotes the node number. ”1” at these bits means that the device receives the physical request packet on the specified node number N when the bus number in the source ID is ”3FFh” or ”NodeID.busNumber”.

3.2.28. Physical Upper Bound

This register specifies the offset address range for a physical request packet. When the offset address range for the received packet is from 48'h"0000_0000_0000" to 48'hphysUpperBoundOffset register value (32- bit)_0000" - 1, the packet is considered as a physical request packet. The maximum offset value set in this register is "0001_0000h".

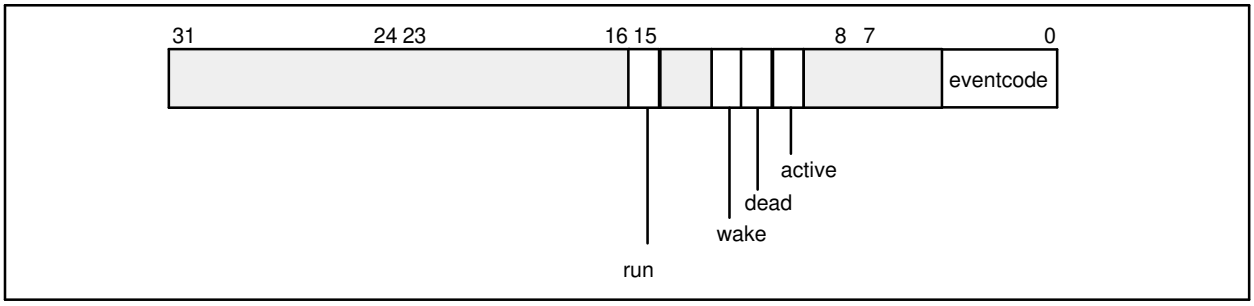
This register contains the previous value when software reset (HCControl.softReset) is done.



Bit	Field Name	rwu	reset	description
31:0	physUpper BoundOffset	rw	0001_ 0000h	These bits specify the upper 32- bit of the offset address that recognizes the phy request packet.

3.2.29. AT ContextControl

This register is used for controlling the AT- CPC. This register is prepared for request and response packets individually.



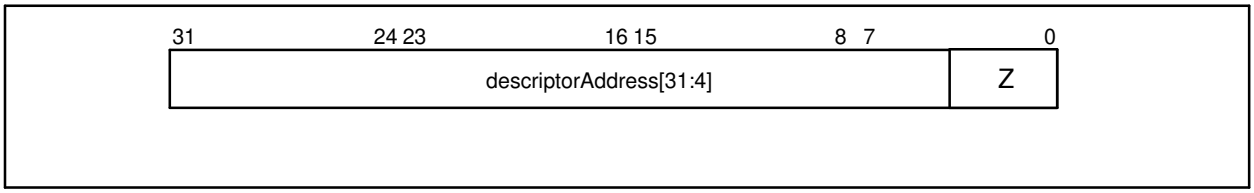
Bit	Field Name	rscu	reset	description
15	run	rscu	0b	Specifying "1" at this bit runs the context program controller (AT- CPC).
12	wake	rsu	undefined	Specifying "1" at this bit wakes up the context program controller to restart the process.
11	dead	ru	0b	This bit indicates "1" when the context program controller can not proceed the operation because an error occurred.
10	active	ru	0b	This bit indicates "1" when the context program controller is in- process.
4:0	eventcode	ru	undefined	These bits contain the result of context program process.

3.2.30. AT DMA CommandPtr

This register specifies the start address of host memory where the first context program is contained and the number of descriptor. Set this register when ATContextControl.run bit and .active bit are cleared.

The context program can be composed of maximum 8 descriptors. However, because the descriptor to specify the packet header needs 32- byte program, please set ”2h” to ”8h” in the Z field.

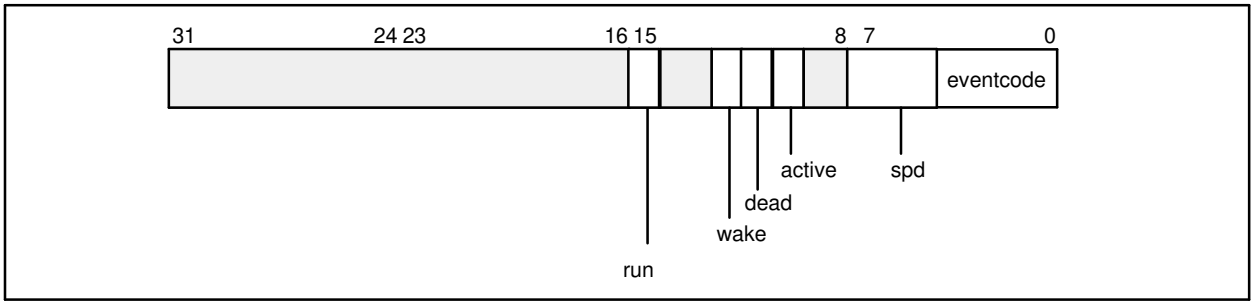
When an error occurred while processing the context program, the descriptor address field indicates the start address in the host memory where the descriptor that was processed is contained. When the context program is normally completed, the address field indicates the host memory start address where the last processed descriptor is contained and Z field indicates ”0”.



Bit	Field Name	rwu	reset	description
31:4	descriptorAddress	rwu	undefined	Specify the start address for the memory where the first context program is stored. (16- byte boundary)
3:0	Z	rwu	undefined	Specify the number of descriptor that forms the context program.

3.2.31. AR ContextControl

This register is used for controlling the AR- CPC. This register is prepared for request and response packets individually.

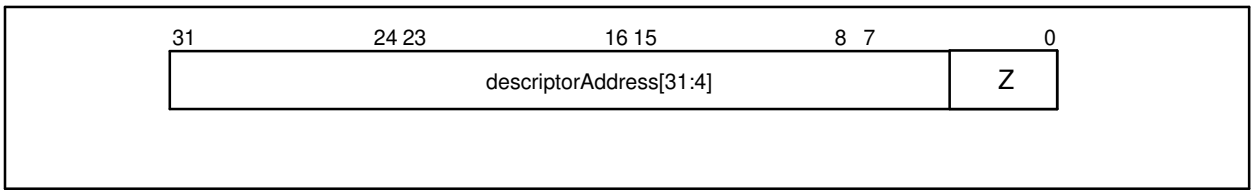


Bit	Field Name	rscu	reset	description
15	run	rscu	0b	Specifying ”1” at this bit runs the context program controller (AR- CPC).
12	wake	rsu	undefined	Specifying ”1” at this bit wakes up the context program controller to restart the process.
11	dead	ru	0b	This bit indicates ”1” when the context program controller cannot proceed the operation because an error occurred.
10	active	ru	0b	This bit indicates ”1” when the context program controller is in- process.
7:5	spd	ru	undefined	These bits indicate the transfer rate at the packet receive.
4:0	eventcode	ru	undefined	These bits contain the result of context program process.

3.2.32. AR DMA CommandPtr

This register specifies the start address of host memory where the first context program is contained and the number of descriptor. Set this register when the ARContextControl.run bit and .active bit are cleared. Since the device supports only buffer- fill mode which means the context program is formed with 1 descriptor, set ”1h” at Z field.

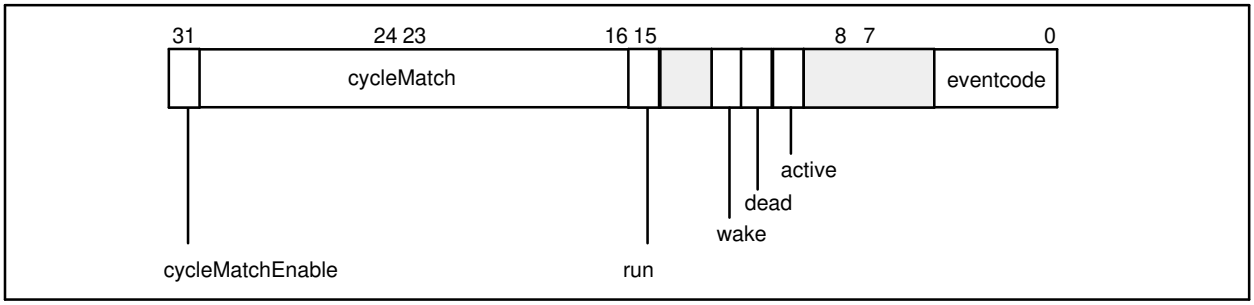
When an error occurred while processing the context program, the descriptor address field indicates the start address in the host memory where the descriptor that was processed is contained. When the context program is normally completed, the address field indicates the host memory start address where the last processed descriptor is contained and Z field indicates ”0”.



Bit	Field Name	rwu	reset	description
31:4	descriptorAddress	rwu	undefined	Specify the start address for the memory where the first context program is stored. (16- byte boundary)
3:0	Z	rwu	undefined	Specify the number of descriptor that forms the context program.

3.2.33. IT ContextControl

This register controls the IT- CPC that transmits isochronous packets. This register is prepared for individual channel of IT- CPC (4 channels). cycleMatchEnable bit is cleared when ITContextControl.active bit is set. The cycleMatch field is valid only when the cycleMatchEnable bit is set.



Bit	Field Name	rscu	reset	description
31	cycleMatchEnable	rscu	undefined	”1” at this bit indicates that the context program process activates when the cycle timer reaches the time specified in the cycleMatch.
30:16	cycleMatch	rsc	undefined	Specify the start time for the context program process.
15	run	rscu	0b	Specifying ”1” at this bit runs the context program controller (IT- CPC).
12	wake	rsu	undefined	Specifying ”1” at this bit wakes up the context program controller to restart the process.
11	dead	ru	0b	This bit indicates ”1” when the context program controller can not proceed the operation because an error occurred.

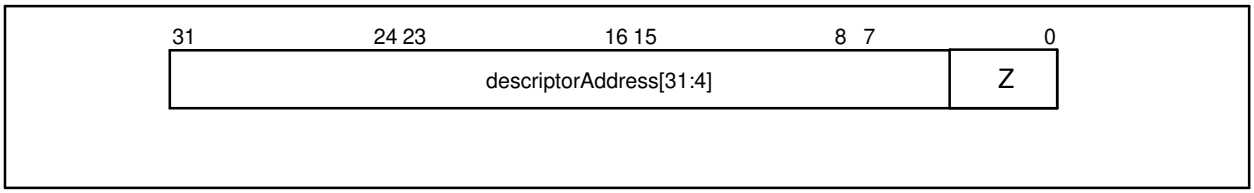
10	active	ru	0b	This bit indicates ”1” when the context program controller is in- process.
4:0	eventcode	ru	undefined	These bits contain the result of context program process.

3.2.34. IT DMA CommandPtr

This register specifies the start address of host memory where the first context program is contained and the number of descriptor. Set this register when ITContextControl.run bit and .active bit are cleared.

The context program can be composed of maximum 8 descriptors. However, because the descriptor to specify the packet header needs 32- byte program, please set ”2h” to ”8h” in the Z field.

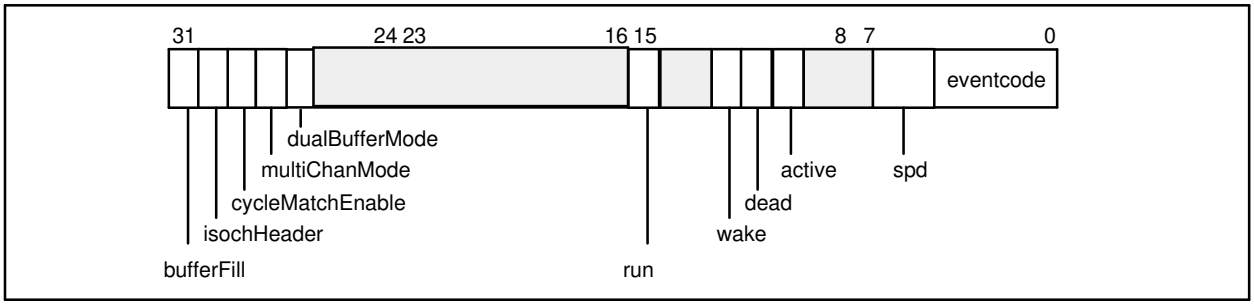
When an error occurred while processing the context program, the descriptor address field indicates the start address in the host memory where the descriptor that was processed is contained. When the context program is normally completed, the address field indicates the host memory start address where the last processed descriptor is contained and Z field indicates ”0”.



Bit	Field Name	rwu	reset	description
31:4	descriptorAddress	rwu	undefined	Specify the start address for the memory where the first context program is stored. (16- byte boundary)
3:0	Z	rwu	undefined	Specify the number of descriptor that forms the context program.

3.2.35. IR ContextControl

This register is used for controlling the IR- CPC. This register is prepared for 4 context programs individually. Set bufferFill, isochHeader, cycleMatchEnable, and multiChanMode bits when IRContextControl.run bit .active bit are cleared. When multiChanMode bit is set, please make sure to set the bufferFill and isochHeader bits too. In this case, the device ignores the IRContextMatch.channelNumber field. cycleMatchEnable bit is cleared when the IRContextControl.active bit is set.
multiChanMode and bufferFill bits are cleared when dualBufferMode bit is set.



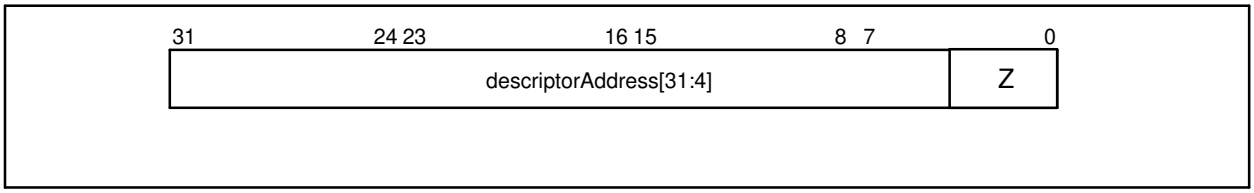
Bit	Field Name	rscu	reset	description
31	bufferFill	rsc	undefined	”1” at this bit indicates that the packet received in buffer-fill- mode is stored in the memory. ”0” at this bit indicates that the packet received in packet-per- buffer- mode is stored in the memory.

30	isochHeader	rsc	undefined	”0” at this bit stores the received packet in the memory removing the header and trailer data.
29	cycleMatchEnable	rscu	undefined	”1” at this bit indicates that the context program process activates when the cycle timer reaches the time specified in the cycleMatch.
28	multiChanMode	rsc	undefined	”1” at this bit, receives the isochronous packet on the channel as set in the IRMultiChanMask register. (Multi-channel mode) ”0” at this bit, receives the isochronous packet on the channel as set in the IRContextMatch.channelNumber. (Single- channel mode)
27	dualBufferMode	rsc	undefined	“1” at this bit, stores the received packet with dual buffer mode. (This bit is not available.)
15	run	rscu	0b	Specifying ”1” at this bit runs the context program controller.
12	wake	rsu	undefined	Specifying ”1” at this bit wakes up the context program controller to restart the process.
11	dead	ru	0b	This bit indicates ”1” when the context program controller can not proceed the operation because an error occurred.
10	active	ru	0b	This bit indicates ”1” when the context program controller is in- process.
7:5	spd	ru	undefined	These bits indicate the transfer rate at the packet receive
4:0	eventcode	ru	undefined	These bits contain the result of context program process.

3.2.36. IR DMA CommandPtr

This register specifies the start address of host memory where the first context program is contained and the number of descriptor. Set this register when the ARContextControl.run bit and .active bit are cleared. When the device is buffer- fill mode which means the context program is formed with 1 descriptor, set ”1h” at Z field. When the device is packet- per- buffer mode, set ”1h” to ”8h” in the Z field because the context program is formed with maximum 8 descriptors.

When an error occurred while processing the context program, the descriptorAddress field indicates the start address in the host memory where the descriptor that was processed is contained. When the context program is normally completed, the address field indicates the host memory start address where the last processed descriptor is contained and Z field indicates ”0”.



Bit	Field Name	rwu	reset	description
31:4	descriptorAddress	rwu	undefined	Specify the start address for the memory where the first context program is stored. (16- byte boundary)

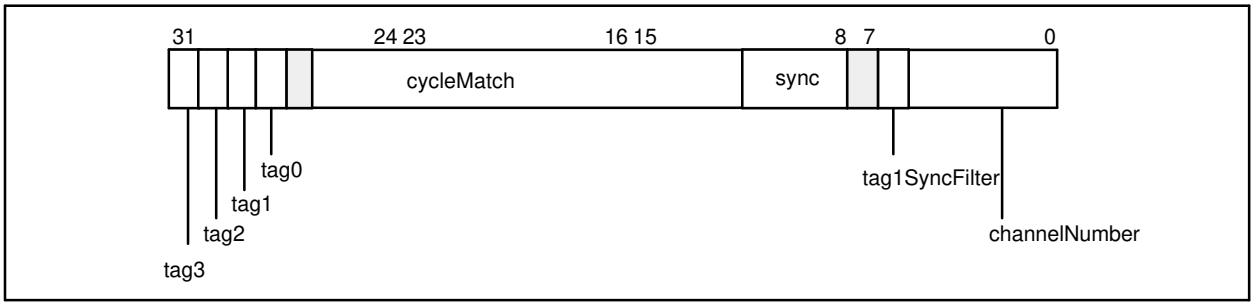
3:0	Z	rwu	undefined	Specify the number of descriptor that forms the context program.

3.2.37. IR ContextMatch

This register specifies the received channel in single- channel mode and the value to be compared with the sync and tag sections in the received packet.

sync field is valid only when " w" field contains "11b" in the first descriptor. The chip discards the received packet even if the tag information and the received packet's channel match with the specified values as long as the sync value is different.

cycleMatch field is valid only when IRContextControl.cycleMatchEnable bit is set. (However, in case IntEvent.cycleInconsistent bit is set, it is not valid.)



Bit	Field Name	rwu	reset	description
31	tag3	rw	undefined	Specifying "1" at this bit enables to receive the isochronous packet that has the tag "11b'.
30	tag2	rw	undefined	Specifying "1" at this bit enables to receive the isochronous packet that has the tag "10b'.
29	tag1	rw	undefined	Specifying "1" at this bit enables to receive the isochronous packet that has the tag "01b'.
28	tag0	rw	undefined	Specifying "1" at this bit enables to receive the isochronous packet that has the tag "00b'.
26:12	cycleMatch	rw	undefined	Specify the start time for the context program process.
11:8	sync	rw	undefined	Specify the value to be compared with the sync on the received isochronous packet.
6	tag1SyncFilter	rw	undefined	Specifying "1" at this bit allows to receive isochronous packet that has only upper 2- bit of sync field contains "00b" and the tag field contains "01b".
5:0	channelNumber	rw	undefined	Specify the channel for the isochronous packet to be received.

4. Context Program Controller (CPC)

Basically Context Program Controller's (CPC) state machine consists of Program Load, Program Analysis, and Interrupt Handle.

CPC starts the operation after the ContextControl.run bit is set by the software. CPC, however, stops the operation when ContextControl.active and .dead bits are cleared as a result of ContextControl.run bit cleared by the software.

In Program Loading Unit (PLU), MB86613S loads the context program from the host memory in which the CommandPtr.descriptorAddress indicates and stores the program into the on- chip work RAM.

In Program Analysis Unit (PAU), MB86613S analyzes the loaded context program and moves the required transmit- packet from/to the work RAM/Host memory to/from FIFO.

In Interrupt Handling Unit (IHU), MB86613S handles/controls the event- code and interrupt as a result from the program analysis. In this case, if Z field in the last descriptor contains "1" or more, the device stores the address set in the branchAddress field of the last descriptor in the CommandPtr.descriptorAddress field in order to store the next context program and returns to the program loading unit.

Figure 4.1 shows the state machine diagram, and Figure 4.2 shows the flag transition of ContextControl Register with the explanation as follows:

1) If the wake bit is set by software, clear the wake bit and set the active bit, then go to the program loading unit. When starting the CPC with the run bit, set the active bit and go to the program loading unit.

2) When the program analysis is completed, go to the interrupt handling unit. If an error occurred in the IHU, set the dead flag and clear the active bit, then wait until the dead flag is cleared by software.

3) If the next context program to be processed is not prepared yet, clear the active bit and wait until the wake bit is set by software; i.e. until the next context program to be processed is prepared.

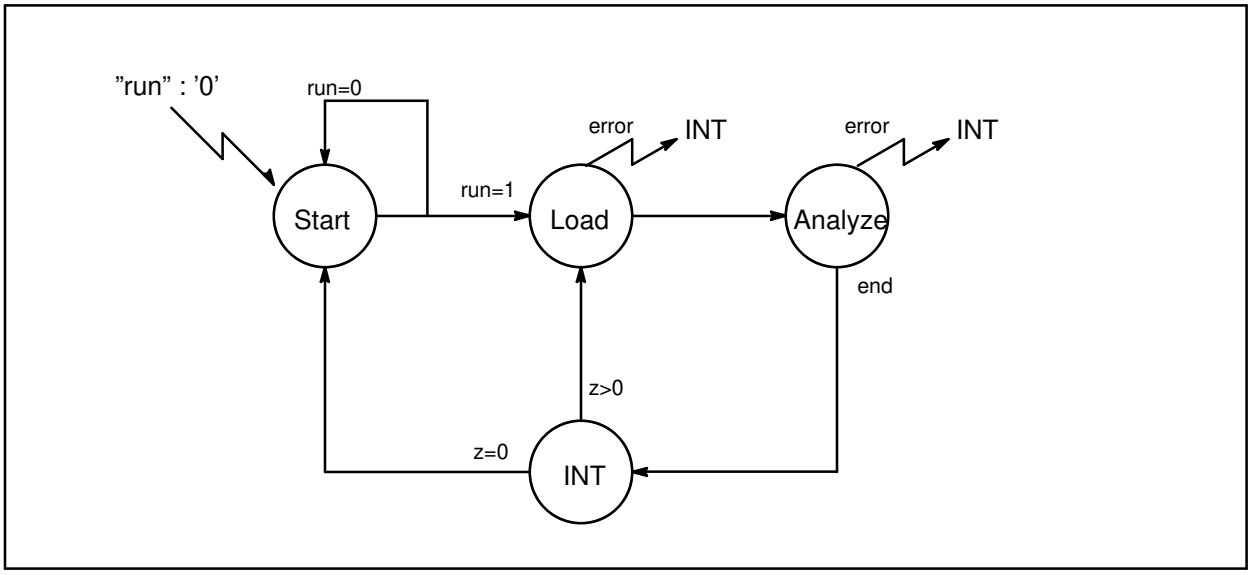


Fig. 4.1 Block Diagram of CPC State Machine

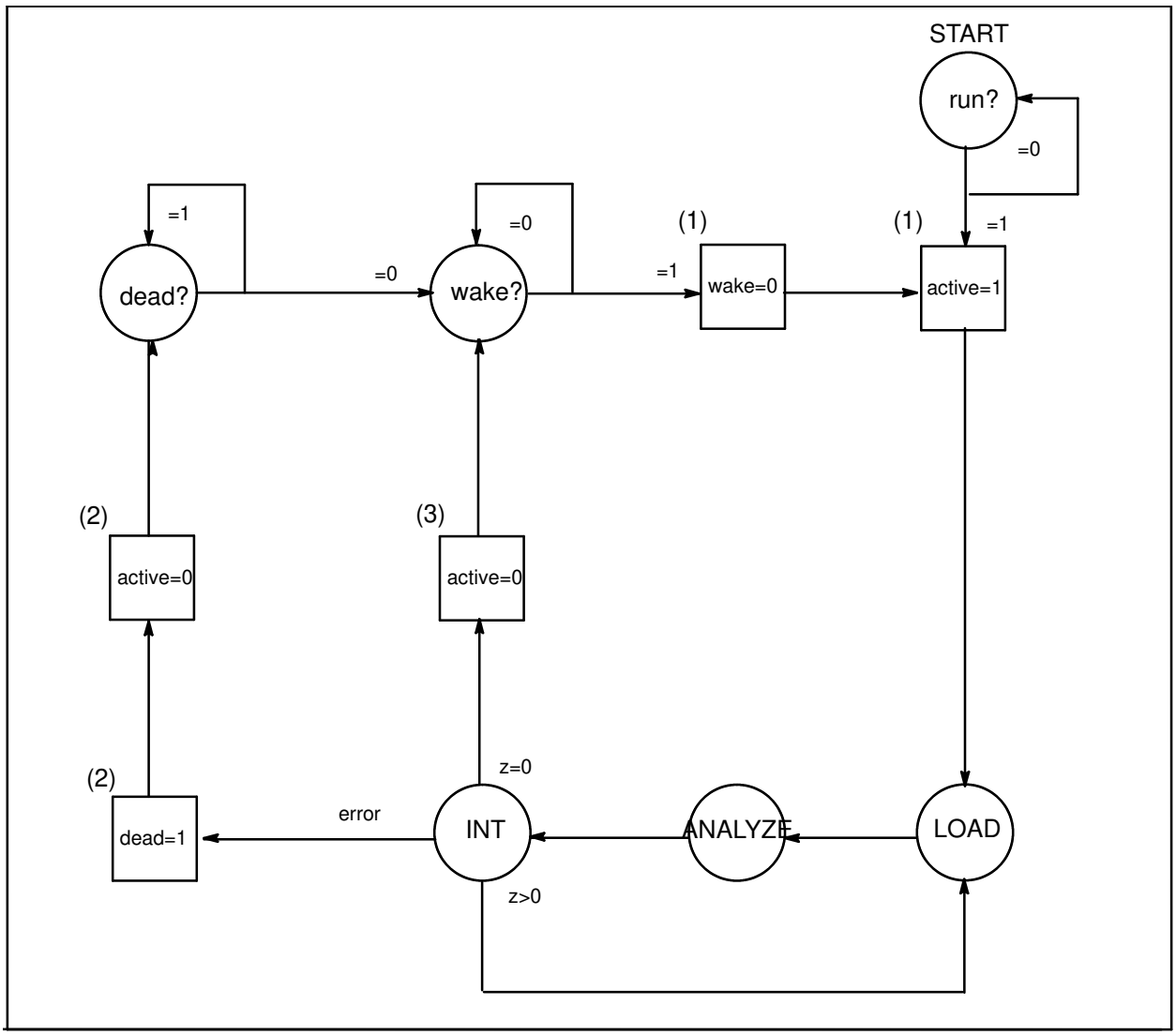


Fig. 4.2 Flag Transition for ContextControl Registers

4.1. Asynchronous Transmit

4.1.1. Program Analysis

Context program is analyzed by the following procedures:

P1 . . . . Move the packet header byte that was stored in the lower 16- byte area for the first descriptor as specified in the descriptor's reqCount area from the work RAM to the AT- FIFO. Then, if the descriptor is the OUTPUT_LAST_Immediate command, go to the interrupt handling after making sure that data transfer has completed. If the descriptor is the OUTPUT_MORE command, go to the procedure P2.

P2 . . . . Move the block data from the host memory in which the address is set in the next descriptor's dataAddress to the AT- FIFO. Then, if the descriptor is the OUTPUT_LAST command, go to the interrupt handling after making sure that data transfer has completed. If the descriptor is the OUTPUT_MORE command, repeat the procedure P2.

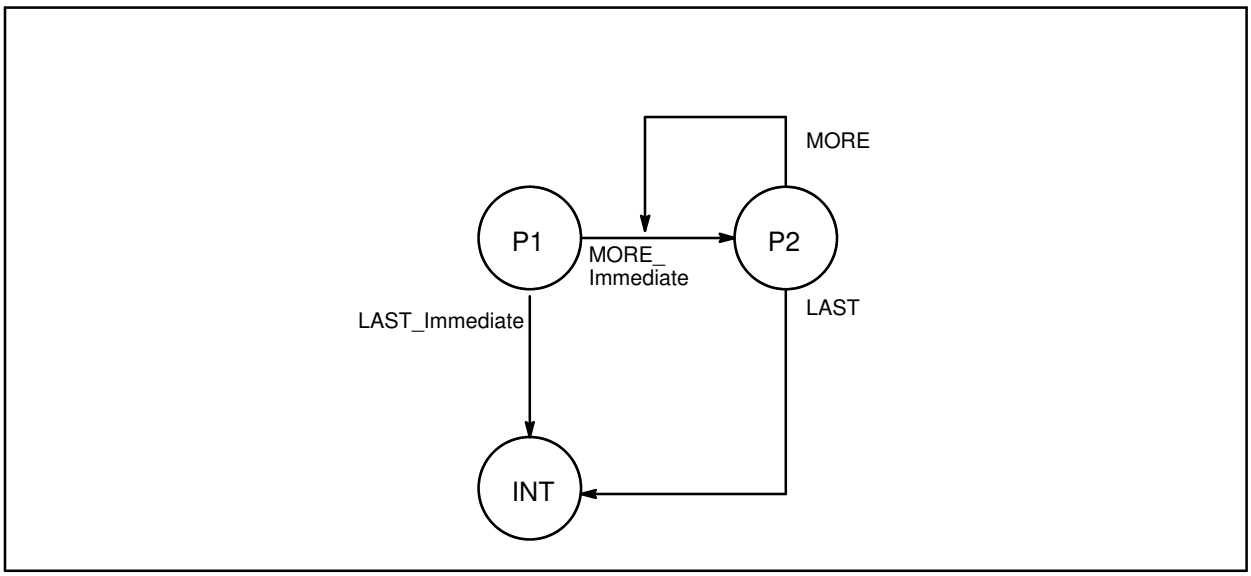


Fig. 4.3 State Machine of Program Analysis for Asynchronous Packet Transmit

4.1.2. Interrupt Handle

There are a number of interrupts possibly occur in the asynchronous transmit. The following lists the error name, code, and the description:

(1) evt_descriptor_read : 06h

when a PCI bus error occurs while the context program moves from the host memory into the work RAM.

(2) evt_unknown : 0Eh

when the context program can not be processed because of some problem with it.

(3) evt_data_read : 07h

when a PCI bus error occurs while the packet data moves from the host memory into the FIFO.

(4) evt_data_write : 08h

when a PCI bus error occurs while the data is put into the xferStatus and timeStamp sections in the descriptor.

(5) evt_tcode_err : 0Bh

when the tcode for the transmit packet is undefined or unknown. (this case, no packet is transmitted.)

(6) evt_timeout : 0Ah

when the time specified in the timeStamp section exceeds the cycle- timer value for the response packet. (this case, no packet is transmitted.)

(7) evt_underrun : 04h

when the FIFO becomes empty while transmitting the packet. (or when the PCI bus error occurs while transmitting the packet.)

(8) evt_flushed : 0Fh

when the bus reset occurs while transmitting the packet.

(9) evt_missing_ack : 03h

when a sub action gap is detected without receiving the acknowledge code for the transmitted packet.

(10) ack_tardy : 1Bh

when the Link or host system that has to receive the packet does not operate.

(11) ack_busy_X : 14h
ack_busy_A : 15h
ack_busy_B : 16h

when the destination's FIFO or host memory is full.

(12) ack_type_error : 1Eh

when the block- write- request packet that exceeds the maximum payload count is transmitted.

(13) ack_data_error : 1Dh

when a dataCRC error or data length error occurs on the transmitted packet.

(14) ack_pending : 12h

when a packet is transmitted normally and the response packet is expected from the packet receiver.

(15) ack_complete : 11h

when a packet is completely transmitted.

The following describes and shows the state machine of interrupt handling:

I1 . . . . Set the interrupt code in the ATContextControl.eventcode field and then further set the interrupt code and cycle timer value in the xferStatus and timeStamp sections in the last descriptor.
After completing these processes, go to the procedure I3 if the context program completed normally. If it didn't normally, go to the procedure I2.

I2 . . . . Clear the FIFO contents. Then, check the " i " field of the last descriptor. If the " i " contains "11b", set the IntEvent.unrecoverableError bit. After that, store the start address of host memory where the error descriptor is contained in the CommandPtr.descriptorAddress field and then return to the Start.

I3 . . . . If the ' i ' field in the last descriptor indicates '11b' or '01b', set the IntEvent.reqTxComplete or .respTxComplete bit. After that, if Z field in the last descriptor contains "0", then return to the Start. If "1" or more, then store the address set in the branchAddress field of last descriptor in the CommandPtr.descriptorAddress field and go to the PLU.

The lower 13- bit of timestamp to be stored in the descriptor is indicated in the IsoCycleTimer.cycleCount and the upper 3- bit is in the lower 3- bit of cycleSeconds. Meaning of the timeStamp varies with the type of packet to be transmitted as follows:

1) request packet Transmit:

Time that packet- transmit is completed is specified in the timeStamp section in the OUTPUT_LAST or OUTPUT_LAST_Immediate command by the cycle- timer value.

2) response packet Transmit:

Time specified in the timeStamp section in the OUTPUT_MORE_Immediate or OUTPUT_LAST_Immediate command is compared with the cycle- timer value to validate the evt_timeout.

3) ping packet Transmit:

When the ' p' flag in the OUTPUT_LAST or OUTPUT_LAST_Immediate command is set, time after the packet is transmitted until the acknowledge or self- ID packet is received is stored in the timeStamp section by the cycle- timer value.

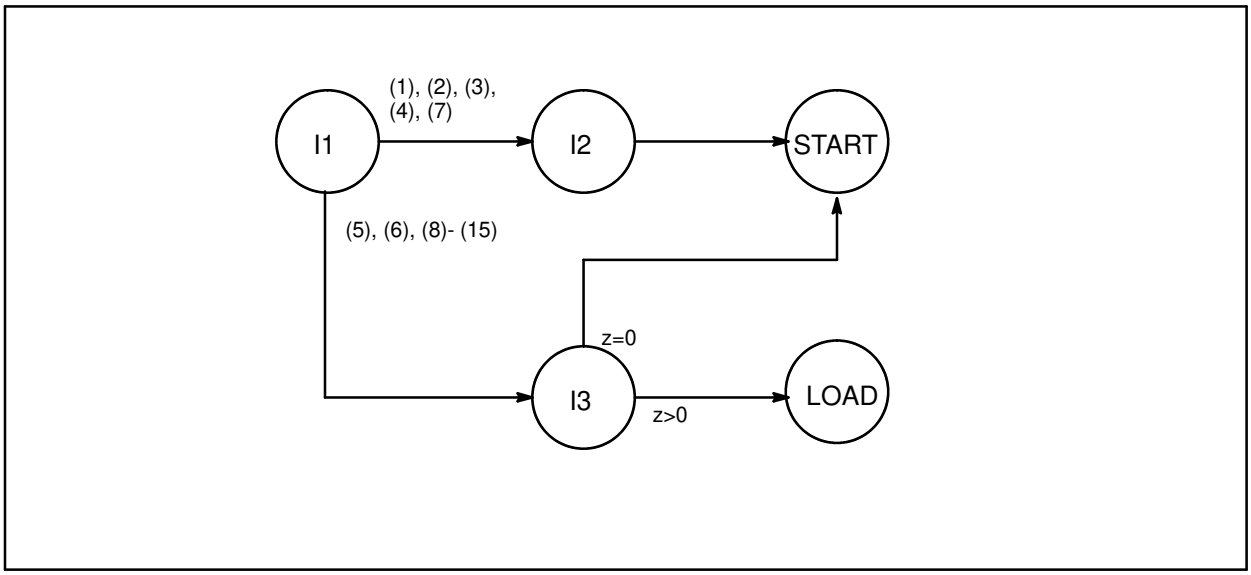


Fig. 4.4 State Machine of Interrupt Handle for Asynchronous Packet Transmit

4.1.3. Packet Format

Figures 4- 5 to 4- 15 show various packets' format when stored in the FIFO from the host memory or work RAM. Link- Tx block converts format of these packets shown from Open HCI to 1394 and transmits the packets onto the 1394 bus.

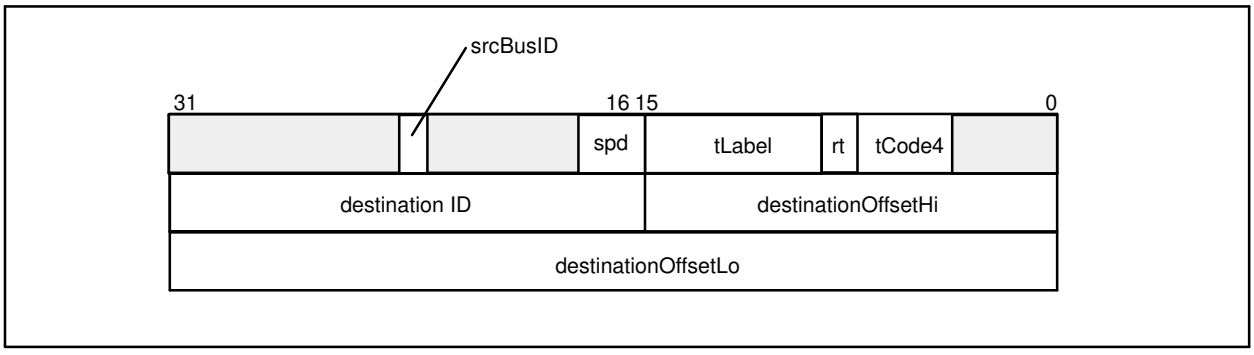


Fig. 4.5 Quadlet Read Request Transmit Packet

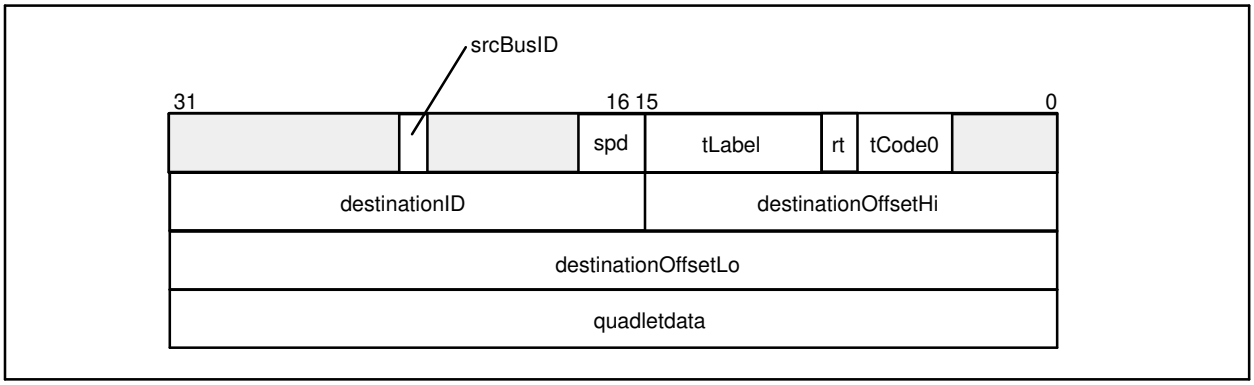


Fig. 4.6 Quadlet Write Request Transmit Packet

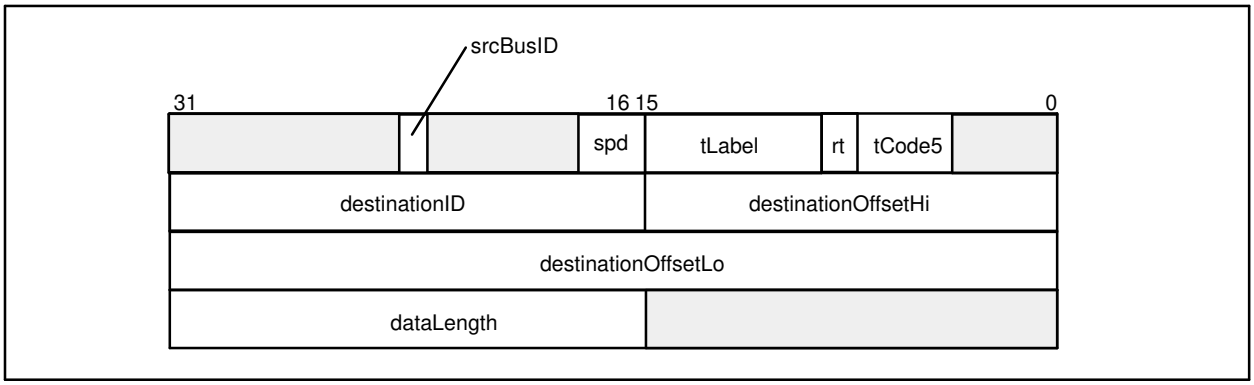


Fig. 4.7 Block Read Request Transmit Packet

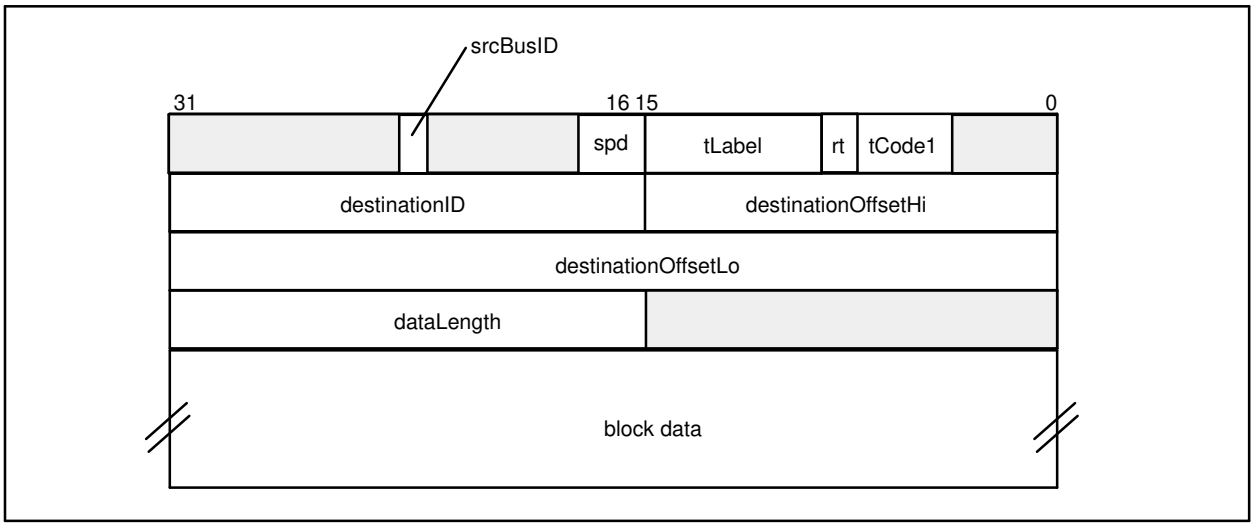


Fig. 4.8 Write Request Transmit Packet

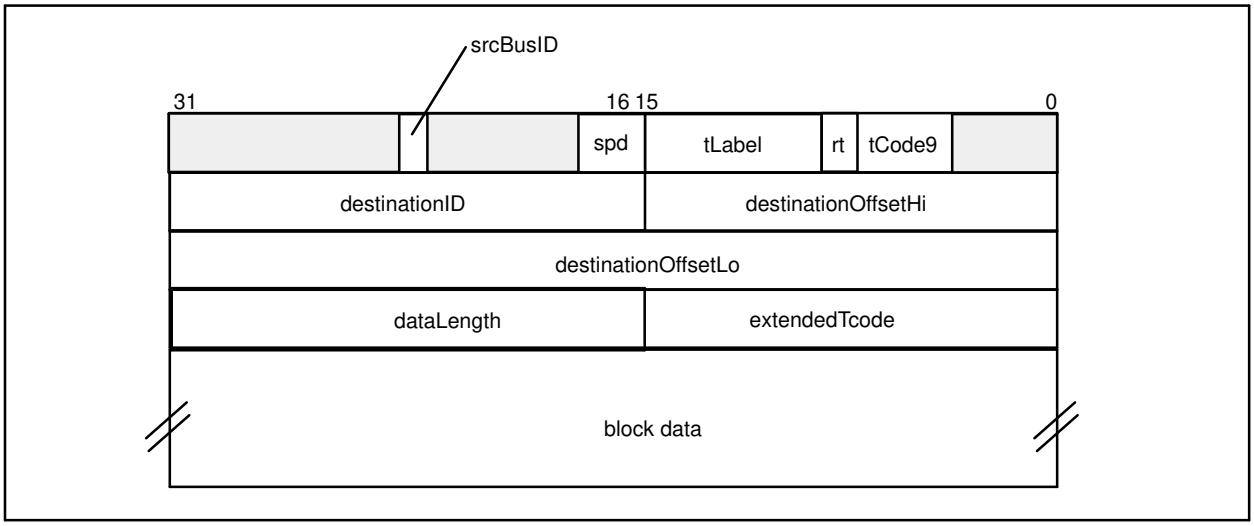


Fig. 4.9 Lock Request Transmit Packet

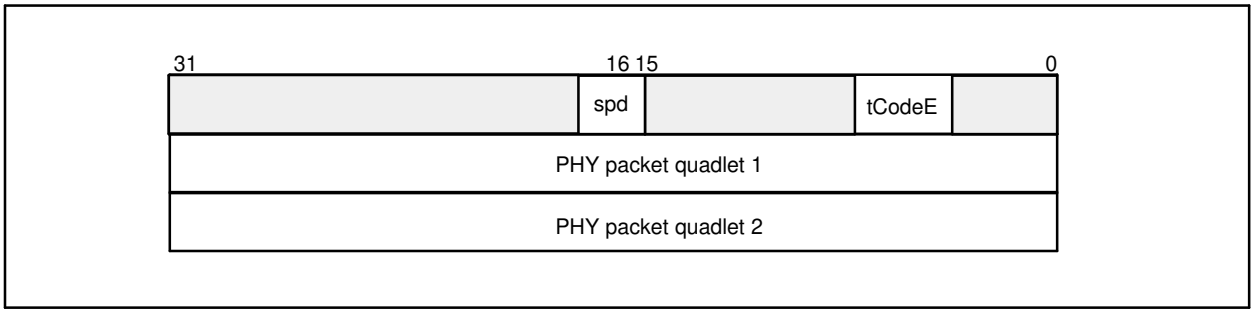


Fig. 4.10 PHY Transmit Packet

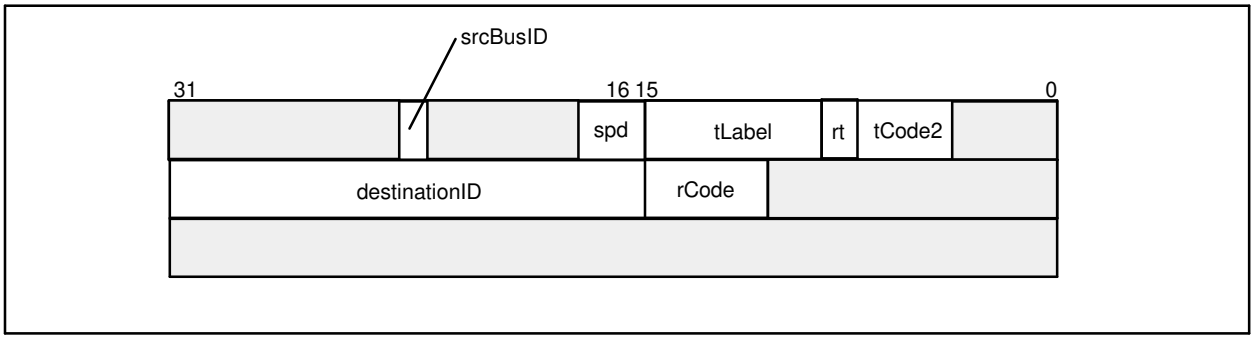


Fig. 4.11 Write Response Transmit Packet

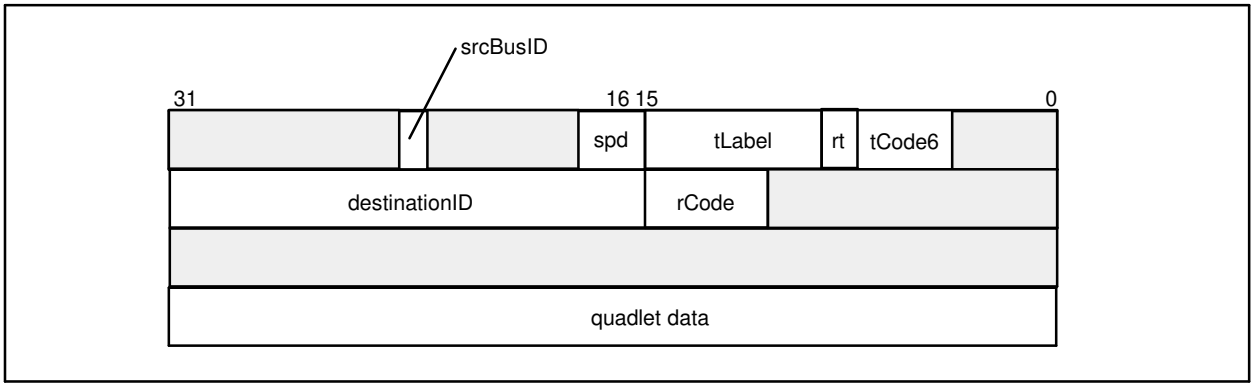


Fig. 4.12 Quadlet Read Response Transmit Packet

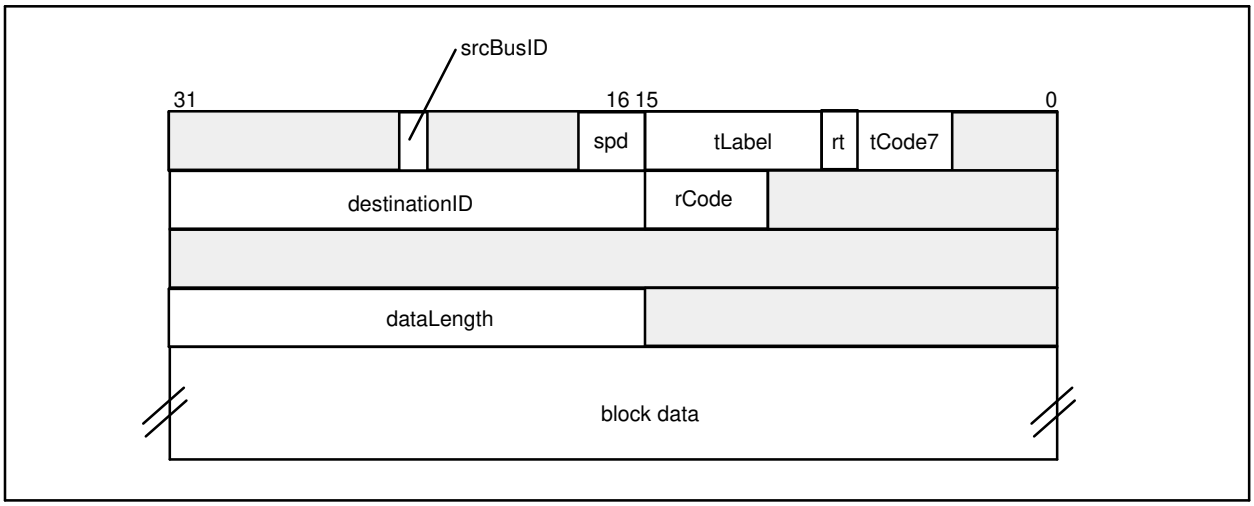


Fig. 4.13 Block Read Response Transmit Packet

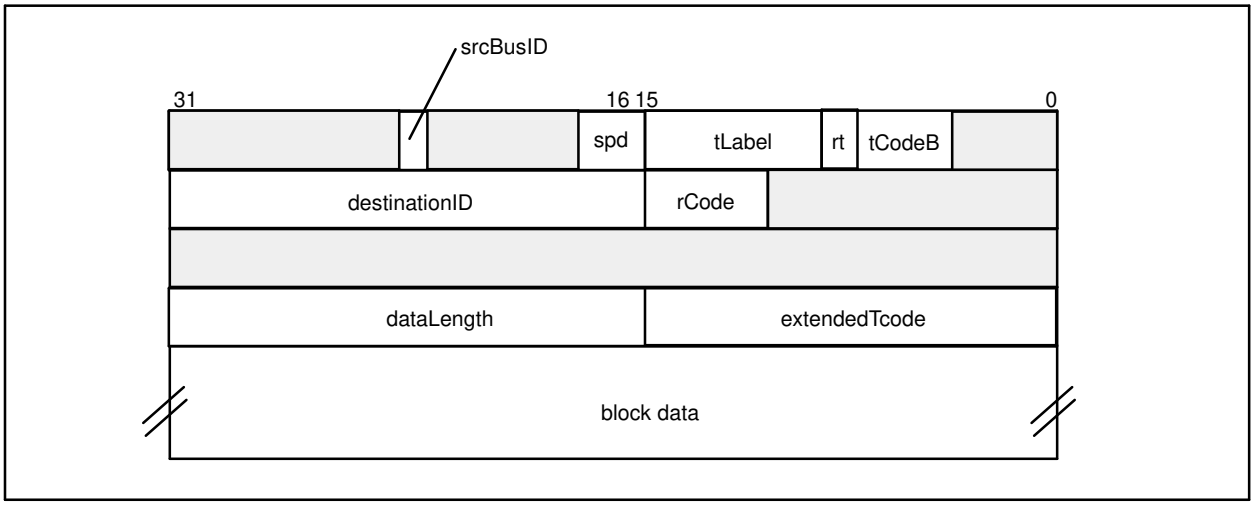


Fig. 4.14 Lock Response Transmit Packet

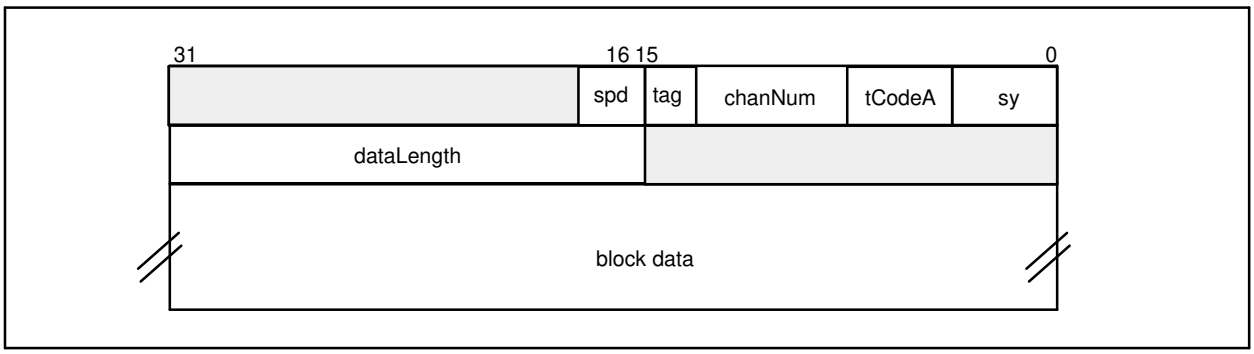


Fig. 4.15 Asynchronous Stream Packet

Bit	Field Name	description
1- bit	srcBusID	”0h” :Set ”3FFh” for the bus number in source ID field. ”1h” : Set the value specified in NodeID.busNumber field for the bus number.
3- bit	spd	Set the transfer speed: ”000b” : 100Mbps ”001b” : 200Mbps ”010b” : 400Mbps
6- bit	tLabel	Set the tLabel (transaction label).
2- bit	rt	Set the rt (retry) code. However, the chip does not support the dual- phase retry. So, always set ”01b” (retryX) in this field.
4- bit	tCode	Set the tcode (transaction code).
16- bit	destinationID	Set the bus number in the upper 10- bit field and set the node number to transmit the packet in the lower 6- bit field.
48- bit	destinationOffset	Set the address to transmit the packet.
16- bit	dataLength	Set the byte count of block data section for the transmit packet.
16- bit	extendedTcode	Set the code that clarifies the lock request/response packet format.
4- bit	rCode	Set the rcode (response code).
2- bit	tag	Set the tag code that clarifies the format of transmit isochronous data.
6- bit	chanNum	Set the channel number of transmit isochronous data.
4- bit	sy	Set the sync bit.

4.2. Asynchronous Receive

4.2.1. Program Analysis

Context program is analyzed by the following procedures:

P1 . . . . Store the received packet from the address specified in the descriptor’s dataAddress field to the host memory in order. If the packet is the last data to be stored, go to the procedure P2. When the host memory has no space to store any more packet(s) while packets are stored in it (resCount=0), go to the interrupt handling process.

P2 . . . . Set the interrupt event code and remained byte count of host memory in the descriptor's xferStatus and resCount fields. Then, set the IntEvent.RQPkt or RSPkt.
After the process is completed, go to the interrupt handling process if no host memory space is available. If the memory has the space, repeat the procedure P1.

The way to store the packet follows buffer- fill mode only. For details, see section 4.2.3 Buffer- Fill Mode.

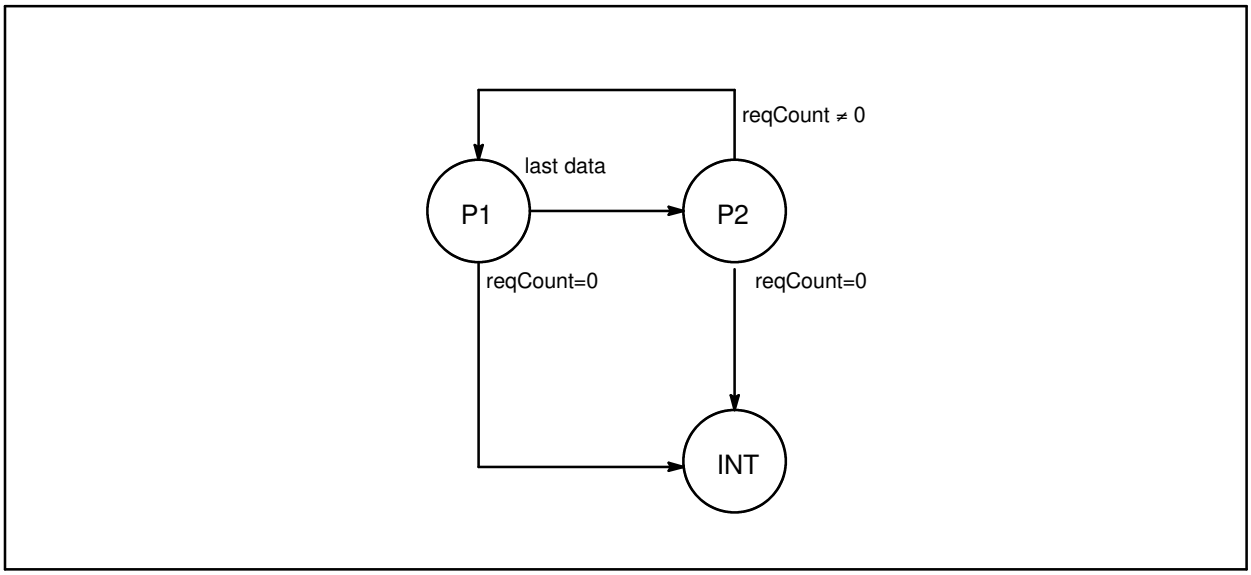


Fig. 4.16 State Machine of Program Analysis for Asynchronous Packet Receive

4.2.2. Interrupt Handle

There are a number of interrupts possibly occur in the asynchronous receive. The following lists the error name, code, and the condition:

Also, it is assumed that interrupt reported every packet received is handled in statemachine P2 in Fig. 4.16.

(1) evt_no_status : 00h

when receiving PHY packet.

(2) evt_descriptor_read : 06h

when a PCI bus error occurs while the context program moves from the host memory into the work RAM.

(3) evt_unknown : 0Eh

when the descriptor in the context program can not be processed due to some error or problem.

(4) evt_data_write : 08h

when a PCI bus error occurs while the packet data is written from AR- FIFO to host memory.

(5) evt_bus_reset : 09h

when the bus reset occurs and the bus reset packet generated is stored in AR- FIFO.

(6) ack_data_error : 1Dh

when a dataCRC error or data length error occurs on the received packet.

(7) ack_type_error : 1Eh

when the block- write- request packet that exceeds the maximum payload count is received. or, when the tcode in the received packet is undefined.

(8) ack_pending : 12h

when a packet is received normally and the response packet needs to be transmitted.

(9) ack_complete : 11h

when a packet is completely received.

The following describes and shows the state machine of interrupt handling:

I1 . . . . Set the interrupt event code in the ContextControl.eventcode field and then further set the interrupt code and remained byte count of the host memory in the xferStatus and resCount fields in the last descriptor. After completing these processes, go to the procedure I3 if the context program has been processed correctly. If it could not be processed correctly then go to the procedure I2.

I2 . . . . Set the IntEvent.unrecoverableError bit if the ' i ' flag in the last descriptor indicates '11b'. Then store the start address of host memory where the error descriptor is contained in the CommandPtr.descriptorAddress field, and return to Start.

I3 . . . . If the ' i ' flag in the last descriptor indicates '11b', set the IntEvent.ARRQ or ARRS bit. After that, if "0" is indicated in the Z field in the last descriptor, return to the START. If "1" is indicated, then store the address set in the branchAddress field in the CommandPtr.descriptorAddress of the last descriptor. After the process completed, go to the program loading process.

The MB86613S device has a function that automatically processes for error when it occurs. This is called "back- out" process. Back- out process removes packets from the host memory when an error is found on the packet such as data length error and data CRC error. This function allows the device to store only the correct packets in the memory, that results in having the proper acknowledge such as ack_complete and ack_pending The following describes some cases where this back- out process is taken:

1) FIFO was full while receiving packets.
2) data_length_error or data_CRC_error occurred on receive packets.
3) Bus reset occurred while receiving packets.

Also, in case where the above error occurs at the packets that are stored over two host memory(s), like the packet- 2 shown in Figure 4.18, the descriptor control is returned to the descriptor- 1's.

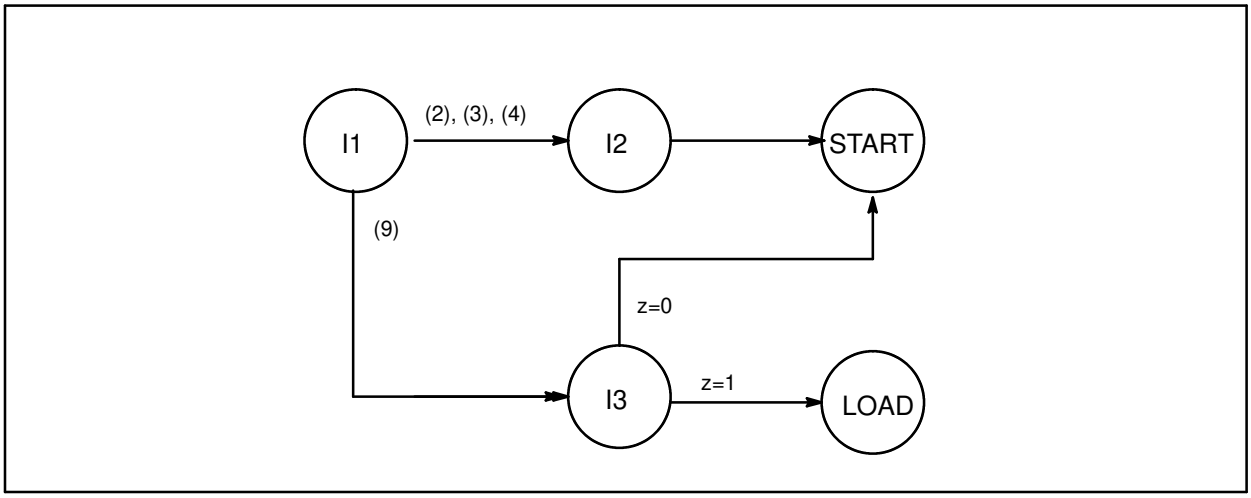


Fig. 4.17 State Machine of Interrupt Handle for Asynchronous Packet Receive

4.2.3. Buffer- Fill Mode

This function is used when storing the received asynchronous packet into the host memory. The buffer- fill mode is to store multiple packets into one memory in sequence. This mode allows to use the host memory effectively although the start point of each packet can be seen from the data_length value only. This mode requires a context program per 1 host memory. Also, the context program uses the INPUT_MORE command. Figure 4.18 shows a reference example of buffer- fill mode. In this figure, the first packet "packet- 1" is stored in the host memory- 1 according to the descriptor- 1. In this case, AR- CPC stores the remaining data byte count of host memory- 1 into resCount field of descriptor- 1. Then, the second packet "packet- 2" is stored in the host memory- 1, but since the host memory- 1 has no space to completely contain the packet- 2, the control ownership of context program goes to the descriptor- 2 and the rest of data of packet- 2 that can not be stored goes into the host memory- 2. At this point, because the host memory- 1 is full already, AR- CPC also stores "0" in the resCount field of descriptor- 1. One packet shall not be in three or more memory(s).

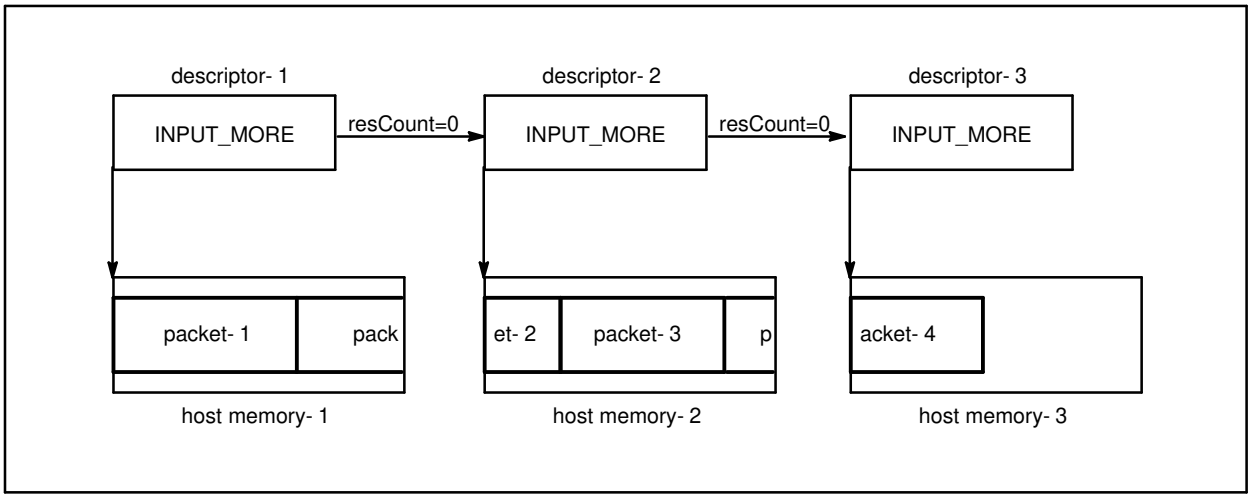


Fig. 4.18 Buffer- Fill Mode (Example)

4.2.4. Packet Format

Figures 4- 19 to 4- 28 show various packets' format, that are received and stored in host memory. Link- Rx block converts format of these packets shown from 1394 to Open HCI and stores the packets into AR- FIFO.

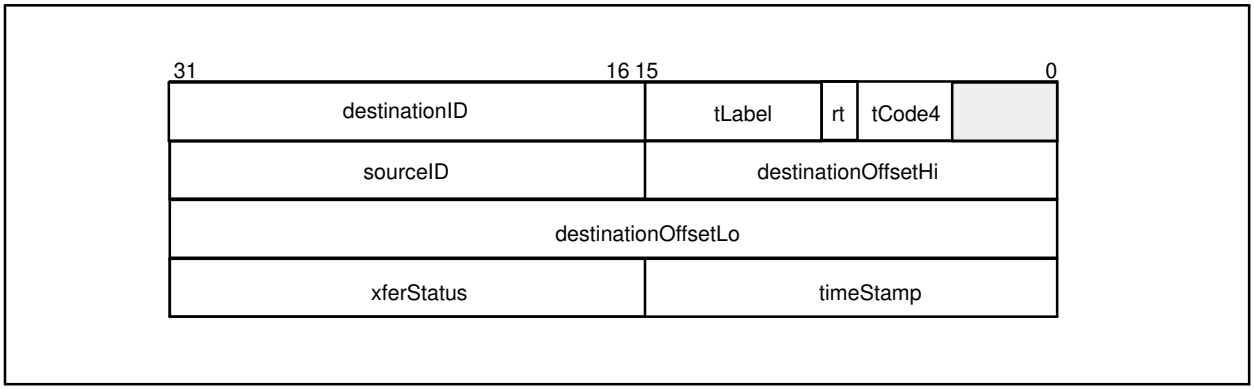


Fig. 4.19 Quadlet Read Request Receive Packet

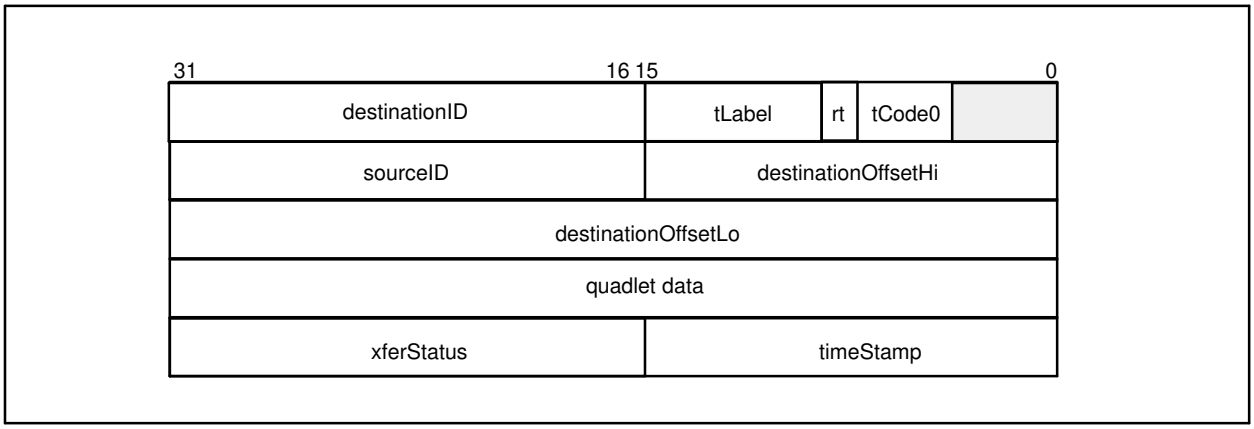


Fig. 4.20 Quadlet Write Request Receive Packet

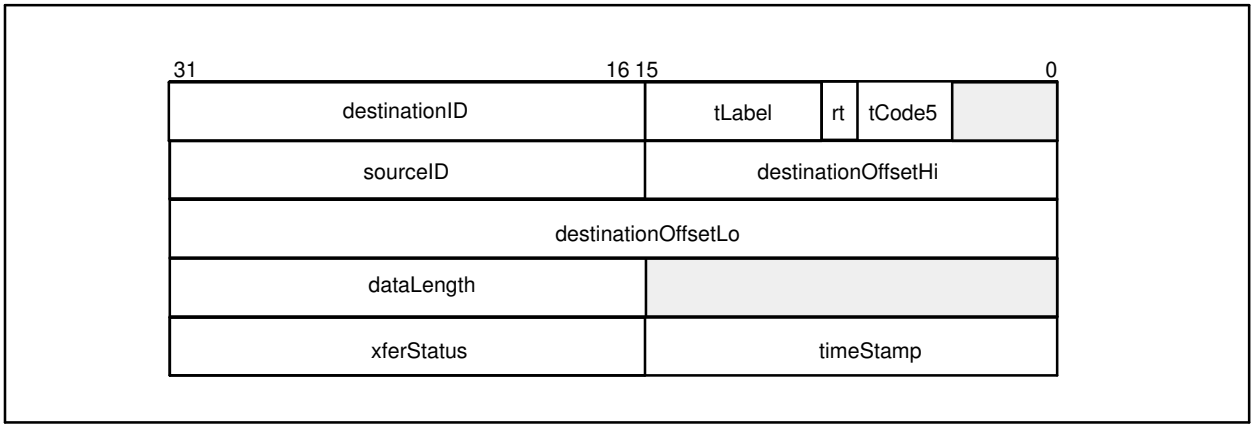


Fig. 4.21 Block Read Request Receive Packet

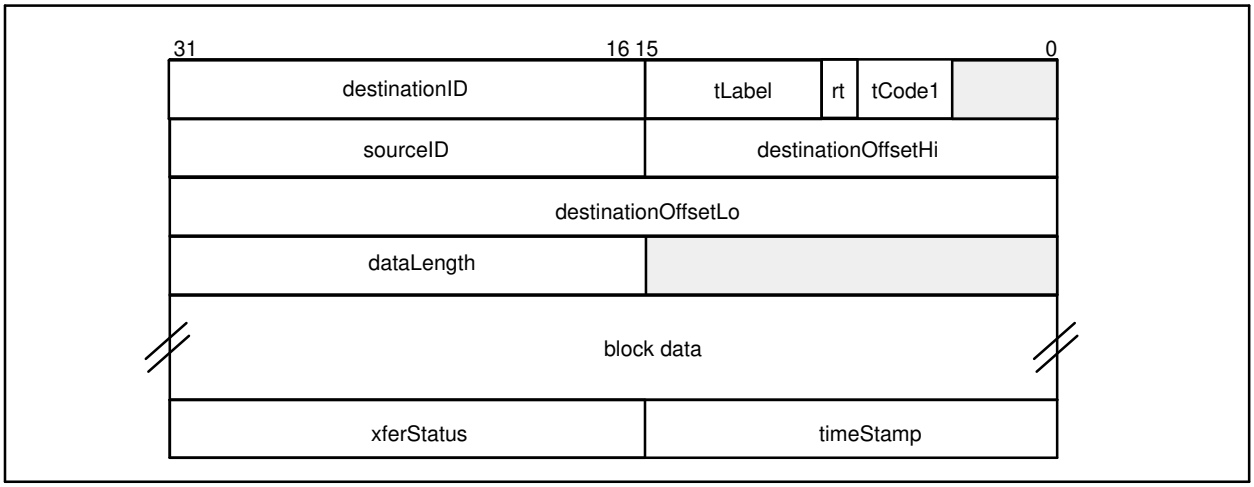


Fig. 4.22 Block Write Request Receive Packet

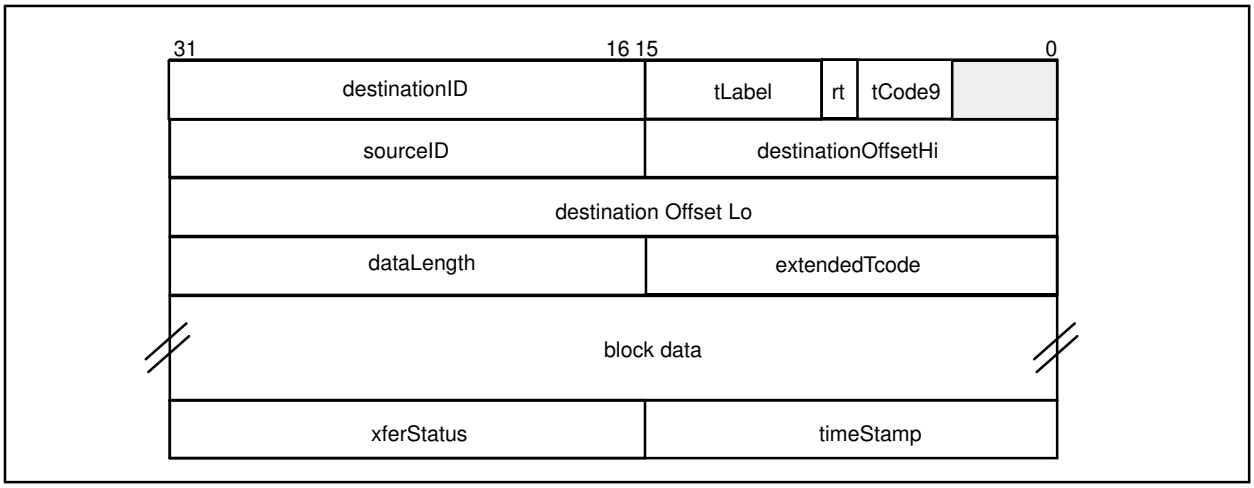


Fig. 4.23 Lock Request Receive Packet

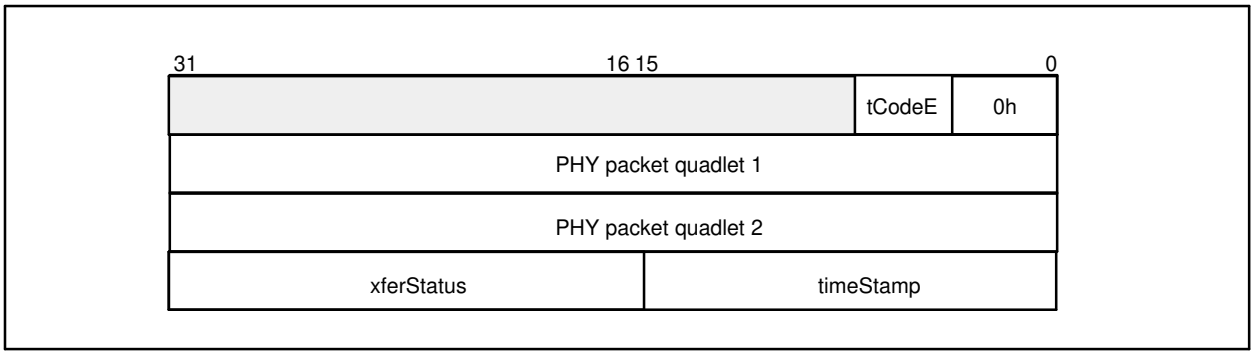


Fig. 4.24 PHY Receive Packet

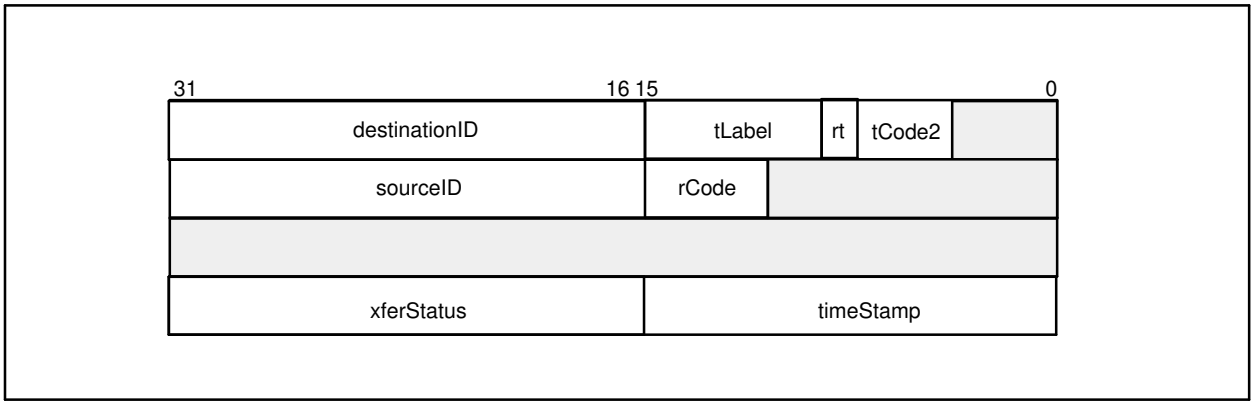


Fig. 4.25 Write Response Receive Packet

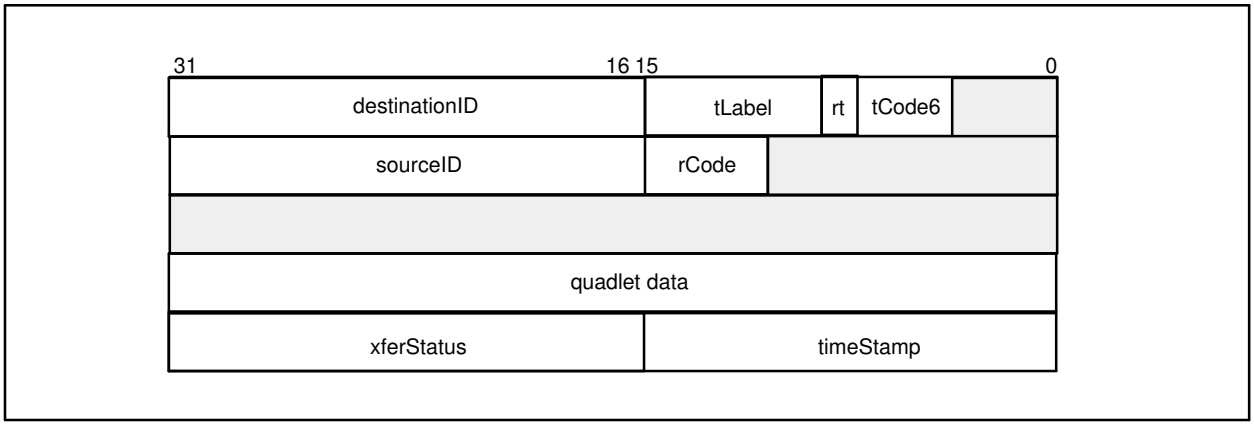


Fig. 4.26 Quadlet Read Response Receive Packet

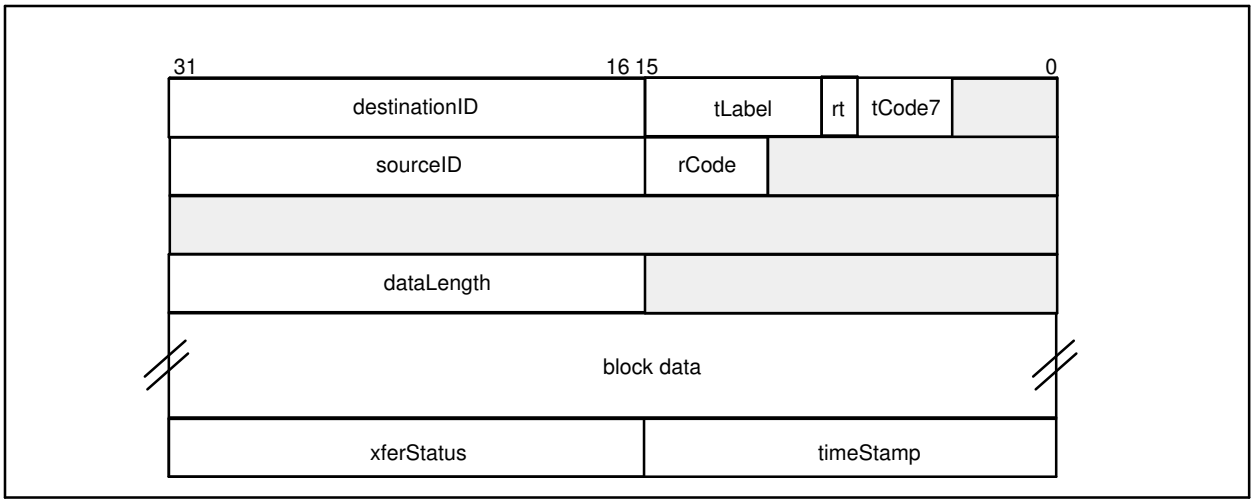


Fig. 4.27 Block Read Response Receive Packet

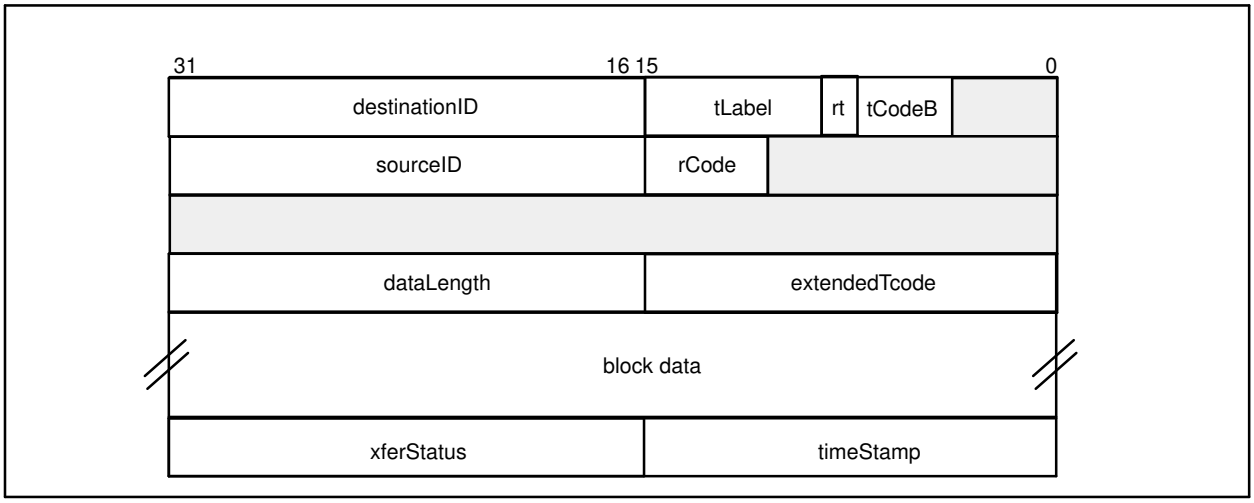


Fig. 4.28 Lock Response Receive Packet

Bit	Field Name	description
16- bit	destinationID	Indicates the bus number in the upper 10- bit field and set the node number of the destination node in the lower 6- bit field.
6- bit	tLabel	Indicates the tLabel (transaction label).
2- bit	rt	Indicates the rt (retry) code as follows: 00b : retry1 01b : retryX 10b : retryA 11b : retryB
4- bit	tCode	Indicates the tCode (transaction code).
16- bit	sourceID	Indicates the bus number in the upper 10- bit field and set the node number of the source node in the lower 6- bit field.
48- bit	destinationOffset	Indicates the address to transmit the packet.
4- bit	rCode	Indicates the rcode (response code).
16- bit	dataLength	Indicates the byte count of block data section for the receive packet.
16- bit	extendedTcode	Indicates the code that clarifies the lock request/response packet format.
16- bit	xferStatus	Indicates the result of packet receive with ARcontextControl register format.
16- bit	timeStamp	Indicates the time when the packet was received with the IsoCycleTimer.cycleCount value and cycle timer value in the lower 3- bit of cycleSecond field.

4.3. Isochronous Transmit

4.3.1. Program Analysis

Context program is analyzed by the following procedures:

P1 . . . . If the first descriptor is STORE_VALUE command, store the 32- bit data adding ”0000h” to the upper case of the value specified in the storeDoublet section into the address specified in dataAddress. If the first descriptor is OUTPUT_MORE_Immediate or _LAST_Immediate command, move the packet header stored in the lower 16- byte of descriptor from the work RAM to the IT- FIFO as specified in the reqCount.

After the processes completed, if that descriptor is STORE_VALUE command, go to the next descriptor process. if it is OUTPUT_LAST_Immediate command, go to the interrupt handling process, and if it is OUTPUT_MORE_Immediate command, go to the procedure P2.

P2 . . . . Move the block data from the host memory where the address is specified in the next descriptor’s dataAddress section to IT- FIFO.
After the processes completed, if that descriptor is OUTPUT_LAST command, go to the interrupt handling process. If it is OUTPUT_MORE command, repeat the P2.

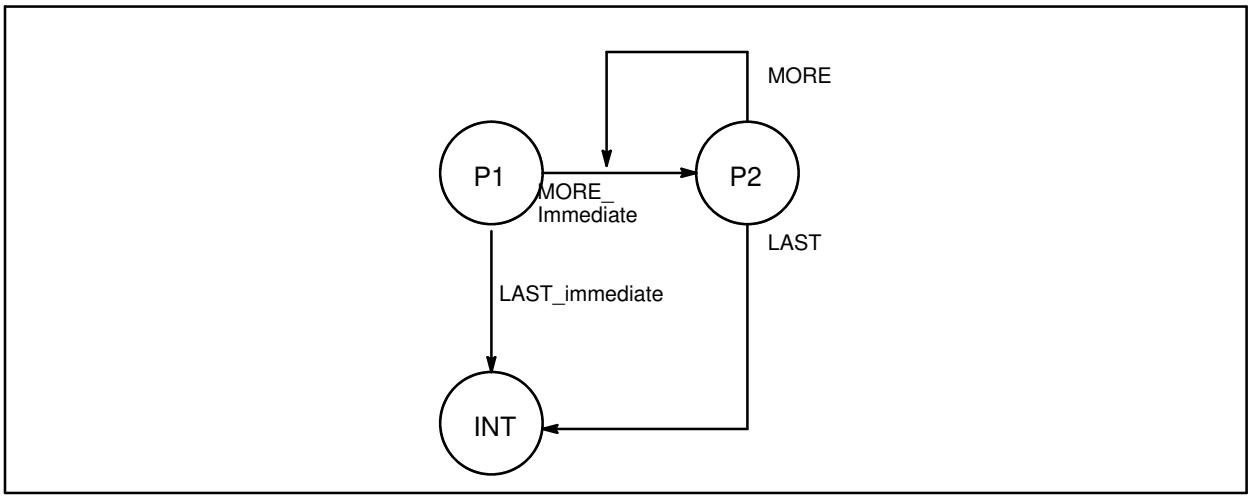


Fig. 4.29 State Machine of Program Analysis for Isochronous Packet Transmit

4.3.2. Interrupt Handle

There are a number of interrupts possibly occur in the isochronous transmit. The following lists the error name and the condition:

(1) evt_descriptor_read : 06h

when a PCI bus error occurs while the context program moves from the host memory into the work RAM.

(2) evt_unknown : 0Eh

when a context program has some problem and it cannot be processed. Or when the device stops the context program process because a long bus reset or cycle lost period is encountered.

(3) evt_data_read : 07h

when a PCI bus error occurs while the packet data moves from the host memory into the IT- FIFO.

(4) evt_data_write : 08h

when a PCI bus error occurs while the data is put into the xferStatus and timeStamp sections in the descriptor, or when a PCI bus error occurs while the STORE_VALUE command is executed.

(5) evt_tcode_err : 0Bh

when the tcode for the transmit packet is not Ah. (This case, no packet is transmitted.)

(6) evt_underrun : 04h

when the IT- FIFO becomes empty while transmitting the packet. (or when the PCI bus error occurs while transmitting the packet.)

(7) ack_complete : 11h

when a packet is completely transmitted.

The following describes and shows the state machine of interrupt handling:

I1 . . . . Set the interrupt event code in the ITContextControl.eventcode field and then further set the event codeand cycle timer value in the xferStatus and timeStamp sections in the last descriptor if the ' s' flag in thelast descriptor is "1b". After completing these processes, go to the procedure I3 if the context programprocessed normally. If the process is not done normally, go to the procedure I2.

I2 . . . . Clear the FIFO contents and set the IntEvent.unrecoverableError bit if the ' i ' flag in the last descriptor indicates '11b'. Then, return to the START after writing the start address of host memory where the erroneous descriptor is stored to the CommandPtr.descriptorAddress.

I3 . . . . If the ' i ' flag in the last descriptor indicates '11b' or '01b', set the IntEvent.isoXmitN field. After that, if 'Z' field indicates "0" in the last descriptor then go to the START. If it indicates "1" or more, then store the address set in the branchAddress field of the last descriptor in the CommandPtr. descriptorAddress field and go to the program loading process.

If the packet can not be transmitted due to an error occurred during the program process, the context program which is stored in the host memory where the address is specified in the first descriptor's skipAddress is executed. This is related with Cycle Loss function noted in section 4.3.3.

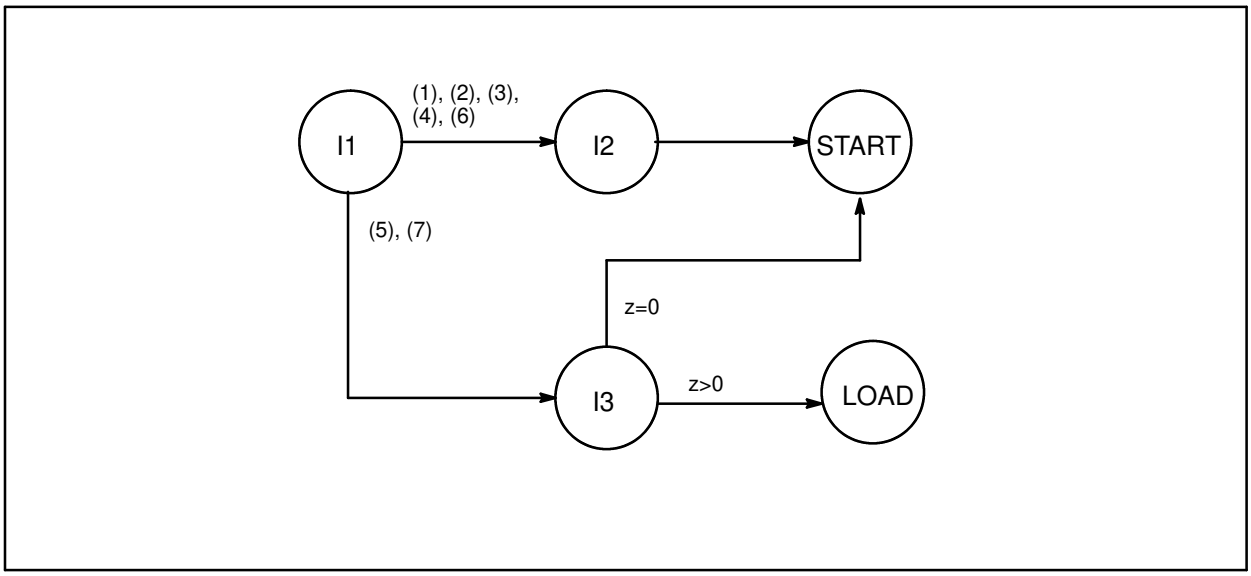


Fig. 4.30 State Machine of Interrupt Handle for Isochronous Packet Transmit

4.3.3. Cycle Loss Function

When isochronous packets cannot be transmitted due to a bus reset or cycle lost generated, IT- CPC executes the context program stored in the address specified in skipAddress field of the first descriptor. The value to be specified in the skipAddress field selects how to transmit the packets as follows:

a) When transmitting the next packet (”A” packet) :
Store the start address of context program that handles the next packet in the skipAddress field.

As shown in an example in Figure 4.31, the chip skip the process to the context program that handles the packet A6 because a bus reset occurred after the completion of process for the packet A4 in cycle 4. However, the bus reset is continuing in cycle 5 and so the chip goes to the process to the context program for A6. Then, the chip processes the packet A6 and stores it in IT- FIFO.

b) When transmitting the same packet (”B” packet) :
Store the start address of context program currently in- process in the skipAddress field.

As shown in an example in Figure 4.31, a bus reset occurred after the completion of process for the packet B4 in cycle 4. However, the bus reset is still continuing in cycle 5 where the chip has to handle the packet B5. In this case, the chip still intends to process the packet B5 to skip the process to the context program for the packet B5. After that, in fact, the chip processes the packet B5 in cycle 6 and stores the packet in IT- FIFO.

c) When transmitting another packet (”C” packet) :
Store the start address of context program that processes another packet in the skipAddress field.

As shown in an example in Figure 4.31, a bus reset occurred after the completion of process for the packet C4 in cycle 4. However, the bus reset is still continuing in cycle 5 and so, the chip skips to the process of context program for Cx. Then, it processes the packet Cx in cycle 6, stores the packet in IT- FIFO, and terminates the process after transmitting the Cx packet onto 1394 bus in cycle 8.

d) When terminating the process (”D” packet) :
Set ”0h” in Z field of skipAddress.

As shown in an example in Figure 4.31, a bus reset occurred after the completion of process for the packet D4 in cycle 4. However, the bus reset is still continuing in cycle 5. Therefore, the chip terminates the process after transmitting the D4 packet onto 1394 bus in cycle 7.

When an error is detected while transmitting a packet, the chip discards the remained packet in the FIFO and processes the context program stored in the specified address in skipAddress field.

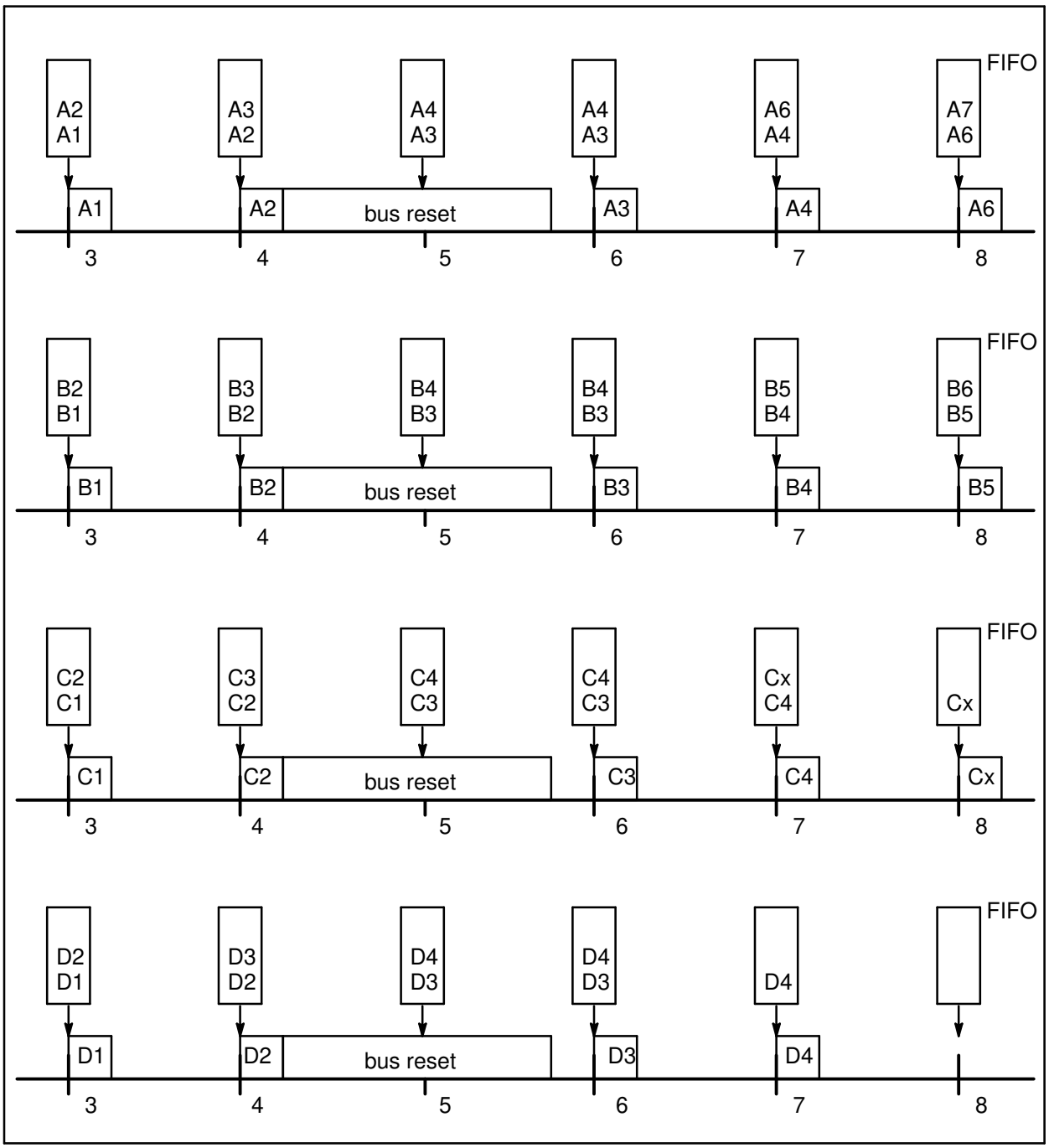


Fig. 4.31 Example of Cycle Loss Function

4.3.4. Packet Format

Figure 4.32 shows the isochronous transmit packet format, that is to be transmitted from the work memory or host memory to the FIFO. Link- Tx block converts format of these packets shown from Open HCI to 1394 and transmits it onto 1394 bus.

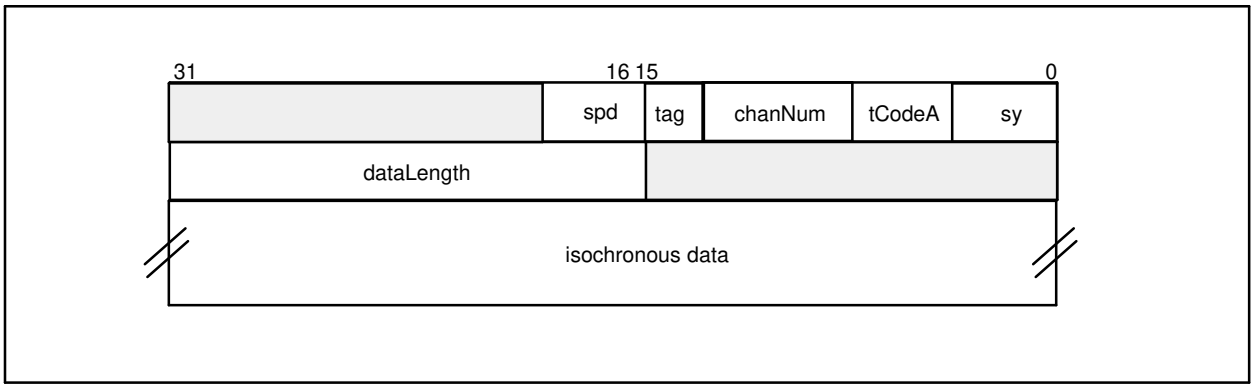


Fig. 4.32 Isochronous Transmit Packet Format

Bit	Field Name	description
3- bit	spd	Set the packet transfer speed: 000b : 100Mbps 001b : 200Mbps 010b : 400Mbps
2- bit	tag	Set the tag code that clarifies the isochronous data format.
6- bit	chanNum	Set the channel of data to be transmitted.
4- bit	tCode	Set the tcode (transaction code).
4- bit	sy	Set the sync bit code.
16- bit	dataLength	Set the byte count of block data section for the transmit packet.

4.4. Isochronous Receive

4.4.1. Program Analysis

Context program is analyzed by the following procedures:

For buffer- fill mode:

P1 . . . . Store the received packet in the host memory that dataAddress field of descriptor indicates the address. If no space is available in the host memory (resCount=0), go to the interrupt handling process.

For packet- per- buffer mode:

P1 . . . . Store the received packet in the host memory that dataAddress field of descriptor indicates the address. If the stored packet is the last one, go to the interrupt handling process. If no space is available in the host memory (resCount=0), go to the procedure P2.

P2 . . . . If the descriptor is INPUT_LAST command, then go to the interrupt handling process and if INPUT_MORE command then return to the procedure P1 again.

The MB86613S chip supports the buffer- fill mode and packet- per- buffer mode. For details, see sections 4.2.3 and 4.4.3 respectively.

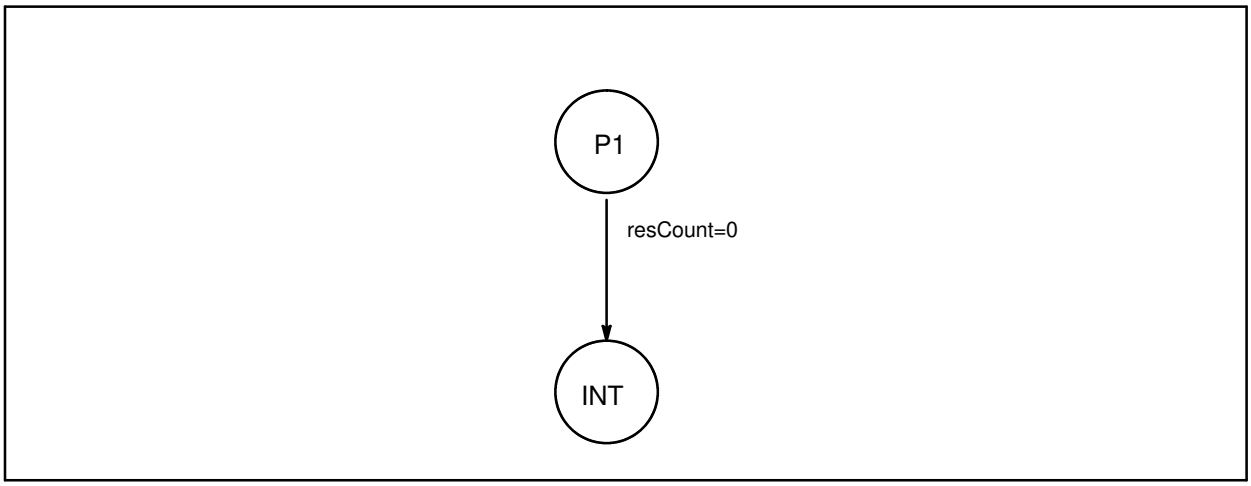


Fig. 4.33 State Machine of Program Analysis for isochronous packet Receive (buffer- fill Mode)

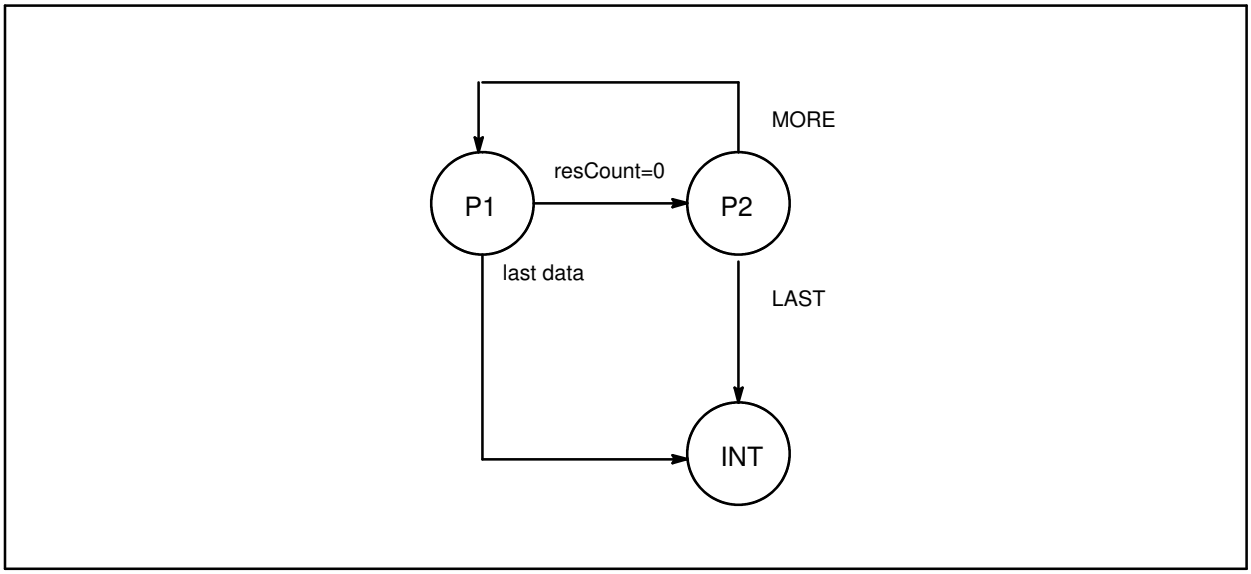


Fig. 4.34 State Machine of Program Analysis for isochronous packet Receive (packet- per- buffer Mode)

4.4.2. Interrupt Handle

There are a number of interrupts possibly occur in the isochronous receive. The following lists the error name and the condition:

(1) evt_descriptor_read : 06h

when a PCI bus error occurs while the context program moves from the host memory into the work RAM.

(2) evt_unknown : 0Eh

when the context program cannot be processed due to an error found. or, when IRContextControl.bufferFill is cleared or IRContextControl.isochHeader is cleared in multi- channel mode .

(3) evt_data_write : 08h

when a PCI bus error occurs while the packet data is written from IR- FIFO to host memory.

(4) evt_overrun : 05h

when the IR- FIFO becomes full while receiving the packet. (PCI bus error occurs while receiving the packet.)

(5) ack_data_error : 1Dh

when a dataCRC error or data length error occurs on the received packet.

(6) evt_long_packet : 02h

In packet- per- buffer mode, when the INPUT_LAST command is completed before storing the last packet.

(8) ack_complete : 11h

when a packet is completely received.

The following describes and shows the state machine of interrupt handling:

I1 . . . . Set the interrupt event code in the IRContextControl.eventcode field. In buffer- fill mode, set the interrupt code and remained byte count of the host memory in the xferStatus and resCount fields in the descriptor. In packet- per- buffer mode, set the interrupt code and remained byte count of the host memory in the xferStatus and resCount fields in the last descriptor if ' s ' flag of the descriptor on which the context program process is done is set.

After completing these processes, go to the procedure I3 if the context program process is completed normally. If it is not the normal completion, go to the procedure I2.

I2 . . . . Set the IntEvent.unrecoverableError bit if the ' i ' flag in the last descriptor indicates '11b'. In this case, return to the START after storing the start address of host memory where the erroneous descriptor is contained in the CommandPtr.descriptorAddress field.

I3 . . . . If the ' i ' flag in the last descriptor indicates '11b', set the isoRecvIntEvent.isoRecvN. Then, if 'Z' field in the last descriptor indicates "0", return to the START and if it indicates "1" or more, go to the program loading process after storing the address specified in the branchAddress field in the last descriptor in the CommandPtr.descriptorAddress field.

The MB86613S chip has a function to automatically handle error if occurred. For details, see section 4.2.2.

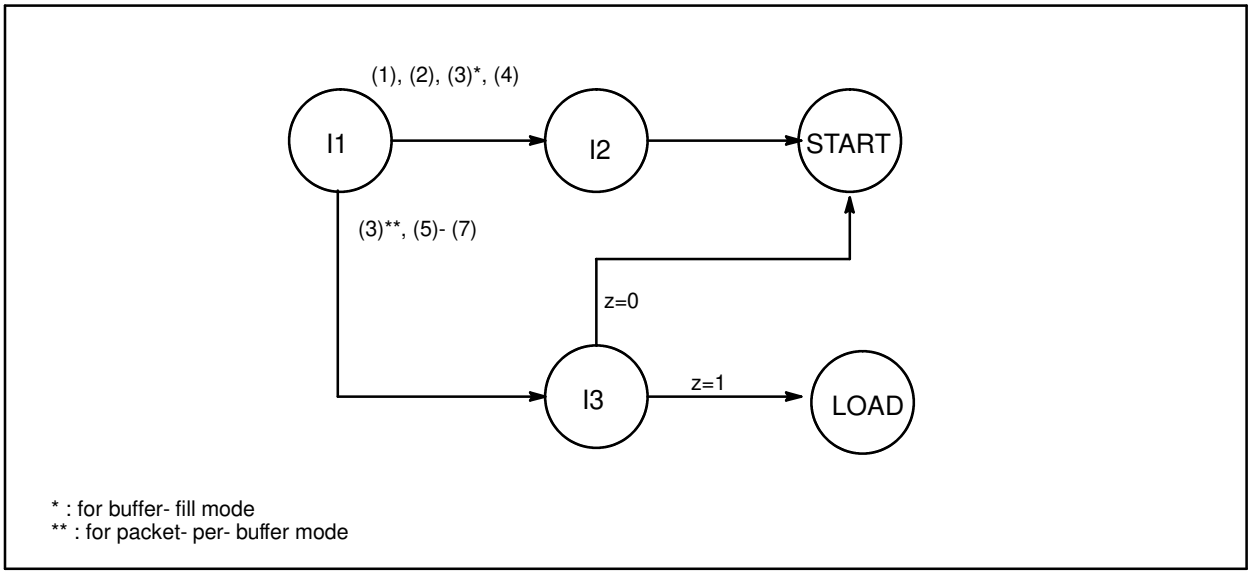


Fig. 4.35 State Machine of Interrupt Handle for Isochronous Packet Receive

4.4.3. Packet- Per- Buffer Mode

Packet- per- buffer mode is used when IRContextControl.bufferFill is cleared for storing a received isochronous packet into the host memory. The vehicle of this mode is that the chip stores the packets to allocate single or multiple space per packet. Even though this mode can not use the host memory efficiently, it makes easy to handle each packet individually.

In this mode, a context program is prepared for one packet, and the context program uses INPUT_MORE and INPUT_LAST commands together. Figure 4.36 shows a reference example of how this mode is used. In that Figure, the first packet, packet- 1, is stored in the host memory- 1 and - 2 according to the descriptor block- 1. In this case, when all the packet data cannot be stored in the host memory- 1, it needs to move to INPUT_LAST command process and to store the rest of data in the host memory- 2. Like the case of packet- 2, if an INPUT_MORE command can completely store the whole packet data in the host memory- 3, the next INPUT_LAST command is not executed and the process moves to the context program that handles the next

packet, packet- 3.

IR- CRC stores the result of process in the xferStatus and resCount fields of descriptor in which the context program has completed the process. This means, the result is stored in the INPUT_LAST command in case of packet- 1 and in the INPUT_MORE command in case of packet- 2.

When a packet size is larger than the size of host memory prepared by one context program, this packet will not completely be stored in the memory and evt_long_packet is reported.

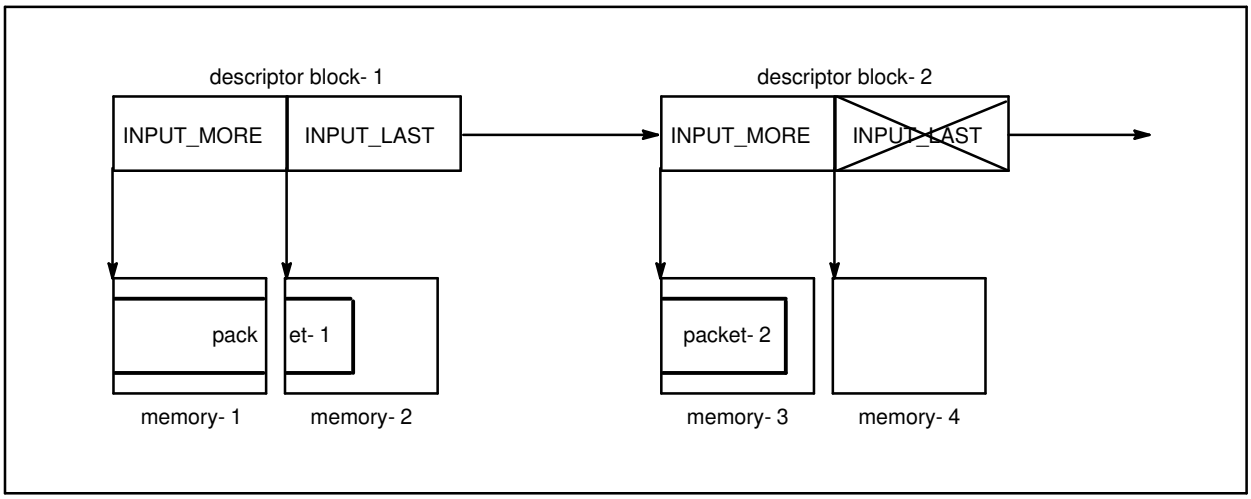


Fig. 4.36 Packet- Per- Buffer Mode (Example)

4.4.4. Packet Format

Figures 4.37 to 4.40 show the isochronous receive packet format, that is to be received and stored in the host memory. Link- Rx block converts format of these packets shown from 1394 to Open- HCI and stores it into IR- FIFO. When storing the received packet into the host memory after removing the packet header and trailer data (i.e., when IRContextControl.isoHeader bit is cleared), the alignment process is taken as described in section 7.1.

Also, like Figure 4.39, if both packer header and trailer data are stored in the memory in packet- per- buffer mode (i.e., IRContextControl.isochHeader bit is set), the alignment process is done only when the packet data and packet header are stored in different memory locations.

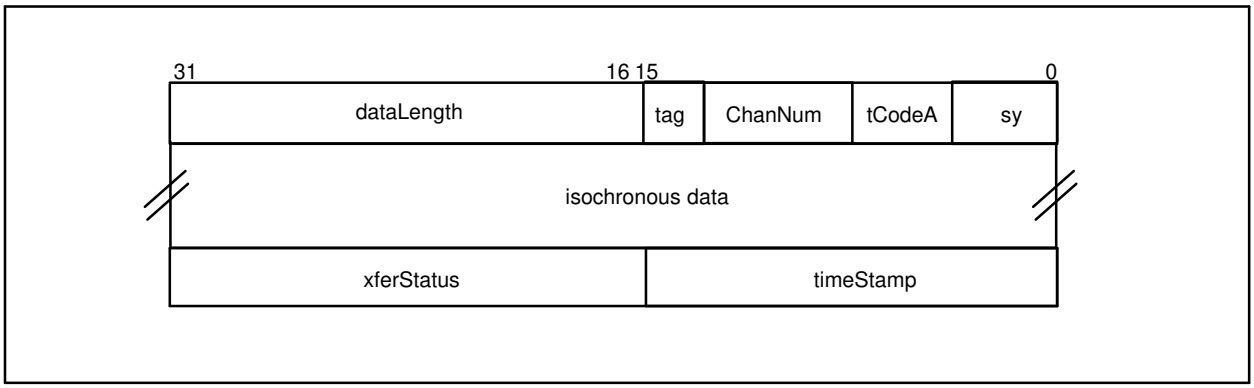


Fig. 4.37 isochronous receive packet Format (buffer- fill Mode with header/trailer)

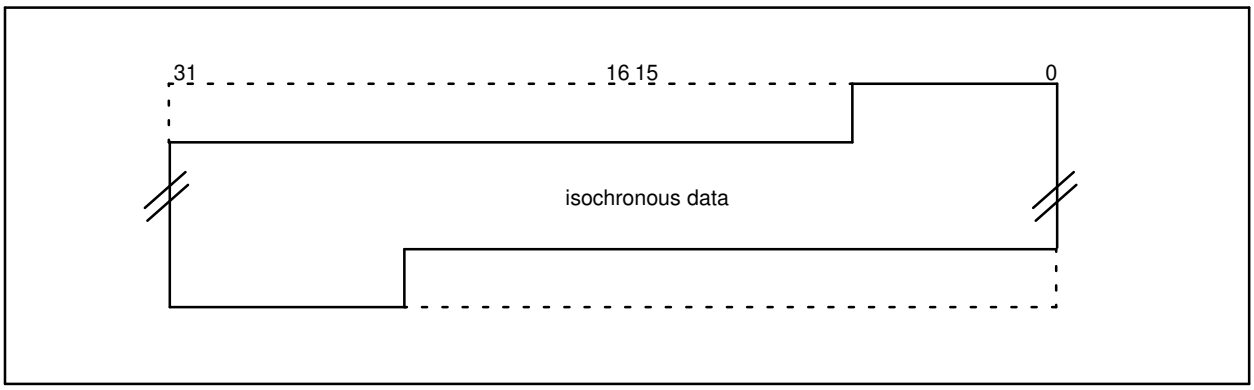


Fig. 4.38 isochronous receive packet Format (buffer- fill Mode without Header/trailer)

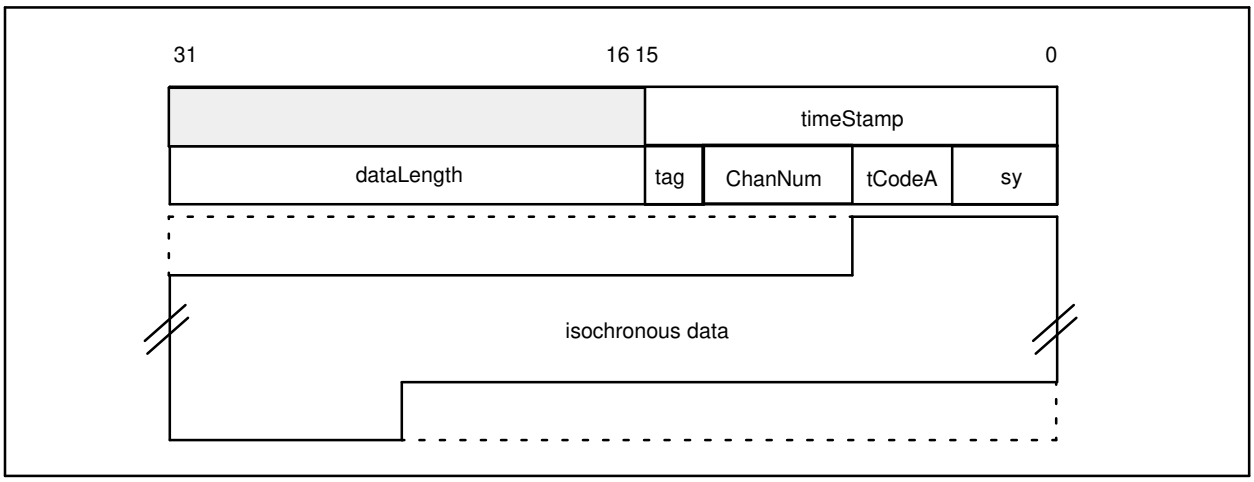


Fig. 4.39 isochronous receive packet Format (Packet- Per- Buffer Mode with header/trailer)

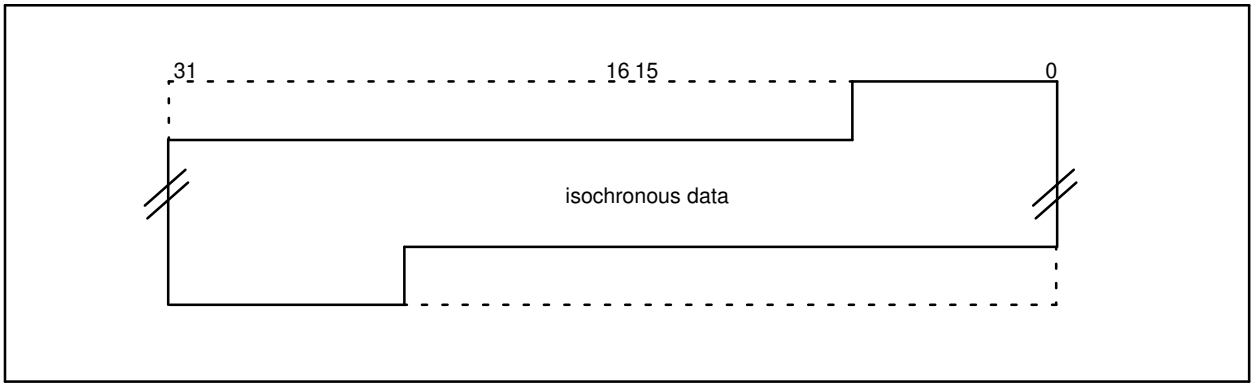


Fig. 4.40 isochronous receive packet Format (Packet- Per- Buffer Mode without Header/trailer)

Bit	Field Name	description
16- bit	dataLength	Indicates the byte count in the isochronous data field of the received packet.
2- bit	tag	Indicates the tag code that clarifies the received isochronous data format.

6- bit	chanNum	Indicates the channel of isochronous data received.
4- bit	tCode	Indicates the tcode (transaction code).
4- bit	sy	Indicates the sync bit code.

4.5. Physical Request

When the chip receives a request packet for the address listed below, it is handled as a physical request packet. This packet is not stored in the host memory and the AR- CPC and AT- CPC will take the following actions automatically:

(1) physical memory : 0000_0000_0000h - PhysicalUpperBound register value (32- bit)+0000h

a) quadlet read request:

The host memory address is determined as the lower 32- bit of offset address for the received packet. 1 quadlet data in the host memory indicated by that address is handled as the quadlet data for the quadlet read response (tcode=8h) and the response packet is transmitted.

b) block read request :

The host memory address is determined as the lower 32- bit of offset address for the received packet. Based on that address, some byte data as specified in the dataLength field are handled as the block data for the block read response (tcode=7h) and the response packet is transmitted.

c) quadlet write request :

The host memory address is determined as the lower 32- bit of offset address for the received packet. The quadlet data are stored in that address.

d) block write request :

The host memory address is determined as the lower 32- bit of offset address for the received packet. The block data are stored from that address in order.

(2) bus management CSR: FFFF_F000_021Ch - FFFF_F000_0228h

a) quadlet read request :

1 quadlet data of the on- chip bus management CSR is handled as the quadlet read response data (tcode=8h) and the response packet is transmitted.

b) lock request :

1 quadlet data of the on- chip bus management CSR is compared with the arg_value of the received packet and the result of comparison is handled as the lock response packet data (tcode=Bh) and the response packet is transmitted.

(3) configuration ROM : FFFF_F000_0400h - FFFF_F000_07FCh

a) quadlet read request :

1 quadlet data of the 1394 configuration ROM is handled as the quadlet read response data (tcode=8h) and the response packet is transmitted. However, when the offset address for the received packet is within FFFF_F000_0400 to _0410, the data in the on- chip ConfigROMHeader, BusID, BusOptions, GUIDHi, and GUIDLo registers are handled as the quadlet data and the response packet is transmitted. When the offset address is within FFFF_F000_0414 to _07FC, the address for the 1394 configuration ROM is calculated from the base address specified in the ConfigROMmapping register and the read out value is handled as the quadlet data and the response packet is transmitted.

5. LINK

5.1. Generation of Acknowledge Code

The following table lists the acknowledge codes automatically reported when the LINK receives an asynchronous packet. The acknowledge code is basically reported only when HCControl.linkEnable bit is set.

acknowledge name	code	description
ack_complete	01h	When and asynchronous response packet was received. When a physical write request packet was received at which HCControl.PostedWriteEnable bit was set and also PostedWrite Address register was not full.
ack_pending	02h	When a physical write request packet was received at which HCControl.PostedWriteEnable bit was cleared. When a physical read packet was received.
ack_busy_x	04h	When a packet was received while the FIFO was full. When a physical write request packet was received at which Posted WriteAddress register was full.
ack_data_error	0Dh	When a data length error occurred. When a data CRC error occurred.
ack_type_error	0Eh	When a tcode error occurred. When a byte count specified in the dataLength field of the received block write request packet was larger than a byte count specified in the max_rec field of the 1394 configuration ROM. When a request packet to the bus management CSR was not either quadlet lock request or quadlet read request packet. When a request packet to the 1394 configuration ROM was not the quadlet read request packet.
ack_tardy	0Bh	When a packet was received while the sytem power was off.

5.2. Open HCI - 1394 Packet Format Conversion

LINK- Tx transmits a packet in the FIFO converting its packet format from Open HCI format to 1394 format. LINK- Rx receives a 1394 formatted packet and stores in the FIFO after converting its format to Open HCI's. For details, please see sections 4.1.3, 4.2.4., 4.3.4, and 4.4.4.

(1) Packet Transmit

a) PHY packet :

- Identify the spd code from the 1st quadlet data.
- Transmit the 2nd and 3rd quadlet data.

b) asynchronous packet:

- Identify the spd code and srcBusID code from the 1st quadlet data.
- Replace the destination ID contained in the 2nd quadlet data with bit31- bit16 code of the 1st quadlet data.
- Set the sourceID which is composed of bus number and node number determined with the previously mentioned srcBusID in a field where the destinationID of 2nd quadlet data is located.

c) isochronous packet:

- Identify the spd code from the 1st quadlet data.
- Replace the dataLength value contained in the 2nd quadlet data with the bit31- bit16 code of the 1st quadlet data.

(2) Packet Receive :

a) PHY packet :

- Insert 1 quadlet data in which tcode ”E”h is set into the head of received packet.
- Insert 1 quadlet trailer data which consists of event code, xferStatus information, and timeStamp value that indicates the time information on which the packet is stored in the FIFO into the end of received packet.

b) asynchronous and isochronous packets :

- Insert 1 quadlet trailer data which consists of event code, xferStatus information, and timeStamp value that indicates the time information on which the packet is stored in the FIFO into the end of received packet.

c) self ID packet (only when LinkControl.rcvSelfID bit is set) :

- Insert 1 quadlet trailer data which consists of selfIDGeneration indicating the number of bus reset generation and timestamp value for the received packet into the packet.
- Store the self ID packet following the trailer data into the FIFO.

IR- CPC takes the responsibility to delete the isochronous packet header and trailer data from the received packet (only when IRContextControl.isochHeader bit is cleared), or to insert a trailer data into the head of received packet (only when packet- per- buffer mode clearing IRContextControl.bufferFill bit is activating).

5.3. Bus Reset

Upon a generation of bus reset, the chip performs the following processes:

(1) asynchronous packet transmit :

It stops the packet transmit and reports ’evt_flush’ event code. However, if no acknowledge was not able to receive because the bus reset has occurred after transmitting the packet, ’eve_ack_missing’ is reported instead.

It also clears the ATContextControl.active bit in order to complete the AT- CPC process clearing all the packets remained in the AT- FIFO.

(2) isochronous packet transmit :

Stops the packet transmit. If the next packet to be transmitted has already been stored in the IT- FIFO, the packet transmit is restarted after the completion of bus reset and receiving the cycle start packet.

(3) asynchronous and Isochronous packet receive :

Inserts a bus reset packet into the AR- FIFO as shown in Figure 5.1. This packet is handled as a kind of PHY packet and the eventcode field in the trailer data contains an event code 09h that is ’evt_bus_reset.

In case the bus reset occurs when the HCControl.linkEnable bit is cleared, the bus reset packet generated and the selfID packet received are not stored into the FIFO.

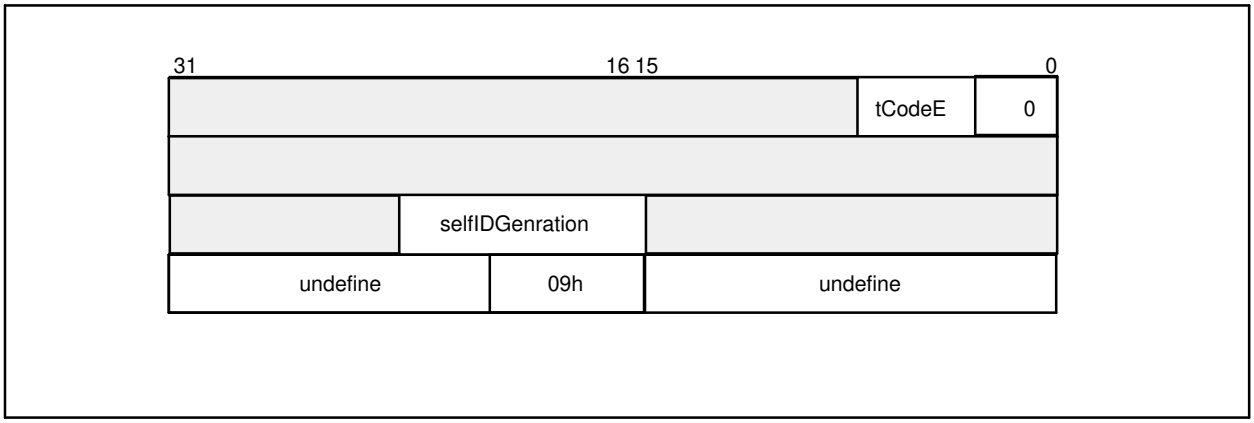


Fig. 5.1 Bus Reset Packet Format

5.4. Physical Configuration Packet Receive

When LINK receives a phy configuration packet, it performs several types of operation depending on the register settings as follows:

(1) If PacketControl.autoPhyPkt bit is cleared while LinkControl.rcvPhyPkt bit is set :

Stores the received phy configuration packet into the FIFO. Software must set the RHB bit and Gap_count field in the PHY register (address 0001b) from the received phy configuration packet as follows:

a) RHB : Set the RHB bit only when a value set in the root_ID field while R bit of phy configuration packet is set is equal to the chip node number. If not equal, clear the RHB bit.

b) Gap_count : Set a value specified in the gap_cnt field while T bit of phy configuration packet is set in the Gap_count field.

(2) If PacketControl.autoPhyPkt bit is set while LinkControl.rcvPhyPkt bit is cleared :

It does not store the received phy configuration packet in the FIFO. LINK analyzes the received phy configuration packet and updates the RHB bit of PHY register (address 0001b) and Gap_count field automatically.

(3) If HCControl.linkEnable bit is cleared:

It does not store the received phy configuration packet in the FIFO. LINK analyzes the received phy configuration packet and updates the RHB bit of PHY register (address 0001b) and Gap_count field automatically.

(4) If PacketControl.autoPhyPkt bit is cleared while LinkControl.rcvPhyPkt bit is cleared :

This condition/setting is prohibited.

Also, please notice that HCControl.linkEnable bit must be cleared if the system of MB86613S is powered off while another node is providing the MB86613S system with the power.

6. PHY

6.1. Support of P1394a Standard Draft

MB86613S device supports the following functions in P1394a standard:

1) arbitration accelerate :

While the enab_accel bit of PHY register is set, arbitration can be started to transmit its own packet after the minimum gap(MIN_IDLE_TIME) is detected.

2) fly by arbitration :

This packet is concatenated with its own packet to repeat the packet received at the child port. This method enables the packet transmission without arbitration.

3) arbitrated short bus reset :

While the normal bus reset period takes 167μs, the period of this bus reset takes 1.3μs. This bus reset can be executed by setting the ISBR bit of PHY register. PHY layer executes this bus reset automatically when one of the following events occurs :

a) when a port that exceeds the connection monitored time (time set in the Connection Timeout register) is detected.

b) when the port is disconnected from a child port.

c) when the bus reset is detected during data receive.

d) when the bus reset is detecting while monitoring the connection time (while counting the Connection Timeout register's counter).

4) PHY registers :

The chip contains the registers as listed in section 6.3.

5) connection hysteresis :

"connection" is detected only when a port connection that exceeds the time set in the ConnectionTimeout register is detected. However, "connection" is detected as soon as a bus reset that exceeds the time set in the BusResetDetect register even if the connection detect time is still within the connection timeout.

6) multi speed packet concatenate :

Multiple concatenating packets can be transmitted while enabling the enab_multi bit of PHY register. The transfer speed of each packet can also be changed. This means the speed signaling operation is performed for every packet to be transmitted. But, please note that you have to make sure not to concatenate a packet with s100 rate after a packet with either s200 or s400 in order to keep the func tional compatibility with IEEE1394- 1995 standard. For such a case, when a packet is transmitted in s100 rate, please transmit the s100 packet and then concatenate the s200 or s400 packet.

7) ping packet :

It transmits the selfID packet after receiving a ping packet.
When it transmits a ping packet, it starts to count the time (in 50MHz) until the selfID packet is received.
When an asynchronous packet is transmitted, it starts to count the time until the acknowledge is received.
The counted time is stored in the timeStamp field of descriptor while "p" flag of OUTPUT_LAST or

OUTPUT_LAST_Immediate command of context program processed by AT- CPC is set.

8) asynchronous stream packet :

It can transmit an asynchronous stream packet that has the same format as an isochronous packet during fairness cycle. The packet is transmitted by AT- CPC.

9) acknowledge :

ack_tardy will be newly supported.
It reports ack_tardy as acknowledge if a packet is received when the system power is OFF.
After that it asserts the PME#.
ack_conflict_error and ack_address_error are not supported.

10) port disable :

For disabling to provide a power to TpBias, set the Disabled bit of PHY register after selecting the port number to be disabled with Port_select field to set ”000b” in Page_select field of PHY register.
If ”1h” (transmit TX_DISABLE_NOTIFY then disable port) is set at cmnd field of this packet when receiving remote command packet, providing TpBias to the port set in port field will be stopped.
If ”5h” (enable port) is set at cmnd field of this packet, providing TpBias to the port set in port field will be started.
TpBias will not be provided if resume is done when Disable bit of PHY register is cleared.

11) priority budget register :

LINK layer executes the priority request as the number specified in ’pri_req’ field of FairnessControl register described in Section 3.20.
Software must support the priority budget register.

12) remote access/reply packet :

The contents of PHY register in other node can be known by transmitting remote access packet.
When receiving remote access packet, remote reply packet is transmitted automatically, which contains information for PHY register set in remote access packet.

13) remote command/confirmation packet :

Transmission of remote command packet executes the command set in cmnd field of this packet for other node.
When remote command packet is received, remote confirmation packet is transmitted after executing the command which is set in cmnd field for the port set in port field of remote command packet.

14) resume packet :

Transmission of resume packet starts resume for each port.
When receiving resume packet, resume for each port starts.

15) suspend/resume :

If ”2h” (initiate suspend) is set at cmnd field of this packet when receiving remote command packet, port connection set in port field will be suspended.
If ”6h” (resume port) is set at cmnd field of this packet when receiving remote command packet, port connection set in port field will be resumed.

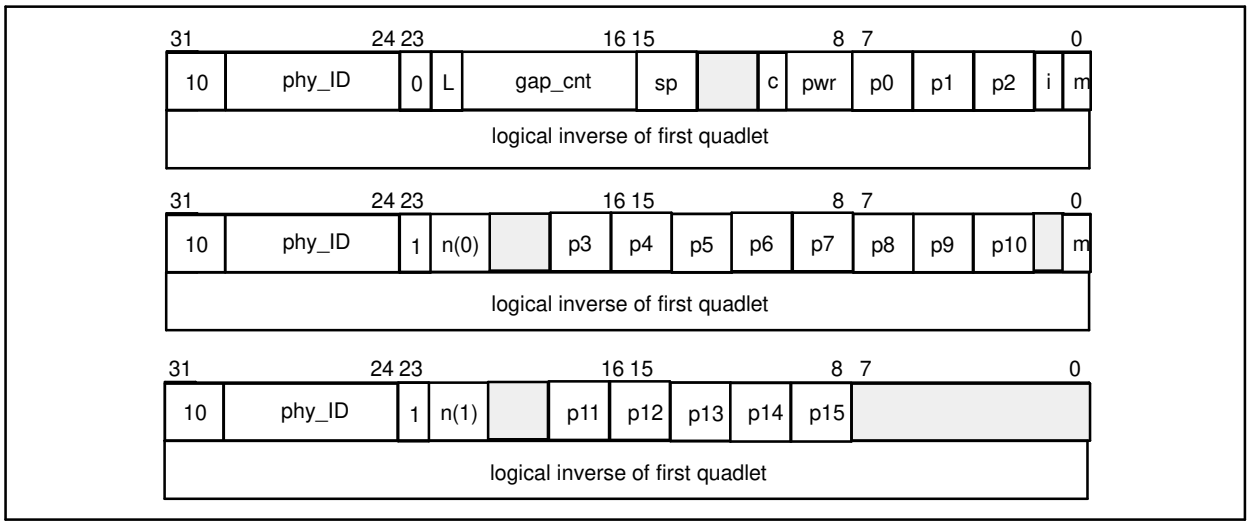


Fig. 6.1 selfID packet receive Format

Bit	Field Name	description
6	phy_ID	This bit specifies the node number of the place from which a packet is transmitted.
1	L	This bit indicates ”1” when LINK of the place from which a packet is transmitted is ON.
6	gap_cnt	This bit specifies the value stored in Gap_count field of PHY register at the place from which a packet is transmitted.
2	sp	This bit specifies the maximum transfer rate of the place from which a packet is transmitted. ”00b” : 100Mbit/sec “01b” : 200Mbit/sec ”10b” : 400Mbit/sec
1	c	This bit indicates ”1” when the place from which a packet is transmitted can be the resource manager.
3	pwr	This bit specifies the power supply type of the place from which a packet is transmitted. ”000b” : Power is not supplied to a cable power. ”001b” : Minimum 15W is supplied to a cable power. ”010b” : Minimum 30W is supplied to a cable power. ”011b” : Minimum 45W is supplied to a cable power. ”100b” : Maximum 1W is consumed from a cable power. ”101b” : Minimum 3W is consumed from a cable power. ”110b” : Minimum 6W is consumed from a cable power. ”111b” : Minimum 10W is consumed from a cable power.
2	p0- p15	This bit specifies the connection type of each port on the place from which a packet is transmitted. ”11b” : connect to child node ”10b” : connect to parent node ”01b” : disconnect ”00b” : unknown
1	i	This bit indicates ”1” when the place from which a packet is transmitted is the source of bus reset.

1	m	This bit indicates ”1” when the serial selfID exists.
3	n	This bit specifies the identification No. of selfID packet.

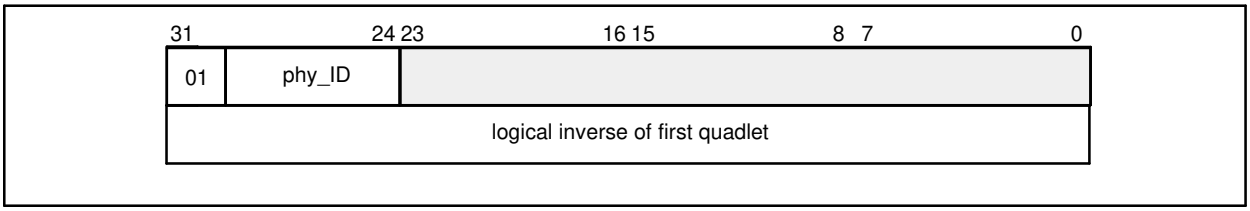


Fig. 6.2 link on packet receive Format

Bit	Field Name	description
6	phy_ID	This bit specifies the node number of the place at which a packet is received.

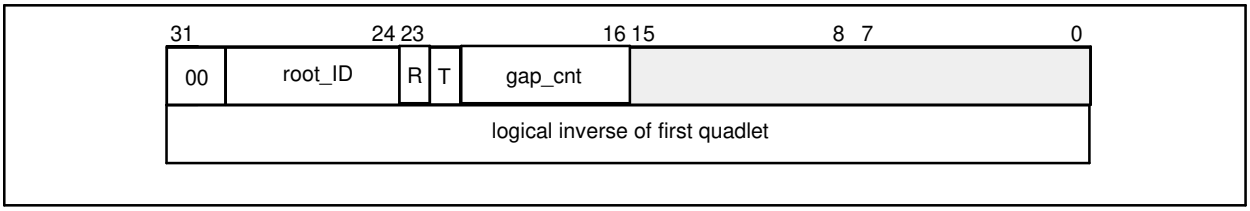


Fig. 6.3 phy configuration receive Format

Bit	Field Name	description
6	root_ID	This bit specifies the node number of the place where a packet goes out.
1	R	This bit indicates ”1” when the root_ ID field has valid data. If its node number is different from that indicated in the root_ ID field when this bit indicates ”1”, RHB bit of the PHY register should be cleared.
1	T	This bit indicates ”1” when the gap_cnt field has valid data. When this bit indicates ”1”, the value contained in the gap_cnt field should be stored in the gap_count field of the PHY register.
6	gap_cnt	This bit indicates the value which should be stored into the gap_count field of the PHY register.

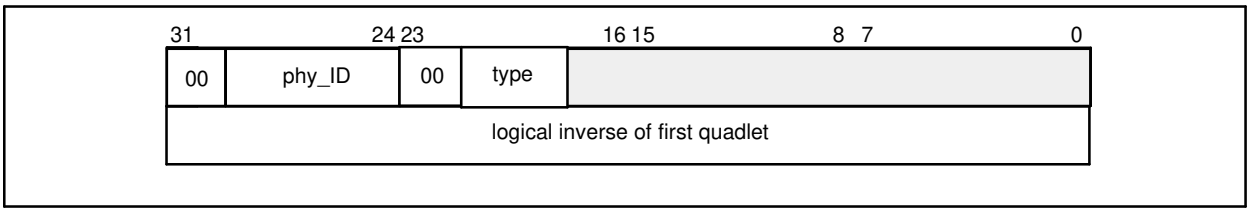


Fig. 6.4 ping packet receive Format

Bit	Field Name	description
6	phy_ID	This bit indicates the node number of the place at which a packet is received.
4	type	”00h” : ping packet

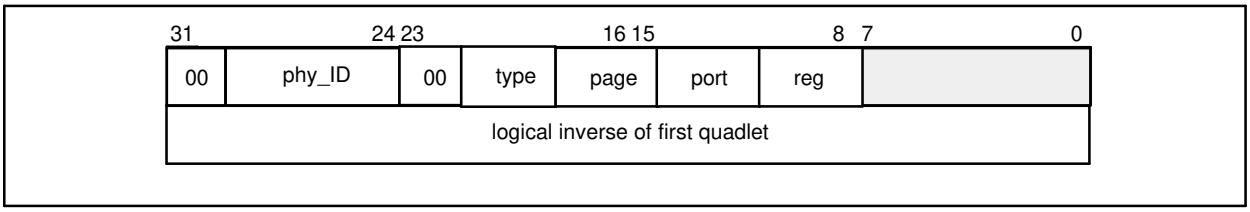


Fig. 6.5 remote access packet Format

Bit	Field Name	description
6	phy_ID	This bit indicates the node number of the place at which a packet is received.
4	type	”01h” : access to read- out 0000b- 0111b of the PHY register. ”05h” : access to read- out 1000b or later of the PHY register.
3	page	This bit corresponds to the Page_select field of the PHY register.
4	port	This bit corresponds to the Port_select field of the PHY register.
3	reg	This bit specifies the address of the PHY register which requests reading out.

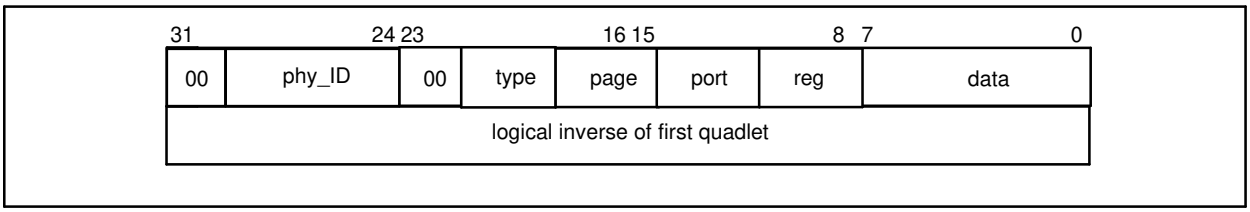


Fig. 6.6 remote reply packet Format

Bit	Field Name	description
6	phy_ID	This bit indicates the node number of the place from which a packet is transmitted.
4	type	”03h” : reply to read- out 0000b- 0111b of the PHY register. ”07h” : reply to read- out 1000b or later of the PHY register.
3	page	This bit corresponds to the Page_select field of the PHY register.
4	port	This bit corresponds to the Port_select field of the PHY register.
3	reg	This bit specifies the address of the PHY register.
8	data	This bit specifies the data of the PHY register at the address indicated in the reg field.

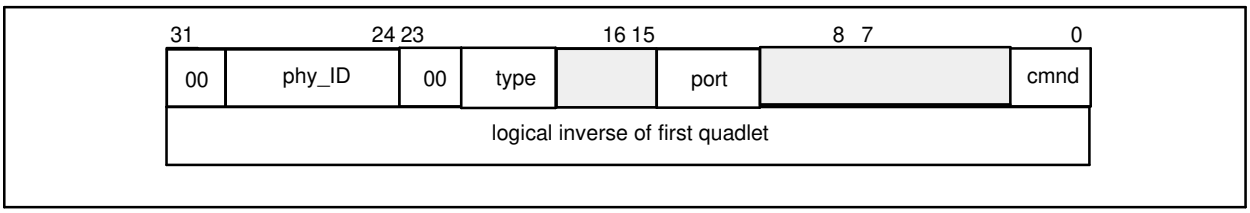


Fig. 6.7 remote command packet Format

Bit	Field Name	description
6	phy_ID	This bit indicates the node number of the place at which a packet is received.
4	type	”08h” : remote command packet.
4	port	This bit specifies the port number where the command indicated in the cmnd field should be executed.
3	cmnd	This bit executes the command indicated in this field to the port indicated in the port field. ”00h” : NOP ”01h” : TpBias providing is stopped after transmitting the TX_DISABLE_NOTIFY signal. ”02h” : Suspend is started. ”04h” : Fault bit is cleared. ”05h” : TpBias providing is started. ”06h” : Resume is started. ”07h” : NOP

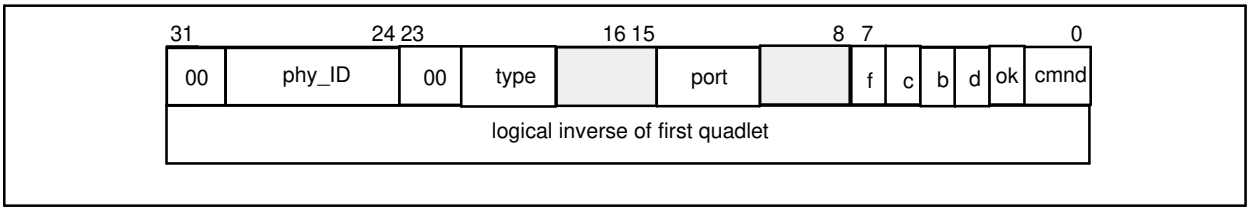


Fig. 6.8 remote confirmation packet Format

Bit	Field Name	description
6	phy_ID	This bit indicates the node number of the place from which a packet is transmitted.
4	type	”0Ah” : remote confirmation packet.
4	port	This bit specifies the port number where the command indicated in the cmnd field has been executed.
1	fault	This bit indicates the Fault bit of the PHY register.
1	connected	This bit indicates the Connected bit of the PHY register.
1	bias	This bit indicates the Bias bit of the PHY register.
1	disabled	This bit indicates the Disabled bit of the PHY register.
1	ok	This bit indicates ”1” when the command indicated in the cmnd field has been executed.
3	cmnd	This bit indicates the command executed to the port indicated in the port field.

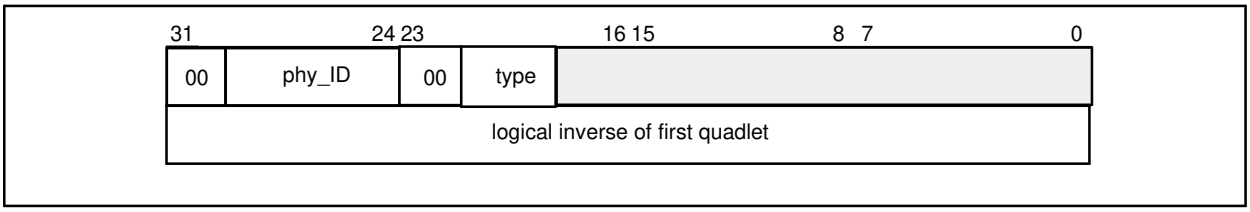


Fig. 6.9 resume packet Format

Bit	Field Name	description
6	phy_ID	This bit indicates the node number of the place from which a packet is transmitted.
4	type	”0Fh” : resume packet.

6.2. Device Reset

MB86613S device has three types of reset as follows:

1) Hardware Reset (Power- on Reset) :

Input ”L” level of signal to the RST pin of device.
This reset makes all the functions to stop and initializes all the registers.

2) Software Reset :

Set HCControl.softReset bit.
This reset stops LINK, CPCs, and PCI operations. But PHY still functions and that retains the bus topology configured.
A part of registers is not initialized by the software reset.

3) Bus Reset :

Set IBR bit of PHY register.
This reset does not affect to all the device functionalities, except for the MB86613S’s internal bus management CSR register which is initialized by the reset.

The MB86613S device contains a power- on reset circuit in order to allow the device operate in a 1394 cable power while the system power is shut- down. Figure 6.1 shows the block diagram of circuit. This circuit automatically resets the device when detecting a 1394 cable power while the system power is shut- down.

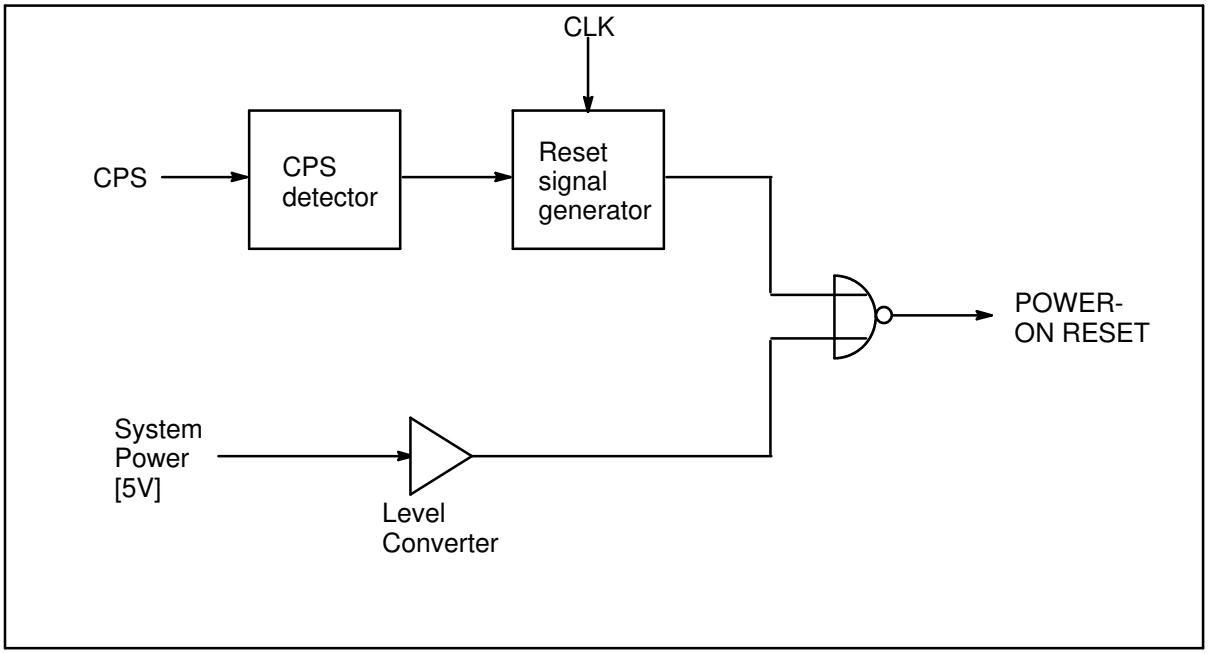


Fig. 6.10 Diagram of Internal Power- On Reset Circuit

6.3. PHY Register

MB86613S contains the following PHY registers. When accessing these registers, use the PHYControl Register in the Open HCI Registers. Each symbol in the description table has the following meaning:

r . . . . . Registers can be read out.
w . . . . . Registers can be written.
c . . . . . Bit can be cleared (”0” indicated) by setting “1” here.
u . . . . . The read out value is undefined per the device operation state.

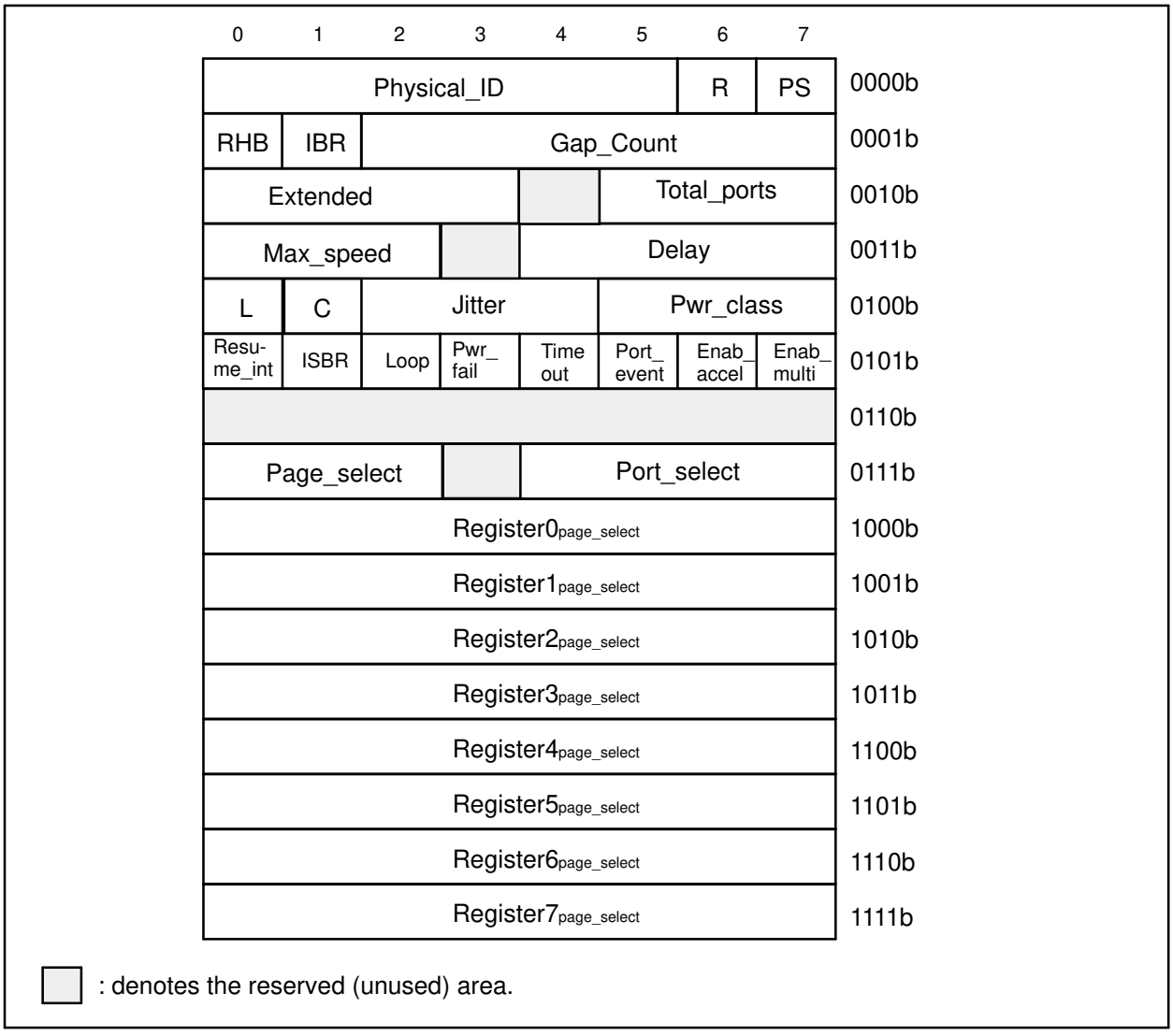


Fig. 6.11 PHY Register Map

Bit size	Field Name	rwcu	reset	description
6	Physical_ID	ru	undefined	This field indicates the node number of this device.
1	PS	ru	undefined	This bit indicates ”1” when detecting a cable power which is greater than a nominal value, 8V.
1	R	ru	undefined	This bit indicates ”1” when the device is a root.

1	RHB	rw	0b	Setting ”1” at this bit makes the device intend to be a root in the next bus reset.
1	IBR	rw	0b	Setting ”1” at this bit executes the bus reset.
6	Gap_count	rw	3Fh	This field is to set and indicate the gap_count time.
4	Extended	r	7h	This field indicates the number of extended registers (7h).
3	Total_ports	r	03h	This field indicates the port number owned by this device (3 ports).
3	Max_speed	r	010b	This field indicates ”010b” since the device supports up to s400 speed in maximum.
4	Delay	r	0h	This field indicates the delay time generated by repeating the packet received.
1	L	rw	1b	Writing ”1” at this bit sets the ”1” at L flag of transmit self ID packet.
1	C	rw	0b	Writing ”1” at this bit sets the ”1” at C flag of transmit self ID packet.
3	Jitter	rw	0h	This field indicates the jitter generated by repeating the packet received.
3	Pwr	rw	100b	This field is stored in Pwr field of transmit selfID packet. ”000”b : Power is not supplied to a cable power. ”001”b : Minimum 15W is supplied to a cable power. ”010”b : Minimum 30W is supplied to a cable power. ”011”b : Minimum 45W is supplied to a cable power. ”100”b : Maximum 1W is consumed from a cable power. ”101”b : Minimum 3W is consumed from a cable power. ”110”b : Minimum 6W is consumed from a cable power. ”111”b : Minimum 10W is consumed from a cable power.
1	Resume_int	rw	0b	This bit indicates ”1” when resume is performed.
1	ISBR	rw	0b	Setting ”1” at this bit executes the arbitrated short bus reset.
1	Loop	rcu	0b	This bit indicates ”1” when the port connection forms a loop.
1	Pwr_fail	rcu	0b	This bit indicates ”1” when a cable power is below the nominal rate (<8V).
1	Timeout	rcu	0b	This bit indicates ”1” when max_arb_state_time is time- out.
1	Port_event	rcu	0b	This bit indicates ”1” when the Connected, Bias, Disabled and/or Fault bit changes with the Int_enable bit set.
1	Enab_accel	rw	0b	Setting ”1” at this bit makes the arbitration accelerate valid.

1	Enab_multi	rw	0b	Setting ”1” at this bit executes multi- speed packet concatenation when transmitting packets continuously.
3	Page_select	rw	undefined	This field is for selecting the page register of port set by Port_select field (bit7:3) : ’000b’ : port status page ’001b’ : vendor identification page ’111b’ : PHY vendors page
4	Port_select	rw	undefined	Set the port number (00h to 02h) in this field.

6.3.1. Page Register

PHY register contains the following three kinds of page register. These page registers allow to set and indicate various information per port.

”000b” : port status page
”001b” : vendor identification page
”111b” : PHY vendors page

Data in page register are shown in 8- byte field of PHY register shown in Figure 6.12 by setting the page register address in Page_select field and port number in Port_select field.

There is no register map defined for PHY vendor page. So, use it as 1- byte x 8 RAM.

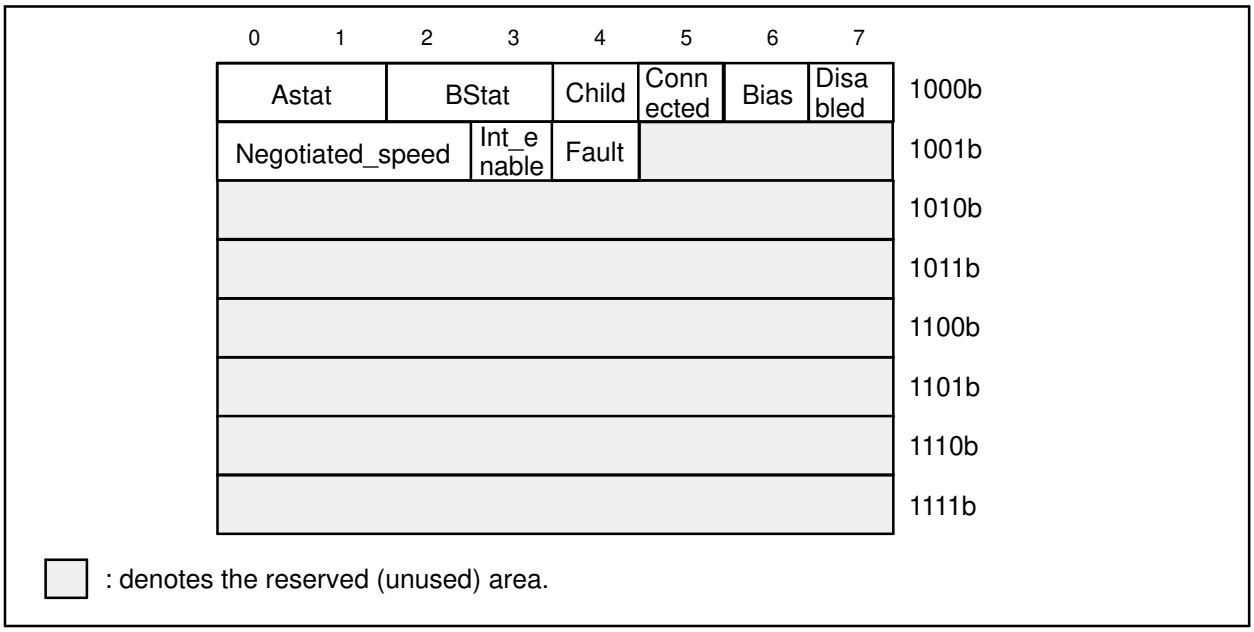


Fig. 6.12 PHY Register Page 0 - Port Status Page Register Map

Bit size	Field Name	rwcu	reset	description
2	Astat	ru	undefined	This field indicates the TPA state: ”01b” : ”1” ”10b” : ”0” ”11b” : ”Z” ”00b” : invalid

2	Bstat	ru	undefined	This field indicates the TPB state: ”01b” : ”1” ”10b” : ”0” ”11b” : ”Z” ”00b” : invalid
1	Child	ru	0b	This bit indicates ”1” when the connected port is a child port.
1	Connected	ru	0b	This bit indicates ”1” when the cable is connected.
1	Bias	ru	0b	However, “1” is indicated only when the TPBias is detected after the time- out of connection_timeout.
1	Disabled	rw	0b	Setting ”1” at this bit stops to provide TPBias.
3	Negotiated_speed	ru	000b	This field indicates the transfer speed that has been exchanged with the destination port to be connected.
1	Int_enable	rw	0b	The Port_event bit indicates ”1” when the Connected, Bias, Disabled and/or Fault bit changes.
1	Fault	rcu	0b	This bit indicates ”1” when an error occurred in performing the suspend or resume.

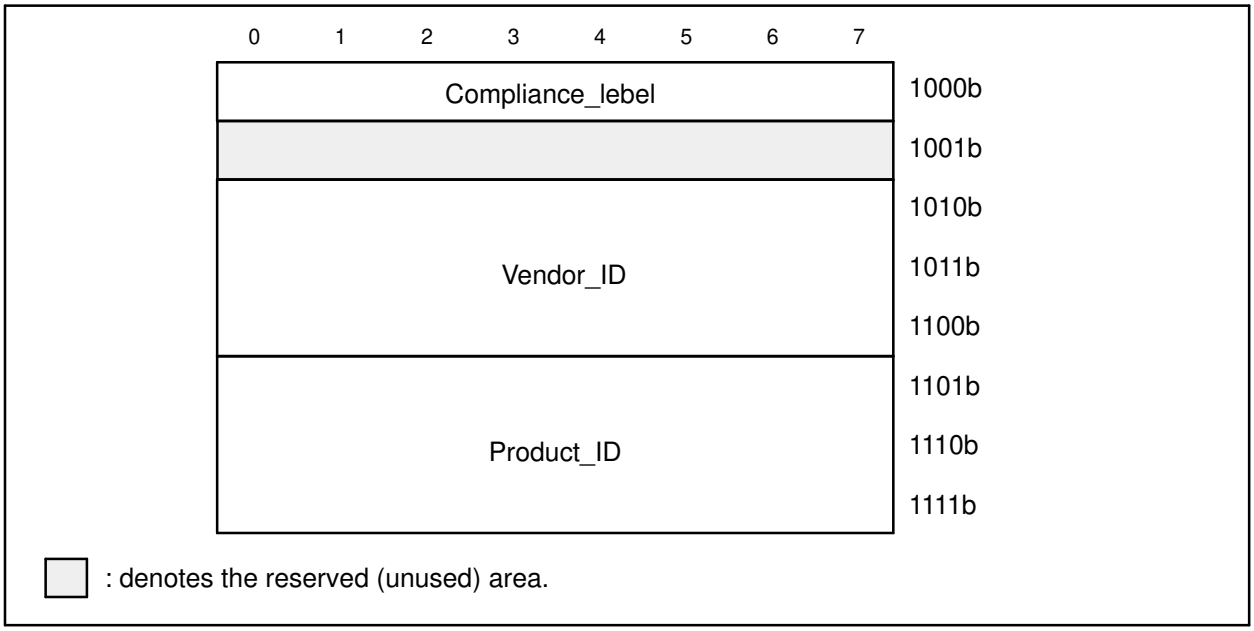


Fig. 6.13 PHY Register Page 1 - Vendor Identification Page Register Map

Bit size	Field Name	rwcu	reset	description
8	Compliance_lebel	r	01h	This field indicates ”01h” because the device supports P1394a.
24	Vendor_ID	r	00000Eh	Fujitsu vendor ID for 1394 Open- HCI ”00000Eh” is indicated in this field.
24	Product_ID	r	086613h	Fujitsu product ID for 1394 Open- HCI ”086613h” is indicated in this field.

7. PCI Interface

7.1. Alignment

When a data length of received isochronous packet is not a multiple of ”4”, there can be an invalid field in the last data section. To connect the next packet data with this invalid field and store the data into a host memory is so called Alignment process. Figure 7.1 shows some examples of Alignment. In Alignment- 1 in the example, the last byte data in the received packet is stored in bit7:0 field in the host memory. If the next packet is stored connecting the previously received data, the head data (byte0) in the next packet is stored in bit15:8 in the host memory. Then, byte3 is stored in the bit7:0 in the next address.

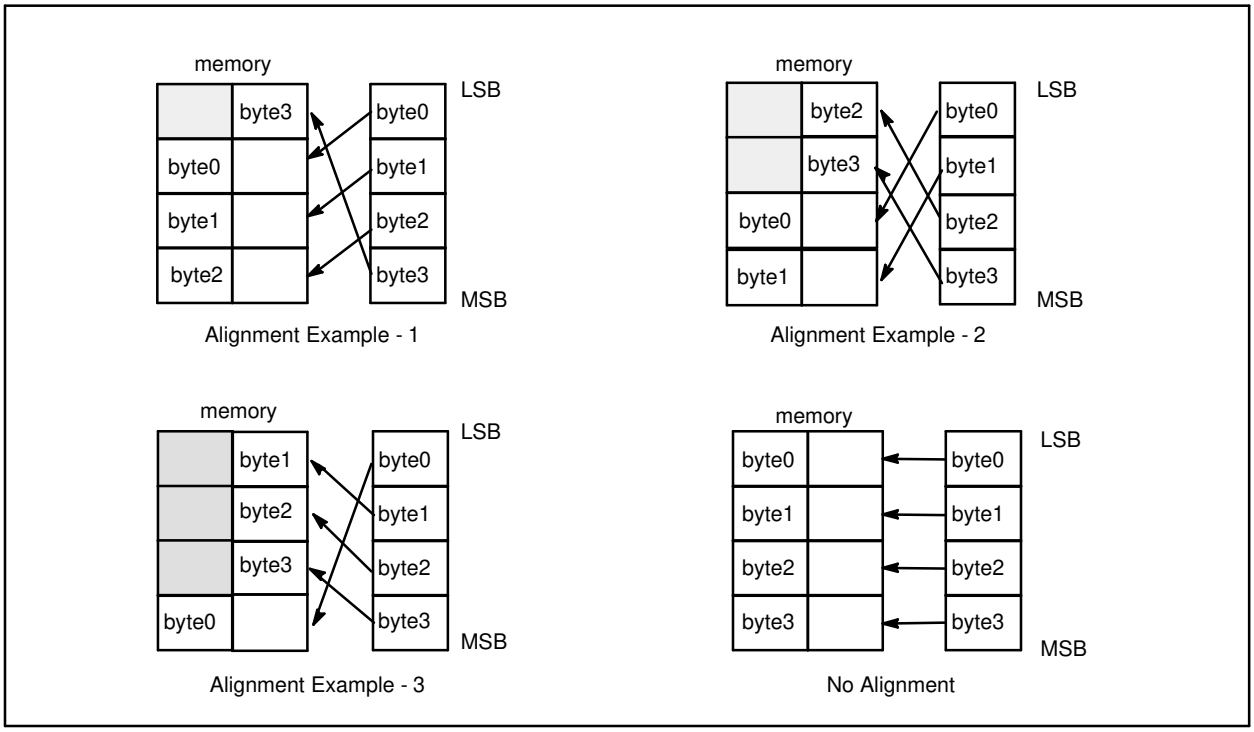


Fig. 7.1 Examples of Alignment

7.2. Byte Swap Dealing

Because of a different type in endian between 1394 interface and PCI bus interface, a byte order swapping is required when transmitting a packet from PCI (little- endian) to 1394 (big- endian) bus and receiving a packet vice versa.

The byte swap is enable only when HCControl.noByteSwapData is cleared or PCI_HCIControl.PCI_Global_Swap is set. The byte swap deal is not applicable to the packet header (quadlet data in Figure 7.3 is excluded.) and PHY packet.

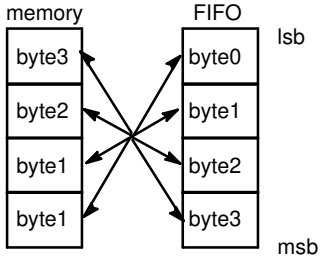


Fig. 7.2 Byte Swap Dealing

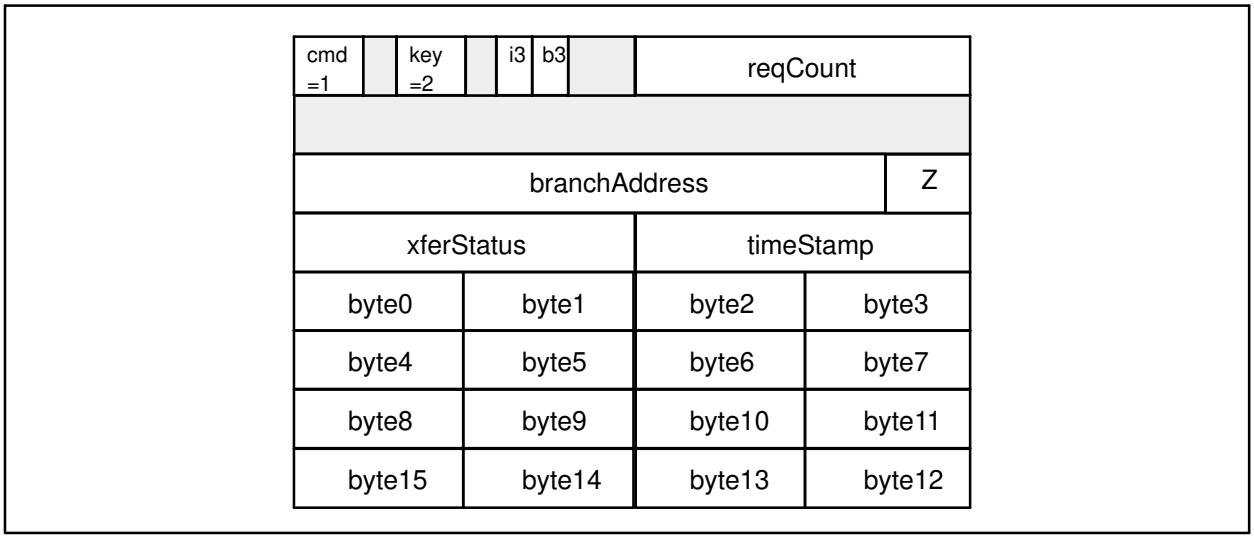


Fig. 7.3 descriptor's Byte Ordering for Transmitting quadlet write request packet

7.3. EEPROM

Figure 7.4 shows the EEPROM address map. The EEPROM needs to contain the vendor ID and the chip ID to be loaded to the Global Unique ID register of the OHCI register, in addition to the data to be loaded to the subsystem ID, subsystem vendor ID of PCI configuration register, the CAP pointer of the power management and CIS pointer of Cadr Bus.

After the device hardware reset, the data in the EEPROM is loaded into PCI configuration register or OHCI register.

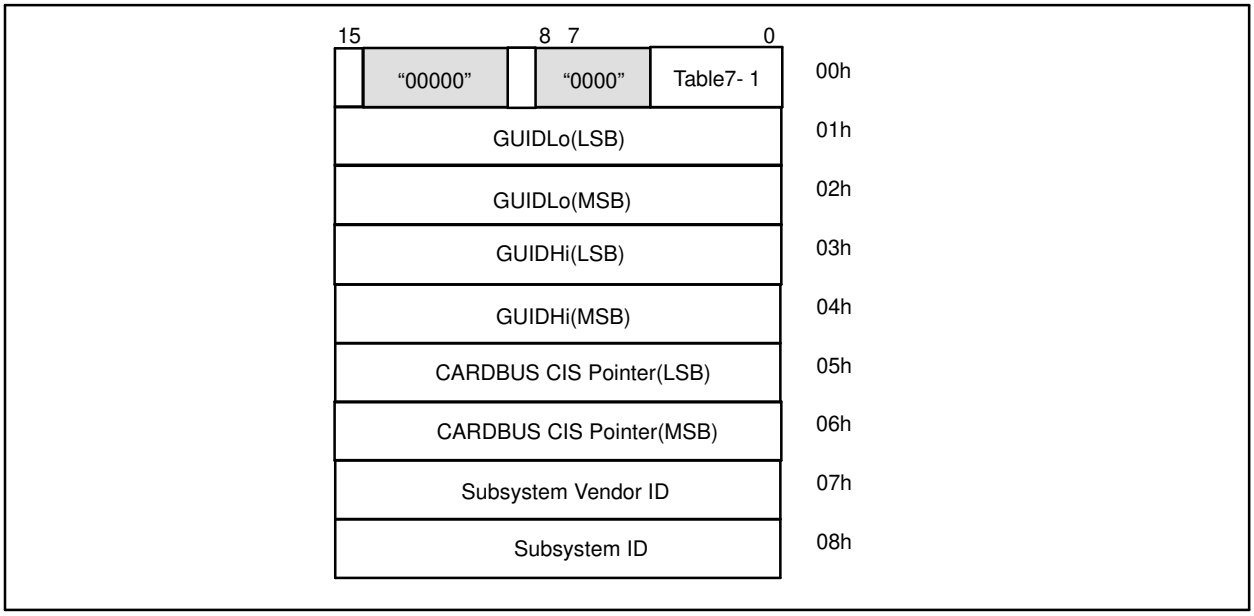


Fig. 7.4 EEPROM Address Map

Table 7- 1 Adr00h bit description

Bit	Field Name	reset	description
15	PWE_EN	0b	“1” at this bit reduces the power consumption during idle (no 1394 connection).
9	OHCImode	0b	Setting “0” at this bit enables OHCI1.0 mode Setting “1” at this bit enables OHCI1.1 mode.
4	PCI_PME#	0b	Setting ”1” at this bit enables the PCI PME# signal.
3	VENDOR_ID	0b	Setting ”0” at this bit shows Fujitsu Vendor ID. Normally set to ”0”.
2	DEBUG_MODE	1b	Setting ”1” at this bit enables to access the debug register and EEPROM configuration write through PCI configuration register. Normally set to ”0”.
1	BYTE_SWAP	1b	Setting ”1” at this bit performs the byte swapping for the data of the 1394 configuration ROM access. Normally set to ”0”.
0	BIOS_EN	1b	Setting ”1” at this bit sets the ExpantionROMBaseAddress, rom_enable bit. MB86613S does not support Expantion ROM, This bit is set to “0”.

Please write “0” to the reserved bits of Adr 00h.

To write with following format through PCI configuration register 14h, MB86613S EEPROM configuration registers are also programmed without external serial EEPROM.

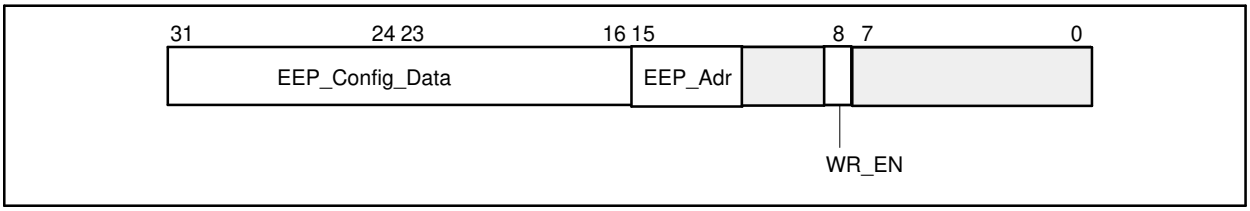


Fig. 7.4.1 EEPROM configuration Register

Bit	Field Name	rwu	reset	description
31:16	EEP_Config_Data	w	00h	Set the write data to EEPROM configuration
15:12	EEP_Adr	w	0h	Set the Address of EEPROM configuration
8	WR_EN	w	0b	“1” enables the data write to EEPROM configuration.

Caution : To use this feature, external EEDO pin must be open.
: EEP_Adr 0h should be written as last access with DEBUG_BIT is “0”.
Even “0” write to DEBUG_BIT of EEP_ADR 0h, this write is available but later access is ignored.

7.3.1 How to access EEPROM

EEPROM can be accessed by using the address space of Open HCI register space field.
Data storing to EEPROM should be done in the following procedures.

1) Write ”00”b at EEPCMD.EEPCmd and ”11”b at the bit 5- 4 in EEPAddr field. Then change the write- operation to ”enable”.

2) The following three settings should be done.

a) Set ”01b” into the EEPCMD.EEPDone field.
b) Set the EEPROM address where the data is stored into the bit 5- 0 of the EEPAddr field.
c) Set the data to be written into the EEPDATA register.

3) Wait until the EEPCMD.EEPDone bit is set.

4) Write ”00”b at EEPCMD.EEPCmd field and ”00”b at the bit 5- 4 in EEPAddr field. Then change the write- operation to ”disable”.

In case EEPCMD.EEPDone bit has not set, set EEPCMD.EEPReset bit to write again.
Reading out EEPROM data can be done as follows;

1) Write ”10”b at EEPCMD.EEPCmd field and set EEPROM address where the data is stored to EEPAddr field.

2) Wait until the EEPCMD.EEPDone bit is set.

3) EEPROM data in the address set at 1) above are displayed on EEPDATA register.

Writing ”AA559966”h and ”669955AA”h continuously to KeyLocation register makes the EEPCMD and EEPDATA registers available.

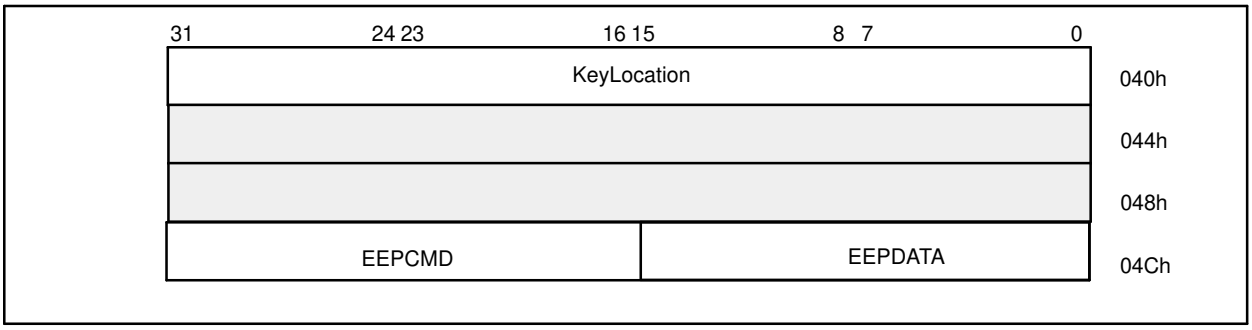


Fig. 7.5 EEPROM Register

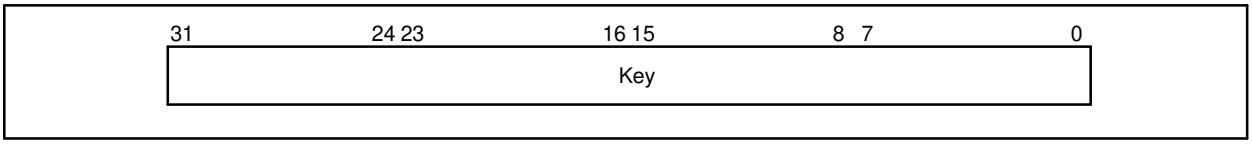


Fig. 7.6 KeyLocation Register Map

Bit size	Field Name	rwcu	reset	description
31:0	Key	w	000000h	Writing ”AA559966”h and ”669955AA”h continuously to KeyLocation register makes EEPCMD and EEPDATA registers available. 1st : ”AA559966”h 2nd :”669955AA”h

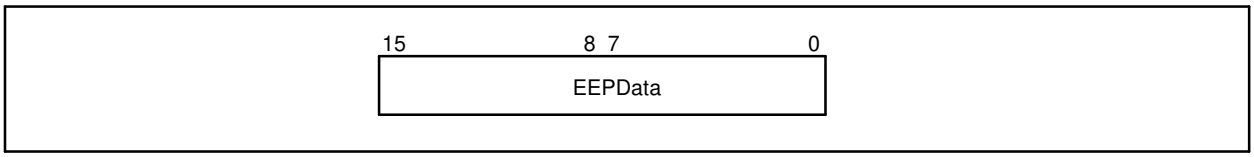


Fig. 7.7 EEPData Register Map

Bit size	Field Name	rwcu	reset	description
15:0	EEPData	rw	undefined	These bits store the data to be written into EEPROM or displays the read data from EEPROM.

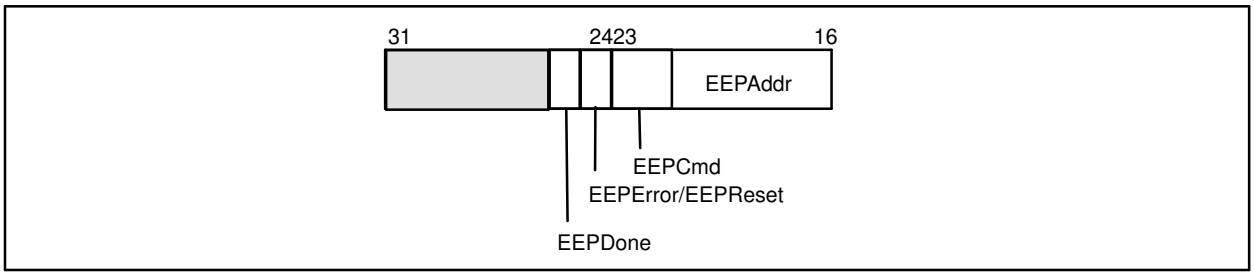


Fig. 7.8 EEPCMD Register Map

Bit size	Field Name	rwcu	reset	description
25	EEPDone	r	1b	This bit indicates ”1” if the data displayed on EEPDATA register is valid when the writing has been completed or the reading out is done.
24	EEPError	r	0b	This bit indicates ”1” when an error occurred during the writing or reading out. It is valid when EEPDone bit indicates ”1”.
24	EEPReset	w	0b	Set this bit to ”1” to execute the writing/reading out again when EEPDone bit does not indicate ”1” after a long wait.
23:22	EEPCmd	rw	00b	Set the command to execute the writing/reading out. Refer to the next table for the commands.
21:16	EEPAddr	rw	00h	Set the EEPROM address to execute the writing/reading out.

Bit 23- 20	description
10XX	read
0011	write enable
01XX	write
0000	disable programming mode

7.4. Power Management

The on- chip PCI bus interface supports the power management of the PCI Bus Power Management Interface Specification (version 1.1).
The MB86613S LSI has a power management register in it as shown in the Figure 7- 9. The initial address of the register is indicated on the Cap_ Ptr register in the PCI configuration register.
The next table shows the four states managed by this power management. Figure7- 10 shows the possible transitions of the states.
When changing the state in D0, or D2, set the state in the PowerStatus field on the PMCSR register.

D0 : The PCI bus can perform a transfer in this state.

D2 : The clock cannot be provided for the inside circuits (can be provided partially) of the MB86613S. 1394 configuration ROM data is not hold. PCI configuration register and GUID register data are hold. INT# is not aserted even cause a interrupt.
A 200us is needed for the sate transition from D2 to D0.

$D3_{hot}$: Set the HCControl.softReset bit for the sate transition from D3hot to D0.
The clock cannot provided for the inside circuits (can be provided partially) of the MB86613S. 1394 configuration ROM and PCI configuration register data are not hold. GUID register data is hold. INT# is not aserted even cause a interrupt. An 10ms is needed for this sate transition.

$D3_{cold}$: The clock and the power cannot be provided for the inside circuits of the MB86613S. After starting to provide the power, assert the PCIRST#.

PME# is asserted if it receives a packet or a bus reset on the D2 or $D3_{hot}$ state with the PMCSR. PME_en bit set. The PME# can be negated by setting ”1” at the PMCSR. PME_status bit or clearing the PME_en bit.

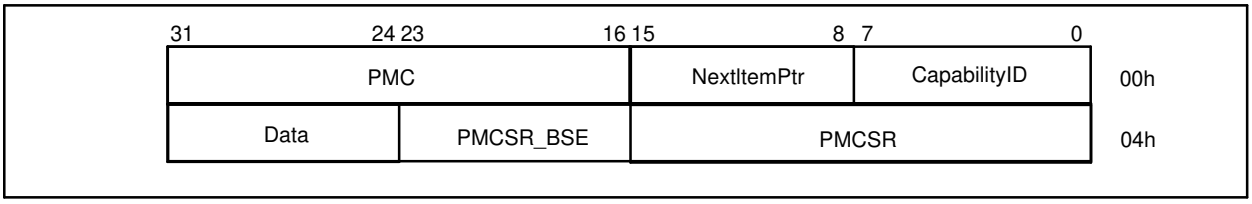


Fig. 7.9 Power Management Register

State	Power	Clock	#INT	PME#
D0	ON	ON	Enable	-
D2	ON	OFF	Disable	Enable
$D3_{hot}$	ON	OFF	Disable	Enable
$D3_{cold}$	OFF	OFF	Disable	Disable

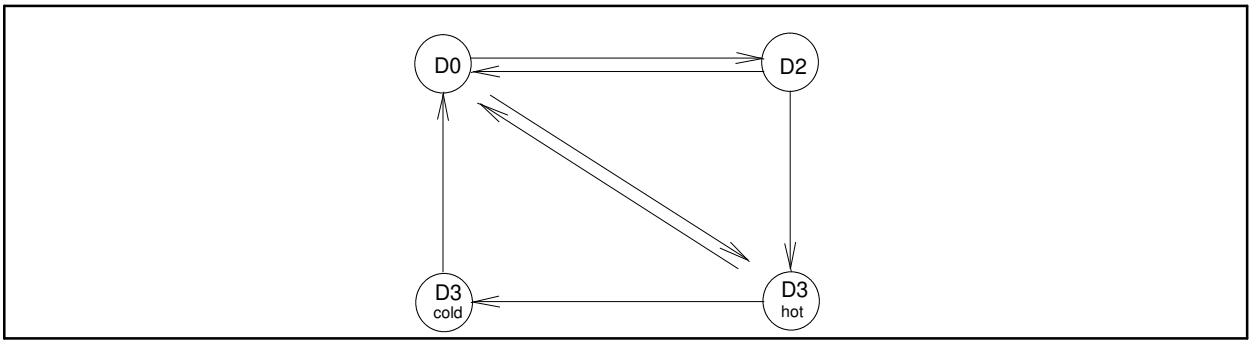


Fig. 7.10 PCI Power Management State Transition

7.4.1 Capability Identifier Register

This register indicates the ID in order to identify two or more power management registers.
”01h” is indicated because the MB86613S contains one power management register.

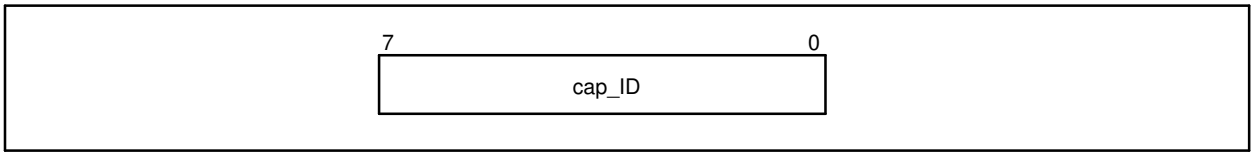


Fig. 7.11 Capability Identifier Register Map

Bit size	Field Name	rwcu	reset	description
7:0	cap_ID	r	01h	These bits indicate ”01h”.

7.4.2 Next Item Pointer Register

This register indicates the initial address of the next power management register if two or more power management registers exist.
”00h” is indicated for the MB86613S because it has one power management register.

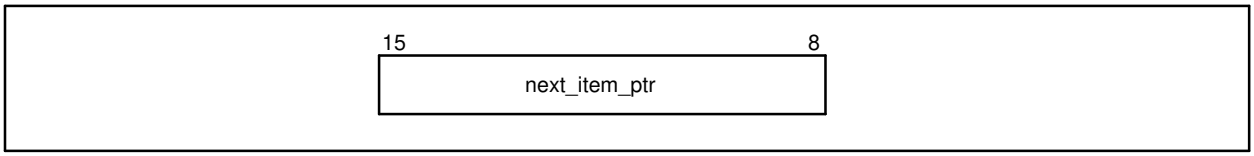


Fig. 7.12 Next Item Pointer Register Map

Bit size	Field Name	rwcu	reset	description
15:8	next_item_ptr	r	00h	These bits indicate ”00h”.

7.4.3 Power Management Capabilities(PMC) Register

This register is used for indicating the contents of the power management managed on the MB86613S.

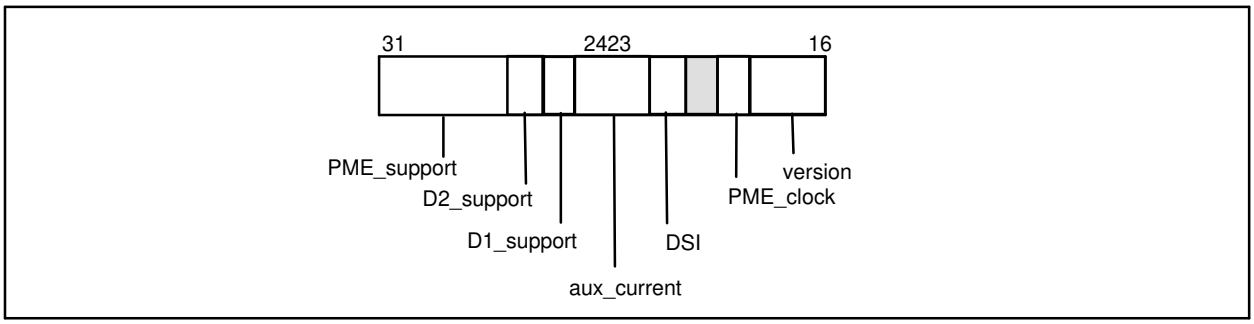


Fig. 7.13 Power Management Capabilities Register Map

Bit size	Field Name	rwcu	reset	description
31:27	PME_support	r	0Ch	These bits indicate ”0Ch” to assert the PME# signal at the state of D2 or D3hot.
26	D2_support	r	1b	This bit indicates ”1b” to support D2 state.
25	D1_support	r	0b	D1 state is not supported.
24:22	aux_current	r	000b	These bits indicate ”000b” because the PME# signal is not asserted at the D3.cold state.
21	DSI	r	0b	This bit indicates ”0b”.
19	PME_clock	r	0b	This bit indicates ”0b” because the clock (PCICLK#) is not necessary to assert the PME# signal.
18:16	version	r	010b	This bit indicates ”010b” because this document complies with the version 1.1 of the PCI Bus Power Management Interface Specification.

7.4.4 Power Management Control/Status(PMCSR) Register

This register indicates the status of the power management.
When initializing the MB86613S, clear the PME_en bit, then set ”1” at the PME_status bit.

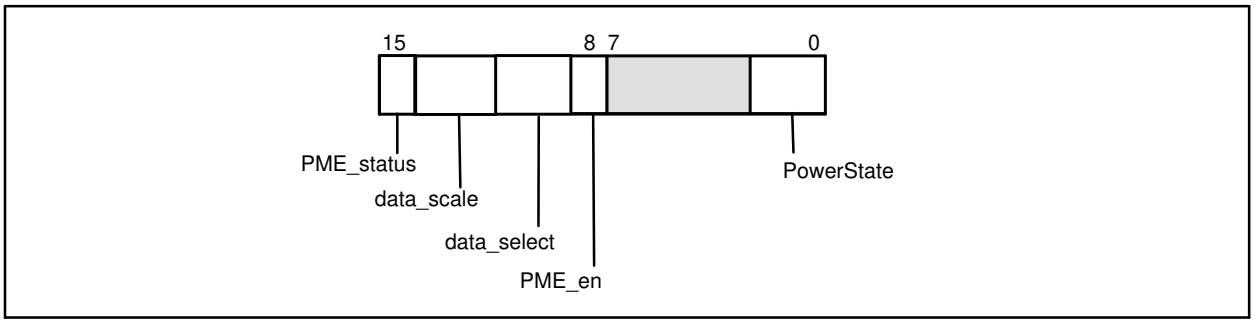


Fig. 7.14 Power Management Control/Status Register Map

Bit size	Field Name	rwcu	reset	description
15	PME_status	rcu	0b	This bit indicates ”1” when the PME# is asserted.
14:13	data_scale	rw	00b	This bit is used to set the data range indicated in the Data register. Refer to the 7.4.6 ”Data Register” for the details.
12:9	data_select	rw	0h	This bit is used to change the windows in the Data register. Refer to the 7.4.6 ”Data Register” for the details.
8	PME_en	rw	undefined	Setting ”1” at this bit enables the PME#. Setting ”0” at this bit does not assert the PME#.
1:0	PowerStatus	rw	00b	This bit sets the state of the power management to be performed. ”00b” : D0 ”01b” : Prohibit to set. ”10b” : D2 ”11b” : D3hot

7.4.5 PMCSR Bridge Support Extension(PMCSR_BSE) Register

This register indicates the function of the power management on the PCI bridge.
It indicates ”00h” because the MB86613S is not equipped with the PCI bridge function.

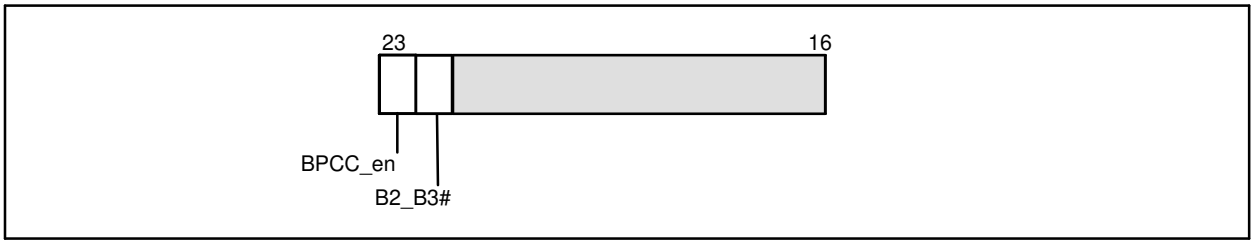


Fig. 7.15 PMCSR Bridge Support Extension Register Map

Bit size	Field Name	rwcu	reset	description
23	BPCC_en	r	0b	This bit indicates ”0b”.
22	B2_B3#	r	0b	This bit indicates ”0b”.

7.4.6 Data Register

This register indicates the consumptive power and the temperature of the MB86613S.
The setting value at the data_select field determines the state of the register as follows.

”0h” : The consumptive power in the D0 state is indicated.
”1h” : The consumptive power in the D1 state is indicated.
”2h” : The consumptive power in the D2 state is indicated.
”3h” : The consumptive power in the D3 state is indicated.
”4h” : The temperature in the D0 state is indicated.
”5h” : The temperature in the D1 state is indicated.
”6h” : The temperature in the D2 state is indicated.
”7h” : The temperature in the D3 state is indicated.
”8h” : The consumptive power in the normal transfer is indicated.

The setting value at the data_scale field determines the range to be indicated as follows.

”0h” : Prohibit to set.
”1h” : 0- 25.5 (W) data X 10^- 1
”2h” : 0- 2.55 (W) data X 10^- 2
”3h” : 0- 255 (mW) data X 10^- 3

The environment of the consumptive power and the temperature indicated in this register is as follows.

Amplifier : 33MHz (PCICLK#)
Power voltage : 5.25V (VDD5/3)
Ambient temperature : 70 degrees C

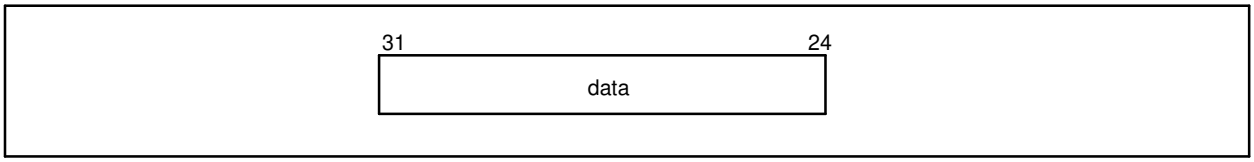


Fig. 7.16 Data Register Map

Bit size	Field Name	rwcu	reset	description
31:24	data	r	undefined	These bits indicate the consumptive power and the temperature of the state set in the data_select field.

8. Software Implementation

8.1. Asynchronous Transmit

An example of software flowchart for transmitting an asynchronous request or response packet is shown in Figure 8.1. The same process should be required for PHY packet (phy configuration packet, linkon packet, and ping packet) and asynchronous stream packet transmission.

1) Check ATContextControl.active bit state and make sure it is cleared.

2) Store the data to transmit in an host memory.

3) Store the context program in the host memory. In this case, also set the packet header in the lower 16- byte field of the first descriptor (immediate command).

4) Set the start address of host memory where the context program is stored, in the ATCommandPtr.descriptorAddress field. Also set the count; (number of descriptor + 1), since the first descriptor is an immediate command, in the ATCommandPtr.Z field.

5) Set the ATContextControl.run bit.

6) Wait until the INTA# is set. (when the last descriptor's " i " field contains 11b or 01b and IntMask.reqTxComplete or .respTxComplete bit is set.)

7) Check the ATContextControl.eventcode field.

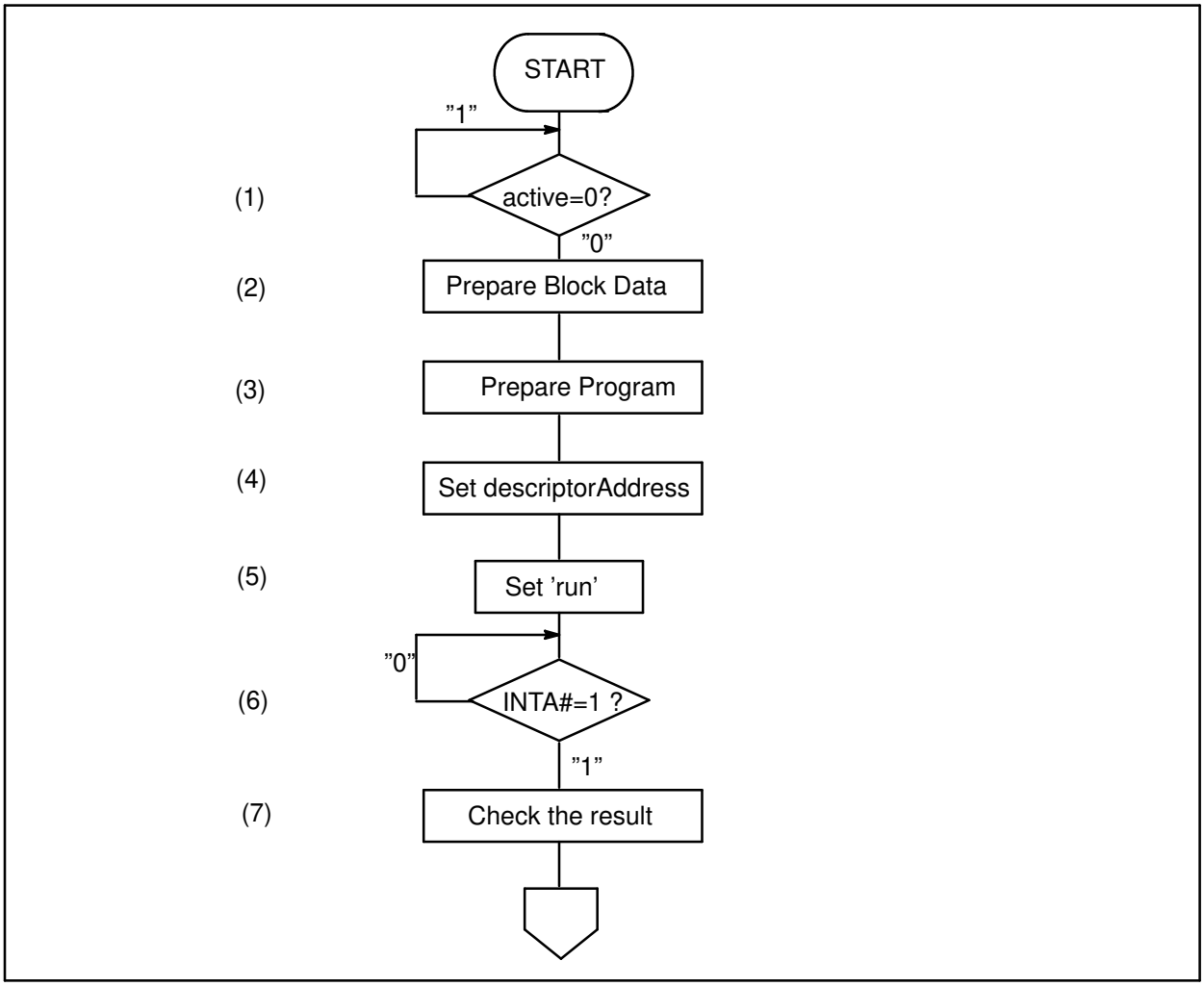


Fig. 8.1 Software Flow Chart for asynchronous packet Transmit

8.1.1. How to Make Context Program

Figure 8.2 shows an example of context program format which is applied when transmitting a block write request packet having 32- byte block.

ATCommandPtr.descriptorAddress field must contain the value "80000000h" in advance that is the start address of host memory where the program is stored. Besides, ATCommandPtr.Z field must contain a value "4h" that is the number of descriptor.

When making a context program for asynchronous transmit, the first descriptor must be OUTPUT_MORE_immediate or OUTPUT_LAST_Immediate command in order to generate the packet header. In the example, it requires a payload and so, the program must start with OUTPUT_MORE_Immediate command.
Because the packet header is 16- byte length, reqCount field in the OUTPUT_MORE_Immediate command must contain "10h" and the lower 4- quadlet field of the command must contain the packet header in OpenHCI format.

In the example, there is an assumption that payload data is split out into 16- byte+16- byte and each byte is stored in individual host memory. Therefore, it is required to use OUTPUT_MORE command following the OUTPUT_MORE_Immediate command. In descriptor- 2, the reqCount field of descriptor for OUTPUT_MORE command must contain "10h" and the dataAddress field must contain the host memory's start address "10000000h" where the block data is stored. In descriptor- 3, it uses the OUTPUT_LAST command

because the last block data is set. The reqCount field must contain “10h” and the dataAddress field must also contain the host memory’s start address:”20000000h” where the block data is stored. In that case, the example is also based on an assumption that another asynchronous request packet is transmitted following the block write request packet. Therefore, the branchAddress field must contain the host memory start address ”90000000h” where the context program to control the next transmit packet is stored, and the Z field must contain ”4h” which is the number of descriptor. The ” i ” field of the OUTPUT_LAST command is being set to ”11b” so that an interrupt event is reported after completion of the block write request packet.

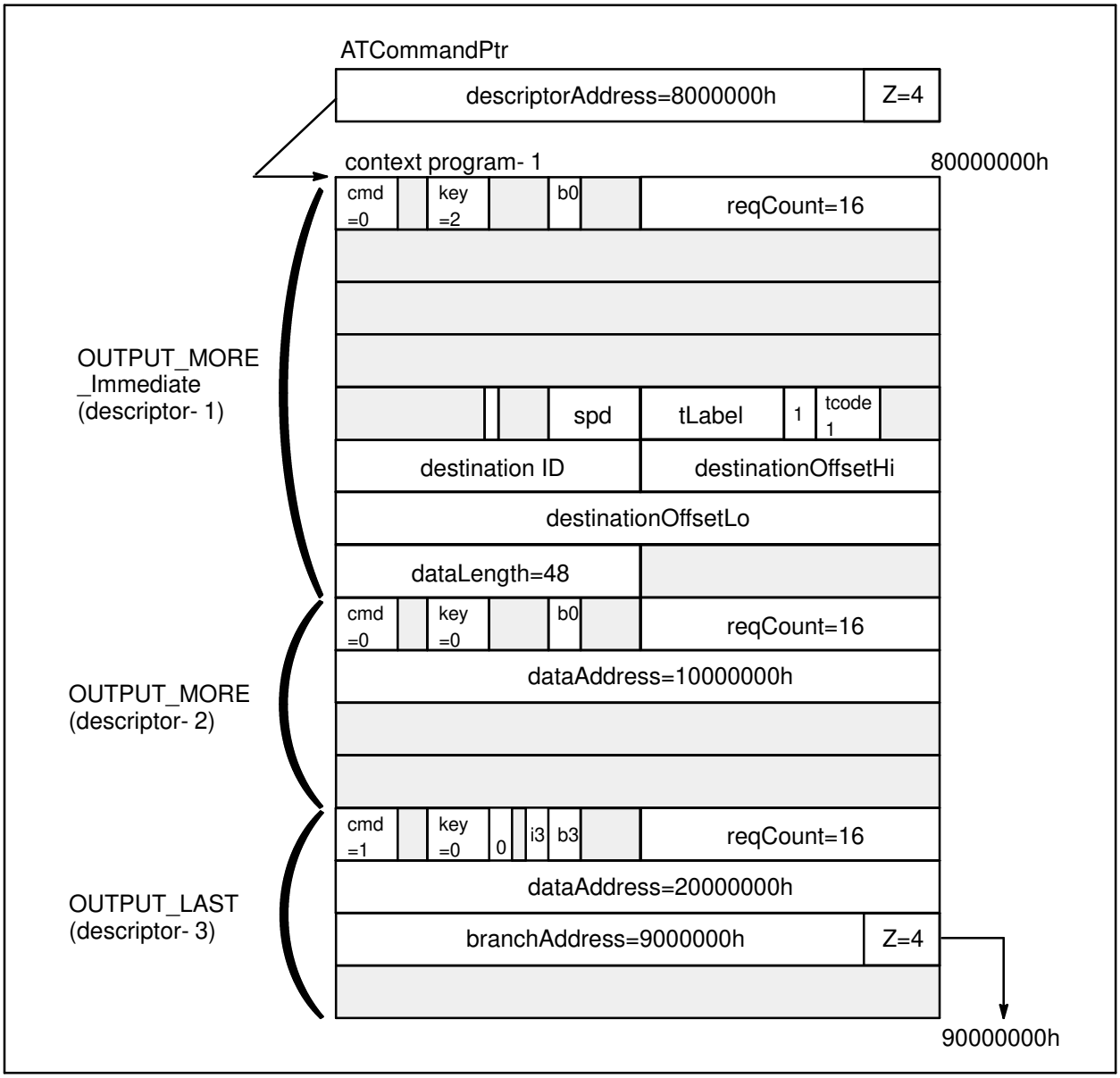


Fig. 8.2 Example of Context Program Format for Block Write Request Packet Transmit

8.1.2. Descriptor

This section describes the format of descriptor processed by AT- CPC. The descriptor(s) that AT- CPC handles are OUTPUT_MORE command, OUTPUT_LAST command, OUTPUT_MORE_Immediate command, and OUTPUT_LAST_Immediate command. For the _Immediate commands, specify the packet header in the lower 16- byte field. Figures 8.3 through 8.6 show the descriptors' format.

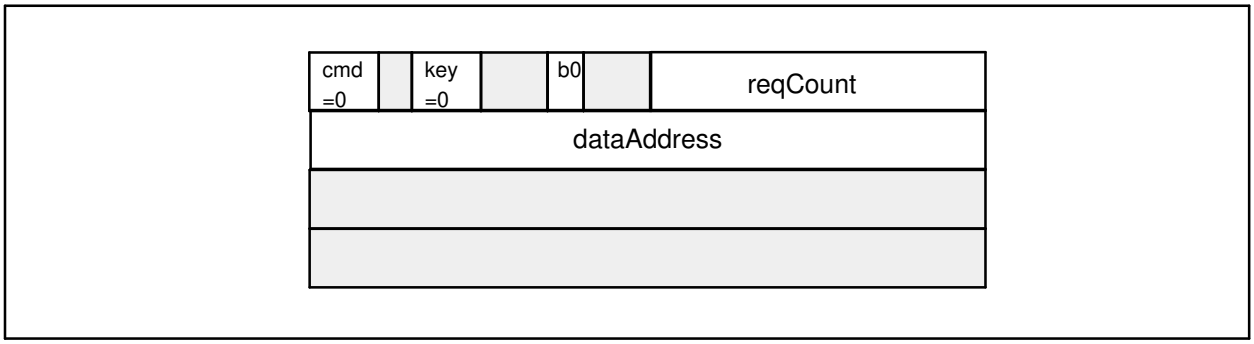


Fig. 8.3 OUTPUT_MORE descriptor Format

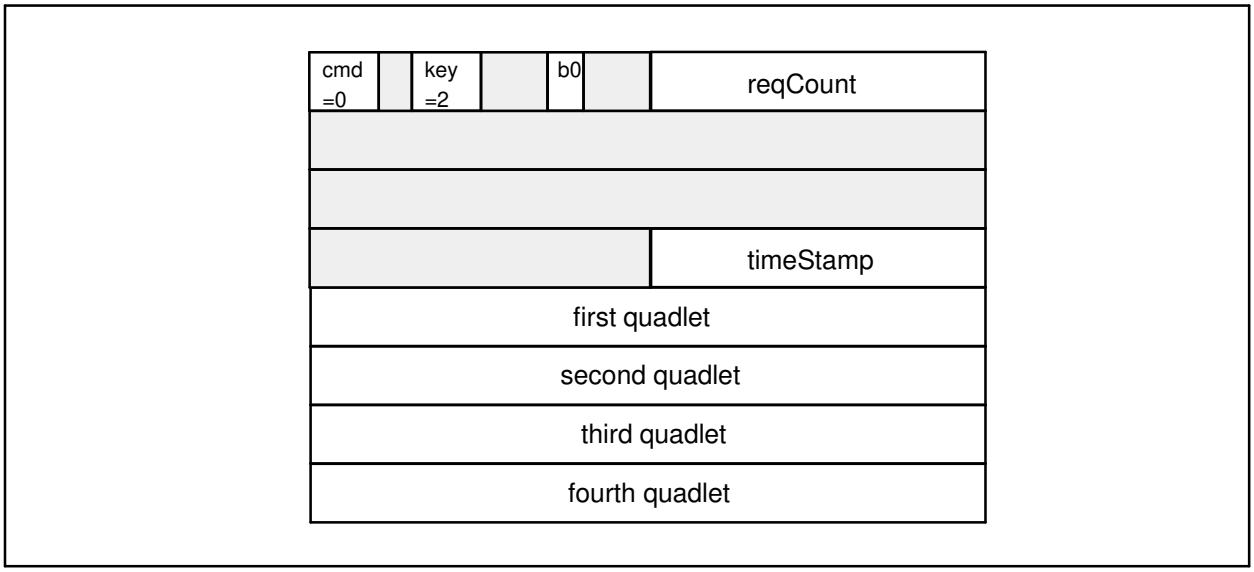


Fig. 8.4 OUTPUT_MORE_Immediate descriptor Format

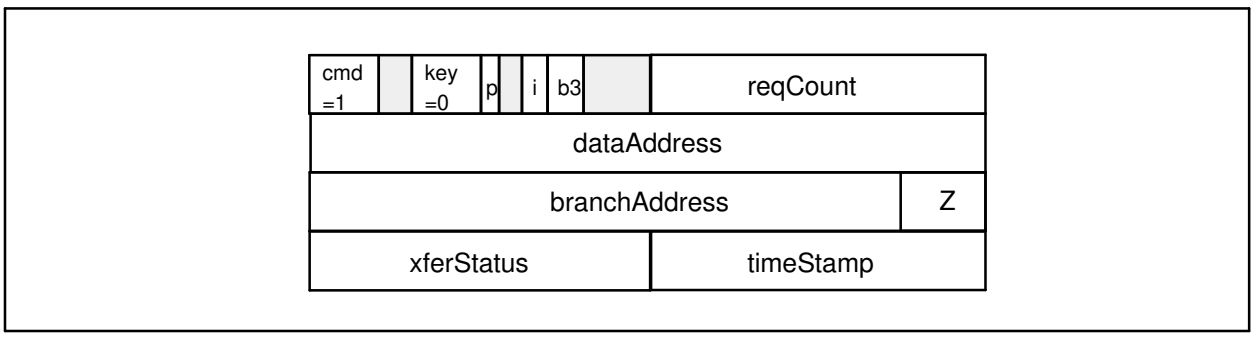


Fig. 8.5 OUTPUT_LAST descriptor Format

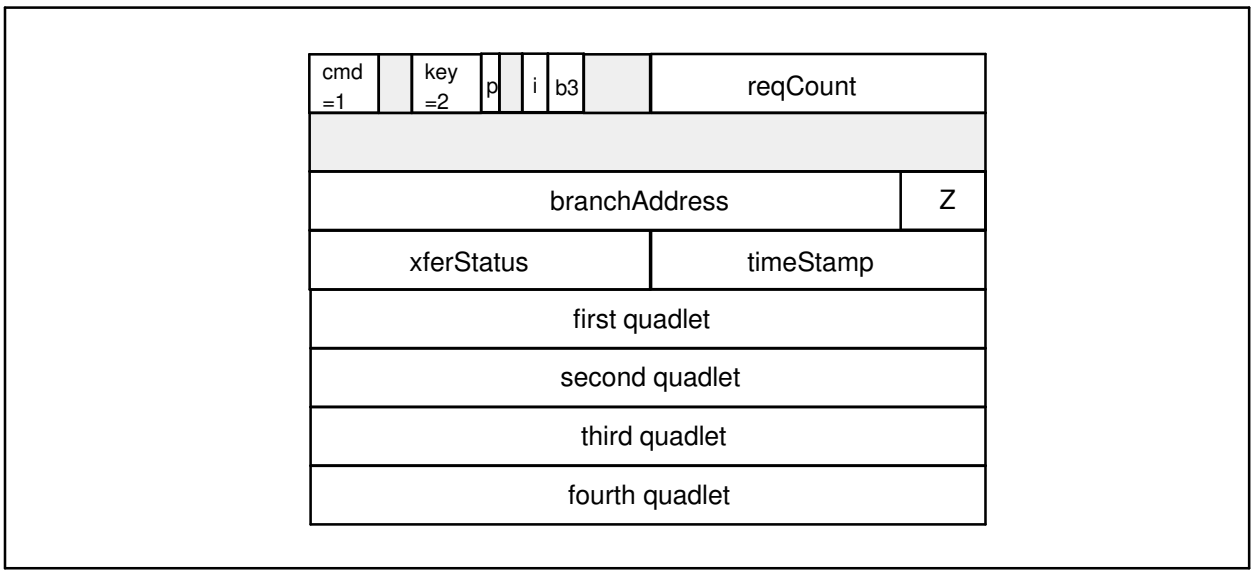


Fig. 8.6 OUTPUT_LAST_Immediate descriptor Format

Bit size	Field Name	description
4	cmd	”0h” means ”MORE” command. ”1h” means ”LAST” command.
3	key	”0h” means ”not immediate” command. ”2h” means ”immediate” command.
1	p	”1b” at this bit means the transmit packet is ping packet. This bit is used to count the time until the selfID packet or acknowledge, or the time until the subaction gap is detected after transmitting a packet. The time is counted in 50MHz and the counted time is indicated in timeStamp field.
2	i	This field is valid only for OUTPUT_LAST and OUTPUT_LAST__Immediate commands. It controls the interrupt event (IntEvent.reqTxComplete or .respTxComplete bit) reported after the descriptor is processed. ”11b” makes to report the interrupt. ”01b” enables the interrupt report only when the acknowledge except for ack_ complete and ack_pending is received. ”00b” does not report the interrupt. ”10b” is an unspecified code.
2	b	Set ”00b” for MORE command. Set ”11b” for LAST command.
16	reqCount	For OUTPUT_MORE and OUTPUT_LAST commands: Set the data byte count to be stored from a host memory into the AT- FIFO. For OUTPUT_MORE_Immediate and OUTPUT_LAST_Immediate commands: Set the byte count of packet header.
32	dataAddress	Set the start address of host memory where data to be stored in AT- FIFO are in.

Bit size	Field Name	description
28	branchAddress	Set the host memory address where the next descriptor to be processed is in. This field specification is valid only when Z field contains ”1” or larger value.
4	Z	Set the number of descriptor to be processed next. If no descriptor to be processed is outstanding, set ”0h” in this field. If any descriptor is existing, set ”2h” to ”8h” as outstanding. However, for the Immediate command, it requires 2 blocks of 16- byte field. Therefore, the minimum value for the Z field is ”2h”.
16	xferStatus	This field indicates the transfer result (status) every packet transmitted, following the ATContextControl register format.
16	timeStamp	For asynchronous response packet transmit : Set the timeout detection time. Packet transmission stops when the cycle timer value at which the packet transmission activates exceeds the timeout value. The time must be specified combining the IsoCycle Timer.cycleCount value + the lower 3- bit of cycleSeconds. This case is valid only for timeStamp field of Immediate command. For asynchronous request packet transmit : This field indicates the time when the packet transmission is completed, using the value for IsoCycleTimer.cycleCount + the lower 3- bit of cycleSeconds. In this case, the time is valid only for timeStamp field of OUTPUT_LAST and OUTPUT_LAST_Immediate commands. For ping packet transmit : This field indicates the time until the selfID packet or acknowledge is received after transmitting the packet or the time until the subaction gap is detected. Valid only for timeStamp field of OUTPUT_LAST and OUTPUT_LAST_Immediate commands.
32	quadlet	Store the packet header in this field.

8.2. Asynchronous Receive

An example of software flowchart for receiving an asynchronous request packet or response packet is shown in Figure 8.7. The same process should be required in the case of PHY packet (phy configuration packet, link on packet, and ping packet), except that LinkControl.rcvPhyPkt bit must be set in advance.

1) Check ARContextControl.active bit and make sure it is cleared.

2) Store the context program in an host memory.

3) Set the address of host memory where the context program is stored, in the ARcommandPtr.descriptorAddress field. Also set the value ”1” in the ARCommandPtr.Z field. (Always setting ”1” means that MB86613S device does support only buffer- fill mode.)

4) Set the ARContextControl.run bit.

5) Wait until the INTA# is set. (when the last descriptor’s ” i ” field contains 11b and IntMask.ARRQ or .ARRS bit is set.)

6) Check the ARContextControl.eventcode field.

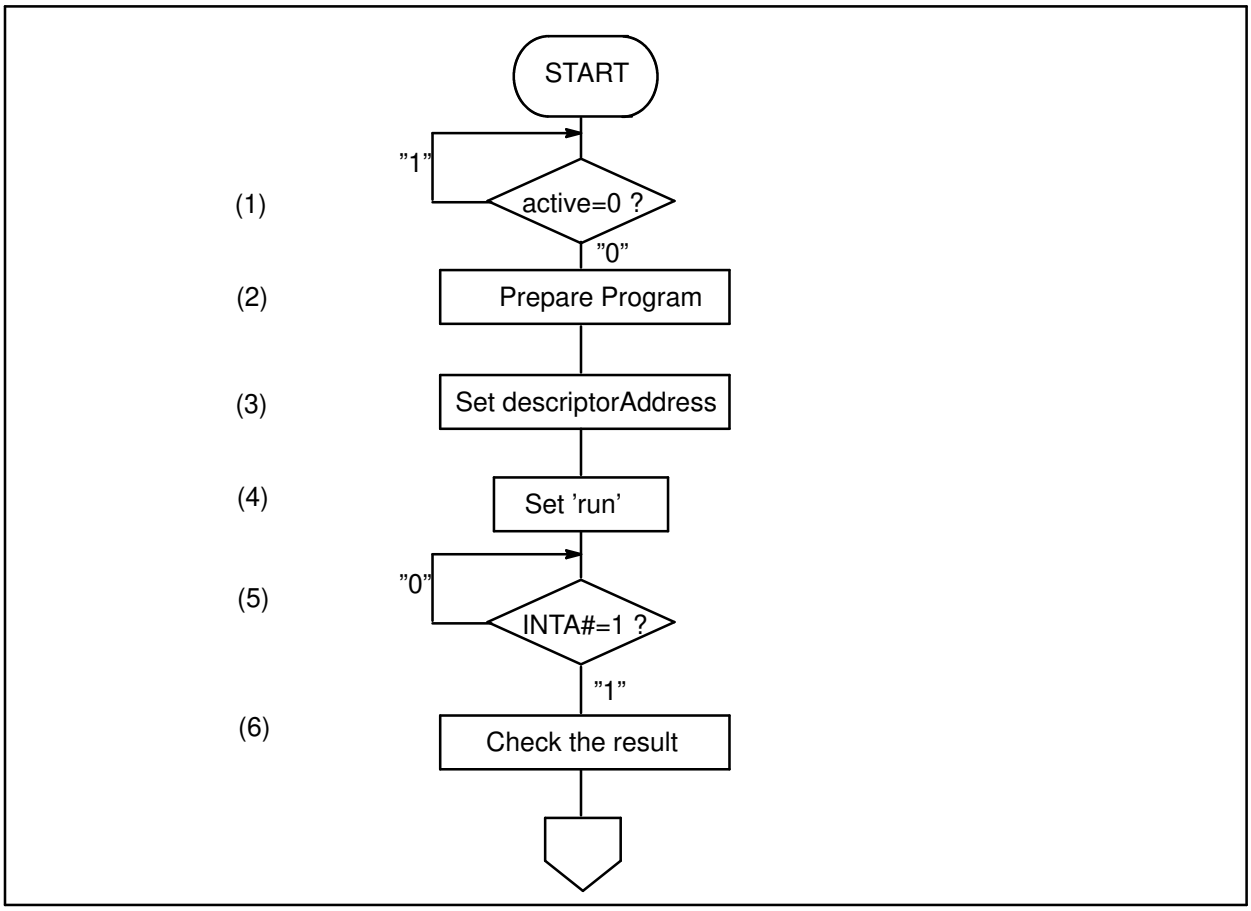


Fig. 8.7 Software Flow Chart for asynchronous packet Receive

8.2.1. How to Make Context Program

Figure 8.8 shows an example of context program format in buffer- fill mode, which is applied when storing a received asynchronous packet into a host memory whose area size is 1024- byte.

ARCommandPtr.descriptorAddress field must contain the value ”80000000h” in advance that is the start address of host memory where the program is stored. Besides, ARCommandPtr.Z field must contain a value ”1h” that is the number of descriptor.

Since the way to store the received packet follows buffer- fill mode, a context program is composed of one INPUT_MORE command. In the example, the reqCount and the resCount field of descriptor- 1 contain ”400h” which is the host memory size and dataAddress field contains ”10000000h” which is the start address of host memory that intends to store the packet. Also, the branchAddress field contains ”90000000h” which is the start address of host memory where the next context program- 2 is stored.

” i ” field contains ”11b” for the INPUT_MORE command in the example. So, the device reports the interrupt event when the program is finished.

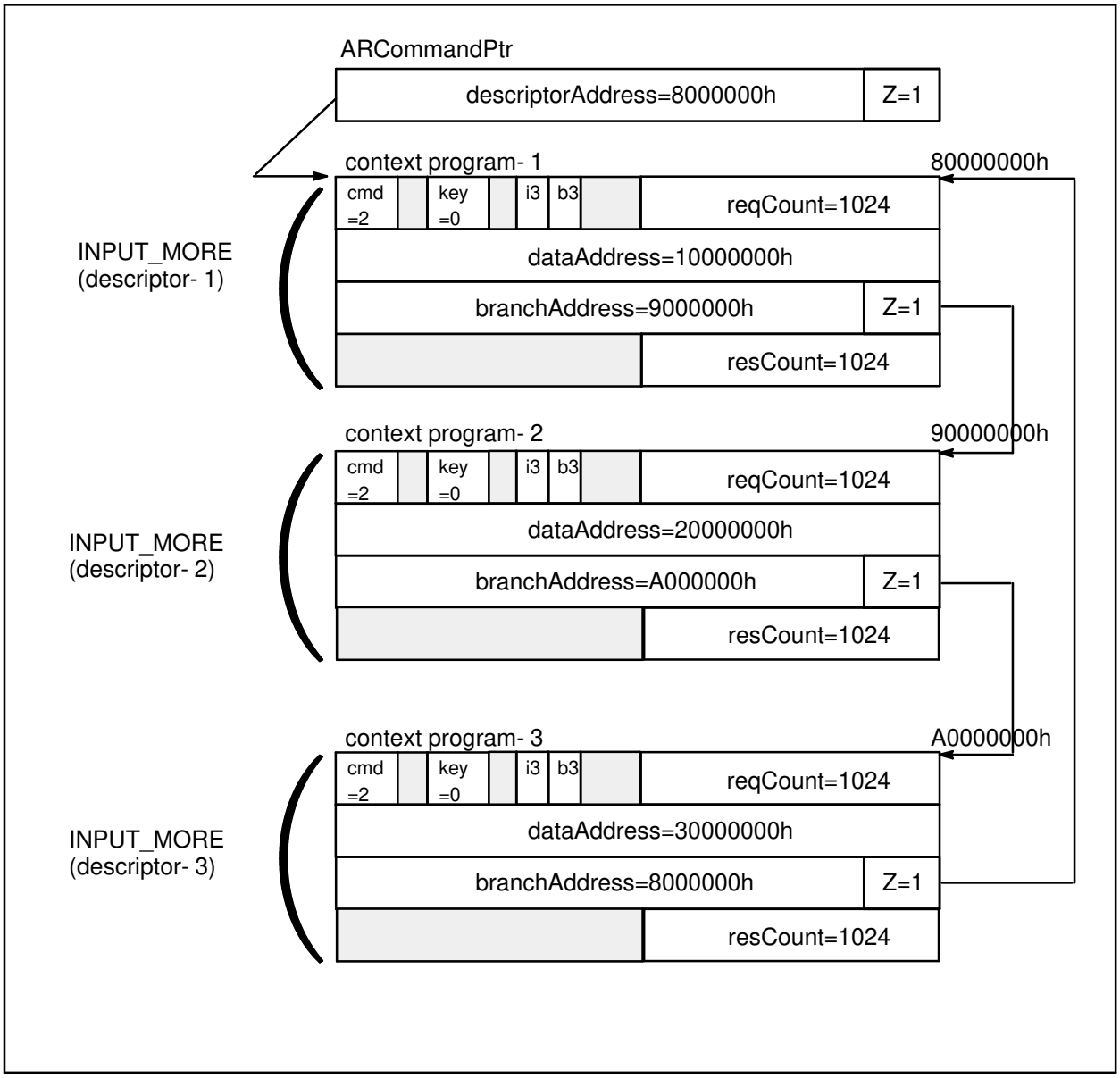


Fig. 8.8 Example of context program Format for asynchronous packet Receive

8.2.2. Descriptor

This section describes the format of descriptor processed by AR- CPC. The descriptor that AR- CPC handles is only INPUT_MORE command.

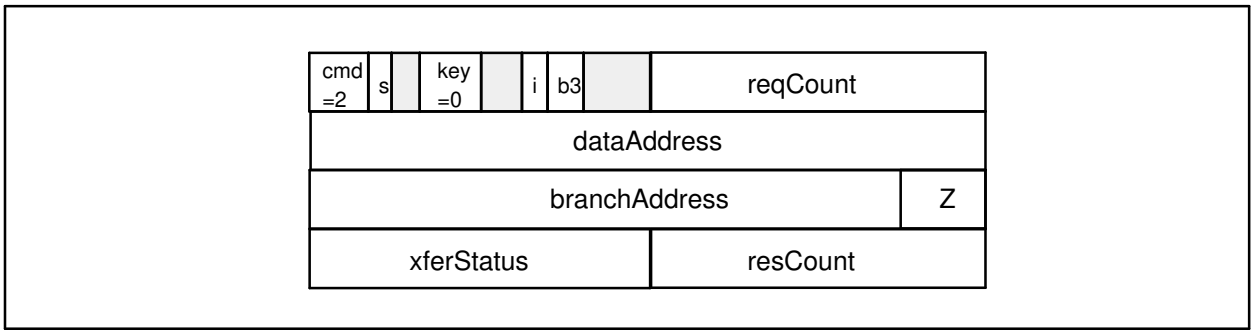


Fig. 8.9 INPUT_MORE descriptor Format

Bit size	Field Name	description
4	cmd	Set ”2h” meaning to ”INPUT MORE” command.
1	s	The result is stored in xferStatus and resCount fields.
3	key	Always set ”0h” in this field.
2	i	It controls the interrupt event (IntEvent.ARRQ and .ARRS bits). reported after the descriptor is processed. ”11b” makes to report the interrupt. ”00b” does not report the interrupt. ”10b” and ”01b” are unspecified codes.
2	b	Always set ”11b” in this field.
16	reqCount	Set the data byte count to be stored in host memory in a multiple of 4.
32	dataAddress	Set the start address of host memory.
28	branchAddress	Set the address of host memory where the next descriptor to be processed is stored. The setting is valid only when Z field contains ”1”.
4	Z	Set ”1h” if the next descriptor to be processed is existing. Set ”0h” if no outstanding descriptor is existing.
16	xferStatus	This field indicates the result of process every packet to store in the host memory, in ARContextControl register format.
16	resCount	Set the byte count of the host memory where the packet is to be stored with a multiple of 4. This field indicates the remaining byte count of host memory while storing the received packet into the host memory.

8.3. Isochronous Transmit

An example of software flowchart for transmitting an isochronous packet is shown in Figure 8.10.

1) Check ITContextControl.active bit and make sure it is cleared.

2) Store the data to transmit in an host memory, except for no data isochronous packet.

3) Store the context program in the host memory. In this case, also set the packet header in the lower 16- bit field of the first descriptor (i.e., immediate command).

4) Set the transmit start time in the ITContextControl.cycleMatch field (only when ITContextControl.cycle MatchEnable bit is set).
5) Set the start address of host memory where the context program is stored, in the ITCommandPtr.descriptorAddress field. Also set the count; (number of descriptor + 1), since the first descriptor is an immediate command, in the ITCommandPtr.Z field.

6) Set the ITContextControl.run bit.

7) Wait until the INTA# is set. (when the last descriptor's " i " field contains 11b or 01b and IsoXmitIntMask.isoXmitN bit is set.)

8) Check the ITContextControl.eventcode field.

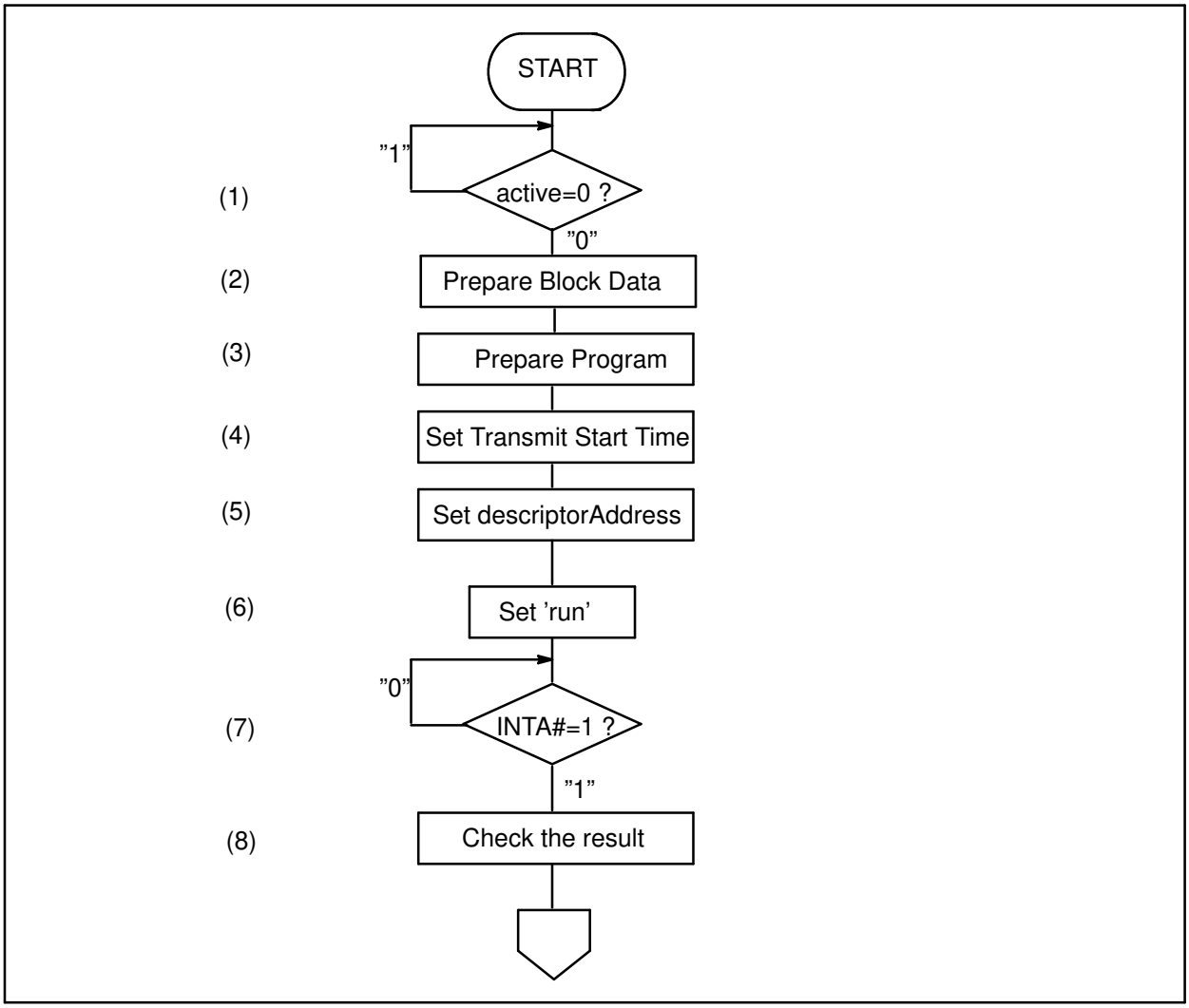


Fig. 8.10 Software Flow Chart for isochronous packet Transmit

8.3.1. How to Make Context Program

Figure 8.11 shows an example of context program format when transmitting an isochronous packet having 256- byte data.

ITCommandPtr.descriptorAddress field must contain the value "80000000h" in advance that is the start address of host memory where the program is stored. Besides, ITCommandPtr.Z field must contain a value "3h" that is the number of descriptor.

When making the context program, the first descriptor must be OUTPUT_MORE_Immediate or OUTPUT_LAST_Immediate command in order to generate the packet header. In the example, it uses the OUTPUT_MORE_Immediate command because the packet data is to be set.

reqCount for the OUTPUT_MORE_Immediate command contains "08h" because the packet header is 8- byte length, and the lower 4- quadlet field of the command contains the packet header in Open HCI format.

Also, in order to create a path to return to the program when a bus reset or cycle lost occurs, skipAddress field contains "80000000h" which is the start address of the program.

In the example, it uses OUTPUT_LAST command following the OUTPUT_MORE_Immediate command because the packet data are contained in one host memory. reqCount field and dataAddress field contain "100h"

and ”10000000h” (start address of host memory where the packet data are contained). Since the example further assumes that another packet is transmitted in the next isochronous cycle, branchAddress contains ”90000000h” which is the start address of host memory where contains a context program to control the packet transmission to be done in the next cycle and Z field contains ”3h” which is the number of descriptor.

The ” i ” field of the OUTPUT_LAST command is being set to ”11b” so that an interrupt event is reported after completion of isochronous packet transmit.

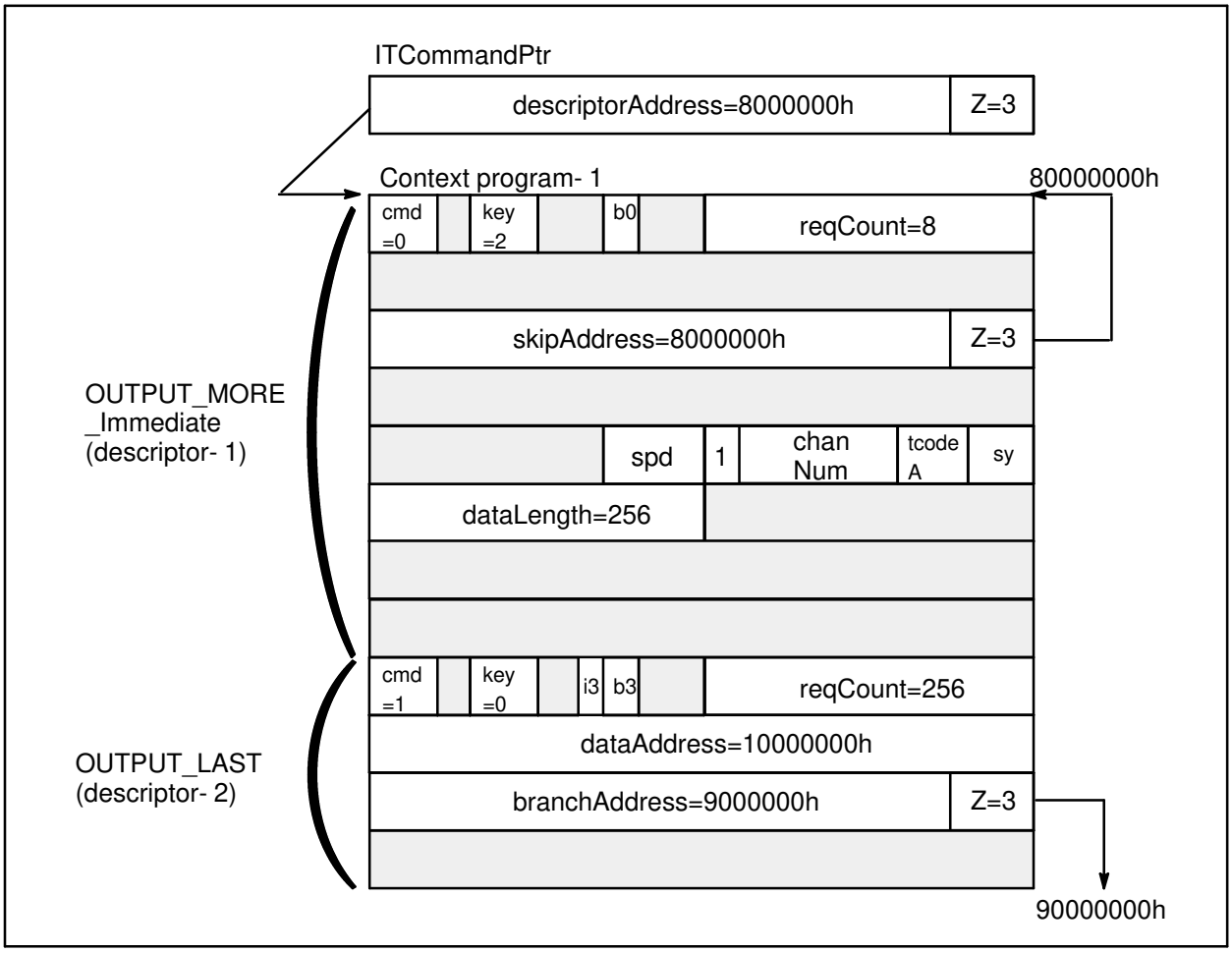


Fig. 8.11 Example of context program Format for isochronous packet Transmit

8.3.2. Descriptor

This section describes the format of descriptor processed by IT- CPC. The descriptor(s) that IT- CPC handles are OUTPUT_MORE command, OUTPUT_LAST command, OUTPUT_MORE_Immediate command, OUTPUT_LAST_Immediate command, and STORE_VALUE command. If the first descriptor is OUTPUT_LAST command and the reqCount field contains ”0”, no packet is transmitted and one cycle is skipped. So, when there is a need that does not want to transmit a packet in certain isochronous cycle, use OUTPUT_LAST command having the reqCount=0. Also, for Immediate command, specify the packet header to the lower 16- byte field.

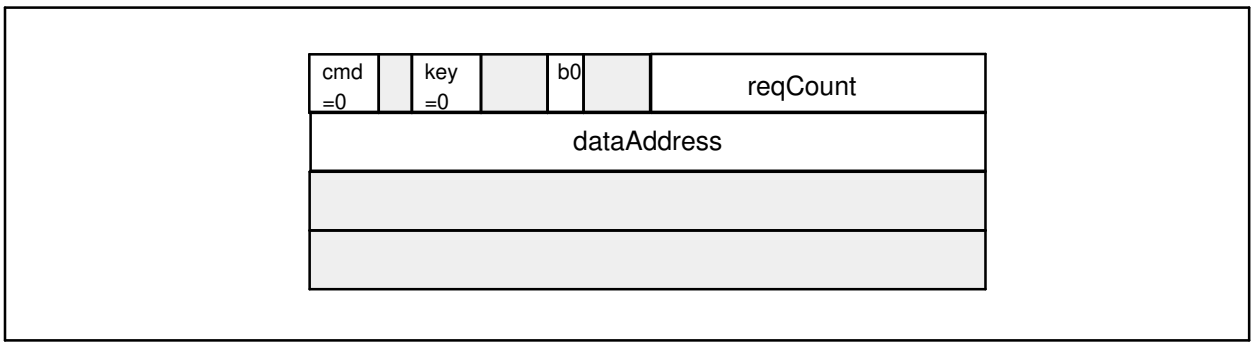


Fig. 8.12 OUTPUT_MORE descriptor Format

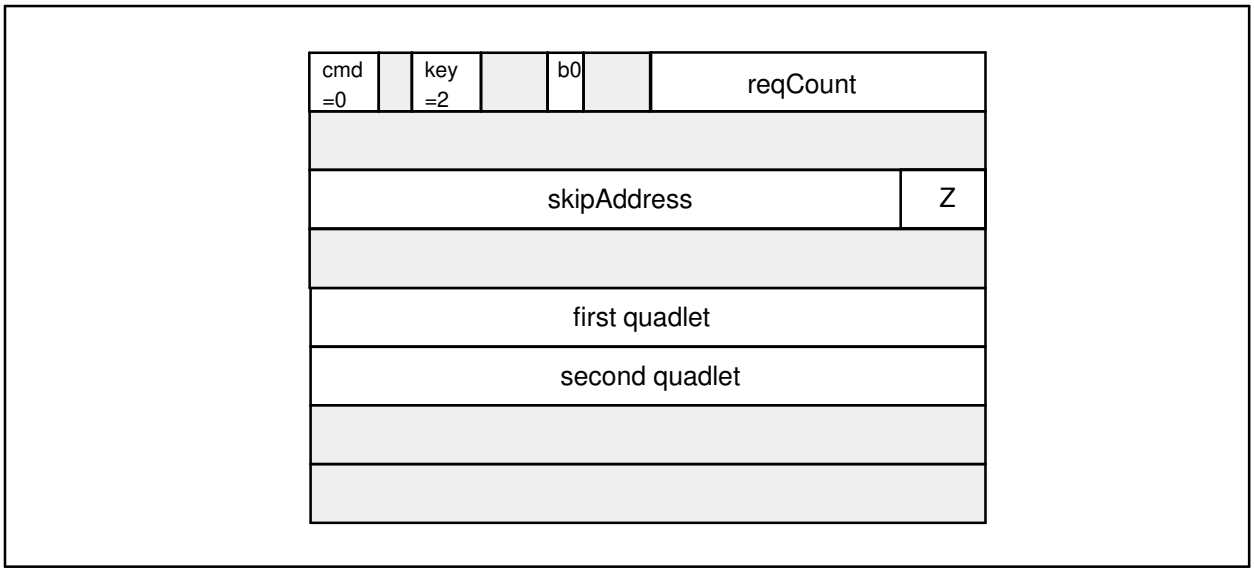


Fig. 8.13 OUTPUT_MORE_Immediate descriptor Format

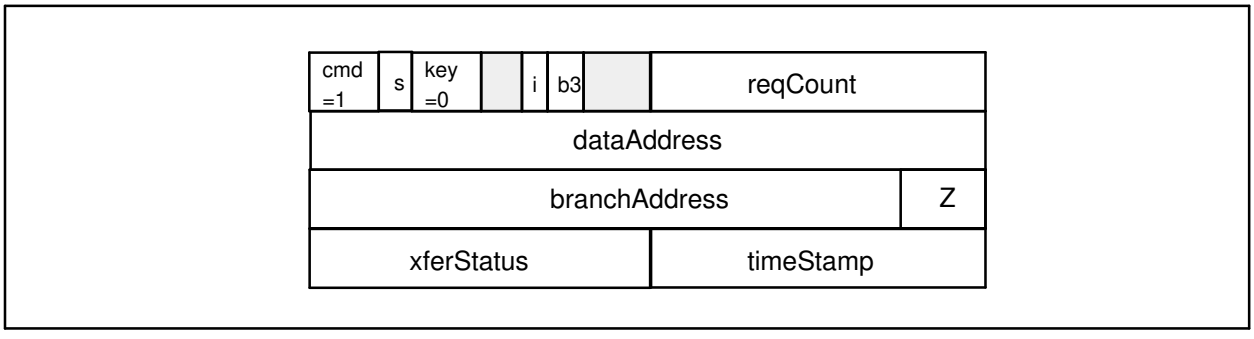


Fig. 8.14 OUTPUT_LAST descriptor Format

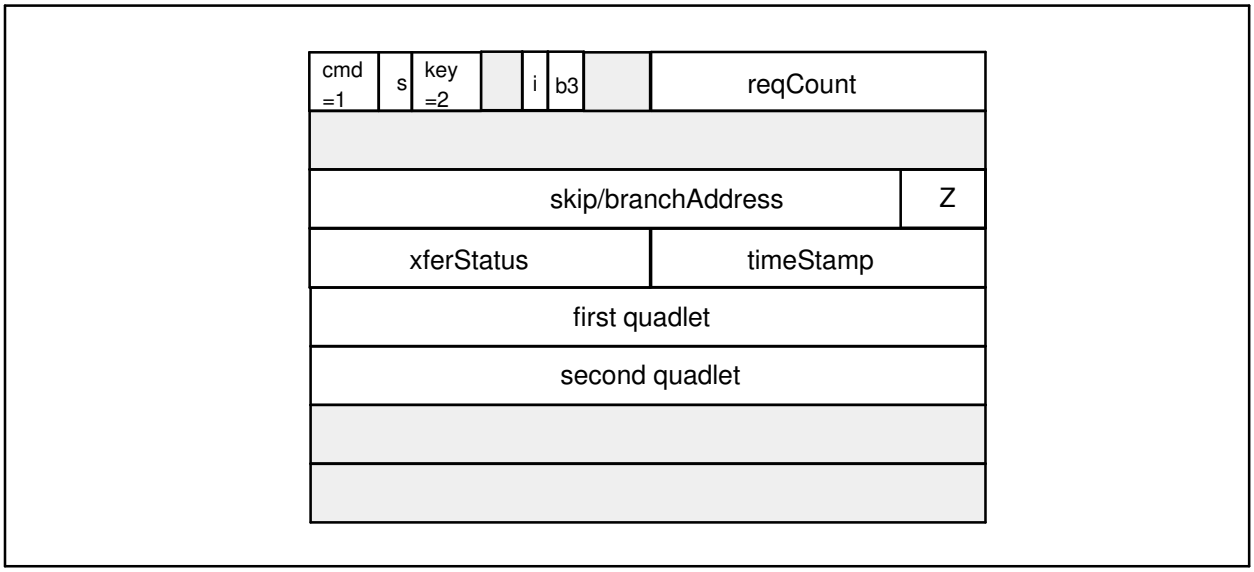


Fig. 8.15 OUTPUT_LAST_Immediate descriptor Format

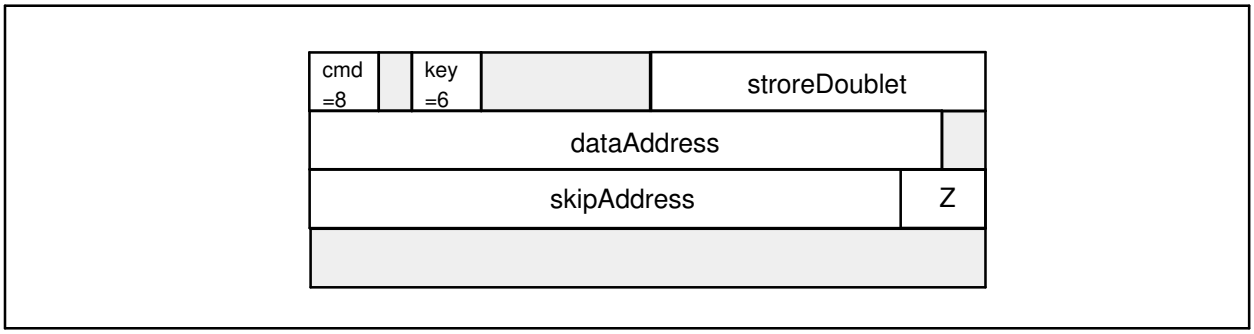


Fig. 8.16 STORE_VALUE descriptor Format

Bit size	Field Name	description
4	cmd	”0h” means ”MORE” command. ”1h” means ”LAST” command.
1	s	This flag is valid only for OUTPUT_LAST and OUTPUT_LAST_Immediate commands. ”1b” stores the result of execution in the xferStatus and resCount fields. ”0b” does not store the result of execution in the xferStatus and resCount fields.
3	key	”0h” means ”not immediate” command. ”2h” means ”immediate” command. ”6h” means STORE_VALUE command.
2	i	This field is valid only for OUTPUT_LAST and OUTPUT_LAST_Immediate commands. It controls the interrupt event (IntEvent.IsochTx bit) reported after the descriptor is processed. ”11b” makes to report the interrupt. ”00b” does not report the interrupt. ”10b” and ”01b” are unspecified codes.

2	b	Always set ”11b”.
16	reqCount	For OUTPUT_MORE and OUTPUT_LAST commands: Set the data byte count to be stored from a host memory into the IT- FIFO. For OUTPUT_MORE_Immediate and OUTPUT_LAST_Immediate commands: Set ”0008h” which is the header’s byte count.
32	dataAddress	For OUTPUT_MORE and OUTPUT_LAST commands: Set the start address of host memory where the data to be stored in IT- FIFO. For STORE_VALUE command : Set the address of host memory where the data stored in the storeDoublet field is in. (The upper 16- bit is specified with ”0”.)
28	branchAddress	Set the address of host memory to be processed next. This setting is valid only when the Z field contains ”1” or greater value.
	skipAddress	Set the address of host memory where the descriptor for bus reset process or cycle lost case is stored. For OUTPUT_LAST command, the skipAddress setting is valid only when the reqCount field contains ”0” and it is the first descriptor.
4	Z	Set the number of descriptor to be processed next. If no descriptor to be processed is outstanding, set ”0h” in this field. If any descriptor is existing, set ”1h” to ”8h” as outstanding. However, setting ”1h” must happen only when the next processed descriptor is OUTPUT_LAST command and that descriptor has ”0” in the reqCount field.
16	xferStatus	This field indicates the transfer result (status) every packet transmitted, following the ITContextControl register format. Valid only when s flag is set.
16	timeStamp	Set the timeout detection time to be stored in the xferStatus field. The time is indicated in a value for (IsoCycleTimer.cycleCount value + Lower 3- bit of cycleSeconds.)
16	storeDoublet	Set the 16- bit data to be stored in host memory. Actually zero data is padded in the upper 16- bit, and the 32- bit data is stored in the memory.
32	quadlet	Store the packet header in this field.

8.4. Isochronous Receive

An example of software flowchart for receiving an isochronous packet is shown in Figure 8.17.

1) Check IRContextControl.active bit and make sure it is cleared.

2) Store the context program in an host memory.

3) Set the address of host memory where the context program is stored, in the IRCommandPtr.descriptorAddress field. Also set the number of descriptor in the IRcommandPtr.Z field. (Always set "1" for buffer- fill mode.)

4) Set the context program start time in IRContextMatch.cycleMatch field. (only when IRContextControl.cycleMatchEnable is set.)

5) For single- channel mode, clear IRContextControl.multiChanMode bit and set the receiving channel in IRContextMatch.channelNumber field.
For multi- channel mode, set the IRContextControl.multiChanMode bit and set the receiving channels in IRMultiChanMask.isoChannelN field.
Then, specify the packet store method (buffer- fill or packet- per- buffer mode) in contextcontrol.bufferFill bit.

6) When storing the received packet after removing the packet header, clear the IRContextControl.isochHeader field. Set the comparison values with the tag and sync data in the IRContextMatch.tag[3:0] bits and sync field respectively.

7) Set the IRContextControl.run bit.

5) Wait until the INTA# is set. (when the last descriptor's " i " field contains 11b and isoRecvIntMask.isoRecvN bit is set.)

6) Check the IRContextControl.eventcode field.

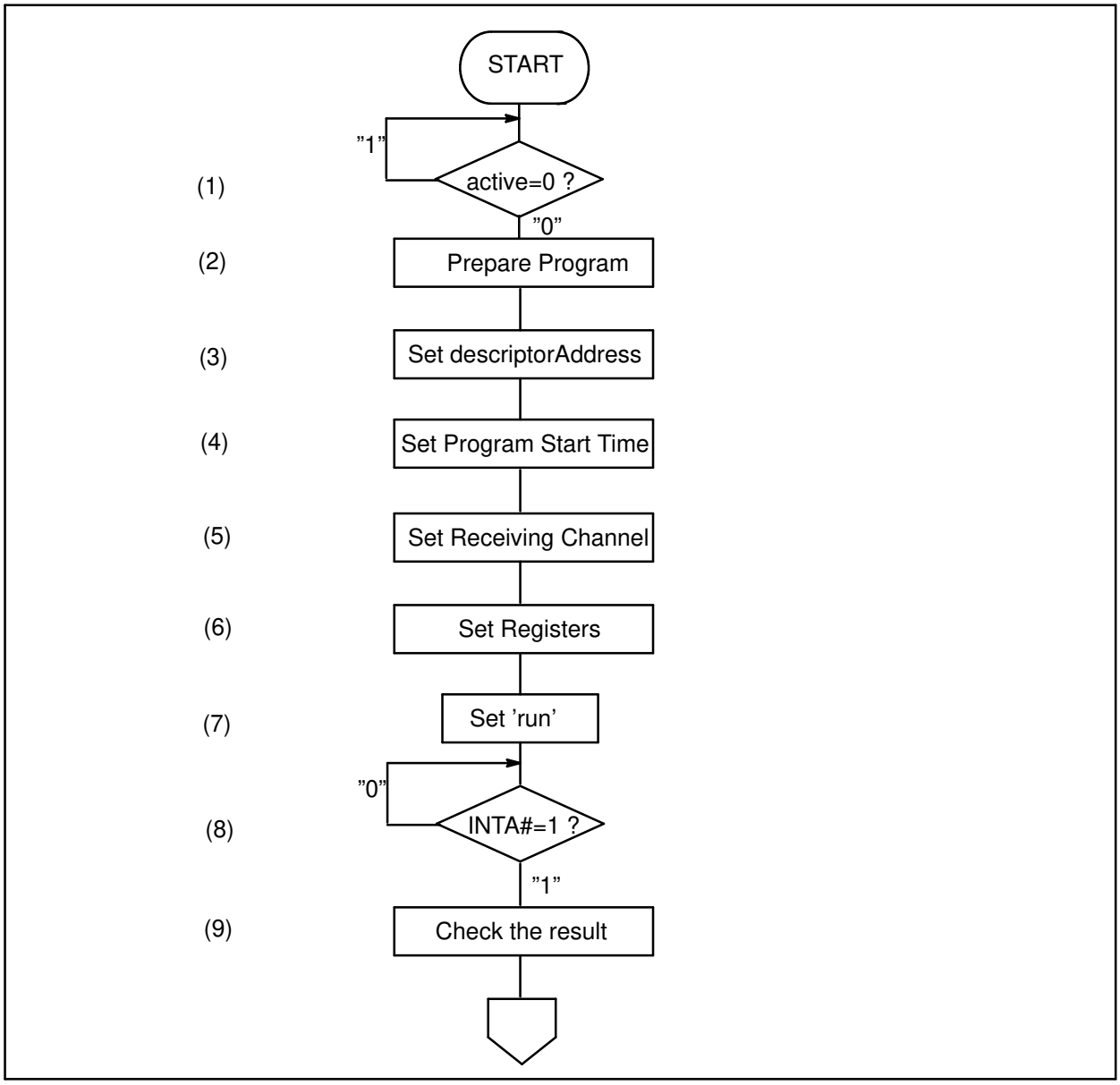


Fig. 8.17 Software Flow Chart for isochronous packet Receive

8.4.1. How to Make Context Program

Figure 8.18 shows an example of context program format when storing an isochronous packet in two host memory areas where each has 1024- byte space, in packet- per- buffer mode.

IRCommandPtr.descriptorAddress field must contain the value "80000000h" in advance that is the start address of host memory where the program is stored. Besides, IRCommandPtr.Z field must contain a value "2h" that is the number of descriptor.

Because the received packet is stored in packet- per- buffer mode, the context program is composed of multiple INPUT_MORE commands and one INPUT_LAST command. In the example, the context program is composed of one INPUT_MORE and one INPUT_LAST commands.

The reqCount and the resCount fields of descriptor- 1 and - 2 contain "400h" which is the host memory size,

and dataAddress field contains ”10000000h” which is the start address of host memory where the packet is stored for descriptor- 1 and ”20000000h” for descriptor- 2. Also, the branchAddress field for descriptor- 2 contains ”90000000h” which is the start address of host memory where the context program- 2 to control the host memory for storing the next packet is stored and the Z field contains ”2h” which is a number of descriptor in context program- 2.

The ” i ” field of the INPUT_LAST command is being set to ”11b” so that an interrupt event is reported after completion of isochronous packet receive process.

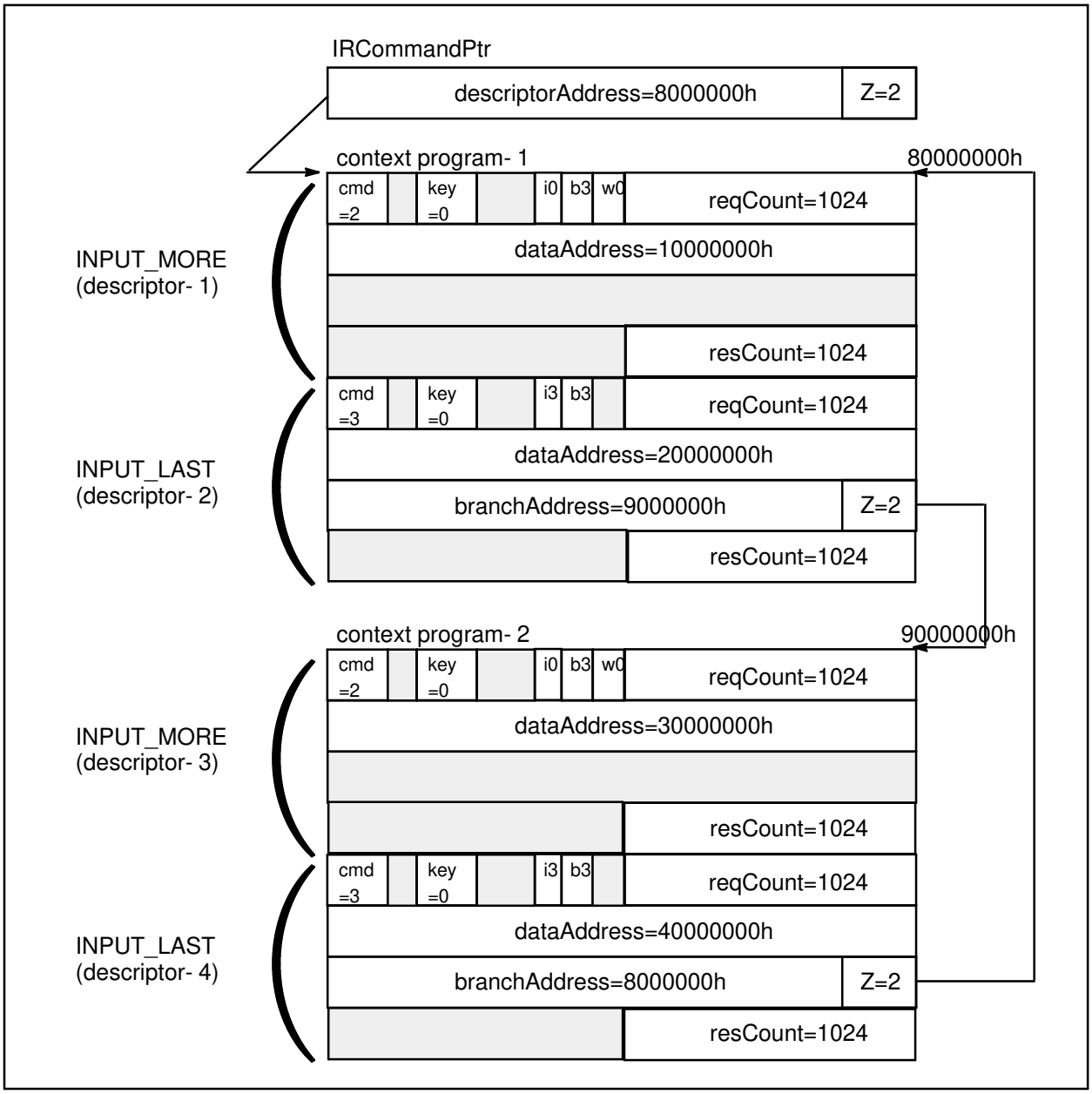


Fig. 8.18 Example of context program Format for isochronous packet Receive

8.4.2. Descriptor

This section describes the format of descriptor processed by IR- CPC. The descriptor(s) that IR- CPC handles are INPUT_MORE command and INPUT_LAST command.

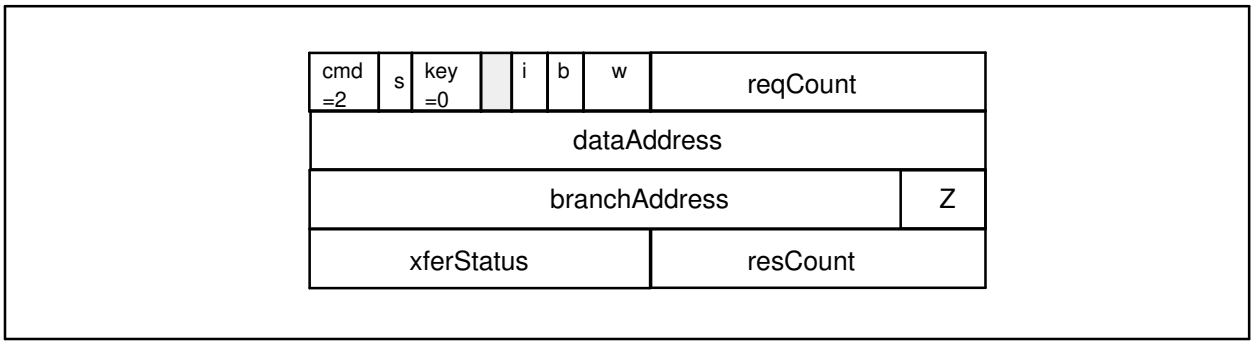


Fig. 8.19 INPUT_MORE descriptor Format

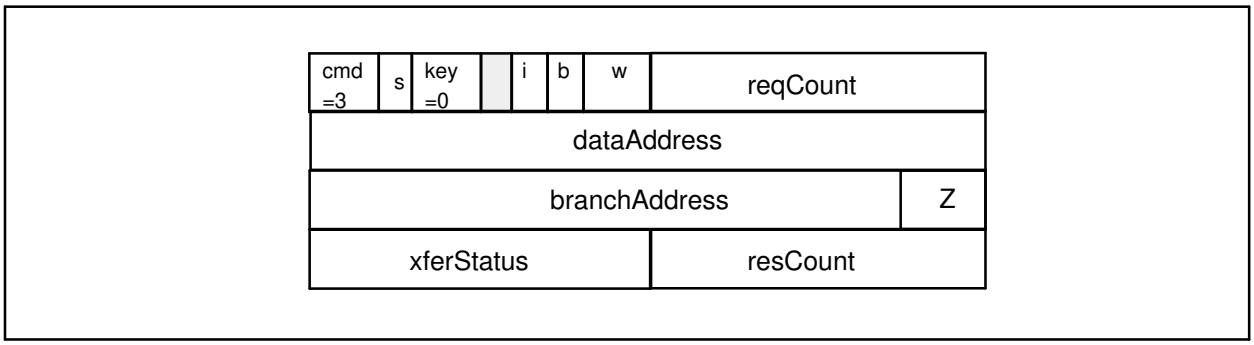


Fig. 8.20 INPUT_LAST descriptor Format

Bit size	Field Name	description
4	cmd	”2h” means ”INPUT_MORE” command. ”3h” means ”INPUT_LAST” command.
1	s	This flag is valid only for packet- per- buffer mode. ”1b” stores the result of execution in the xferStatus and resCount fields. ”0b” does not store the result of execution in the xferStatus and resCount fields.
3	key	Always set ”0h”.
2	i	This field controls the interrupt event (IntEvent.IsochRx bit) reported after the descriptor is processed. ”11b” makes to report the interrupt. ”00b” does not report the interrupt. ”10b” and ”01b” are unspecified codes. In packet- per- buffer mode, only the ” i ” field of INPUT_LAST command is valid.
2	b	For buffer- fill mode: Always set ”11b”. For packet- per- buffer mode : Set ”00b” for INPUT_MORE command. Set ”11b” for INPUT_LAST command.

2	w	”11b” does not start to store packets until a packet specified in IRContextMatch. sync field is received. ”00b” starts to store packets when incoming. ”01b” and ”10b” are unspecified codes. In packet- per- buffer mode, only the ”w” field at the first descriptor of context program is valid.
16	reqCount	Set the data byte count of host memory to store the packets.
32	dataAddress	Set the start address of host memory to store the packets.
28	branchAddress	Set the address of host memory where the next descriptor to be processed is stored. This setting is valid only when the Z field contains ”1” or greater value.
4	Z	For buffer- fill mode : Set ”1h” if there is the next descriptor to be processed. Set ”0h” if there is no descriptor next. For packet- per- buffer mode : Set the number of descriptor to be processed next. Only ”Z” field in INPUT_LAST command is valid. Set ”0h” if there is no descriptor next. Set ”1h” to ”8h” as the number if there is (are) descriptor(s).
16	xferStatus	This field indicates the transfer result (status) every packet stored in the memory, following the IRContextControl register format. However, for packet- per- buffer mode, the setting is valid only when s flag is set.
16	resCount	Set the byte count of the host memory where the packet is to be stored. This field indicates the remaining byte count. However, for packet- per- buffer mode, the setting is valid only when s flag is set.

8.5. Self ID

An example of software flowchart for receiving a self ID packet is shown in Figure 8.21. There is no context program to receive self ID packets.

1) Clear LinkControl.rcvSelfID bit.

2) Set the host memory base address (in 2048- byte boundary) where the received packet is stored in the selfIDBufferPointer register.

3) Set LinkControl.rcvSelfID bit.

4) Wait until the INTA# is set. (when IntMask.selfIDComplete bit)

5) Check the SelfIDCount.selfIDError bit.

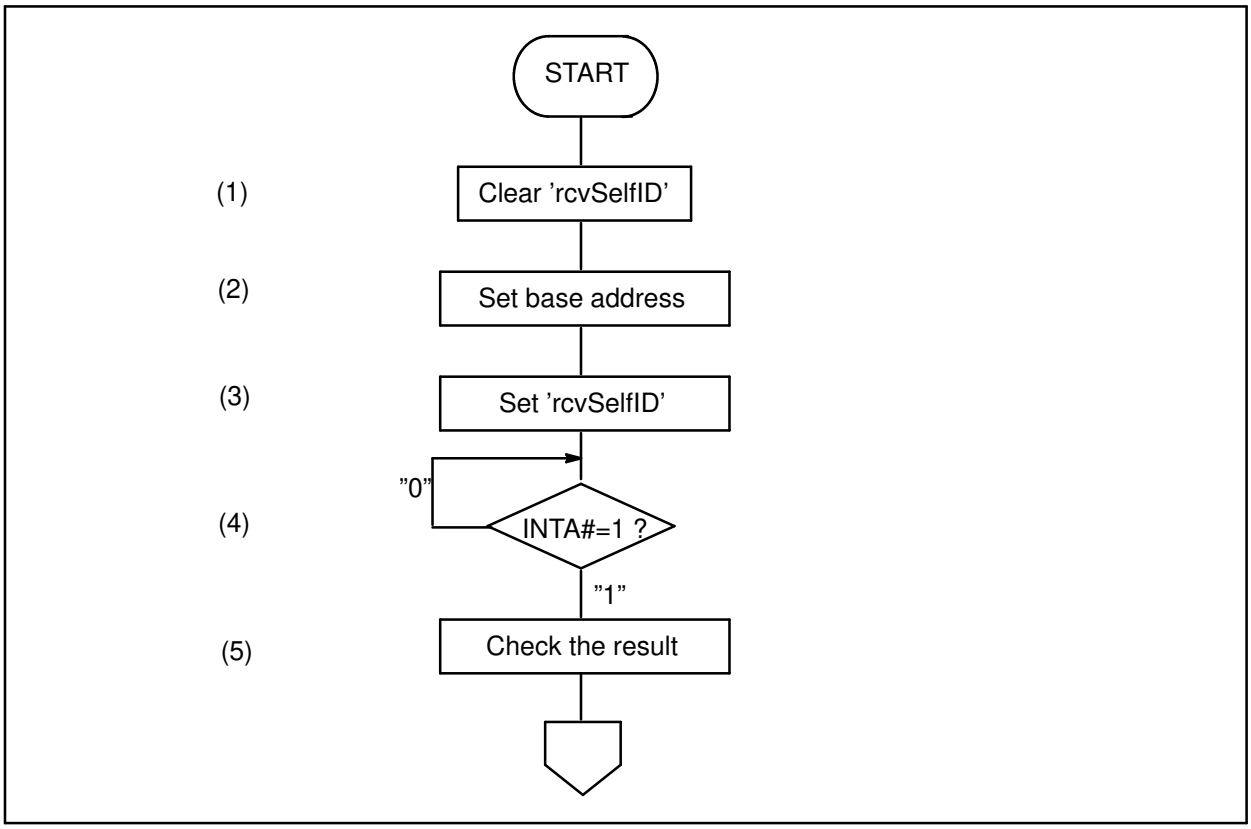


Fig. 8.21 Software Flow Chart for Self ID Packet Receive

8.5.1. Packet Format

Figure 8.22 shows the format of self ID packet when it is stored in a host memory.

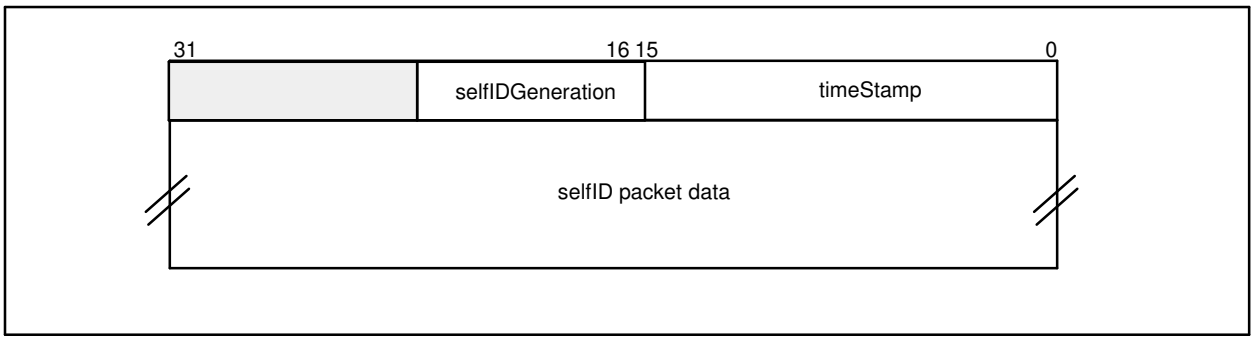


Fig. 8.22 Self ID Packet Format

Bit size	Field Name	description
8	selfIDGeneration	This field indicates the number of bus reset occurrence.
16	timeStamp	This field shows the time indicated with (IsoCycleTimer.cycleCount value + the lower 3- bit of cycleSeconds).

8.6. Notes on Software Implementation

This section describes the software implementations and the corresponding event happened.

1) Bus Reset:

   Set the IBR bit of PHY register :
   1) when the received self ID packet is abnormal.
   2) when IntEvent.cycleTooLong is detected.
   3) when a topology needs to change at bus manager operation.

2) Generation of Response Packet :

   when a read request packet is received for speed map or topology map is received.

3) Cycle Master Setting :

   Set LinkControl.cycleMaster bit :
   1) when the device is a root and STATE_CLEAR.cmstr bit (”FFFF_F000_0000h”) is set by the write request packet from a bus manager.

4) Power Management (when at bus manager operation) :

   Transmit a link- on packet:
   when enough cable power can be provided based on a calculation of pwr field in selfID packet.

5) Cycle Master (when at bus manager operation) :

   Set the force_root bit by sending the phy configuration packet to a root node, and set STATE_CLEAR. cmstr bit by sending the write request packet.

   Make sure that cmc bit in bus_Info_block where 1394 configuration ROM of a root node contains is set.

6) Change Root Node :

   Search for the node who has it’s cmc bit = ”1” in the 1394 configuration ROM and set the force_root bit of found node by the phy configuration packet :
   when the current root does not have the cycle master function.

7) Construction of Speed Map and Topology Map :

   When the self ID packet is received.

8) Intention to be Bus Manager :

   Intend to be a bus manager by sending a lock request packet to the resource manager:
   when it want to be a bus manager.

9) Management of priority budget Register :

   Store the pri_max value of priority budget register in the pri_req field of Fairness Control register:
   when the write request packet is received from the bus manager to priority budget register (”FFFF_F000_0218h”).

10) Management of LINK ON (when at bus manager operation) :

    Clear PacketControl.autoLinkPkt bit :
    when it wants to be able to transmit or receive the Link- on packet.

11) Verification of Self ID Packet (when at resource manager) :

Need to verify the following items for the received self ID packet :

a) The inverted data of the first quadlet equals to the second quadlet.
b) All the ports on first received self ID packet are child ports.
c) Self ID packet is received in order of node number ”0”.

8.7. Device Activation

When activating the MB86613S device, software needs to take the following steps :

1) Execute hardware reset.
2) Initialize the device internal registers.
3) Enable the LINK, by setting HCControl.linkEnable bit.
4) Execute the bus reset, setting IBR bit of PHY register.

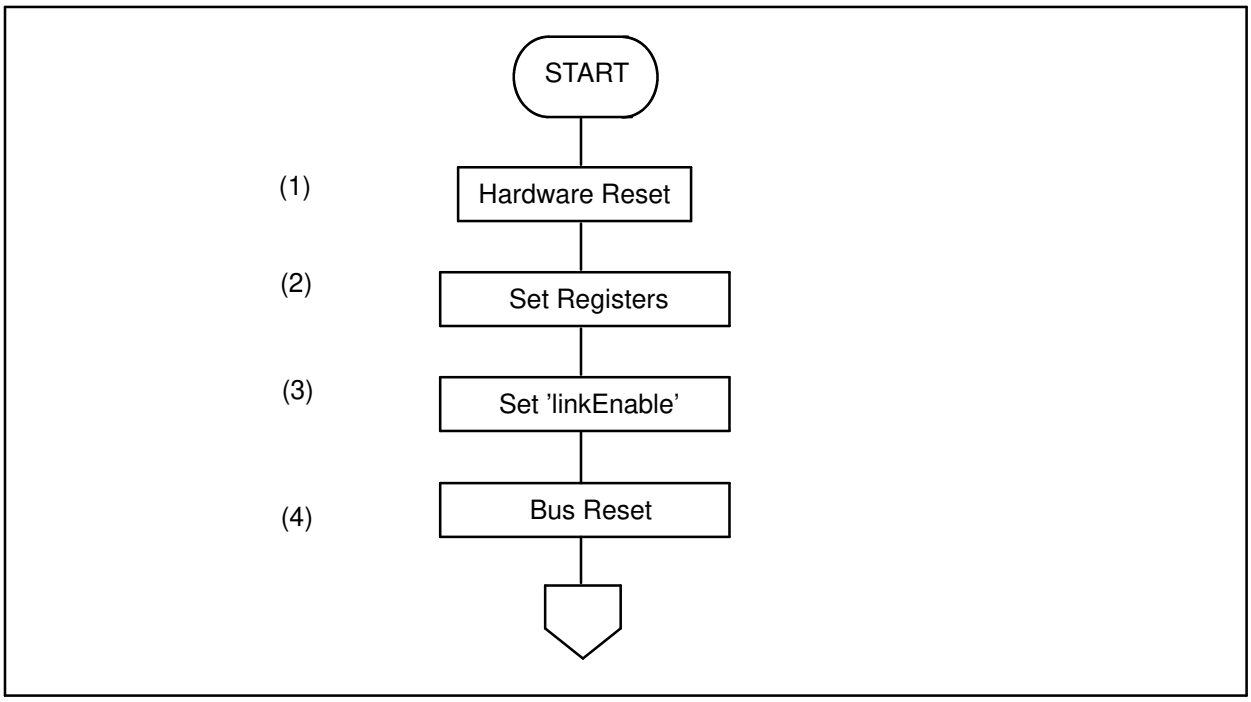


Fig. 8.23 Software Flow Chart for Device Activation

Annex 1. Debug Registers

Debug register consists of Vendor Option register (128 bytes) and PCI- 1394 Direct Transfer Mode register (128 bytes).

Base address that determines the addressing space for these register set is located in the last 2048- byte space of 4096- byte which is set in the MEM Base Address register in PCI configuration register.

In order to enable these registers, set the DEBUG_MODE bit of the EEPROM.

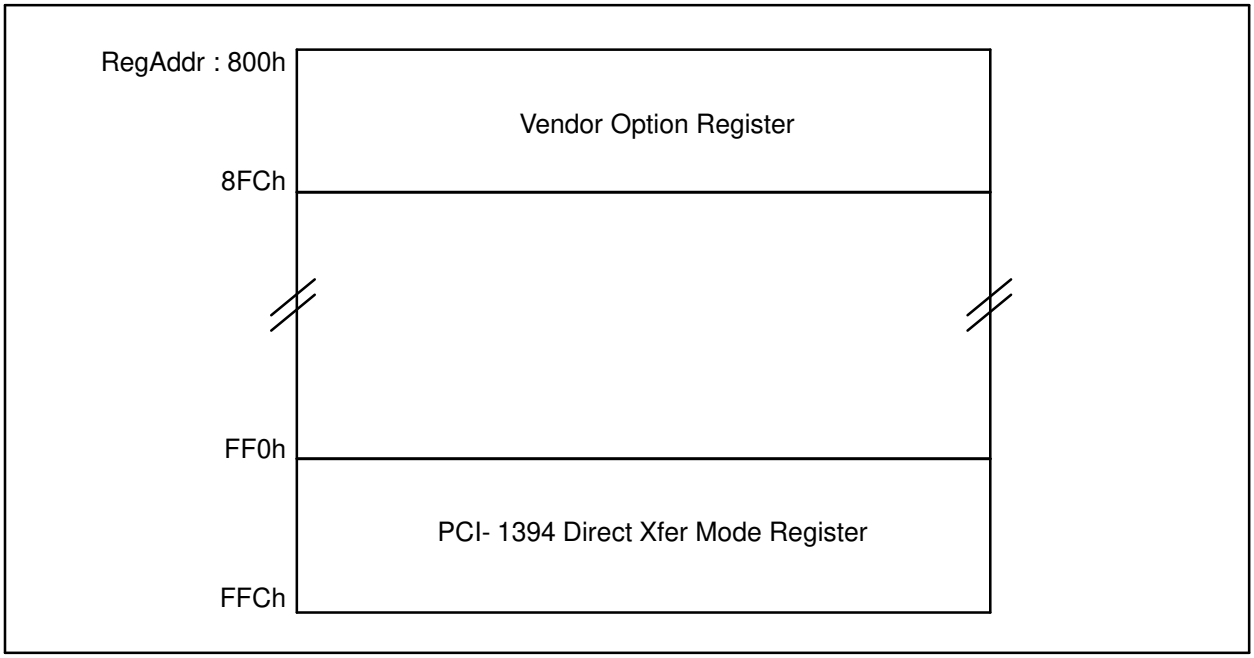


Fig. C.1 Address Space for Debug Register

Annex 1.1. Vendor Option Register

Vendor option register consists of various mode setting registers for packet transfer. The followings are the letters used in the register descriptions.

”r” denotes the register that can be read out.
”w” denotes the register that can be written.
”s” denotes the register that ”1” can be written.
”c” denotes the register whose value can be cleared to ”0”.
”u” denotes the register whose value is undefined depending on the MB86613S device status.

Figure C.2 lists the Vendor Option Register.

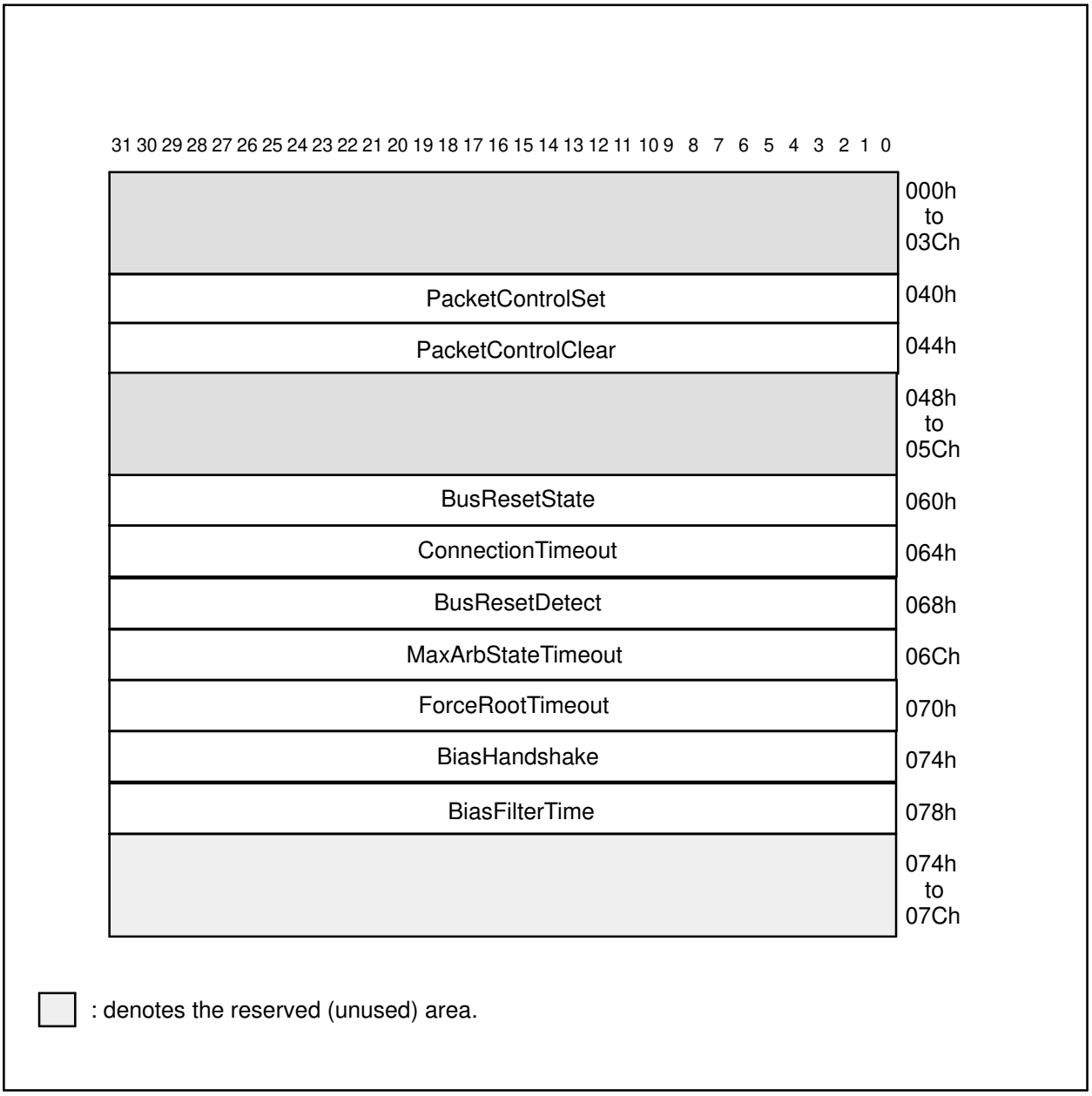
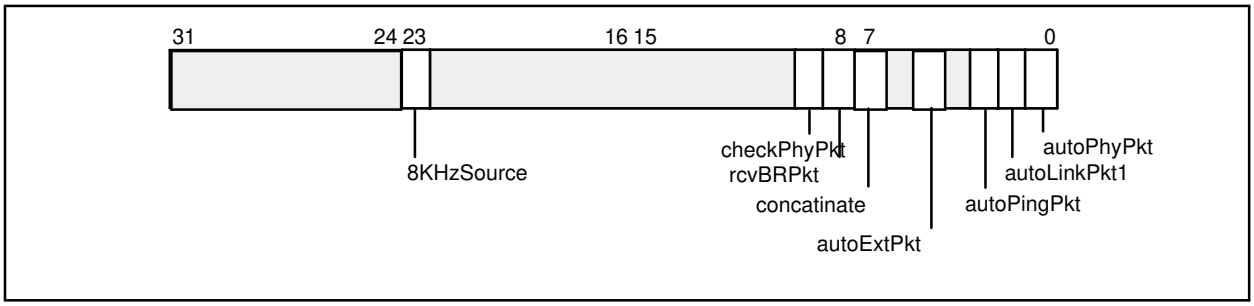


Fig. C.2 Vendor Option Register

Annex 1.1.1. PacketControl Register

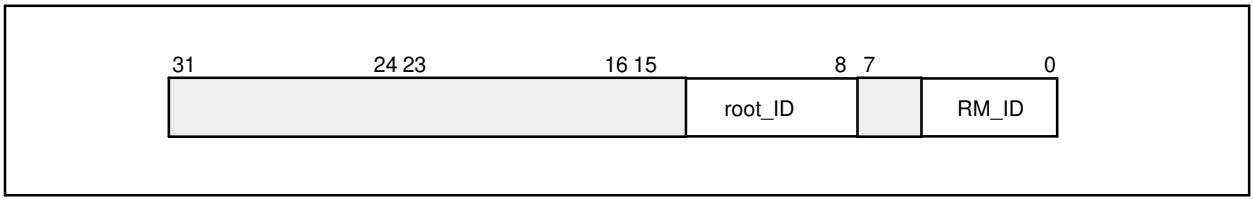
This register manages the LINK state or sets the operation mode when a PHY packet is received. Refer to 5.4 “Physical Configuration Packet Receive” for the use of autoPhyPkt bit.



Bit	Field Name	rscu	reset	description
23	8KHzSource	rsc	0b	This bit is valid only when LinkControl.cycleSource bit is set. Setting ”1” at this bit increments the IsoCycleTimer. cycleCount field by the 8KHz clock(CSCLK) input from the external. Setting”0” at this bit increments the IsoCycle Timer.cycleCount field by the 8KHz clock generated by the 24.576MHz.
9	checkPhyPkt	rsc	0b	Setting ”1” at this bit allows to check the logical inverse section of the received PHY packet.
8	rcvBRPkt	rsc	1b	Setting ”1” at this bit generates a bus reset packet.
7	contcatinate	rsc	0b	Setting “1” at this field concatinates two or more packets when they are transmitted.
4	autoExtPkt	rsc	1b	Setting ”1” at this bit automatically reports a remote reply packet or a remote confirmation packet when receiving the remote access packet or a remote command packet.
2	autoPingPkt	rsc	1b	Setting ”1” at this bit transmits a self ID packet when receiving a ping packet.
1	autoLinkPkt	rsc	0b	Setting ”1” at this bit enables LINK packet transmit and receive after receiving a link on packet. Setting ”0” at this bit enables LINK packet transmit and receive even if no link on packet is received. (But it requires HCControl.linkEnable bit to be set.)
0	autoPhyPkt	rsc	0b	Setting ”1” at this bit replaces the Gap_count and RHB fields in PHY register when receiving a phy configuration packet. Setting ”0” at this bit does not replace the PHY register values when receiving a phy configuration packet.

Annex 1.1.2. BusResetState Register

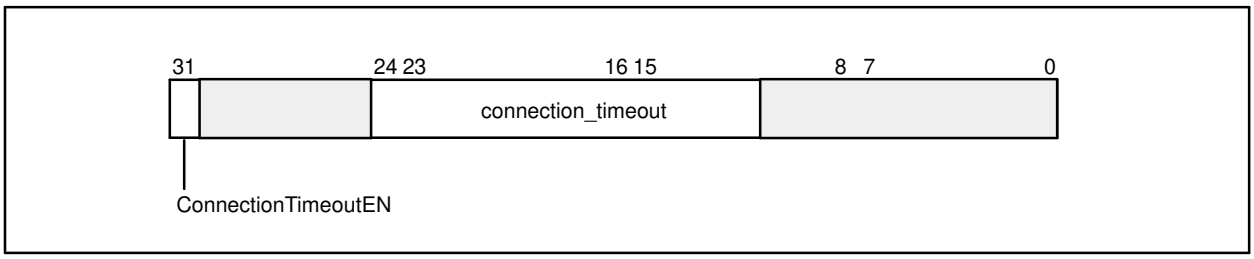
This register indicates the node number of root and resource manager determined in the bus reset.



Bit	Field Name	rwu	reset	description
13:8	root_ID	ru	undefined	This field indicates the root node number.
5:0	RM_ID	ru	3Fh	This field indicates the resource manager's node number.

Annex 1.1.3. ConnectionTimeout Register

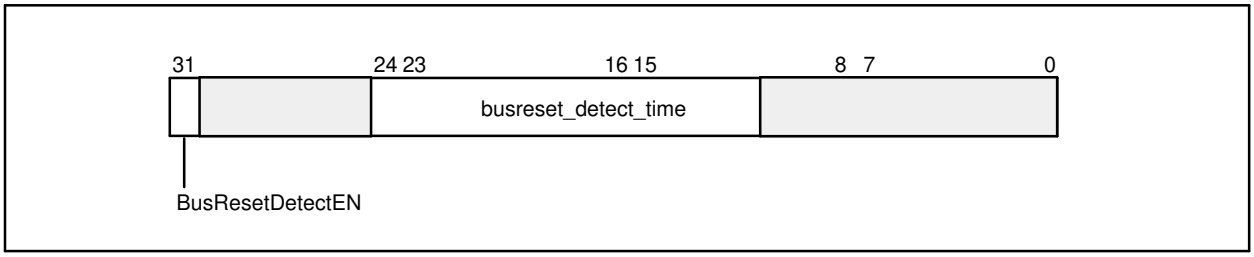
This register sets the time of detections between other node's bias voltage and the connection status. Do not set "0" for the connection timeout field in this register.



Bit	Field Name	rwu	reset	description
31	ConnectionTimeoutEN	rw	1b	Setting "0" at this bit does not count the time as set in the connection_timeout field (bit24:11).
24:11	connection_timeout	rw	0A80h	Set the connection timeout. The calculation is (Setting Value) x 125μs.

Annex 1.1.4. BusResetDetect Register

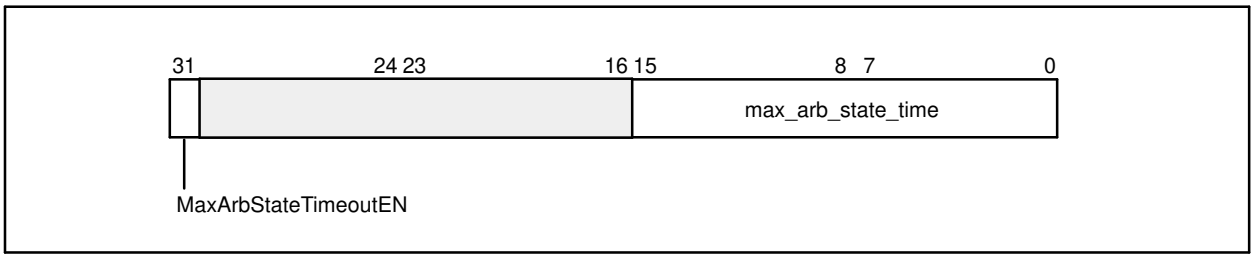
This register sets the time between the bus reset detected (received) and the response made. Do not set "0" for the busreset_detect_time field in this register.



Bit	Field Name	rwu	reset	description
31	BusResetDetectEN	rw	1b	Setting ”0” at this bit does not count the time as set in the busreset_detect_time field (bit24:11).
24:11	busreset_detect_ time	rw	0280h	Set the busreset_detect_time. The calculation is (Setting value) x 125μs.

Annex 1.1.5. MaxArbStateTimeout Register

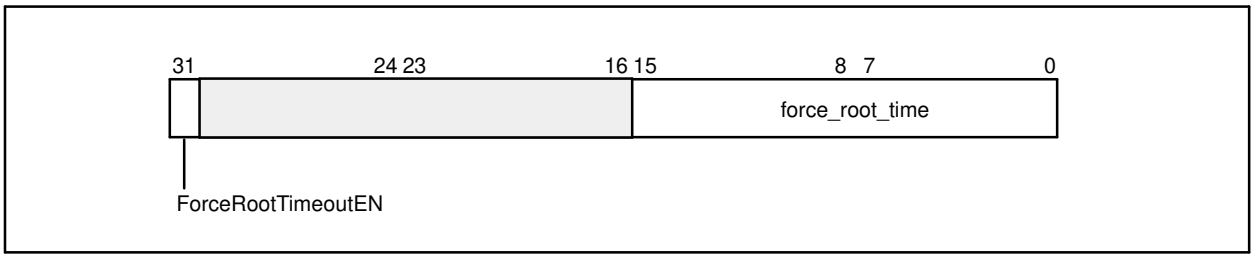
This register controls the Timeout bit of PHY register. When a PHY is staying in certain state exceeding a time specified in max_arb_state_time field, the Timeout bit is set. Do not set ”0” for the max_arb_state_time field in this register.



Bit	Field Name	rwu	reset	description
31	MaxArbState TimeoutEN	rw	1b	Setting ”0” at this bit does not count the time as set in the max_arb_state_time field (bit15:0).
15:0	max_arb_state_ time	rw	2710h	Set the max_arb_state_time. The calculation is (Setting value) x 40ns (=24.576MHz).

Annex 1.1.6. ForceRootTimeout Register

This register is used to set the time to start tree_identify process after setting the RHB bit of PHY register. Do not set ”0” for force_root_time field in this register.



Bit	Field Name	rwu	reset	description
31	ForceRootTImeout EN	rw	1b	Setting ”0” at this bit does not count the time as set in the force_root_time field (bit15:0).
15:0	force_root_time	rw	0F8Dh	Set the force_root_time. The calculation is (Setting value) x 40ns (=24.576MHz).

Annex 1.1.7. BiasHandshake Register

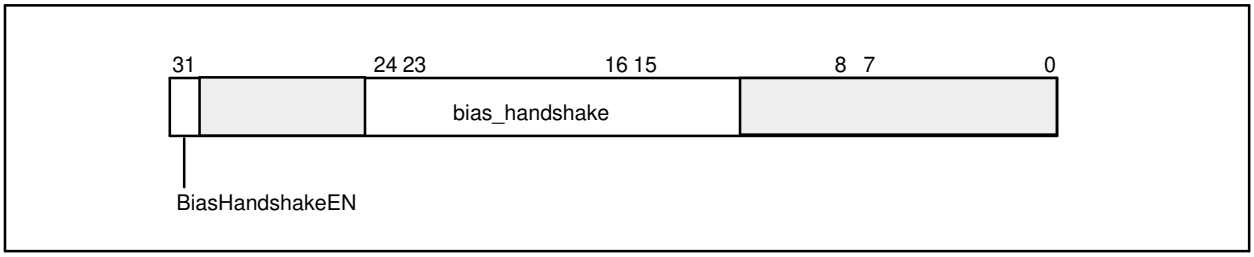
This register sets the times shown in below:

1) Between the suspend start and the verification of the TpBias for the other device.

2) Between the detection of the other device's TpBias and the providing stop of its own TpBias.

3) Between the resume start and the providing stop of its own TpBias.

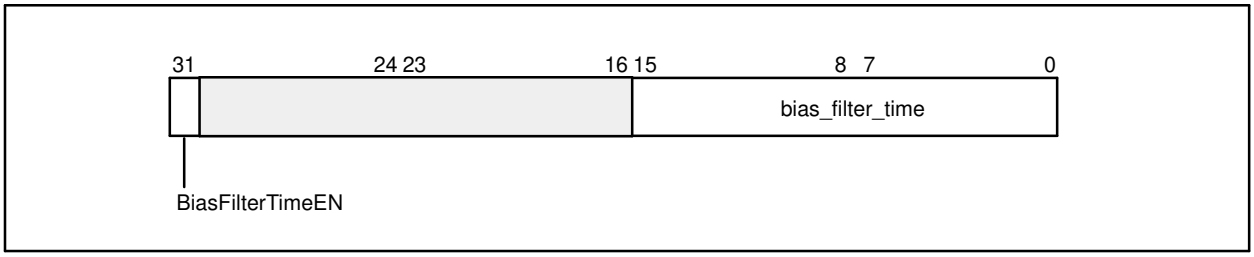
Do not set "0" for the bias_handshake field in this register.



Bit	Field Name	rwu	reset	description
31	BiasHand shakeEN	rw	1b	Setting "0" at this bit does not count the time set in the bias_handshake field(bit24:21).
24:11	busreset_ detect_time	rw	002Bh	Set the time of the bias_handshake. The calculation is; (Setting value) x 125us.

Annex 1.1.8. BiasFilterTime Register

This register is used to set the time between the detection of the other device's TpBias and the setting of the Bias bit on the PHY register.
Do not set "0" for the bias_filter_time field in this register.



Bit	Field Name	rwu	reset	description
31	BiasFilterTime EN	rw	1b	Setting "0" at this bit does not count the time set in the bias_filter_time field(bit15:0).
15:0	bias_filter_time	rw	0410h	Set the time of the bias_filter_time. The calculation is; (Setting value) x 40ns (24.576MHz).