Z16C30

CMOS USC Universal Serial Controller

Product Specification

DS007902-0708

PRELIMINARY

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

 Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore!, Z8 Encore! XP, Z8 Encore! MC, Crimzon, eZ80, and ZNEO are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.

ISO 9001:2000
FS 507510

Zilog products are designed and manufactured under an ISO registered 9001:2000 Quality Management System. For more details, please visit www.zilog.com/quality.

Revision History

Each instance in Revision History reflects a change to this document from its previous revision. For more details, refer to the corresponding pages and appropriate links in the table below.

Date	Revision Level	Description	Page No
July 2008	02	Updated as per latest template and style guide.	All
Jan 2000	01	Original issue	

Table of Contents

Architectural Overview

Features

The key features of Zilog's Z16C30 device include:

- Two Independent 0-to-10 Mbps Full-Duplex Channels, each with Two Baud Rate Generators and One digital phase-locked loop (DPLL) for Clock Recovery
- 32-byte Data FIFO's for each Receiver and Transmitter
- 110 ns Bus Cycle Time, 16-bit Data Bus Bandwidth
- Multi-Protocol Operation under Program Control with Independent Mode Selection for Receiver and Transmitter
- Async Mode with 1 to 8 Bits/Character, 1/16 to 2 Stop Bits/Character in 1/16-bit Increments, Programmable Clock Factor, Break Detect and Generation, Odd, Even, Mark, Space or no Parity and Framing Error Detection, Supports One Address/Data Bit and MIL STD 1553B Protocols
- Byte Oriented Synchronous Mode with One to Eight Bits/Character, Programmable Idle Line Condition, Optional Receive Sync Stripping; Optional Preamble Transmission, 16- or 32-bit CRC, and Transmit-to-Receive Slaving (for X.21)
- Bisync Mode with 2- to 16-bit Programmable Sync Character, Programmable Idle Line Condition, Optional Receive Sync Stripping, Optional Preamble Transmission, 16- or 32-bit CRC
- Transparent Bisync Mode with EBCDIC or ASCII Character Code, Automatic CRC Handling, Programmable Idle Line Condition, Optional Preamble Transmission, Automatic Recognition of DLE, SYN, SOH, ITX, ETX, ETB, EOT, ENQ, and ITB
- External Character Sync Mode for Receive
- HDLC/SDLC Mode with Eight-Bit Address Compare, Extended Address Field Option, 16- or 32-bit CRC, Programmable Idle Line Condition, Optional Preamble Transmission and Loop Mode
- DMA Interface with Separate Request and Acknowledge for Each Receiver and Transmitter
- Channel Load Command for DMA Controlled Initialization
- Flexible Bus Interface for Direct Connection to Most Microprocessors, User Programmable for 8 or 16 Bits Wide, Directly Supports 680X0 Family or 8X86 Family Bus Interfaces
- Low Power CMOS
- 68-Pin PLCC/100-Pin VQFP Packages

General Description

Zilog's Z16C30 USC Universal Serial Controller is a dual-channel multi-protocol data communications peripheral designed for use with any conventional multiplexed or non-multiplexed bus. The USC functions as a serial-to-parallel, parallel-to-serial converter/controller and may be software configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including two baud rate generators per channel, one digital phase-locked loop (DPLL) per channel, character counters for both receive and transmit in each channel and 32-byte data FIFO's for each receiver and transmitter (Figure 1 on page 3).

Zilog now offers a high speed version of the USC with improved bus bandwidth. CPU bus accesses have been shortened from 160 ns per access to 110 ns per access. The USC has a transmit and receive clock range of up to 10 MHz (20 MHz when using the DPLL, BRG, or CTR) and data transfer rates as high as 10 Mbits/sec full duplex.

The USC handles asynchronous formats, synchronous byte-oriented formats such as BISYNC, and synchronous bit-oriented formats such as HDLC. This device supports virtually any serial data transfer application.

The device can generate and check CRC in any synchronous mode and can be programmed to check data integrity in various modes. The USC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls may be used for general-purpose I/O (GPIO). The same is true for most of the other pins in each channel.

Interrupts are supported with a daisy-chain hierarchy, with the two channels having completely separate interrupt structures.

High-speed data transfers through DMA are supported by a Request/Acknowledge signal pair for each receiver and transmitter. The device supports automatic status transfer through DMA and also allows device initialization under DMA control.

Note: *When written to, all reserved bits must be programmed to 0.*

To aid in efficiently programming the USC, support tools are available. The Technical Manual describes in detail all features presented in this Product Specification and gives programming sequence hints. The Programmer's Assistant is a MS-DOS disk-based programming initialization tool to be used in conjunction with the Technical Manual. There are also available assorted application notes and development boards to assist in the hardware/software development.

All Signals with an overline, are active Low. For example: B/$\overline{W}$, in which WORD is active Low, and $\overline{B}$/W, in which BYTE is active Low.

Power connections follow these conventional descriptions:

Table 1. Power connection conventions

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

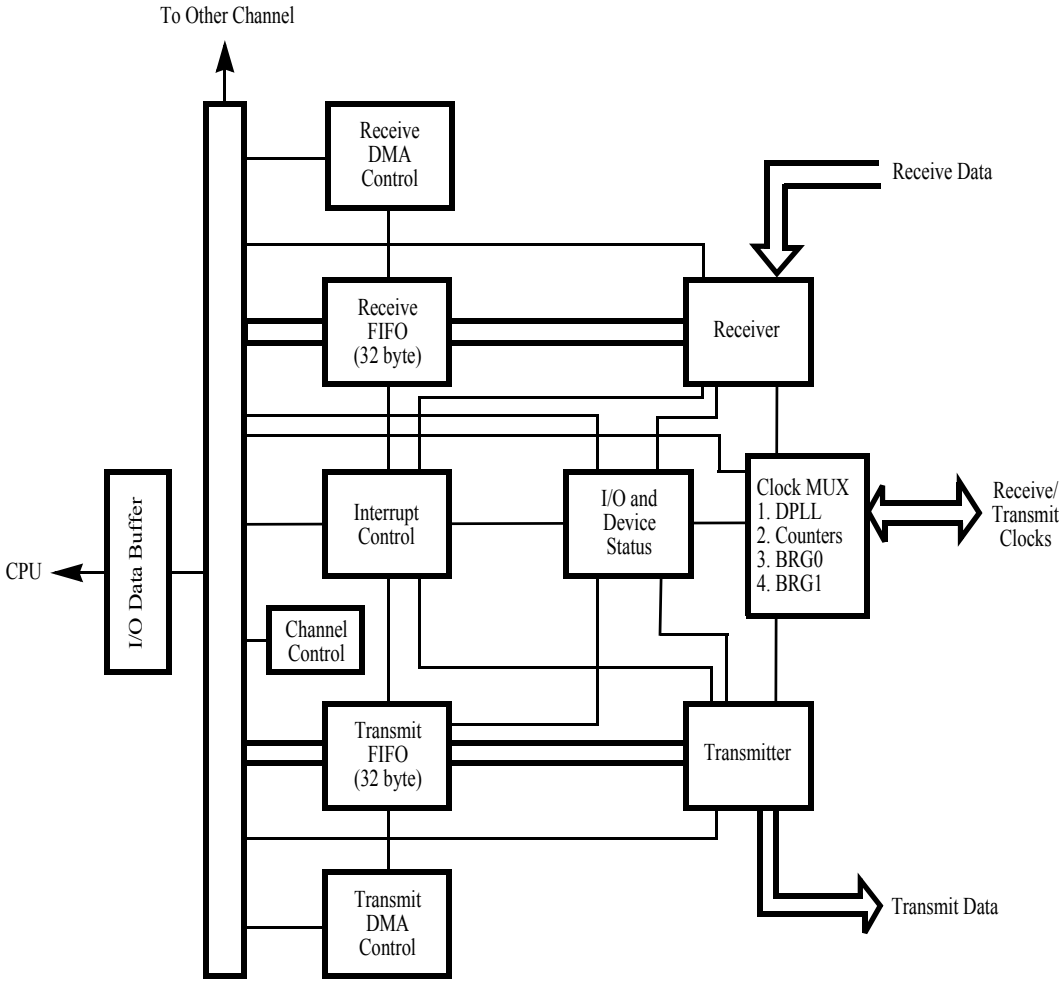


Figure 1. Z16C30 Block Diagram

Pin Description

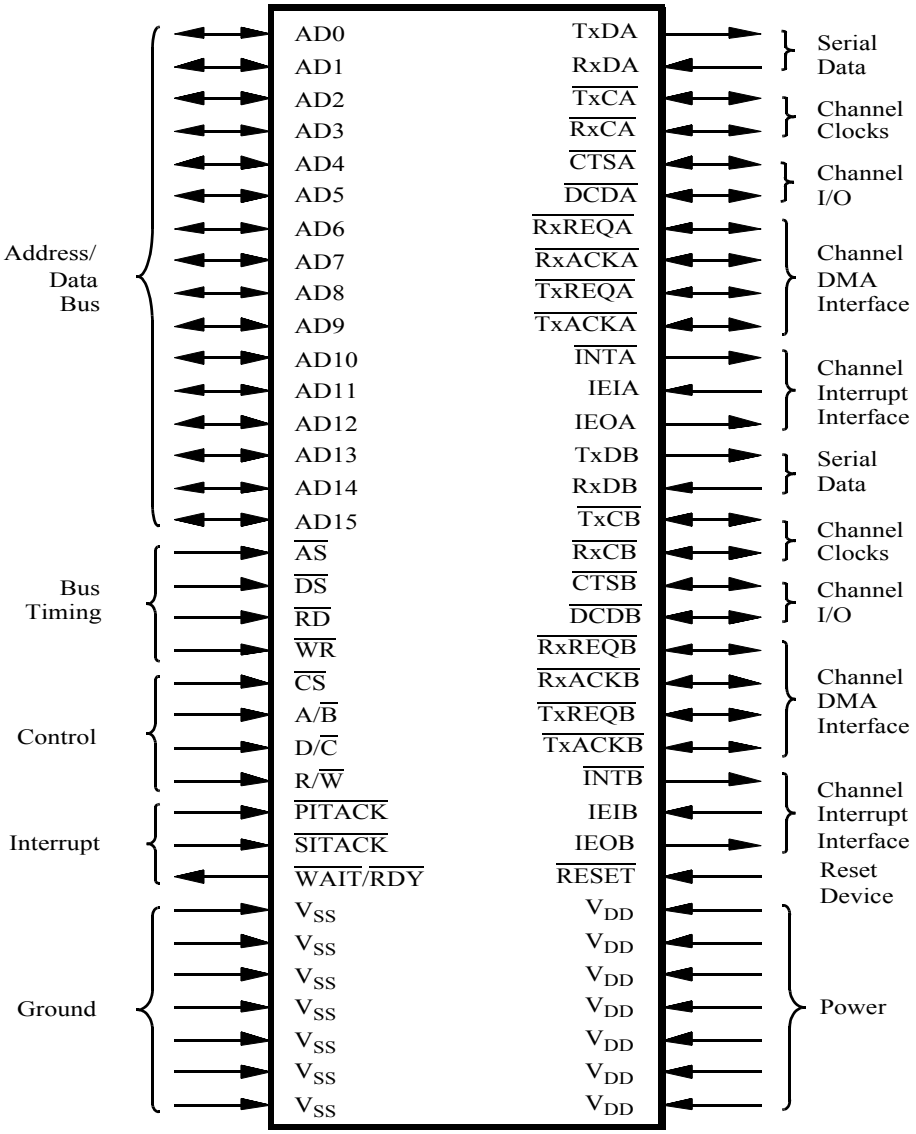


Figure 2. Z16C30 Pin Functions

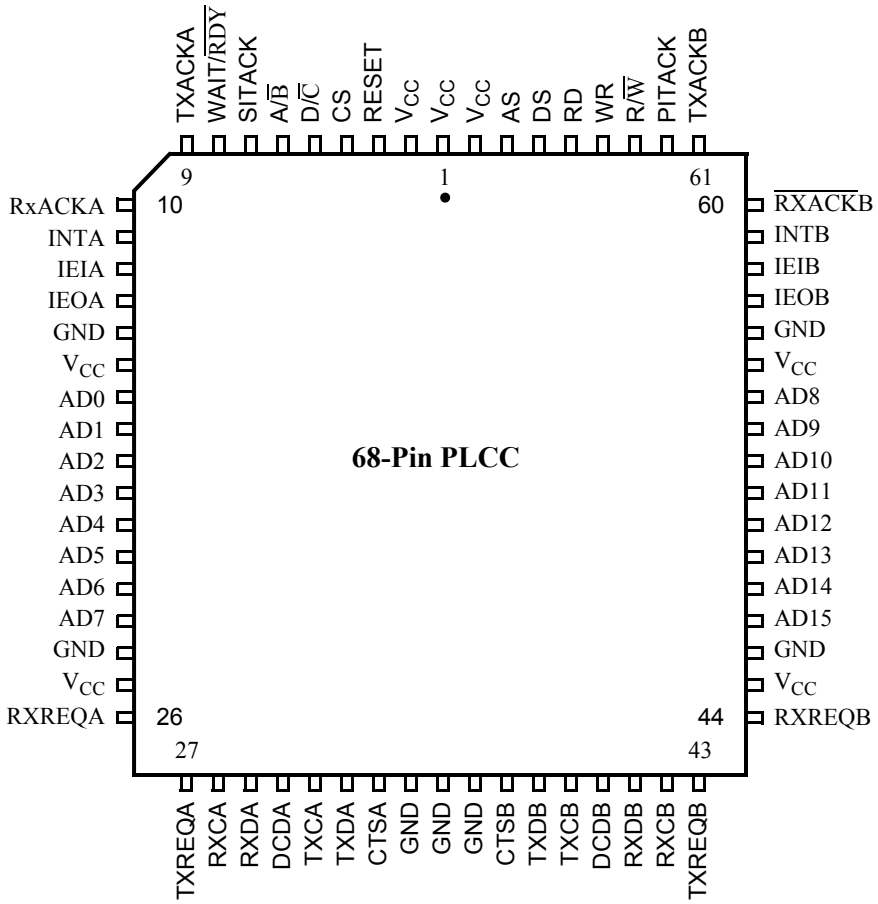


Figure 3. Z16C30 68-Pin PLCC Pin Assignments

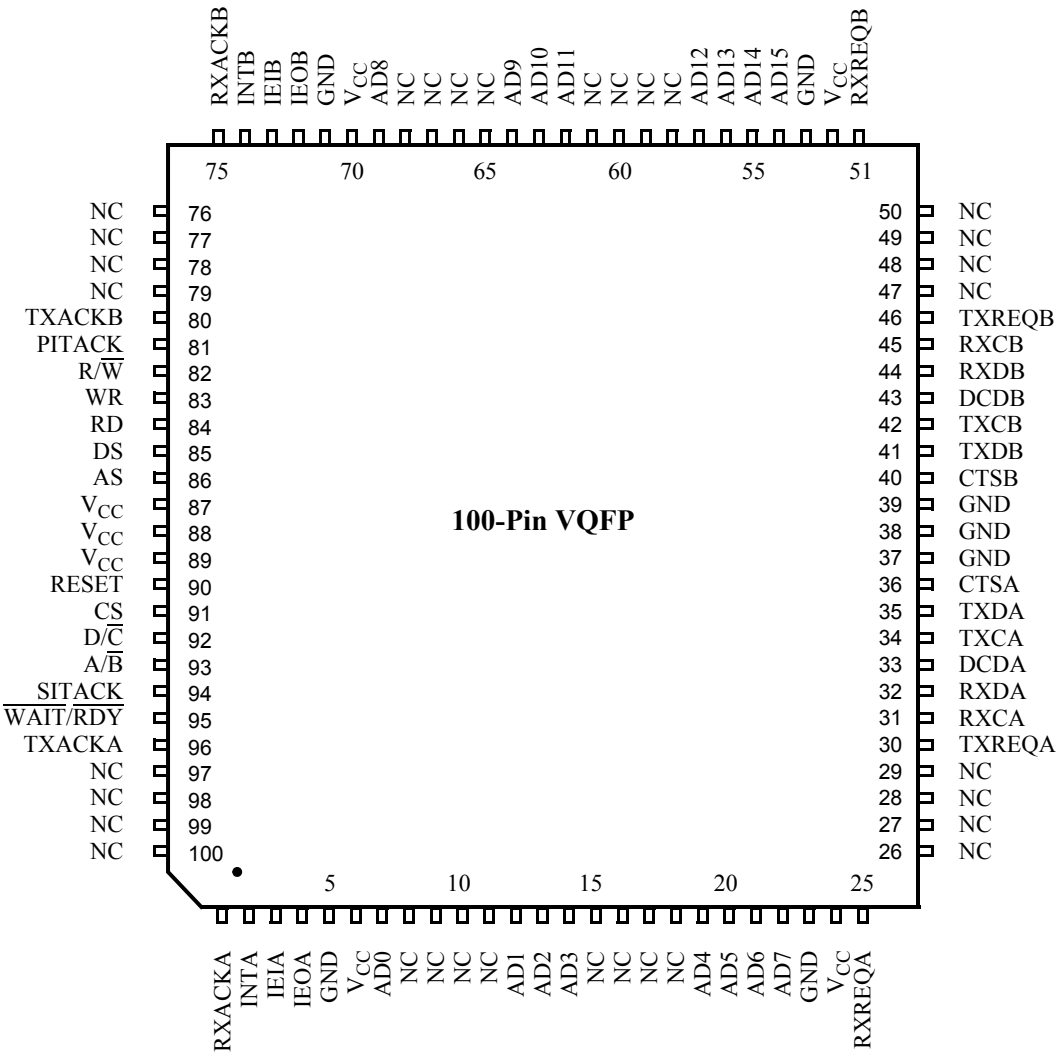


Figure 4. 100-Pin VQFP Pin Assignments

The Z16C30 contains 13 pins per channel for channel I/O, 16 pins for address and data, 12 pins for CPU handshake, and 14 pins for power and ground.

Three separate bus interface types are available for the device. The Bus Configuration Register (BCR) and external connections to the AD bus control selection of the bus type. A 16-bit bus is selected by setting BCR bit 2 to a 1. The 8-bit bus is selected by setting BCR bit 2 to 0 and tying AD15–AD8 to V_{SS}.

The 8-bit bus with separate address is selected by setting BCR bit 2 to 0 and, during the BCR write, forcing AD15 to a 1 and forcing AD14–AD8 to 0.

The multiplexed bus is selected for the USC if there is an Address Strobe prior to or during the transaction which writes the BCR. If no Address Strobe is present prior to or during the transaction which writes the BCR, a nonmultiplexed bus is selected (see Figure 29 on page 49).

Pin Functions

$\overline{\text{RESET}}$ Reset (input, active Low)—This signal resets the device to a known state. The first write to the USC after a reset accesses the BCR to select additional bus options for the device.

$\overline{\text{AS}}$ Address Strobe (input, active Low)—This signal is used in the multiplexed bus modes to latch the address on the AD lines. The $\overline{\text{AS}}$ signal is not used in the nonmultiplexed bus modes and should be tied to V_{DD}.

$\overline{\text{DS}}$ Data Strobe (input, active Low)—This signal strobes data out of the device during a read and may strobe an interrupt vector out of the device during an interrupt acknowledge cycle. $\overline{\text{DS}}$ also strobes data into the device on the state of R/$\overline{\text{W}}$.

$\overline{\text{RD}}$ Read Strobe (input, active Low)—This signal strobes data out of the device during a read and may strobe an interrupt vector out of the device during an interrupt acknowledge cycle.

$\overline{\text{WR}}$ Write Strobe (input, active Low)—This signal strobes data into the device during a write.

R/$\overline{\text{W}}$ Read/Write (input)—This signal determines the direction of data transfer for a read or write cycle in conjunction with $\overline{\text{DS}}$.

$\overline{\text{CS}}$ Chip Select (input, active Low)—This signal selects the device for access and must be asserted for read and write cycles, but is ignored during interrupt acknowledge and fly-by DMA transfers. In the case of a multiplexed bus interface, $\overline{\text{CS}}$ is latched by the rising edge of $\overline{\text{AS}}$.

A/$\overline{\text{B}}$ Channel A/Channel B Select (input)—This signal selects between the two channels in the device. High selects channel A and Low selects channel B. This signal is sampled and the result is latched during the BCR (Bus Configuration Register) write. It programs the sense of the $\overline{\text{WAIT}}$/$\overline{\text{RDY}}$ signal appropriate for different bus interfaces.

D/$\overline{\text{C}}$ Data/Control Select (input)—This signal, when High, provides for direct access to the RDR and TDR. In the case of a multiplexed bus interface, D/$\overline{\text{C}}$ High overrides the address provided to the device.

$\overline{\text{SITACK}}$ Status Interrupt Acknowledge (input, active Low)—This signal is a status signal that indicates that an interrupt acknowledge cycle is in progress. The device is capable of returning an interrupt vector that may be encoded with the type of interrupt pending during this acknowledge cycle. This signal is compatible with 680X0 family microprocessors.

$\overline{\text{PITACK}}$ Pulsed Interrupt Acknowledge (input, active Low)—This signal is a strobe signal that indicates that an interrupt acknowledge cycle is in progress. The device is capable of returning an interrupt vector that may be encoded with the type of interrupt pending during this acknowledge cycle. $\overline{\text{PITACK}}$ may be programmed to accept a single pulse or double pulse acknowledge type. This programming is done in the BCR. With the double pulse type selected, the first $\overline{\text{PITACK}}$ is recognized but no action takes place. The interrupt vector is returned on the second pulse if the no vector option is not selected. The double pulse type is compatible with 8X86 family microprocessors.

$\overline{\text{WAIT}}$/$\overline{\text{RDY}}$ Wait/Data Ready (output, active Low)—This signal serves to indicate when the data is available during a read cycle, when the device is ready to receive data during a write cycle, and when a valid vector is available during an interrupt acknowledge cycle. It may be programmed to function either as a Wait signal or a Ready signal using the state of the A/$\overline{\text{B}}$ pin during the BCR write. When A/$\overline{\text{B}}$ is High during the BCR write, this signal functions as a wait output and thus supports the READY function of 8X86 family microprocessors. When A/$\overline{\text{B}}$ is Low during the BCR write, this signal functions as a ready output and thus supports the DTACK function of 680X0 family microprocessors.

AD15–AD0 Address/Data Bus (bidirectional, active High, tri-state)—The AD signals carry addresses to, and data to and from, the device. When the 16-bit nonmultiplexed bus is selected, AD15–AD0 carry data to and from the device. Addresses are provided using a pointer within the device that is loaded with the desired register address. When selecting the 8-bit nonmultiplexed bus (without separate address) only AD7–AD0 are used to transfer data. The pointer is used for addressing, with AD15–AD8 unused. When selecting the 8-bit nonmultiplexed bus (with separate address), AD7–AD0 are used to transfer data with AD15–AD8 used as address bus. When the 16-bit multiplexed bus is selected, addresses are latched from AD7–AD0 and data transfers are sixteen bits wide. When selecting the 8-bit multiplexed bus (without separate address) only AD7–AD0 are used to transfer addresses and data, with AD15–AD8 unused. When the 8-bit multiplexed bus with separate address is selected, only AD7–AD0 are used to transfer data, while AD15–AD8 are used as an address bus.

$\overline{\text{INTA}}$, $\overline{\text{INTB}}$ Interrupt Request (outputs, active Low)—These signals indicate that the channel has an interrupt condition pending and is requesting service. These outputs are NOT open-drain.

IEIA, IEIB Interrupt Enable In (inputs, active High)—The IEI signal for each channel is used with the accompanying IEO signal to form an interrupt daisy chain. An active IEI indicates that no device having higher priority is requesting or servicing an interrupt.

IEOA, IEOB Interrupt Enable Out (outputs, active High)—The IEO signal for each channel is used with the accompanying IEI signal to form an interrupt daisy chain. IEO is Low if IEI is Low, an interrupt is under service in the channel, or an interrupt is pending during an interrupt acknowledge cycle.

$\overline{\text{TxACKA}}$, $\overline{\text{TxACKB}}$ Transmit Acknowledge (inputs or outputs, active Low)—The primary function of these signals is to perform fly-by DMA transfers to the transmit FIFOs. They may also be used as bit inputs or outputs.

$\overline{\text{RxACKA}}$, $\overline{\text{RxACKB}}$ Receive Acknowledge (inputs or outputs, active Low)—The primary function of these signals is to perform fly-by DMA transfers from the receive FIFOs. They may also be used as bit inputs or outputs.

TxDA, TxDB Transmit Data (outputs, active High, tri-state)—These signals carry the serial transmit data for each channel.

RxDA, RxDB Receive Data (inputs, active High)—These signals carry the serial receive data for each channel.

$\overline{\text{TxCA}}$, $\overline{\text{TxCB}}$ Transmit Clock (inputs or outputs, active Low)—These signals are used as clock inputs for any of the functional blocks within the device. They may also be used as outputs for various transmitter signals or internal clock signals.

$\overline{\text{RxCA}}$, $\overline{\text{RxCB}}$ Receive Clock (inputs or outputs, active Low)—These signals are used as clock inputs for any of the functional blocks within the device. They may also be used as outputs for various receiver signals or internal clock signals.

$\overline{\text{TxREQA}}$, $\overline{\text{TxREQB}}$ Transmit Request (inputs or outputs, active Low)—The primary function of these signals is to request DMA transfers to the transmit FIFOs. They may also be used as simple inputs or outputs.

$\overline{\text{RxREQA}}$, $\overline{\text{RxREQB}}$ Receive Request (inputs or outputs, active Low)—The primary function of these signals is to request DMA transfers from the receive FIFOs. They may also be used as simple inputs or outputs.

$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$ Clear To Send (inputs or outputs, active Low)—These signals are used as enables for the respective transmitters. They may also be programmed to generate interrupts on either transition or used as simple inputs or outputs.

$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$ Data Carrier Detect (inputs or outputs, active Low)—These signals are used as enables for the respective receivers. They may also be programmed to generate interrupts on either transition or used as simple inputs or outputs.

Electrical Characteristics

Table 2. Absolute Maximum Ratings

Symbol	Description	Min	Max	Units
V_{CC}	Supply Voltage (*)	–0.3	+7.0	V
T_{STG}	Storage Temp.	–65°	+150°	C
T_A	Oper Ambient Temp		†	C
	Power Dissipation		2.2	W

*Voltage on all pins with respect to GND.
†See Ordering Information on page 97.

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Standard Test Conditions

The DC Characteristics and Capacitance section below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin (Figure 5 on page 11). Standard conditions are as follows:

- +4.5 V < V_{CC} < +5.5 V
- GND = 0 V
- T_A as specified in Ordering Information on page 97

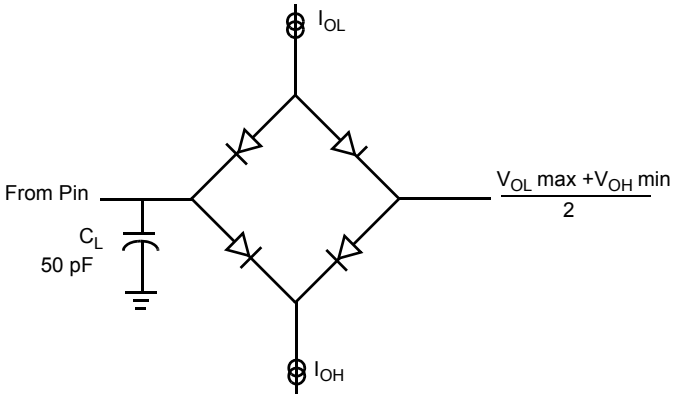


Figure 5. Test Load Diagram

Capacitance

Table 3. Capacitance

Symbol	Parameter	Min	Max	Unit	Condition
C_{IN}	Input Capacitance		10	pF	Unmeasured Pins
C_{OUT}	Output Capacitance		15	pF	Returned to Ground.
CI/O	Bidirectional Capacitance		20	pF	
Note: f = 1 MHz over specified temperature range.					

Miscellaneous

Transistor Count: 174,000

Temperature Ratings

Standard = 0 °C to ±70 °C
Extended = –40 °C to +85 °C

DC Characteristics

Table 4. Z16C30 DC Characteristics

Symbol	Parameter	Min	Typ	Max	Unit	Condition
V_{IH}	Input High Voltage	2.2		V_{CC}+0.3	V	
V_{IL}	Input Low Voltage	–0.3		0.8	V	
V_{OH1}	Output High Voltage	2.4			V	I_{OH} = –1.6 mA
V_{OH2}	Output High Voltage	V_{CC}–0.8			V	I_{OH} = –250 µA
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = +2.0 mA
I_{IL}	Input Leakage			±10.00	µA	0.4 < V_{IN} < +2.4 V
I_{OL}	Output Leakage			±10.00	µA	0.4 < V_{OUT} < +2.4 V
I_{CCI}	V_{CC} Supply Current		7	50	mA	V_{CC} = 5 V V_{IH} = 4.8 V V_{IL} = 0.2V
Note: V_{CC}= 5 V ±10% unless otherwise specified, over specified temperature range.						

AC Characteristics

Table 5. Z16C30 AC Characteristics

No	Symbol	Parameter	Min	Max	Units	Note
1	Tcyc	Bus Cycle Time	110		ns	
2	TwASl	$\overline{AS}$ Low Width	30		ns	
3	TwASh	$\overline{AS}$ High Width	60		ns	
4	TwDSl	$\overline{DS}$ Low Width	60		ns	
5	TwDSh	$\overline{DS}$ High Width	50		ns	
6	TdAS(DS)	$\overline{AS}$ Rise to $\overline{DS}$ Fall Delay Time	5		ns	
7	TdDS(AS)	$\overline{DS}$ Rise to $\overline{AS}$ Fall Delay Time	5		ns	
8	TdDS(DRa)	$\overline{DS}$ Fall to Data Active Delay	0		ns	
9	TdDS(DRv)	$\overline{DS}$ Fall to Data Valid Delay		60	ns	
10	TdDS(DRn)	$\overline{DS}$ Rise to Data Not Valid Delay	0		ns	
11	TdDS(DRz)	$\overline{DS}$ Rise to Data Float Delay		20	ns	

Table 5. Z16C30 AC Characteristics (Continued)

No	Symbol	Parameter	Min	Max	Units	Note
12	TsCS(AS)	$\overline{CS}$ to $\overline{AS}$ Rise Setup Time	15		ns	
13	ThCS(AS)	$\overline{CS}$ to $\overline{AS}$ Rise Hold Time	5		ns	
14	TsADD(AS)	Direct Address to $\overline{AS}$ Rise Setup Time	15		ns	1
15	ThADD(AS)	Direct Address to $\overline{AS}$ Rise Hold Time	5		ns	1
16	TsSIA(AS)	$\overline{SITACK}$ to $\overline{AS}$ Rise Setup Time	15		ns	
17	ThSIA(AS)	$\overline{SITACK}$ to $\overline{AS}$ Rise Hold Time	5		ns	
18	TsAD(AS)	Address to $\overline{AS}$ Rise Setup Time	15		ns	
19	ThAD(AS)	Address to $\overline{AS}$ Rise Hold Time	5		ns	
20	TsRW(DS)	R/$\overline{W}$ to $\overline{DS}$ Fall Setup Time	0		ns	
21	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ Fall Hold Time	25		ns	
22	TsDSf(RRQ)	$\overline{DS}$ Fall to $\overline{RxREQ}$ Inactive Delay		60	ns	4
23	TdDSr(RRQ)	$\overline{DS}$ Rise to $\overline{RxREQ}$ Active Delay	0		ns	
24	TsDW(DS)	Write Data to $\overline{DS}$ Rise Setup Time	30		ns	
25	ThDW(DS)	Write Data to $\overline{DS}$ Rise Hold Time	0		ns	
26	TdDSf(TRQ)	$\overline{DS}$ Fall to $\overline{TxREQ}$ Inactive Delay		65	ns	5,6
27	TdDSr(TRQ)	$\overline{DS}$ Rise to $\overline{TxREQ}$ Active Delay	0		ns	
28	TwRDl	$\overline{RD}$ Low Width	60		ns	
29	TwRDh	$\overline{RD}$ High Width	50		ns	
30	TdAS(RD)	$\overline{AS}$ Rise to $\overline{RD}$ Fall Delay Time	5		ns	
31	TdRD(AS)	$\overline{RD}$ Rise to $\overline{AS}$ Fall Delay Time	5		ns	
32	TdRD(DRa)	$\overline{RD}$ Fall to Data Active Delay	0		ns	
33	TdRD(DRv)	$\overline{RD}$ Fall to Data Valid Delay		60	ns	
34	TdRD(DRn)	$\overline{RD}$ Rise to Data Not Valid Delay	0		ns	

Table 5. Z16C30 AC Characteristics (Continued)

No	Symbol	Parameter	Min	Max	Units	Note
35	TdRD(DRz)	$\overline{RD}$ Rise to Data Float Delay		20	ns	
36	TdRDf(RRQ)	$\overline{RD}$ Fall to $\overline{RxREQ}$ Inactive Delay		60	ns	4
37	TdRDr(RRQ)	$\overline{RD}$ Rise to $\overline{RxREQ}$ Active Delay	0		ns	
38	TwWRl	$\overline{WR}$ Low Width	60		ns	
39	TwWRh	$\overline{WR}$ High Width	50		ns	
40	TdAS(WR)	$\overline{AS}$ Rise to $\overline{WR}$ Fall Delay Time	5		ns	
41	TdWR(AS)	$\overline{WR}$ Rise to $\overline{AS}$ Fall Delay Time	5		ns	
42	TsDW(WR)	Write Data to $\overline{WR}$ Rise Setup Time	30		ns	
43	ThDW(WR)	Write Data to $\overline{WR}$ Rise Hold Time	0		ns	
44	TdWRf(TRQ)	$\overline{WR}$ Fall to $\overline{TxREQ}$ Inactive Delay		65	ns	5
45	TdWRr(TRQ)	$\overline{WR}$ Rise to $\overline{TxREQ}$ Active Delay	0		ns	
46	TsCS(DS)	$\overline{CS}$ to $\overline{DS}$ Fall Setup Time	0		ns	2
47	ThCS(DS)	$\overline{CS}$ to $\overline{DS}$ Fall Hold Time	25		ns	2
48	TsADD(DS)	Direct Address to $\overline{DS}$ Fall Setup Time	5		ns	1,2
49	ThADD(DS)	Direct Address to $\overline{DS}$ Fall Hold Time	25		ns	1,2
50	TsSIA(DS)	$\overline{SITACK}$ to $\overline{DS}$ Fall Setup Time	5		ns	2
51	ThSIA(DS)	$\overline{SITACK}$ to $\overline{DS}$ Fall Hold Time	25		ns	2
52	TsCS(RD)	$\overline{CS}$ to $\overline{RD}$ Fall Setup Time	0		ns	2
53	ThCS(RD)	$\overline{CS}$ to $\overline{RD}$ Fall Hold Time	25		ns	2
54	TsADD(RD)	Direct Address to $\overline{RD}$ Fall Setup Time	5		ns	1,2
55	ThADD(RD)	Direct Address to $\overline{RD}$ Fall Hold Time	25		ns	1,2
56	TsSIA(RD)	$\overline{SITACK}$ to $\overline{RD}$ Fall Setup Time	5		ns	2
57	ThSIA(RD)	$\overline{SITACK}$ to $\overline{RD}$ Fall Hold Time	25		ns	2
58	TsCS(WR)	$\overline{CS}$ to $\overline{WR}$ Fall Setup Time	0		ns	2

Table 5. Z16C30 AC Characteristics (Continued)

No	Symbol	Parameter	Min	Max	Units	Note
59	ThCS(WR)	$\overline{CS}$ to $\overline{WR}$ Fall Hold Time	25		ns	2
60	TsADD(WR)	Direct Address to $\overline{WR}$ Fall Setup Time	5		ns	1,2
61	ThADD(WR)	Direct Address to $\overline{WR}$ Fall Hold Time	25		ns	1,2
62	TsSIA(WR)	$\overline{SITACK}$ to $\overline{WR}$ Fall Setup Time	5		ns	2
63	ThSIA(WR)	$\overline{SITACK}$ to $\overline{WR}$ Fall Hold Time	25		ns	2
64	TwRAKl	$\overline{RxACK}$ Low Width	60		ns	
65	TwRAKh	$\overline{RxACK}$ High Width	50		ns	
66	TdRAK(DRa)	$\overline{RxACK}$ Fall to Data Active Delay	0		ns	
67	TdRAK(DRv)	$\overline{RxACK}$ Fall to Data Valid Delay		60	ns	
68	TdRAK(DRn)	$\overline{RxACK}$ Rise to Data Not Valid Delay	0		ns	
69	TdRAK(DRz)	$\overline{RxACK}$ Rise to Data Float Delay		20	ns	
70	TdRAKf(RRQ)	$\overline{RxACK}$ Fall to $\overline{RxREQ}$ Inactive Delay		60	ns	4
71	TdRAKr(RRQ)	$\overline{RxACK}$ Rise to $\overline{RxREQ}$ Active Delay	0		ns	
72	TwTAKl	$\overline{TxACK}$ Low Width	60		ns	
73	TwTAKh	$\overline{TxACK}$ High Width	50		ns	
74	TsDW(TAK)	Write Data to $\overline{TxACK}$ Rise Setup Time	30		ns	
75	ThDW(TAK)	Write Data to $\overline{TxACK}$ Rise Hold Time	0		ns	
76	TdTAKf(TRQ)	$\overline{TxACK}$ Fall to $\overline{TxREQ}$ Inactive Delay		65	ns	5
77	TdTAKr(TRQ)	$\overline{TxACK}$ Rise to $\overline{TxREQ}$ Active Delay	0		ns	
78	TdDSf(RDY)	$\overline{DS}$ Fall (INTACK) to $\overline{RDY}$ Fall Delay		200	ns	
79	TdRDY(DRv)	$\overline{RDY}$ Fall to Data Valid Delay		40	ns	
80	TdDSr(RDY)	$\overline{DS}$ Rise to $\overline{RDY}$ Rise Delay		40	ns	

Table 5. Z16C30 AC Characteristics (Continued)

No	Symbol	Parameter	Min	Max	Units	Note
81	TsIEI(DSI)	IEI to $\overline{DS}$ Fall (INTACK) Setup Time	10		ns	
82	ThIEI(DSI)	IEI to $\overline{DS}$ Rise (INTACK) Hold Time	0		ns	
83	TdIEI(IEO)	IEI to IEO Delay		30	ns	
84	TdAS(IEO)	$\overline{AS}$ Rise (Intack) to IEO Delay		60	ns	
85	TdDSI(INT)	$\overline{DS}$ Fall (INTACK) to $\overline{INT}$ Inactive Delay		200	ns	7
87	TdDSI(Wr)	$\overline{DS}$ Fall (INTACK) to $\overline{WAIT}$ Rise Delay		200	ns	
88	TdW(DRv)	$\overline{WAIT}$ Rise to Data Valid Delay		40	ns	
89	TdRDf(RDY)	$\overline{RD}$ Fall (INTACK) to $\overline{RDY}$ Fall Delay		200	ns	
90	TdRDr(RDY)	$\overline{RD}$ Rise to $\overline{RDY}$ Rise Delay		40	ns	
91	TsIEI(RDI)	IEI to $\overline{RD}$ Fall (INTACK) Setup Time	10		ns	
92	ThIEI(RDI)	IEI to $\overline{RD}$ Rise (INTACK) Hold Time	0		ns	
93	TdRDI(INT)	$\overline{RD}$ Fall (INTACK) to $\overline{INT}$ Inactive Delay		200	ns	
94	TdRDI(Wf)	$\overline{RD}$ Fall (INTACK) to $\overline{WAIT}$ Fall Delay		40	ns	
95	TdRDI(Wr)	$\overline{RD}$ Fall (INTACK) to $\overline{WAIT}$ Rise Delay		200	ns	
96	TwPIAl	$\overline{PITACK}$ Low Width	60		ns	
97	TwPIAh	$\overline{PITACK}$ High Width	50		ns	
98	TdAS(PIA)	$\overline{AS}$ Rise to $\overline{PITACK}$ Fall Delay Time	5		ns	
99	TdPIA(AS)	$\overline{PITACK}$ Rise to $\overline{AS}$ Fall Delay Time	5		ns	
100	TdPIA(DRa)	$\overline{PITACK}$ Fall to Data Active Delay	0		ns	
101	TdPIA(DRn)	$\overline{PITACK}$ Rise to Data Not Valid Delay	0		ns	
102	TdPIA(DRz)	$\overline{PITACK}$ Rise to Data Float Delay		20	ns	

Table 5. Z16C30 AC Characteristics (Continued)

No	Symbol	Parameter	Min	Max	Units	Note
103	TsIEI(PIA)	IEI to $\overline{\text{PITACK}}$ Fall Setup Time	10		ns	
104	ThIEI(PIA)	IEI to $\overline{\text{PITACK}}$ Rise Hold Time	0		ns	
105	TdPIA(IEO)	$\overline{\text{PITACK}}$ Fall to IEO Delay		60	ns	
106	TdPIA(INT)	$\overline{\text{PITACK}}$ Fall to $\overline{\text{INT}}$ Inactive Delay		200	ns	
107	TdPIAf(RDY)	$\overline{\text{PITACK}}$ Fall to $\overline{\text{RDY}}$ Fall Delay		200	ns	
108	TdPIAr(RDY)	$\overline{\text{PITACK}}$ Rise to $\overline{\text{RDY}}$ Rise Delay		40	ns	
109	TdPIA(Wf)	$\overline{\text{PITACK}}$ Fall to $\overline{\text{WAIT}}$ Fall Delay		40	ns	
110	TdPIA(Wr)	$\overline{\text{PITACK}}$ Fall to $\overline{\text{WAIT}}$ Rise Delay		200	ns	
111	TdSIA(INT)	$\overline{\text{SITACK}}$ Fall to IEO Inactive Delay		200	ns	2
112	TwSTBh	$\overline{\text{Strobe}}$ High Width	50		ns	3
113	TwRESl	$\overline{\text{RESET}}$ Low Width	170		ns	
114	TwRESh	$\overline{\text{RESET}}$ High Width	60		ns	
115	Tdres(STB)	$\overline{\text{RESET}}$ Rise to $\overline{\text{STB}}$ Fall	60		ns	3
116	TdDSf(RDY)	$\overline{\text{DS}}$ Fall to $\overline{\text{RDY}}$ Fall Delay		50	ns	
117	TdWRf(RDY)	$\overline{\text{WR}}$ Fall to $\overline{\text{RDY}}$ Fall Delay		50	ns	
118	TdWRr(RDY)	$\overline{\text{WR}}$ Rise to $\overline{\text{RDY}}$ Rise Delay		40	ns	
119	TdRDf(RDY)	$\overline{\text{RD}}$ Fall to $\overline{\text{RDY}}$ Fall Delay		50	ns	
120	TdRAKf(RDY)	$\overline{\text{RxACK}}$ Fall to $\overline{\text{RDY}}$ Fall Delay		50	ns	
121	TdRAKr(RDY)	$\overline{\text{RxACK}}$ Rise to $\overline{\text{RDY}}$ Rise Delay		40	ns	
122	TdTAKf(RDY)	$\overline{\text{TxACK}}$ Fall to $\overline{\text{RDY}}$ Fall Delay		50	ns	

Table 5. Z16C30 AC Characteristics (Continued)

No	Symbol	Parameter	Min	Max	Units	Note
123	TdTAKr(RDY)	TxACK Rise to RDY Rise Delay		40	ns	

Notes
1. Direct address is any of A/B, D/C, or AD15–AD8 used as an address bus.
2. The parameter applies only when AS is not present.
3. Strobe (STB) is any of DS, RD, WR, PITACK, RxACK or TxACK.
4. Parameter applies only if read empties the receive FIFO.
5. Parameter applies only if write fills the transmit FIFO.
6. For extended temperature part TdDSI(Wf) max = 220 ns.
7. For extended temperature part TdDSF(TRQ) max = 75 ns.

USC Timing

The USC interface timing is similar to that found on a static RAM, except that it is much more flexible. Up to eight separate timing strobe signals may be present on the interface: DS, RD, WR, PITACK, RxACKA, RxACKB, TxACKA, and TxACKB. Only one of these timing strobes may be active at any time. Should the external logic activate more than one of these strobes at the same time the USC will enter a pre-reset state that is only exited by a hardware reset. Do not allow overlap of timing strobes. The timing diagrams beginning on the next page illustrate the different bus transactions possible with the necessary setup hold and delay times.

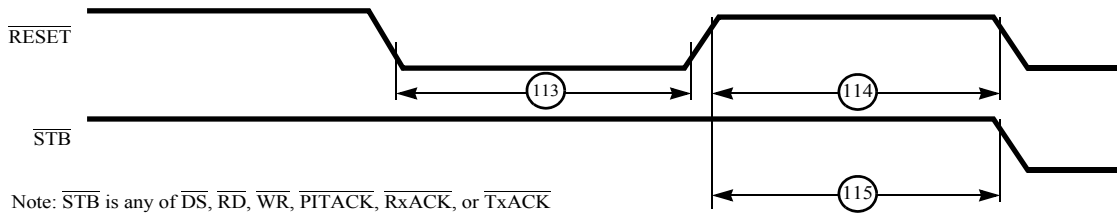


Figure 6. Reset Timing

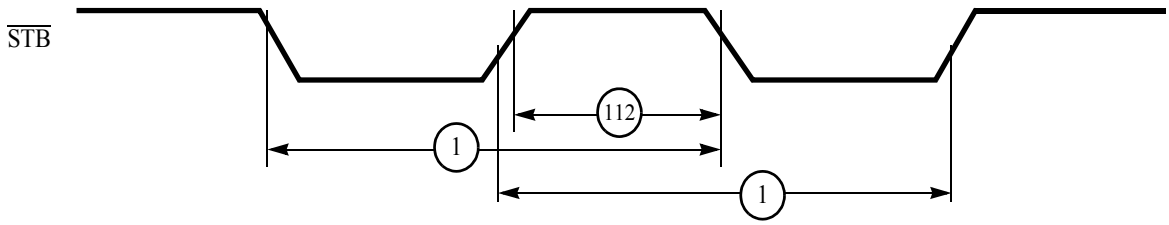
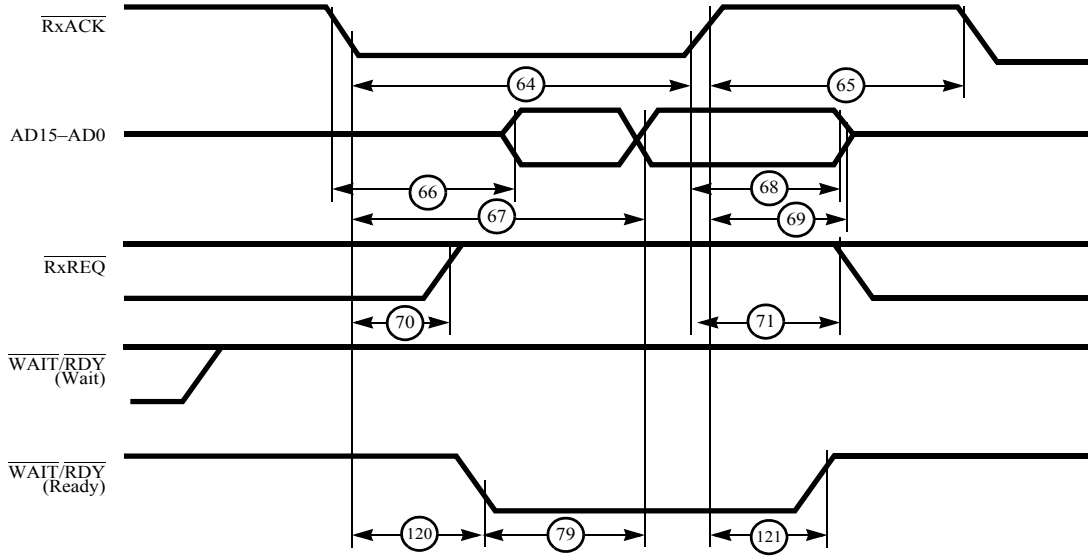


Figure 7. Bus Cycle Timing



Note: STB is any of DS, RD, WR, PITACK, RxACK, or TxACK

Figure 8. DMA Read Cycle

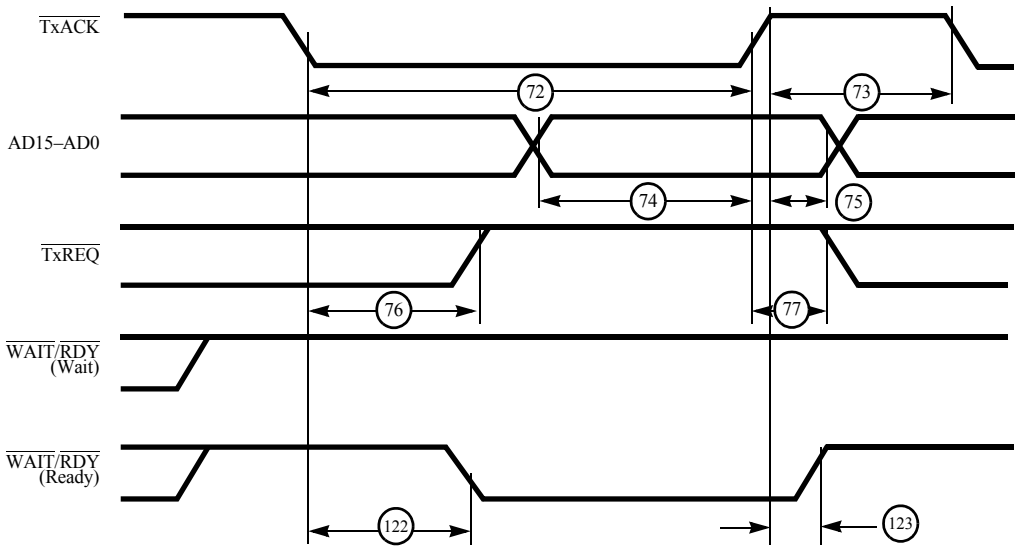


Figure 9. DMA Write Cycle

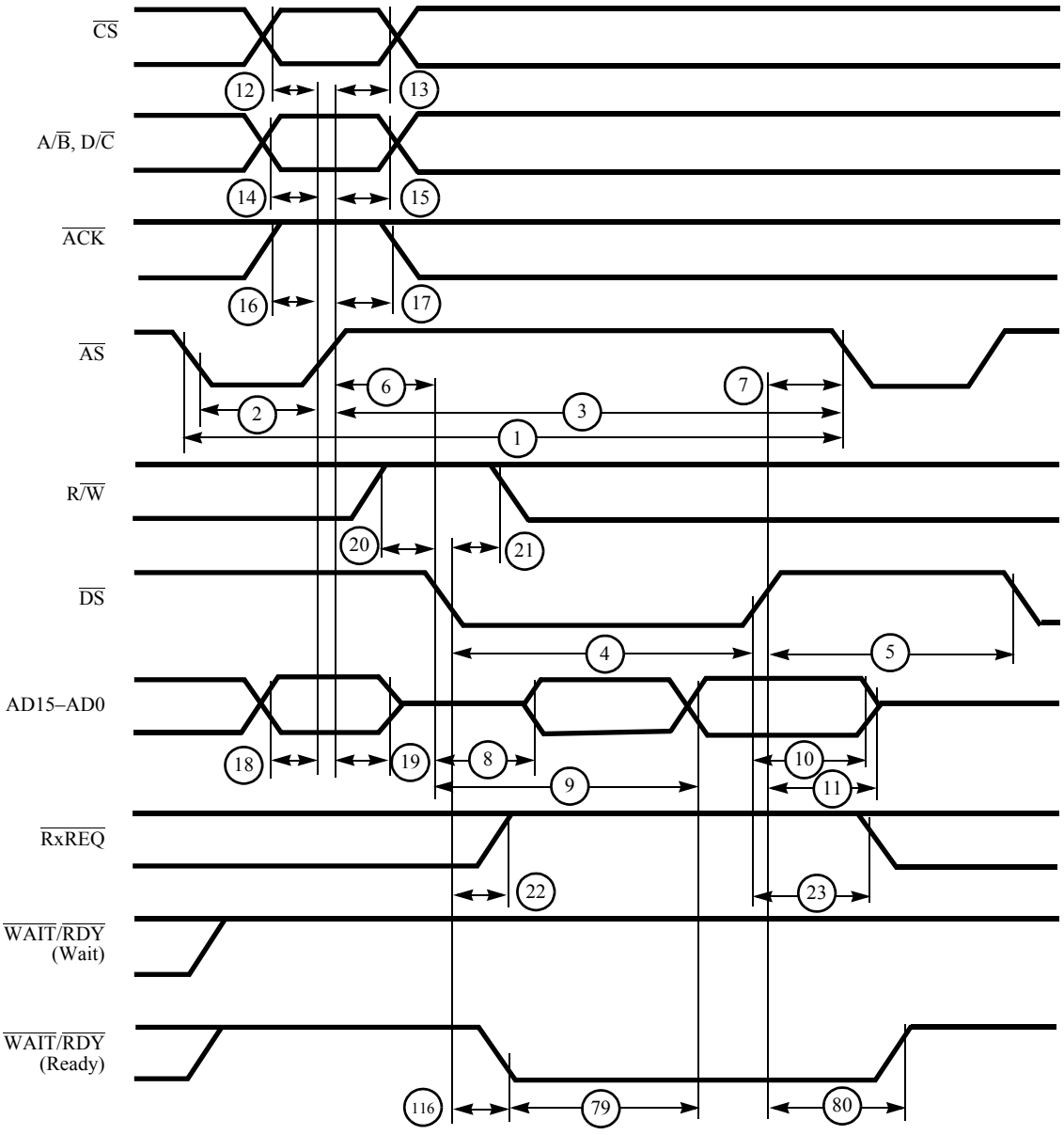


Figure 10. Multiplexed DS Read Cycle

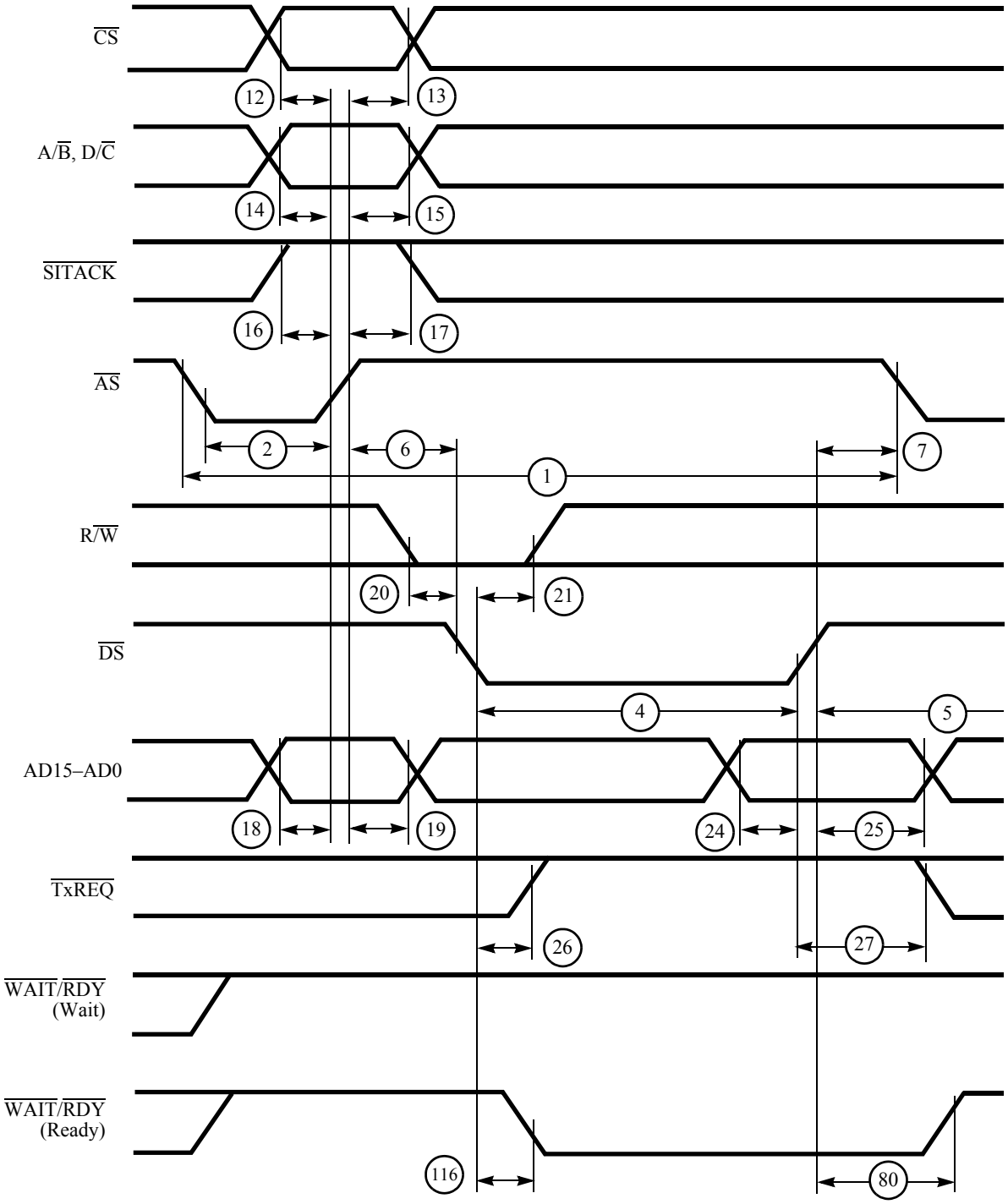


Figure 11. Multiplexed DS Write Cycle

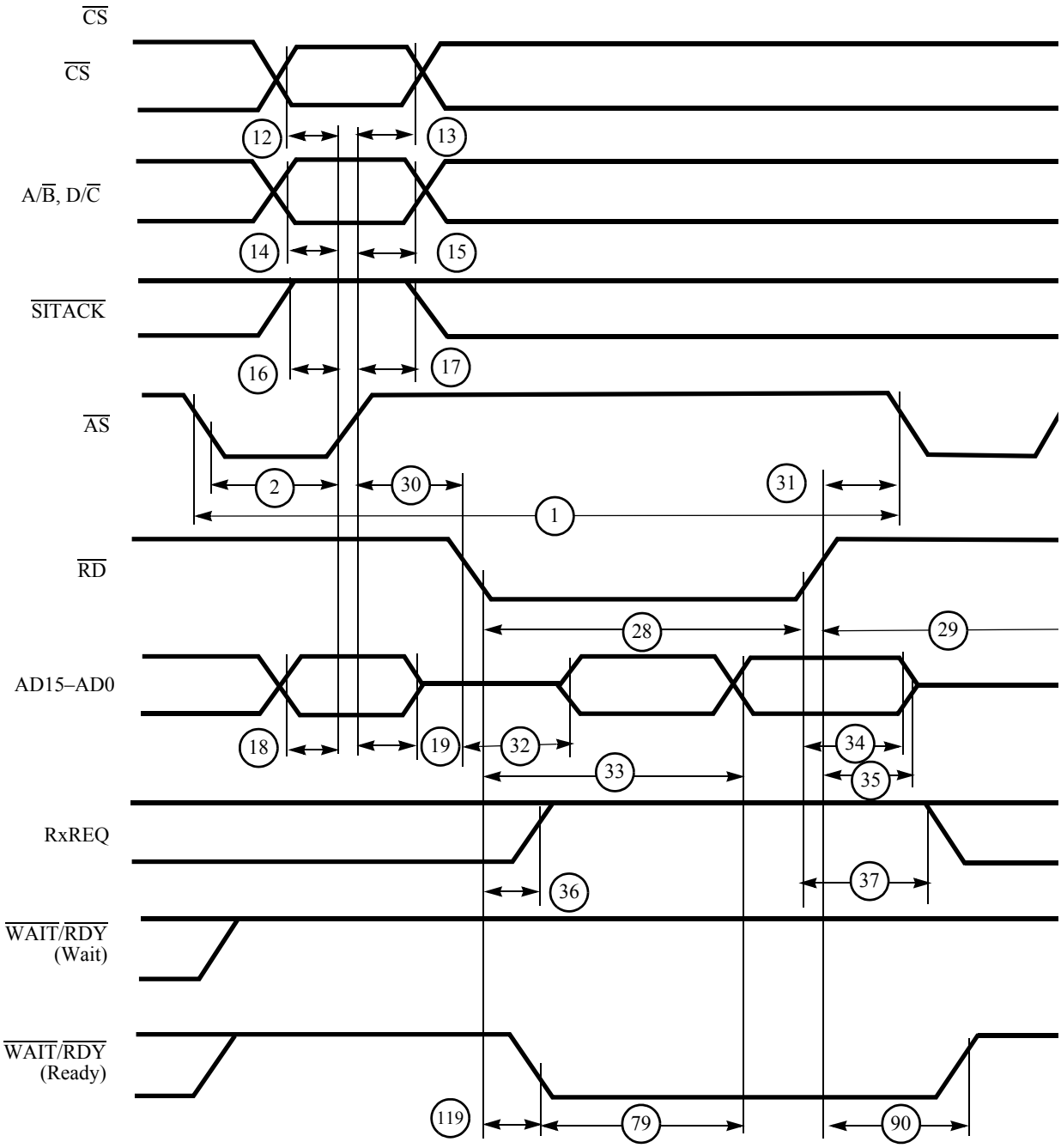


Figure 12. Multiplexed RD Read Cycle

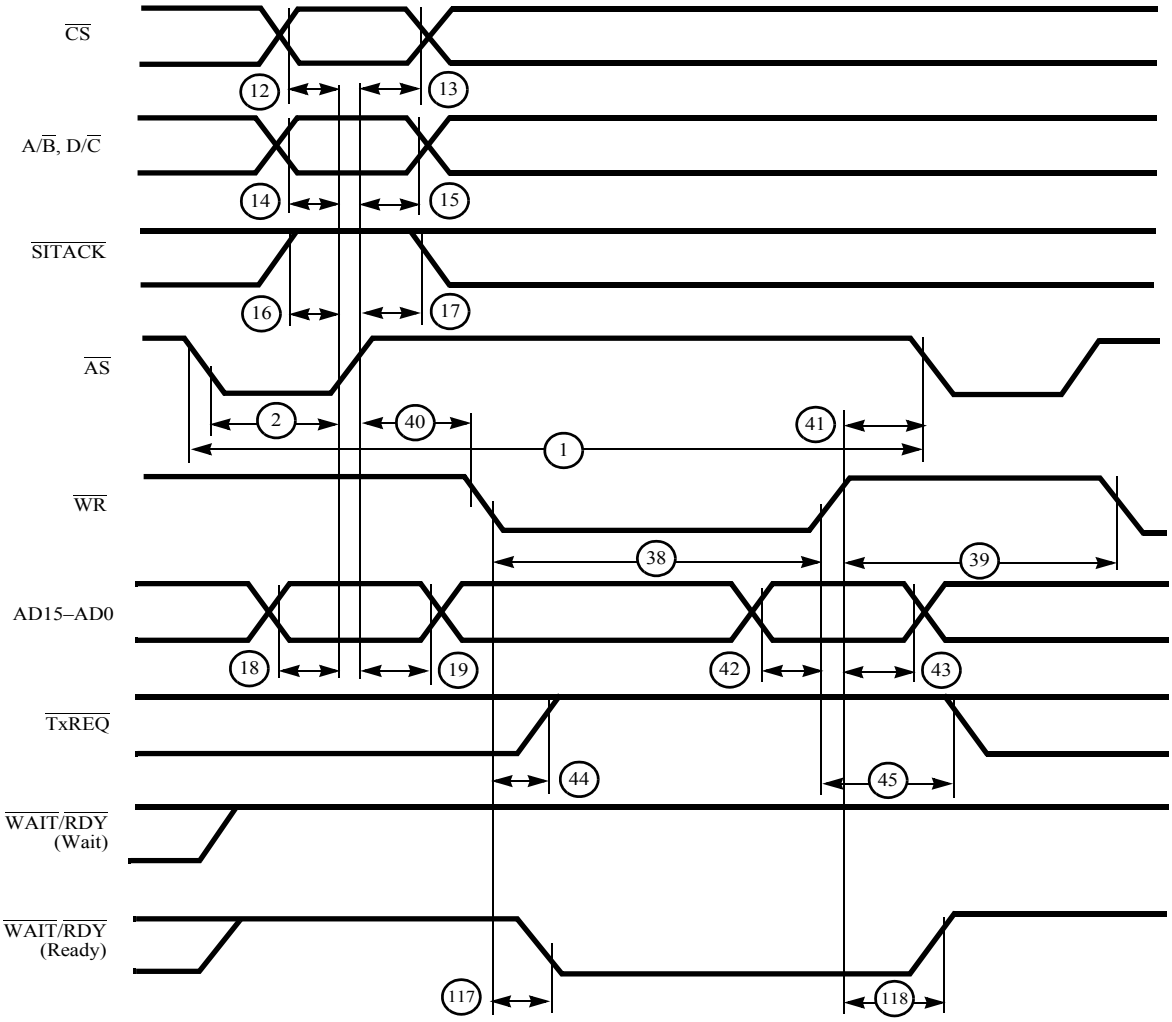


Figure 13. Multiplexed WR Write Cycle

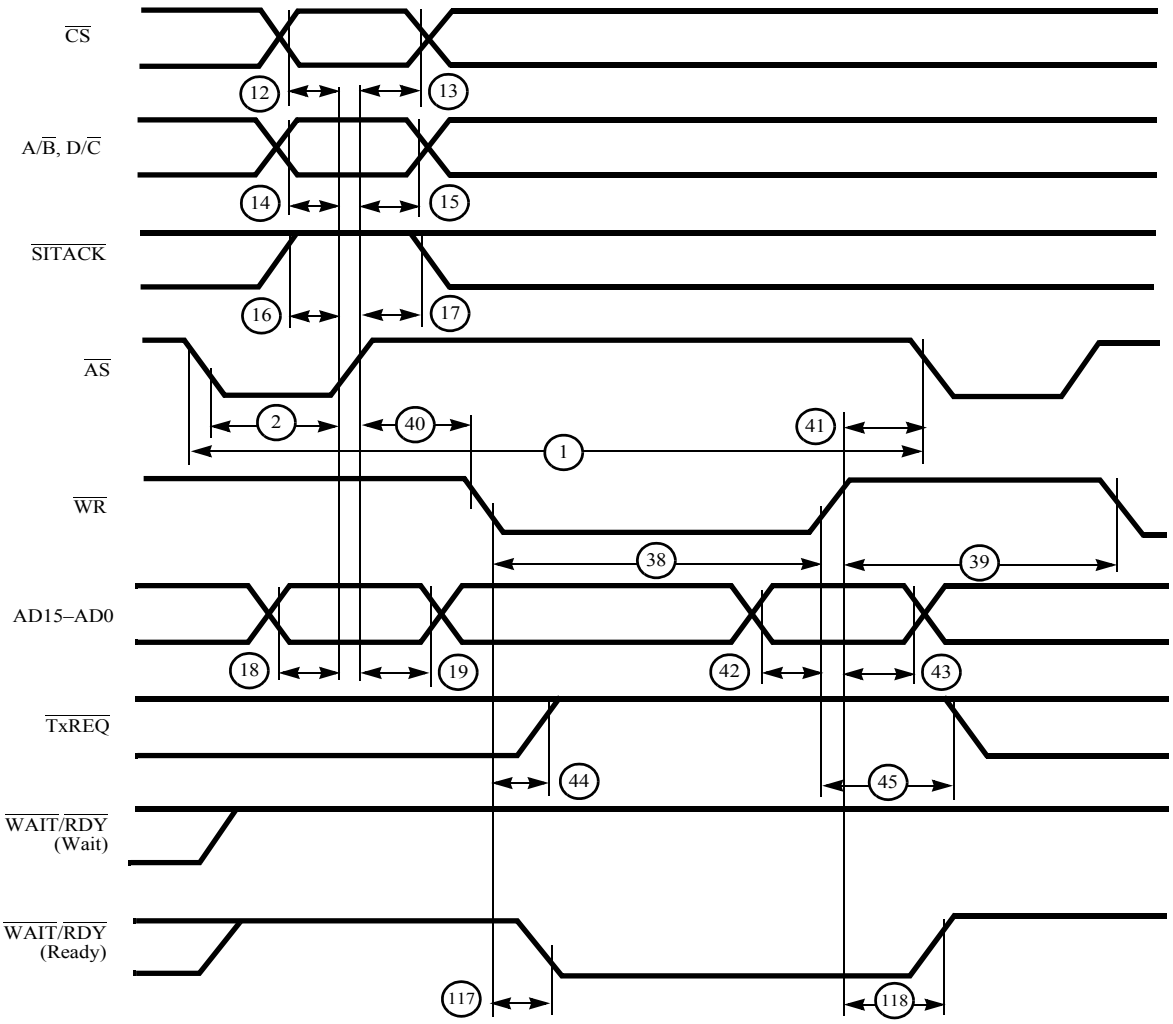


Figure 14. Nonmultiplexed DS Read Cycle

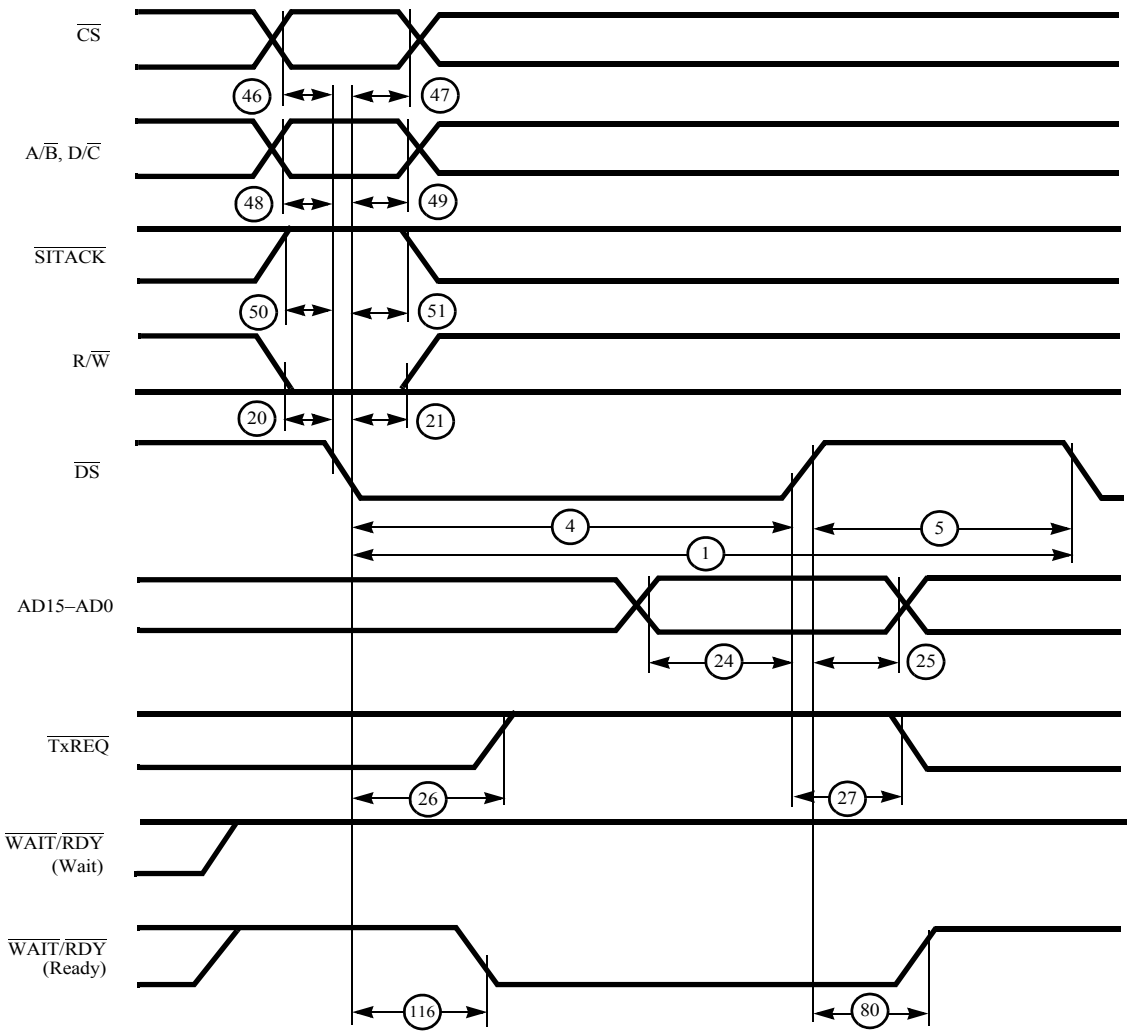


Figure 15. Nonmultiplexed DS Write Cycle

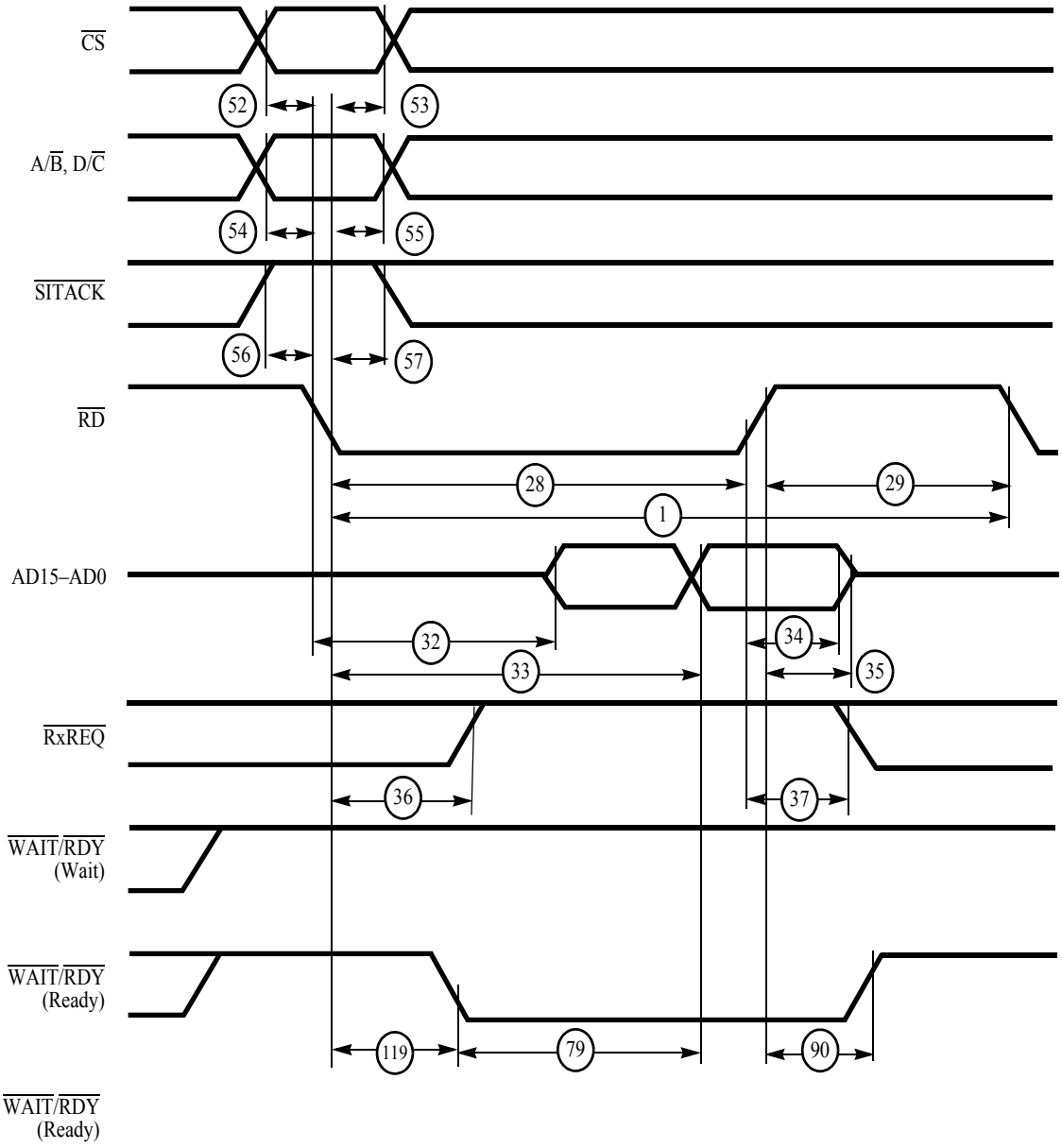


Figure 16. Nonmultiplexed RD Read Cycle

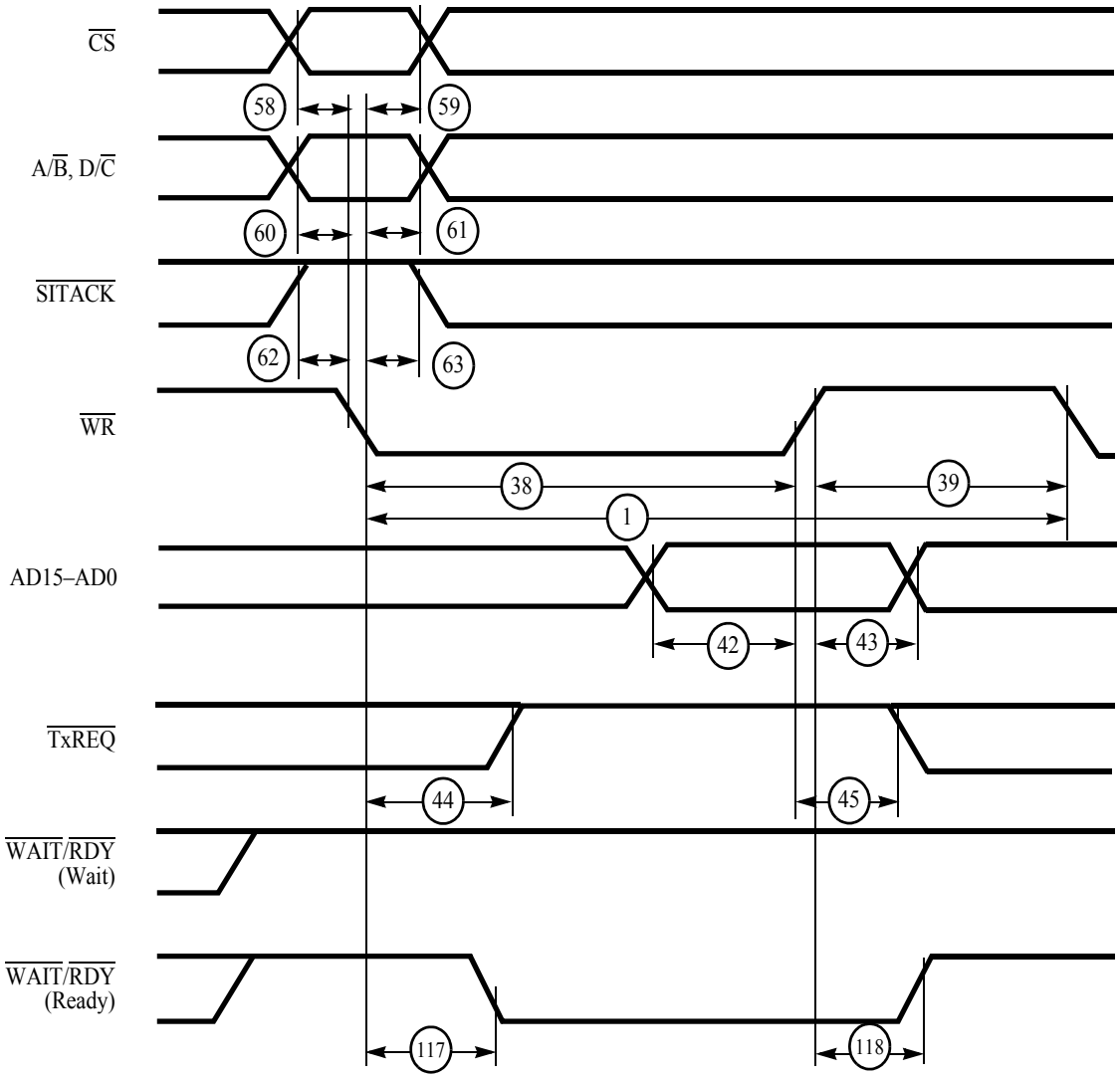


Figure 17. Nonmultiplexed WR Write Cycle

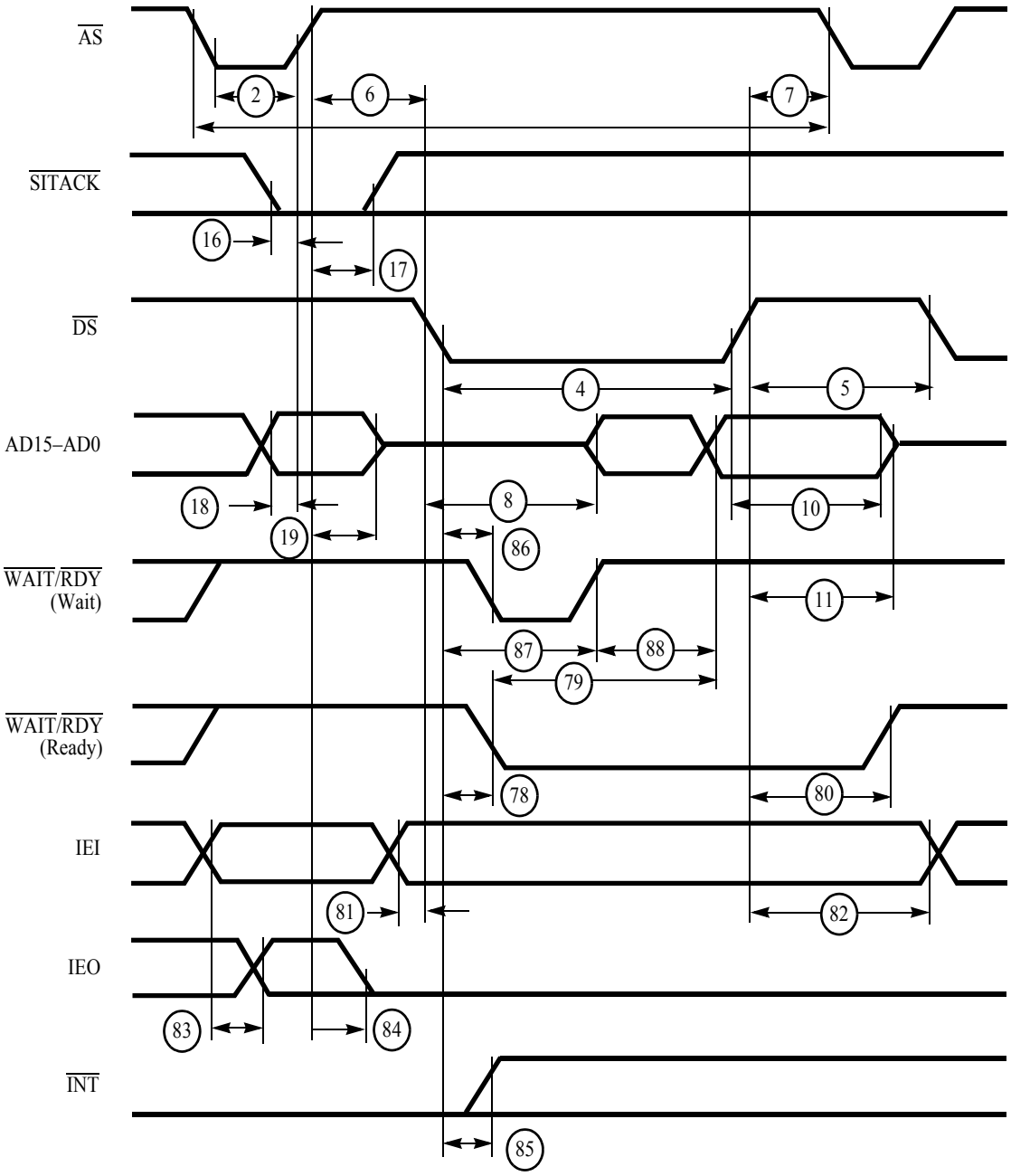


Figure 18. Multiplexed $\overline{DS}$ Interrupt Acknowledged Cycle

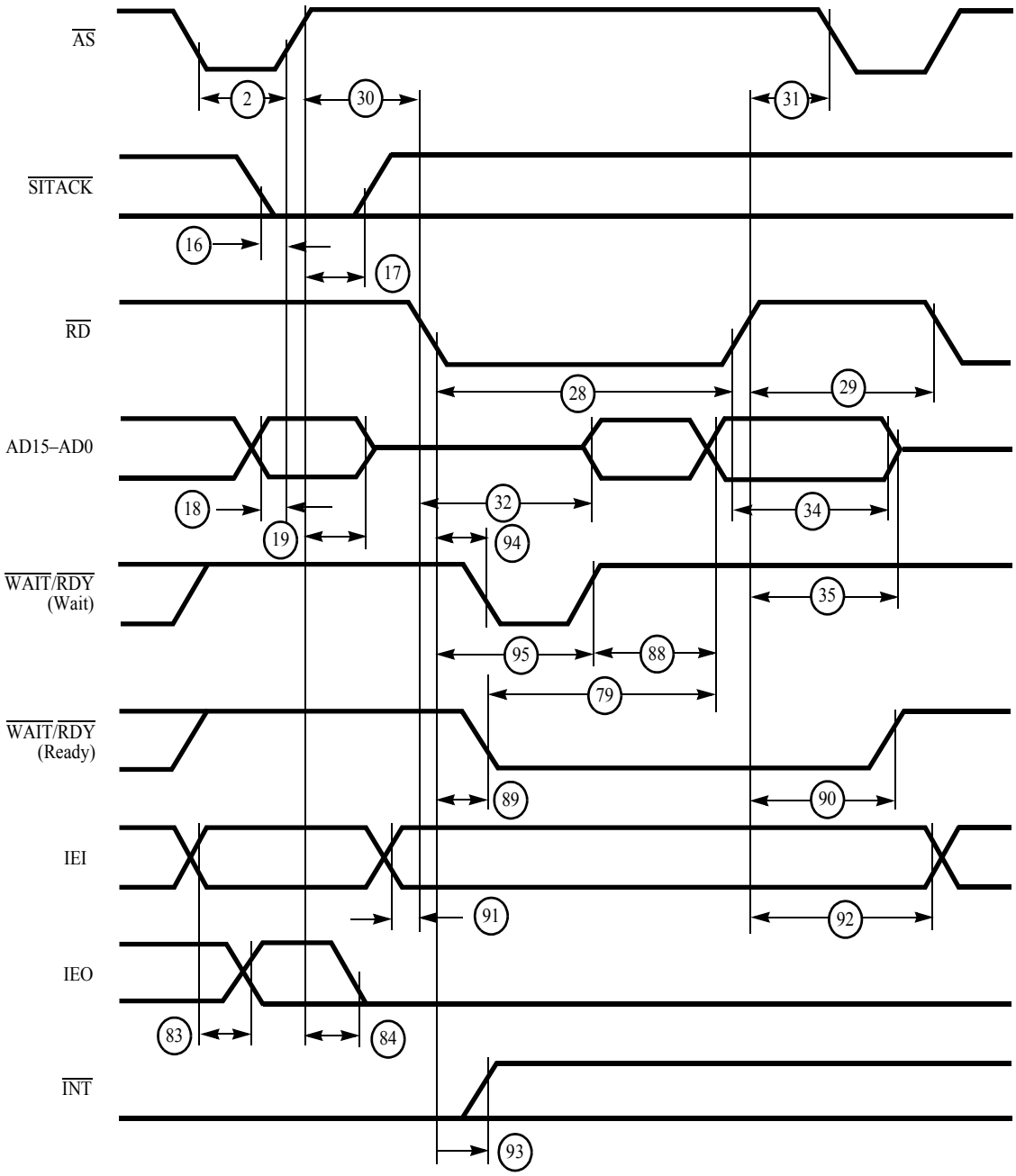


Figure 19. Multiplexed $\overline{RD}$ Interrupt Acknowledge Cycle

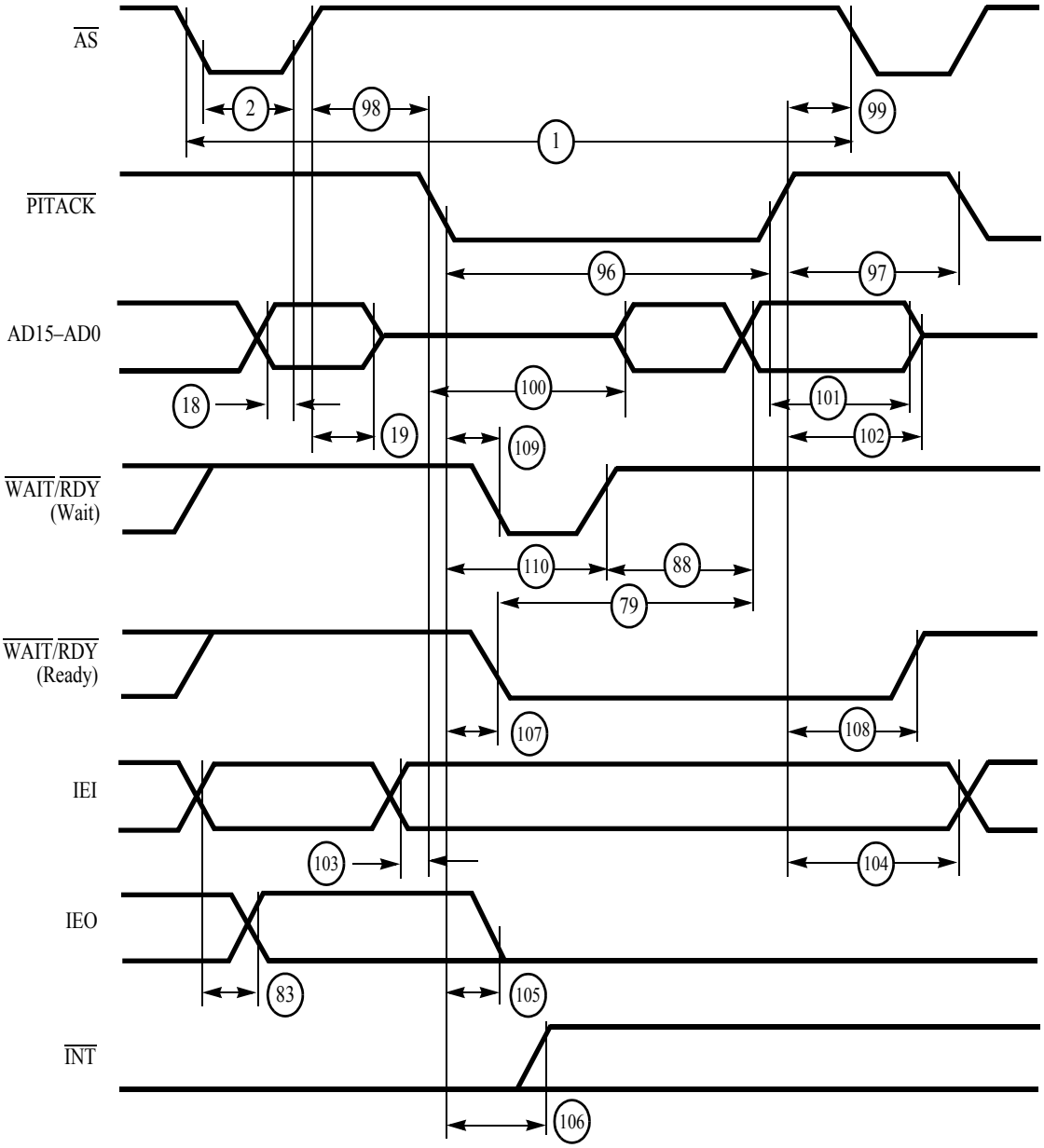


Figure 20. Multiplexed Pulsed Interrupt Acknowledge Cycle

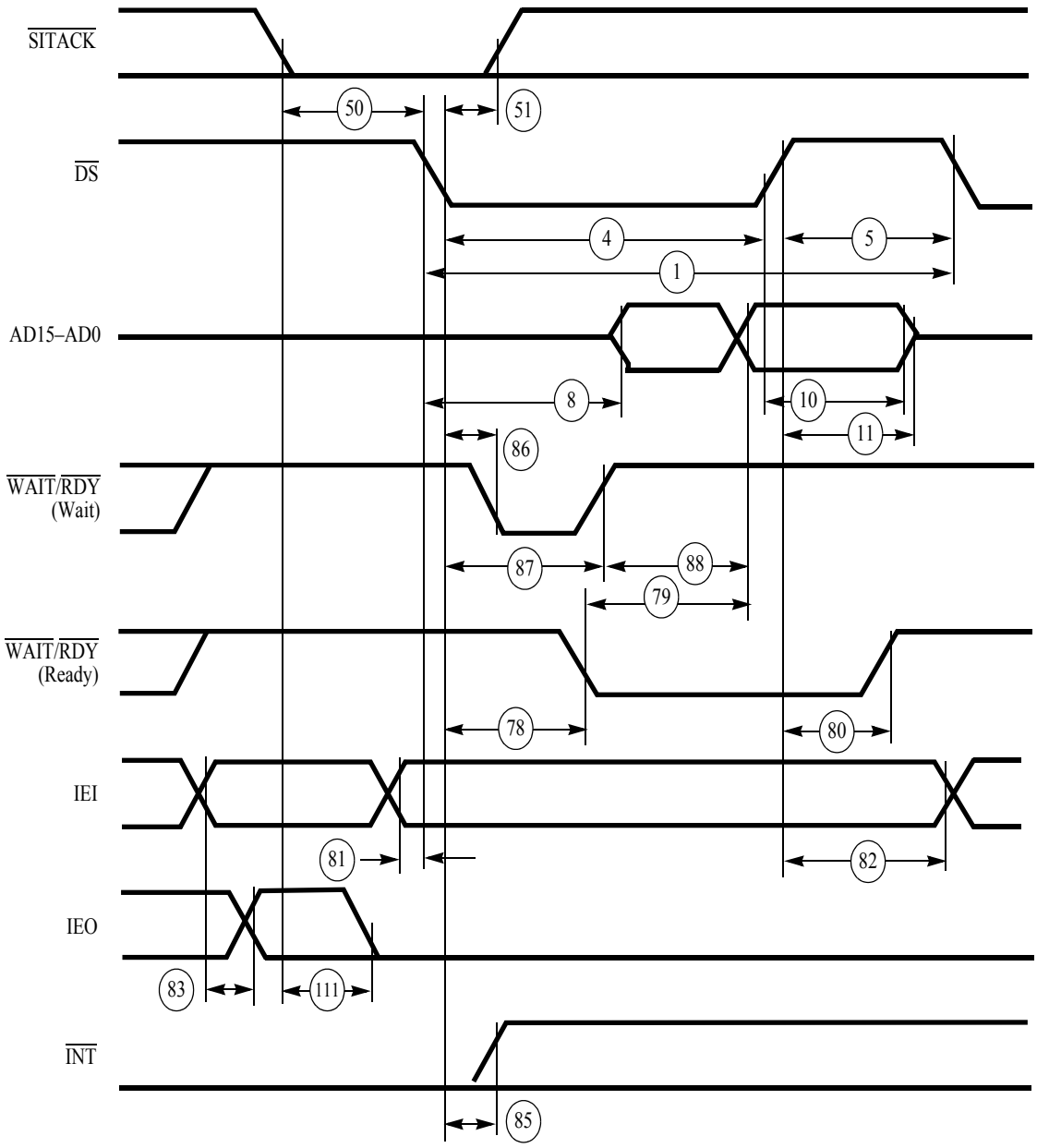


Figure 21. Nonmultiplexed DS Interrupt Acknowledge Cycle

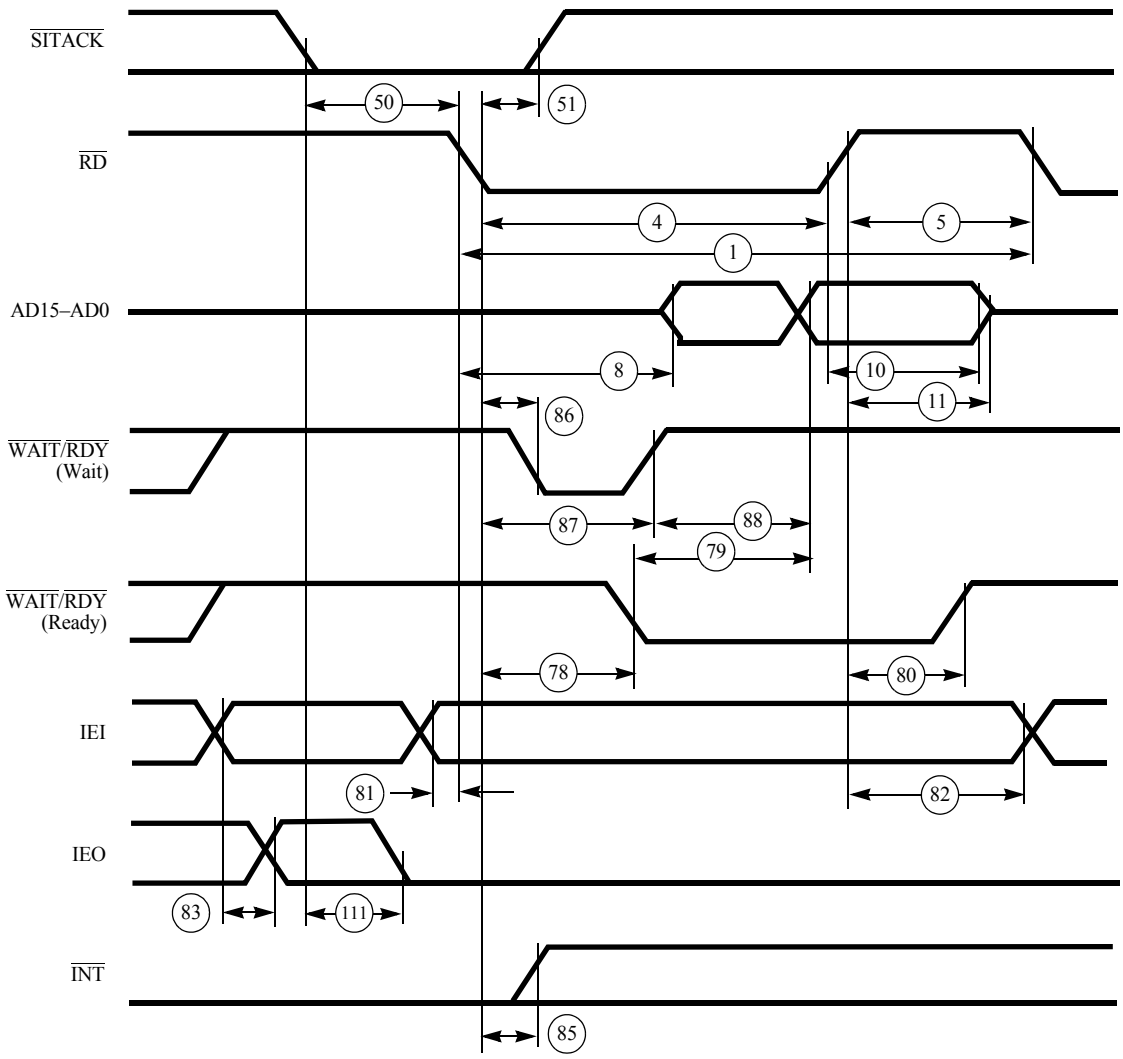


Figure 22. Nonmultiplexed $\overline{RD}$ Interrupt Acknowledge Cycle

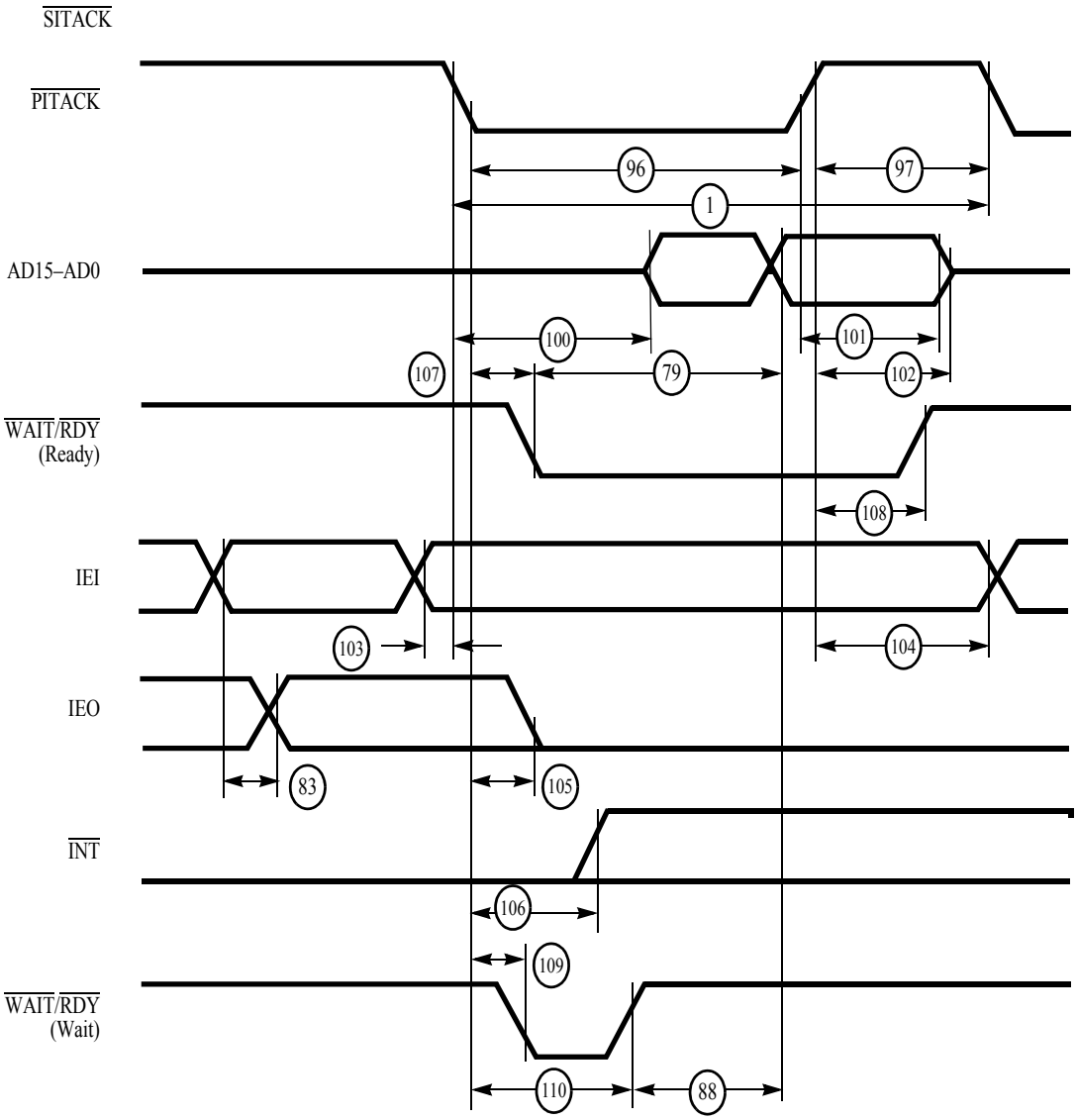


Figure 23. Nonmultiplexed Pulsed Interrupt Acknowledge Cycle

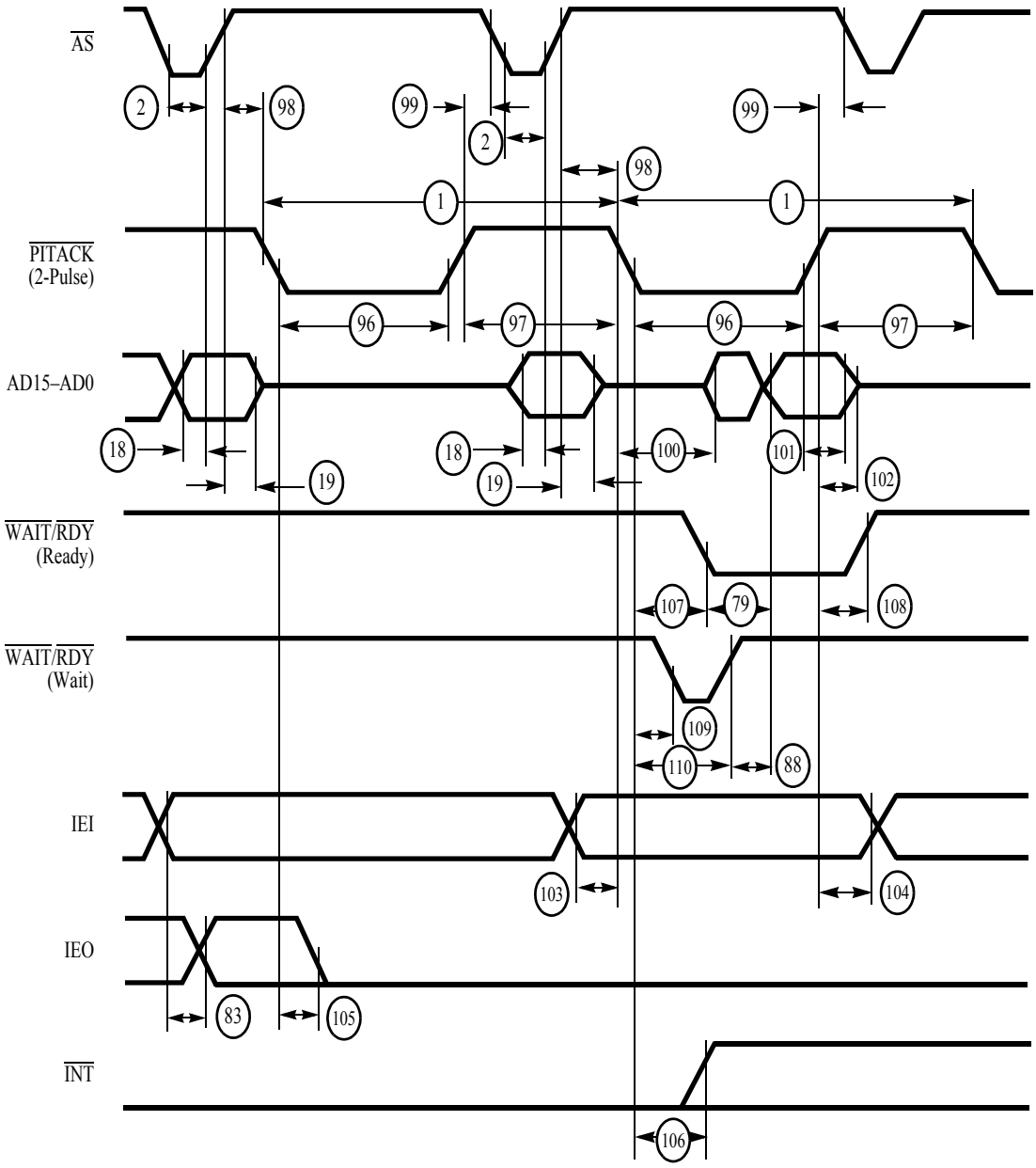


Figure 24. Multiplexed Double-Pulse Intack Cycle

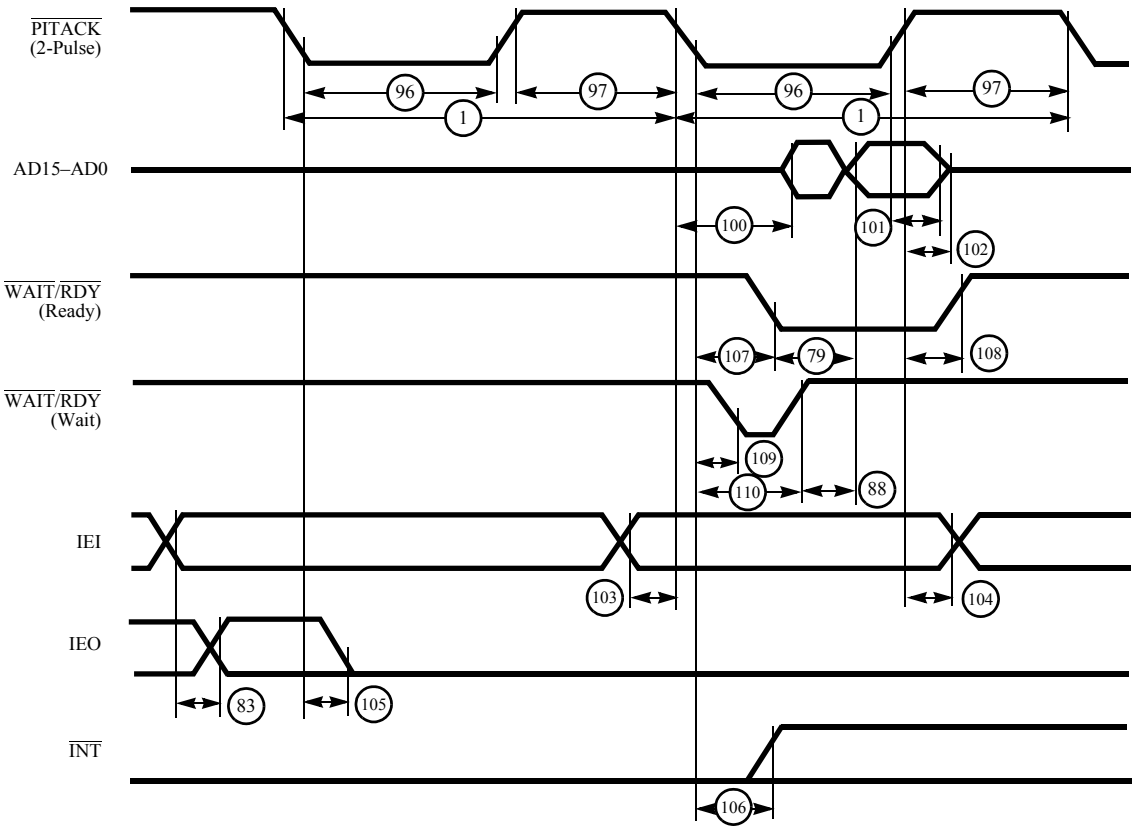


Figure 25. Nonmultiplexed Double-Pulse Intack Cycle

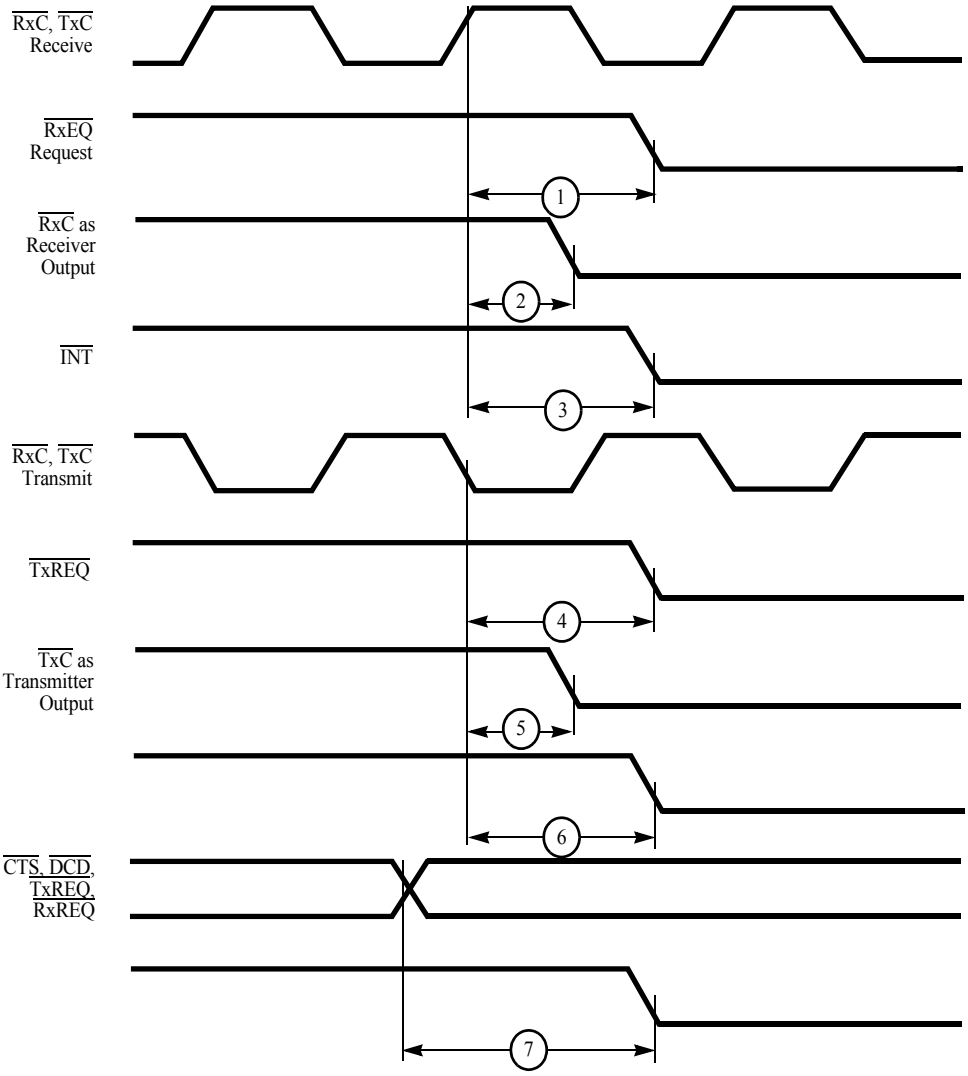
AC Characteristics

Table 6 lists Z16C30 General Timing.

Table 6. Z16C30 General Timing

No	Symbol	Parameter	Min	Max	Units	Notes
1	TsRxD(RxCr)	RxD to $\overline{RxC}$ Rise Setup Time (x1 Mode)	0		ns	1
2	ThRxD(RxCr)	RxD to $\overline{RxC}$ Rise Hold Time (x1 Mode)	40		ns	1
3	TsRxd(RxCf)	RxD to $\overline{RxC}$ Fall Setup Time (x1 Mode)	0		ns	1,3
4	ThRxD(RxCf)	RxD to $\overline{RxC}$ Fall Hold Time (x1 Mode)	40		ns	1,3
5	TsSy(RxC)	$\overline{DCD}$ as $\overline{SYNC}$ to $\overline{RxC}$ Rise Setup Time	0		ns	1
6	ThSy(RxC)	$\overline{DCD}$ as $\overline{SYNC}$ to $\overline{RxC}$ Rise Hold Time (x1 Mode)	40		ns	1
7	TdTxCf(TxD)	$\overline{TxC}$ Fall to TxD Delay		50	ns	2
8	TdTxCr(TxD)	$\overline{TxC}$ Rise to TxD Delay		50	ns	2,3
9	TwRxCh	$\overline{RxC}$ High Width	40		ns	1
11	TcRxC	$\overline{RxC}$ Cycle Time	100		ns	1
12	TwTxCh	$\overline{TxC}$ High Width	40		ns	2
13	TwTxCl	$\overline{TxC}$ Low Width	40		ns	2
14	TcTxC	$\overline{TxC}$ Cycle Time	100		ns	2
15	TwExT	$\overline{DCD}$ or CTS Pulse Width	70		ns	
16	TWSY	$\overline{DCD}$ as $\overline{SYNC}$ Input Pulse Width	70		ns	
17	TwCLKh	$\overline{CLK}$ High Width	20		ns	4
18	TwCLKl	$\overline{CLK}$ High Width	20		ns	4
19	TcCLK	$\overline{CLK}$ Cycle Time	50		ns	4

Notes
1. $\overline{RxC}$ is $\overline{RxC}$ or $\overline{TxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TxC}$ or $\overline{RxC}$, whichever is supplying the transmit clock.
3. Parameter applies only to FM encoding/decoding.
4. $\overline{CLK}$ is $\overline{RxC}$ or $\overline{TxC}$, when supplying DPLL, BRG, or CTR clock.



Note: CLK is RxC or TxC when supplying DPLL, BRG, or CTR clock.

Figure 26. Z16C30 System Timing

Table 7 lists Z16C30 System Timing

Table 7. Z16C30 System Timing

No	Symbol	Parameter	Min	Max	Units	Notes
1	TdRxC(REQ)	$\overline{RxC}$ Rise to $\overline{RxREQ}$ Valid Delay		100	ns	1
2	TdRxC(RxC)	$\overline{TxC}$ Rise to $\overline{RxC}$ as Receiver Output Valid Delay		100	ns	1
3	TdRxC(INT)	$\overline{RxC}$ Rise to $\overline{INT}$ Valid Delay		100	ns	1
4	TdTxC(REQ)	$\overline{TxC}$ Fall to $\overline{TxREQ}$ Valid Delay		100	ns	2
5	TdTxC(TxC)	$\overline{RxC}$ Fall to $\overline{TxC}$ as Transmitter Output Valid Delay		100	ns	2
6	TdTxC(INT)	$\overline{TxC}$ Fall to $\overline{INT}$ Valid Delay		100	ns	2
7	TdEXT(INT)	$\overline{CTS}$, $\overline{DCD}$, $\overline{TxREQ}$, $\overline{RxREQ}$ transition to $\overline{INT}$ Valid Delay		100	ns	

Notes
1. $\overline{RxC}$ is $\overline{RxC}$ or $\overline{TxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TxC}$ or $\overline{RxC}$, whichever is supplying the transmit clock.

Architecture

The USC internal structure includes two completely independent full-duplex serial channels, each with two baud rate generators, a digital phase-locked loop for clock recovery, transmit and receive character counters and a full-duplex DMA interface. The two serial channels share a common bus interface. The bus interface is designed to provide easy interface to most microprocessors, whether they employ a multiplexed or nonmultiplexed, 8-bit or16-bit bus structure. Each channel is controlled by a set of thirty 16-bit registers, nearly all of which are readable and writable. There is one additional 16-bit register in the bus interface used to configure the nature of the bus interface. The BCR functions are shown as follows:

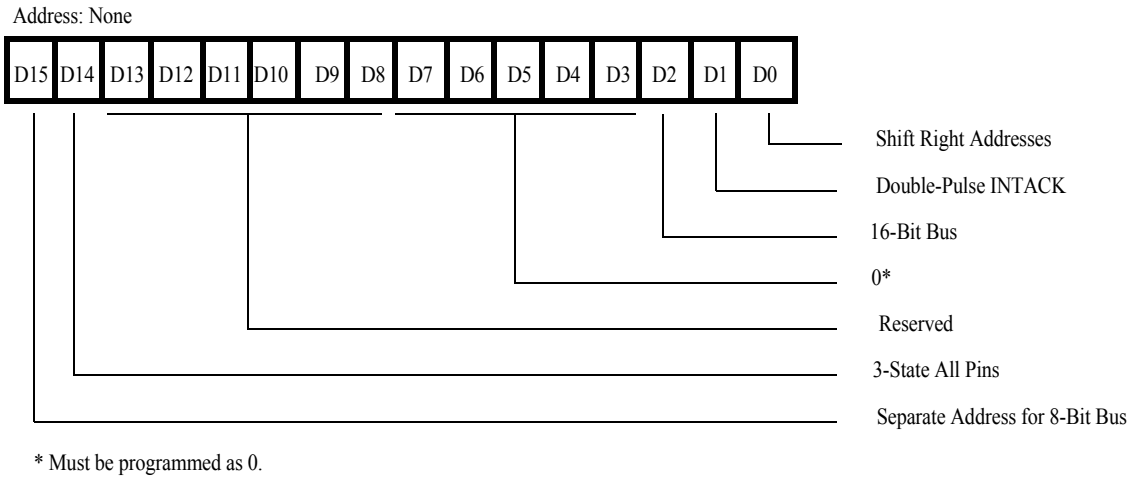


Figure 27. Bus Configuration Register

Data Path

Both the transmitter and the receiver in the channel are actually microcoded serial processors. As the data shifts through the transmit or receive shift register, the microcode watches for specific bit patterns, counts bits, and at the appropriate time transfers data to or from the FIFOs. The microcode also checks status and generates status interrupts as appropriate.

Functional Description

The functional capabilities of the USC are described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, the USC offers such features as read/write registers, a flexible bus interface, DMA interface support, and vectored interrupts.

Data Communications Capabilities

The USC provides two independent full-duplex channels programmable for use in any common data communication protocol. The receiver and transmitter modes are completely independent, as are the two channels. Each receiver and transmitter is supported by a 32-byte deep FIFO and a 16-bit message length counter. All modes allow optional even, odd, mark or space parity. Synchronous modes allow the choice of two 16-bit or one 32-bit CRC polynomial. Selection of from one to eight bits-per-character is available in both receiver and transmitter, independently. Error and status conditions are carried with the data in the receive and transmit FIFOs to greatly reduce the CPU overhead required to send or receive a message. Specific, appropriately timed interrupts are available to signal such conditions as overrun, parity error, framing error, end-of-frame, idle line received, sync acquired, transmit underrun, CRC sent, closing sync/flag sent, abort sent, idle line sent, and preamble sent. In addition, several useful internal signals such as receive FIFO load, received sync, transmit FIFO read and transmission complete may be sent to pins for use by external circuitry.

Asynchronous Mode—The receiver and transmitter can handle data at a rate of 1/16, 1/32, or 1/64 the clock rate. The receiver rejects start bits less than one-half a bit time and will not erroneously assemble characters following a framing error. The transmitter is capable of sending one, two, or anywhere in the range of 1/16 to two stop bits per character in 1/16 bit increments.

External Sync Mode—The receiver is synchronized to the receive data stream by an externally-supplied signal on a pin for custom protocol applications.

Isochronous Mode—Both transmitter and receiver may operate on start-stop (async) data using a 1x clock. The transmitter can send one or two stop bits.

Asynchronous With Code Violations—This is similar to Isochronous mode except that the start bit is replaced by a three bit-time code violation pattern as in MIL-STD 1553B. The transmitter can send zero, one or two stop bits.

Monosync Mode—In this mode, a single character is used for synchronization. The sync character can be either eight bits long with an arbitrary data character length, or programmed to match the data character length. The receiver is capable of automatically stripping sync characters from the received data stream. The transmitter may be pro-

grammed to automatically send CRC on either an underrun or at the end of a programmed message length.

Bisync Mode—This mode is identical to monosync mode except that character synchronization requires two successive characters for synchronization. The two characters need not be identical.

HDLC Mode—In this mode, the receiver recognizes flags, performs optional address matching, accommodates extended address fields, 8- or 16-bit control fields and logical control fields, performs zero deletion and CRC checking. The receiver is capable of receiving shared-zero flags, recognizes the abort sequence and can receive arbitrary length messages. The transmitter automatically sends opening and closing flags, performs zero insertion and can be programmed to send an abort, an extended abort, a flag or CRC, and a flag on transmit underrun. The transmitter can also automatically send the closing flag with optional CRC at the end of a programmed message length. Shared-zero flags are selected in the transmitter and a separate character length may be programmed for the last character in the frame.

Bisync Transparent Mode—In this mode, the synchronization pattern is DLE–SYN, programmable selected from either ASCII or EBCDIC encoding. The receiver recognizes control character sequences and automatically handles CRC calculation without CPU intervention. The transmitter can be programmed to send either SYN, DLE–SYN, CRC–SYN, or CRC–DLE–SYN upon underrun and can automatically send the closing DLE–SYN with optional CRC at the end of a programmed message length.

NBIP Mode—This mode is identical to async except that the receiver checks for the status of an additional address/data bit between the parity bit and the stop bit. The value of this bit is FIFO'ed along with the data. This bit is automatically inserted in the transmitter with the value that is FIFO'ed with the transmit data.

802.3 Mode—This mode implements the data format of IEEE 802.3 with 16-bit address compare. In this mode, $\overline{\text{DCD}}$ and $\overline{\text{CTS}}$ are used to implement the carrier sense and collision detect interactions with the receiver and transmitter.

Slaved Monosync Mode—This mode is available only in the transmitter and allows the transmitter (operating as though it were in monosync mode) to send data that is byte-synchronous to the data being received by the receiver.

HDLC Loop Mode—This mode is also available only in the transmitter and allows the USC to be used in an HDLC loop configuration. In this mode, the receiver is programmed to operate in HDLC mode so that the transmitter echoes received messages. Upon receipt of a particular bit pattern (actually a sequence of seven consecutive ones) the transmitter breaks the loop and inserts its own frame(s).

Data Encoding

The USC may be programmed to encode and decode the serial data in any of eight different ways as displayed in Figure 28 on page 44. The transmitter encoding method is selected independently of the receiver decoding method.

NRZ—In NRZ, a 1 is represented by a High level for the duration of the bit cell and a 0 is represented by a Low level for the duration of the bit cell.

NRZB—Data is inverted from NRZ.

NRZI-Mark—In NRZI-Mark, a 1 is represented by a transition at the beginning of the bit cell. That is, the level present in the preceding bit cell is reversed. A 0 is represented by the absence of a transition at the beginning of the bit cell.

NRZI-Space—In NRZI-Space, a 1 is represented by the absence of a transition at the beginning of the bit cell. That is, the level present in the preceding bit cell is maintained. A 0 is represented by a transition at the beginning of the bit cell.

Biphase-Mark—In Biphase-Mark, a 1 is represented by a transition at the beginning of the bit cell and another transition at the center of the bit cell. A 0 is represented by a transition at the beginning of the bit cell only.

Biphase-Space—In Biphase-Space, a 1 is represented by a transition at the beginning of the bit cell only. A 0 is represented by a transition at the beginning of the bit cell and another transition at the center of the bit cell.

Biphase-Level—In Biphase-Level, a 1 is represented by a High during the first half of the bit cell and a Low during the second half of the bit cell. A 0 is represented by a Low during the first half of the bit cell and a High during the second half of the bit cell.

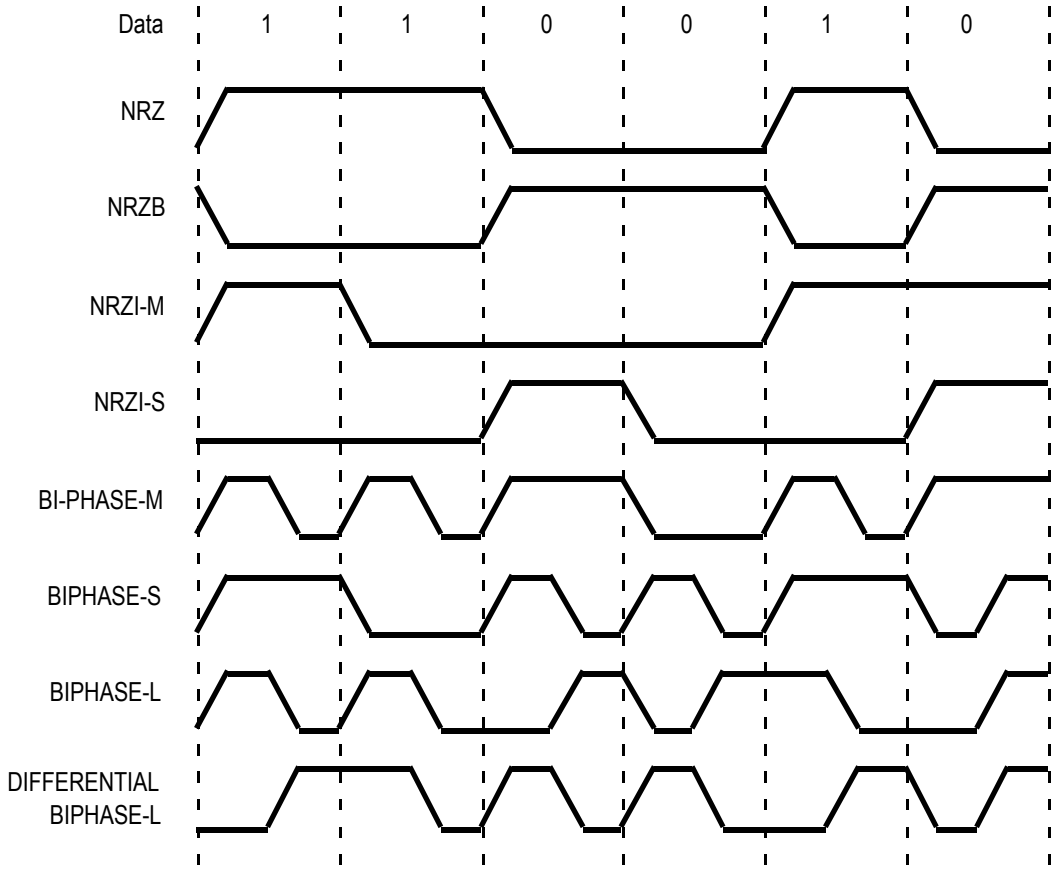


Figure 28. Data Encoding

Differential Biphase-Level—In Differential Biphase-Level, a 1 is represented by a transition at the center of the bit cell, with the opposite polarity from the transition at the center of the preceding bit cell. A 0 is represented by a transition at the center of the bit cell with the same polarity as the transition at the center of the preceding bit cell. In both cases there may be transitions at the beginning of the bit cell to set up the level required to make the correct center transition.

Character Counters

Each channel in the USC contains a 16-bit character counter for both receiver and transmitter. The receive character counter may be preset either under software control or automatically at the beginning of a receive message. The counter decrements with each receive character and at the end of the receive message the current value in the counter is automat-

ically loaded into a four-deep FIFO. This allows DMA transfer of data to proceed without CPU intervention at the end of a received message, as the values in the FIFO allow the CPU to determine message boundaries in memory. Similarly, the transmit character counter is loaded either under software control or automatically at the beginning of a transmit message. The counter is decremented with each write to the transmit FIFO. When the counter has decremented to 0, and that byte is sent, the transmitter automatically terminates the message in the appropriate fashion (usually CRC and the closing flag or sync character) without requiring CPU intervention.

Baud Rate Generators

Each channel in the USC contains two baud rate generators. Each generator consists of a 16-bit time constant register and a 16-bit down counter. In operation, the counter decrements with each baud rate generator clock, with the time constant automatically reloaded when the count reaches zero. The output of the baud rate generator toggles when the counter reaches a count of one-half of the time constant and again when the counter reaches zero.A new time constant may be written at any time but the new value will not take effect until the next load of the counter. The outputs of both baud rate generators are sent to the clock multiplexer for use internally or externally. The baud rate generator output frequency is related to the baud rate generator input clock frequency by the following equation:

Output frequency = Input frequency/(time constant + 1)

This allows an output frequency in the range of 1 to 1/65536 of the input frequency, inclusive.

Digital Phase-Locked Loop

Each channel in the USC contains a Digital Phase-Locked Loop (DPLL) to recover clock information from a data stream with NRZI or Biphase encoding. The DPLL is driven by a clock that is nominally 8, 16 or 32 times the receive data rate. The DPLL uses this clock, along the data stream, to construct a clock for the data. This clock may then be routed to the receiver, transmitter, or both, or to a pin for use externally. In all modes, the DPLL counts the input clock to create nominal bit times. As the clock is counted, the DPLL watches the incoming data stream for transitions. Whenever a transition is detected, the DPLL makes a count adjustment (during the next counting cycle), to produce an output clock which tracks the incoming bit cells. The DPLL provides properly phased transmit and receive clocks to the clock multiplexer.

Counters

Each channel contains two 5-bit counters, which are programmed to divide an input clock by 4, 8, 16, or 32. The inputs of these two counters are sent to the clock multiplexer. The counters are used as prescalers for the baud rate generators, or to provide a stable transmit clock from a common source when the DPLL is providing the receive clock.

Clock Multiplexer

The clock multiplexer in each channel selects the clock source for the various blocks in the channel and selects an internal clock signal to potentially be sent to either the $\overline{\text{RxC}}$ or $\overline{\text{TxC}}$ pin.

Test Modes

The USC can be programmed for local loopback or auto echo operation. In local loopback, the output of the transmitter is internally routed to the input of the receiver. This allows testing of the USC data paths without any external logic. Auto echo connects the RxD pin directly to the TxD pin. This is useful for testing serial links external to the USC.

I/O Interface Capabilities

The USC offers the choice of polling, interrupt (vectored or nonvectored) and block transfer modes to transfer data, status and control information to and from the CPU.

Polling

All interrupts are disabled. The registers in the USC are automatically updated to reflect current status. The CPU polls the Daisy Chain Control Register (DCCR) to determine status changes and then reads the appropriate status register to find and respond to the change in status. USC status bits are grouped according to function to simplify this software action.

Interrupt

When a USC responds to an interrupt acknowledge from the CPU, an interrupt vector may be placed on the data bus. This vector is held in the Interrupt Vector Register (IVR). To speed interrupt response time, the USC modifies three bits in this vector to indicate which type of interrupt is being requested.

Each of the six sources of interrupts in each channel of the USC (Receive Status, Receive Data, Transmit Status, Transmit Data, I/O Status, and Device Status) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt-Under-Service (IUS), and Interrupt Enable (IE). If the IE bit for a given source is set, that source can request interrupts. Note that individual sources within the six groups also have interrupt enable bits which are set for the particular source. In addition, there is a Master Interrupt Enable (MIE) bit in each channel which globally enables or disables interrupts within the channel.

The other two bits are related to the interrupt priority chain. A channel in the USC may request an interrupt only when no higher priority interrupt source is requesting one, e.g., when IEI is High for the channel. In this case the channel activates the $\overline{\text{INT}}$ signal. The CPU then responds with an interrupt acknowledge cycle, and the interrupting channel places a vector on the data bus.

In the USC, the IP bit signals that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority within the channel and external to the channel are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the channel being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an interrupt acknowledge cycle if there are no higher priority devices requesting interrupts.

There are six sources of interrupt in each channel: Receive Status, Receive Data, Transmit Status, Transmit Data, I/O Status, and Device Status, prioritized in that order within the channel. There are six sources of Receive Status interrupt, each individually enabled: exited hunt, idle line, break/abort, code violation/end-of-transmission/end-of-frame, parity error, and overrun error. The Receive Data interrupt is generated whenever the receive FIFO fills with data beyond the level programmed in the Receive Interrupt Control Register (RICR).

There are six sources of Transmit Status interrupt, each individually enabled: preamble sent, idle line sent, abort sent, end-of-frame/end-of-transmission sent, CRC sent, and underrun error. The Transmit Data interrupt is generated whenever the transmit FIFO empties below the level programmed in the Transmit Interrupt Control Register (TICR). The I/O Status interrupt serves to report transitions on any of six pins. Interrupts are generated on either or both edges with separate selection and enables for each pin. The pins programmed to generate I/O Status interrupts are $\overline{\text{RxC}}$, $\overline{\text{TxC}}$, $\overline{\text{RxREQ}}$, $\overline{\text{TxREQ}}$, $\overline{\text{DCD}}$, and $\overline{\text{CTS}}$. These interrupts are independent of the programmed function of the pins. The Device Status interrupt has four separately enabled sources: receive character count FIFO overflow, DPLL sync acquired, BRG1 zero count, and BRGO zero count.

Block Transfer Mode

The USC accommodates block transfers through DMA through the $\overline{\text{RxREQ}}$, $\overline{\text{TxREQ}}$, $\overline{\text{RxACK}}$, and $\overline{\text{TxACK}}$ pins. The $\overline{\text{RxREQ}}$ signal is activated when the fill level of the receive FIFO exceeds the value programmed in the RICR. The DMA may respond with either a normal bus transaction or by activating the $\overline{\text{RxACK}}$ pin to read the data directly (fly-by transfer). The $\overline{\text{TxREQ}}$ signal is activated when the empty level of the transmit FIFO falls below the value programmed in the TICR. The DMA may respond either with a normal bus transaction or by activating the $\overline{\text{TxACK}}$ pin to write the data directly (fly-by transfer). The $\overline{\text{RxACK}}$ and $\overline{\text{TxACK}}$ pin functions for this mode are controlled by the Hardware Configuration Register (HCR). Then using the $\overline{\text{RxACK}}$ and $\overline{\text{TxACK}}$ pins to transfer data, no chip select is necessary; these are dedicated strobes for the appropriate FIFO.

Programming

The registers in each USC channel are programmed by the system to configure the channels. Before this can occur, however, the system must program the bus interface by writing to the Bus Configuration Register (BCR). The BCR has no specific address and is only

accessible immediately after a hardware reset of the device. The first write to the USC, after a hardware reset, programs the BCR. From that time on, the normal channel registers may be accessed. No specific address need be presented to the USC for the BCR write because the first write after a hardware reset is automatically programmed for the BCR.

In the multiplexed bus case, all registers are directly addressable through the address latched by $\overline{AS}$ at the beginning of a bus transaction. The address is decoded from either AD6–AD0 or AD7–AD1. This is controlled by the Shift Right/Shift Left bit in the BCR. The address maps for these two cases are listed in Table 8. The D/$\overline{C}$ pin is still used to directly access the receive and transmit data registers (RDR and TDR) in the multiplexed bus; if D/$\overline{C}$ is High the address latched by $\overline{AS}$ is ignored and an access of RDR or TDR is performed.

In the nonmultiplexed bus case, the registers in each channel are accessed indirectly using the address pointer in the Channel Command/Address Register (CCAR) in each channel. The address of the desired register is first written to the CCAR and then the selected register is accessed; the pointer in the CCAR is automatically cleared after this access. The RDR and TDR are accessed directly using the D/$\overline{C}$ pin, without disturbing the contents of the pointer in the CCAR.

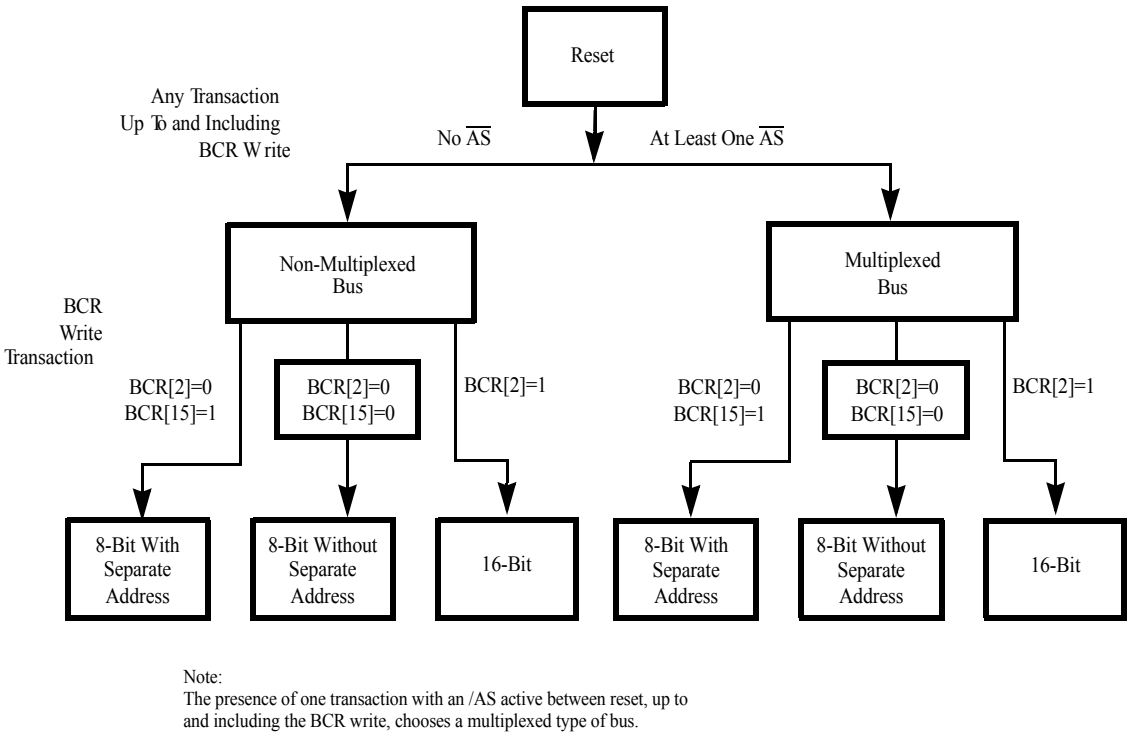
Table 8. Multiplexed Bus Address Assignments

Address Signal	Shift Left	Shift Right
Byte/Word Access	AD7	AD6
Address 4	AD6	AD5
Address 3	AD5	AD4
Address 2	AD4	AD3
Address 1	AD3	AD2
Address 0	AD2	AD1
Upper/Lower Byte Select	AD1	AD0

Notes:

1. *The Channel Reset bit in the CCAR places the channel in the reset state. To exit this reset state either a word of all zeros must be written to the CCAR (16-bit bus), or a byte of all zeros must be written to the lower byte of the CCAR (8-bit bus).*
2. *After reset, the transmit and receive clocks are not connected. The first thing that should be done in any initialization sequence is a write to the Clock Mode Control Register (CMCR) to select a clock source for the receiver and transmitter.*

The register addressing is listed in Table 9 on page 50 while the bit assignments for the registers are displayed in Figure 29.



Note:
The presence of one transaction with an /AS active between reset, up to and including the BCR write, chooses a multiplexed type of bus.

Figure 29. BCR Reset Sequence and Bit Assignments

Table 9. Register Address List

Address A4–A0		
00000	CCAR	Channel Command/Address Register
00001	CMR	Channel Mode Register
00010	CCSR	Channel Command/Status Register
00011	CCR	Channel Control Register
00110	TMDR	Test Mode Data Register
00111	TMCR	Test Mode Control Register
01000	CMCR	Clock Mode Control Register
01001	HCR	Hardware Configuration Register
01010	IVR	Interrupt Vector Register
01011	IOCR	I/O Control Register
01100	ICR	Interrupt Control Register
01101	DCCR	Daisy-Chain Control Register
01110	MISR	Misc Interrupt Status Register
01111	SICR	Status Interrupt Control Register
1X000	RDR	Receive Data Register (Read Only)
10001	RMR	Receive Mode Register
10010	RCSR	Receive Command/Status Register
10011	RICR	Receive Interrupt Control Register
10100	RSR	Receive Sync Register
10101	RCLR	Receive Count Limit Register
10110	RCCR	Receive Character Count Register
10111	TC0R	Time Constant 0 Register
1X000	TDR	Transmit Data Register (Write Only)
11001	TMR	Transmit Mode Register
11010	TCSR	Transmit Command/Status Register
11011	TICR	Transmit Interrupt Control Register
11100	TSR	Transmit Sync Register
11101	TCLR	Transmit Count Limit Register
11110	TCCR	Transmit Character Count Register
11111	TC1R	Time Constant 1 Register
XXXXX	BCR	Bus Configuration Register

Control Registers

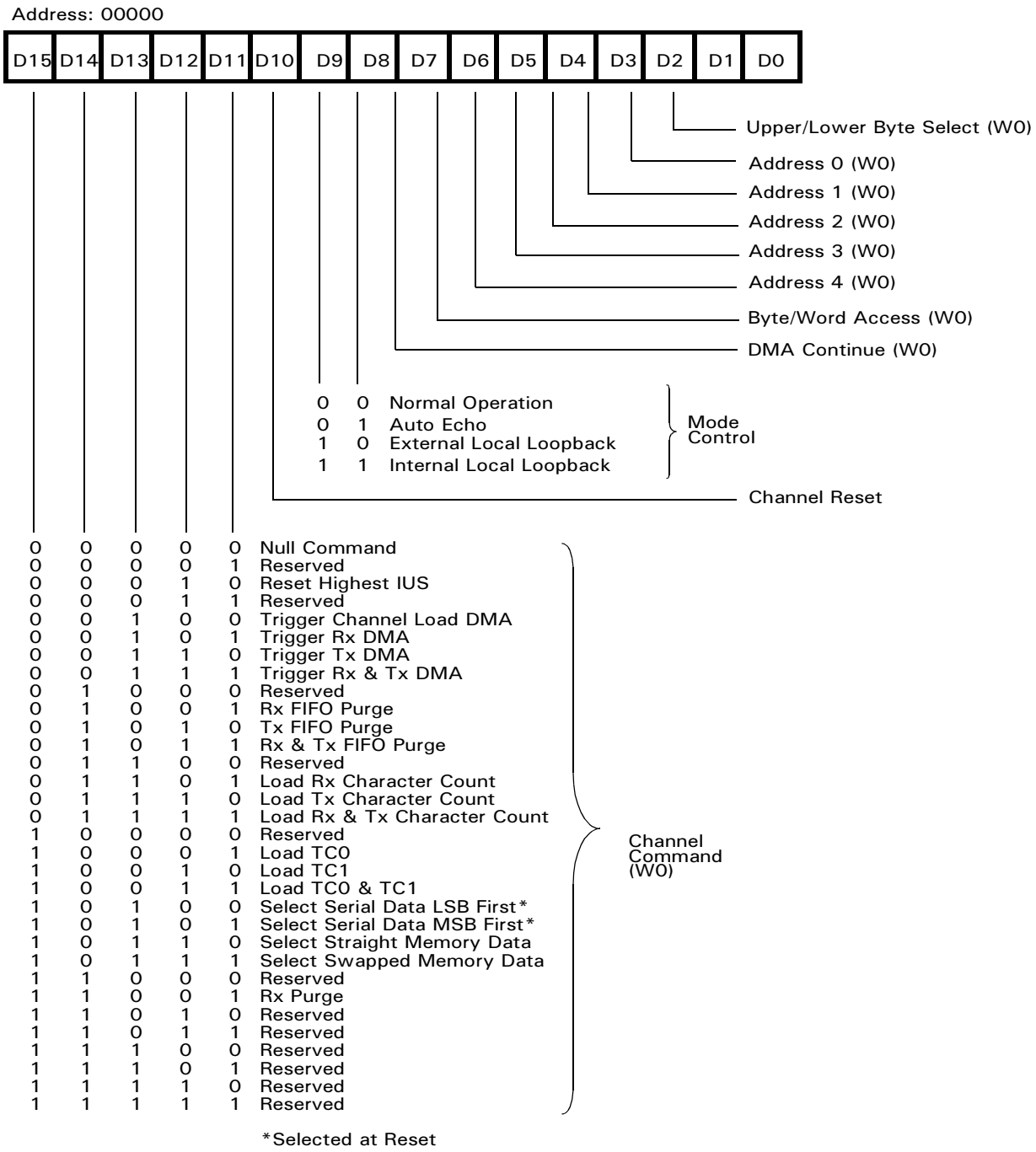


Figure 30. Channel Command/Address Register

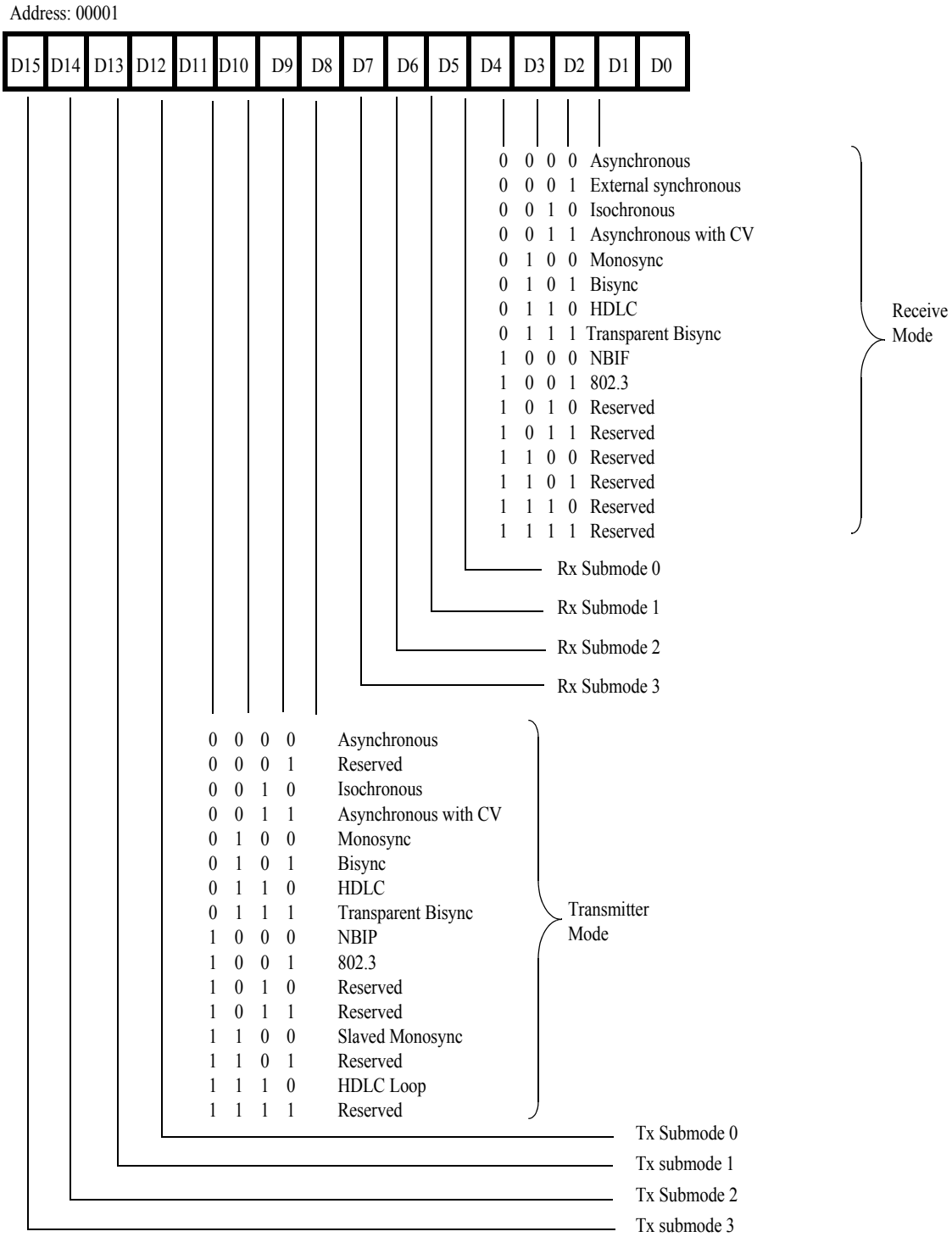


Figure 31. Channel Mode Register

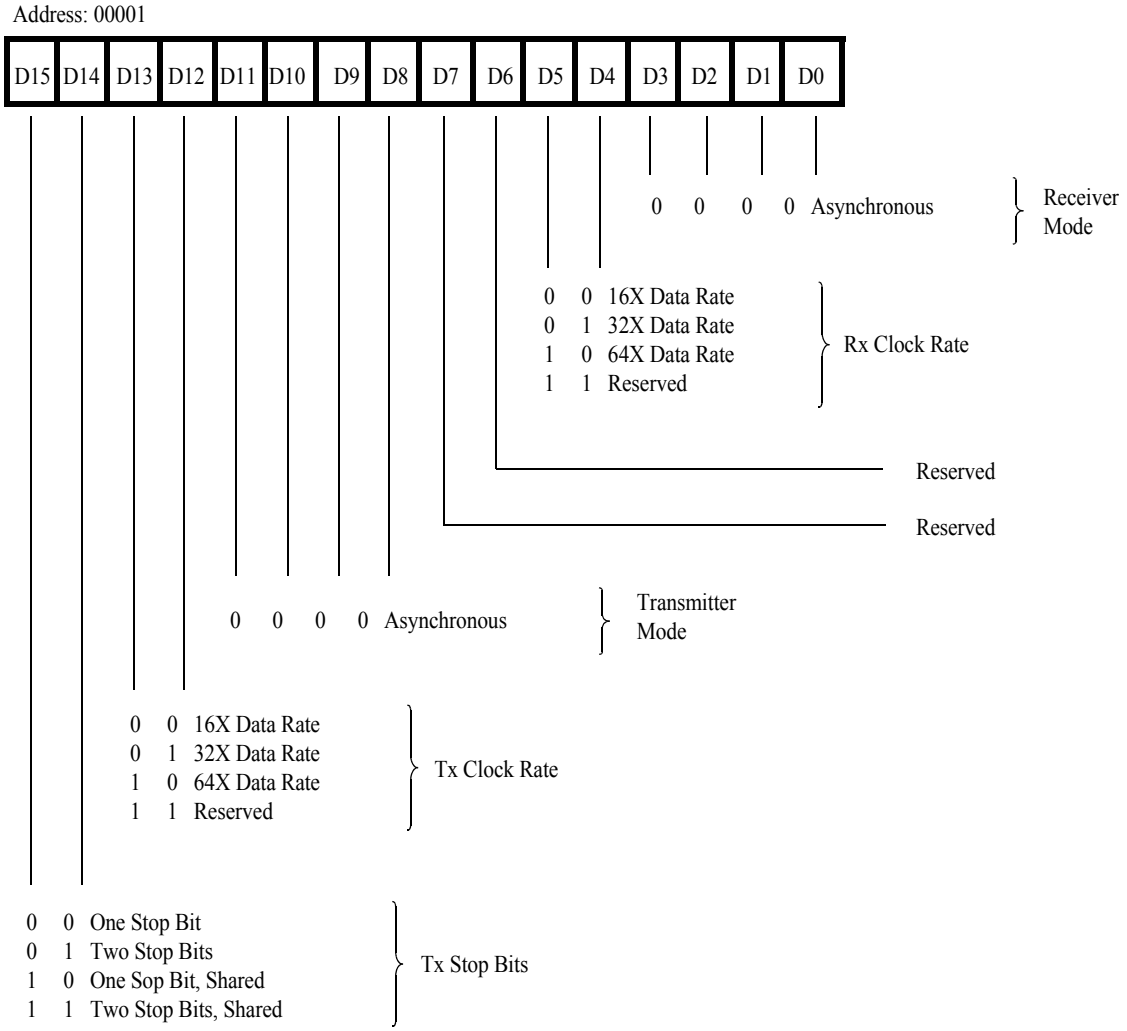


Figure 32. Channel Mode Register, Asynchronous Mode

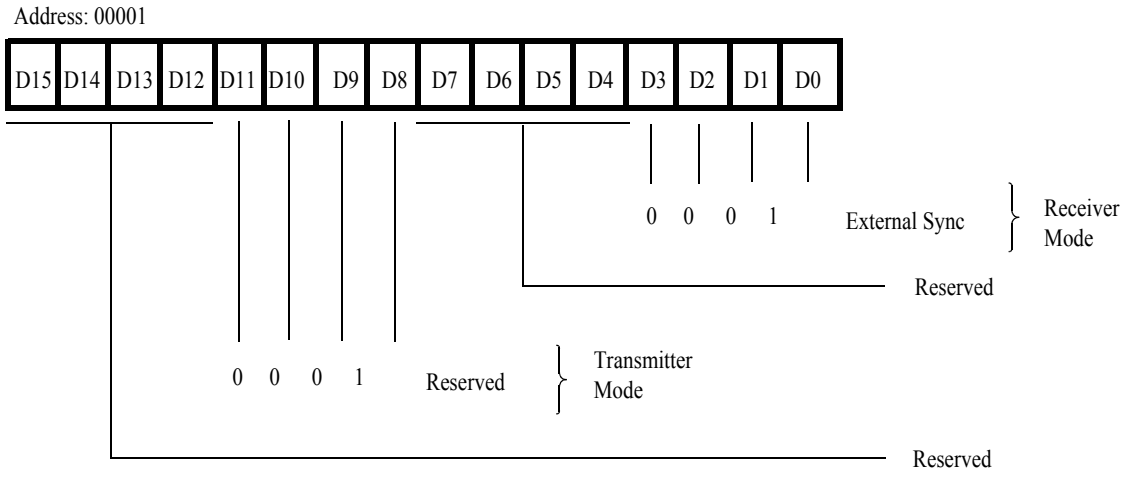


Figure 33. Channel Mode Register, External Sync Mode

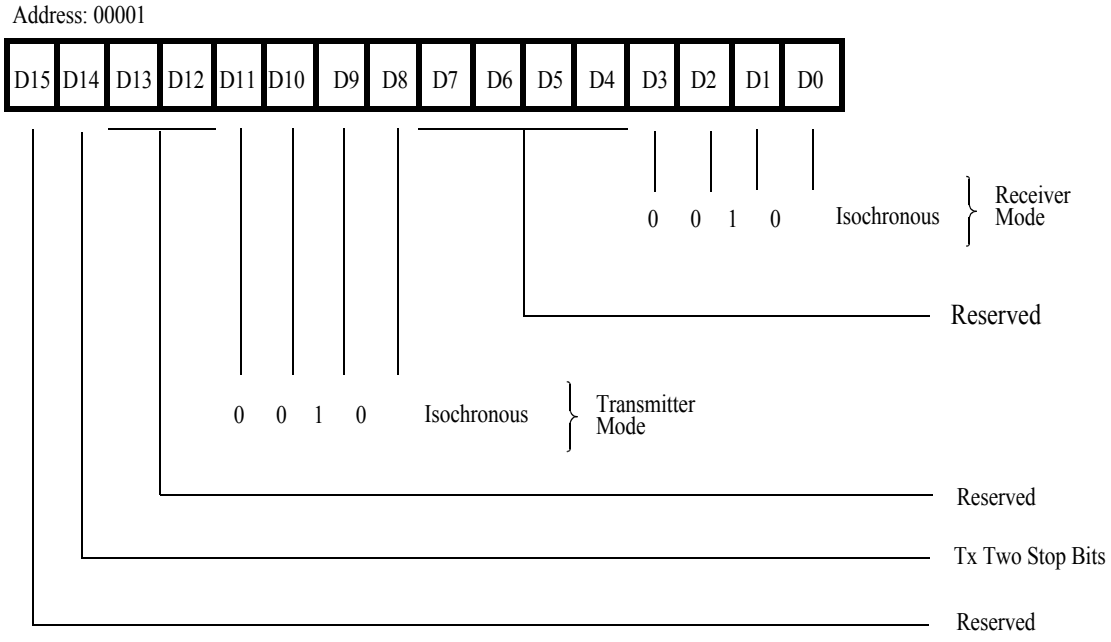


Figure 34. Channel Mode Register, Isochronous Mode

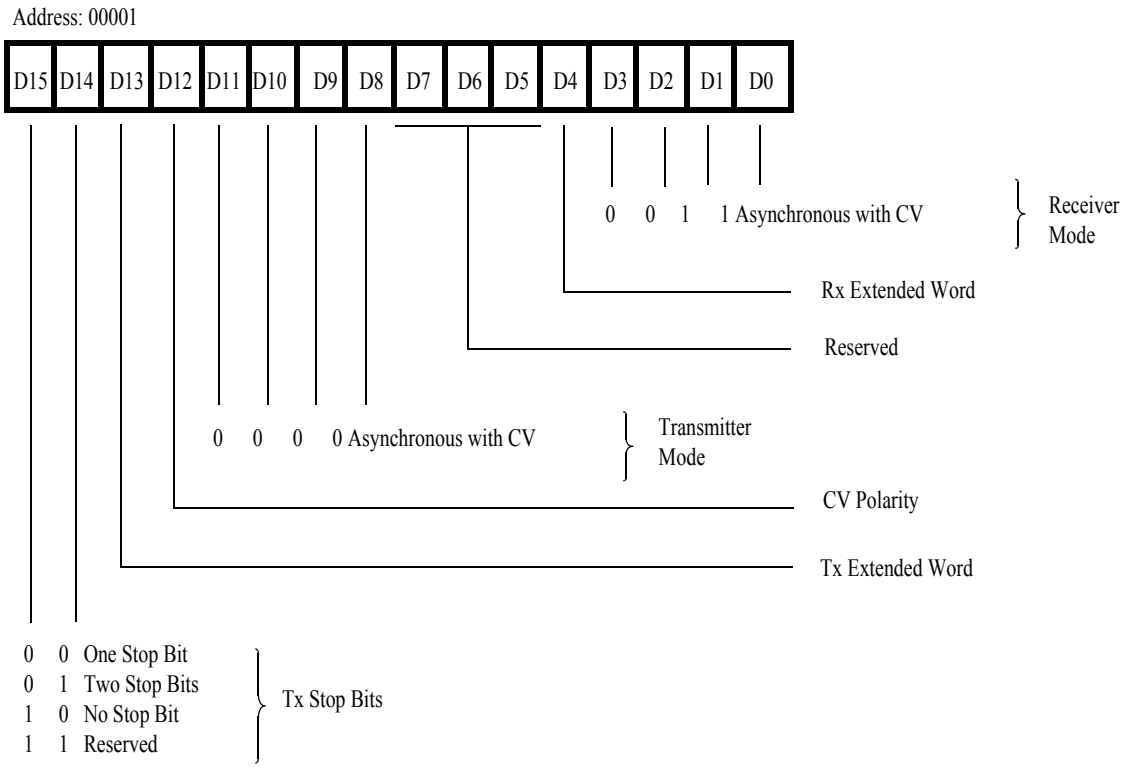


Figure 35. Channel Mode Register, Asynchronous Mode with Code Violation (MIL STD 1553)

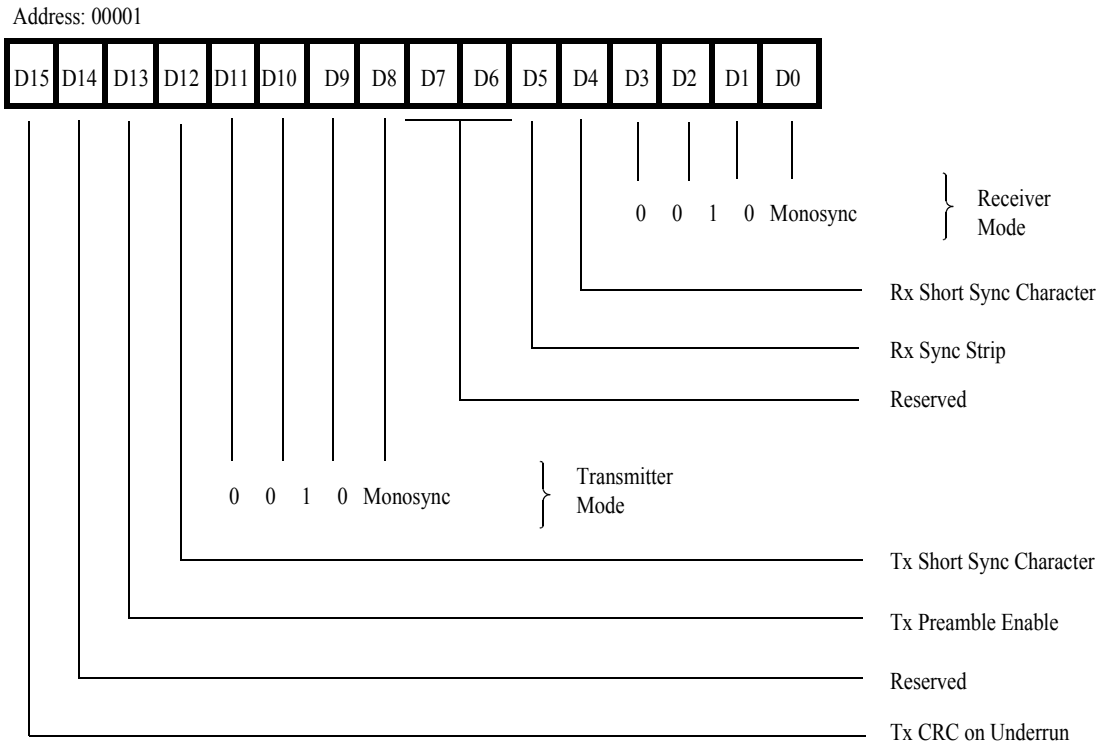


Figure 36. Channel Mode Register, Monosync Mode

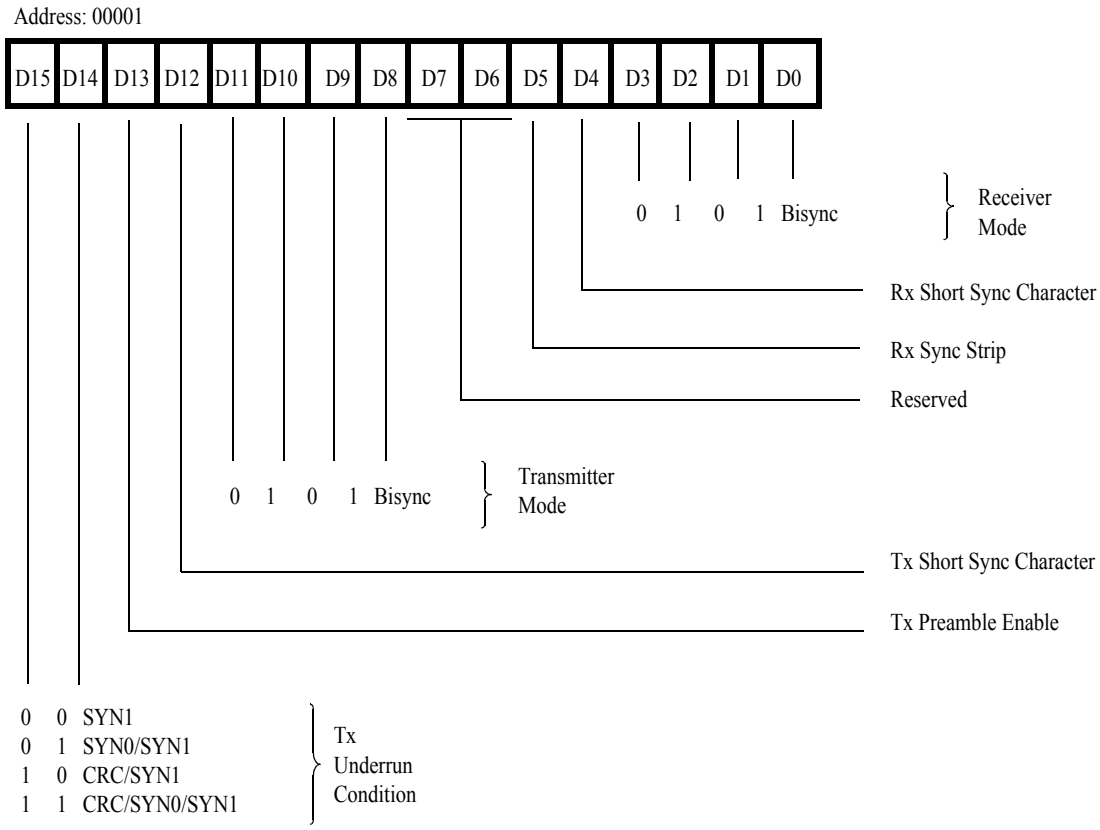


Figure 37. Channel Mode Register, Bisync Mode

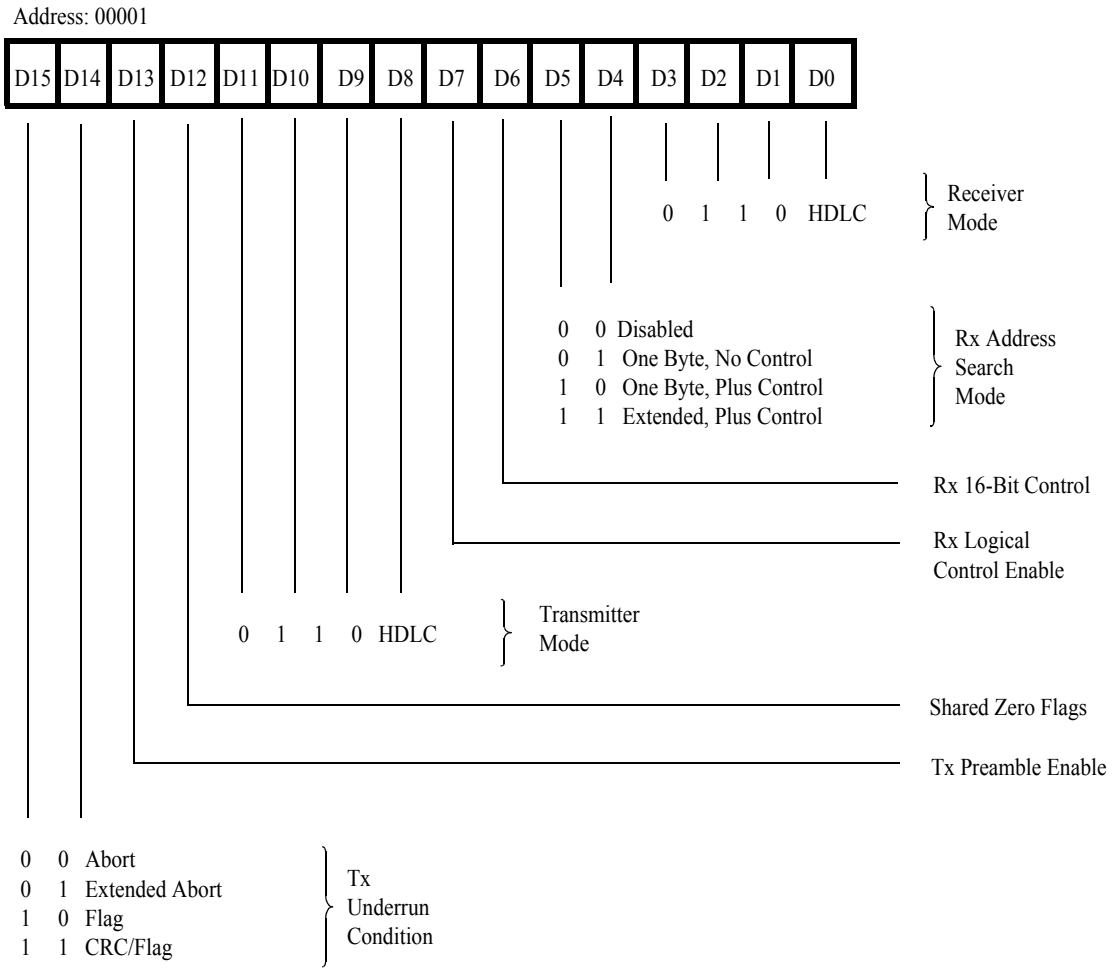


Figure 38. Channel Mode Register, HDLC Mode

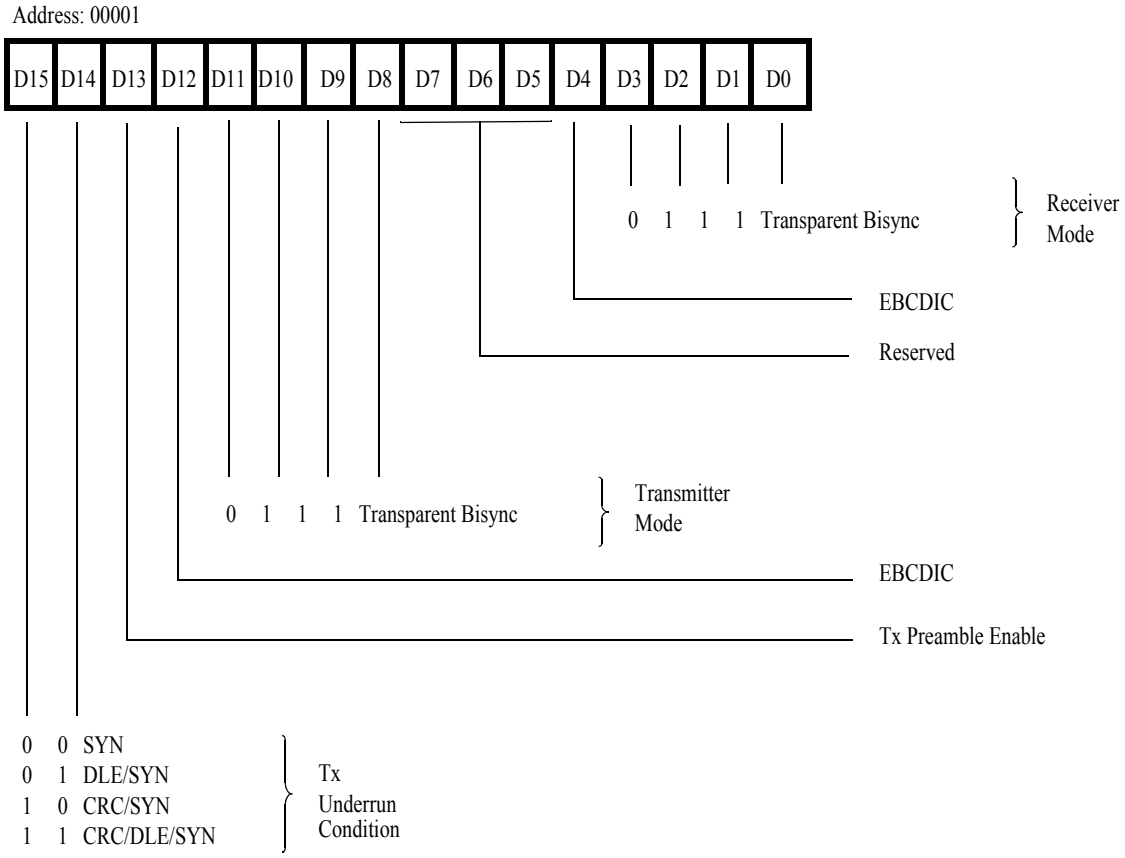


Figure 39. Channel Mode Register, Transparent Bisync Mode

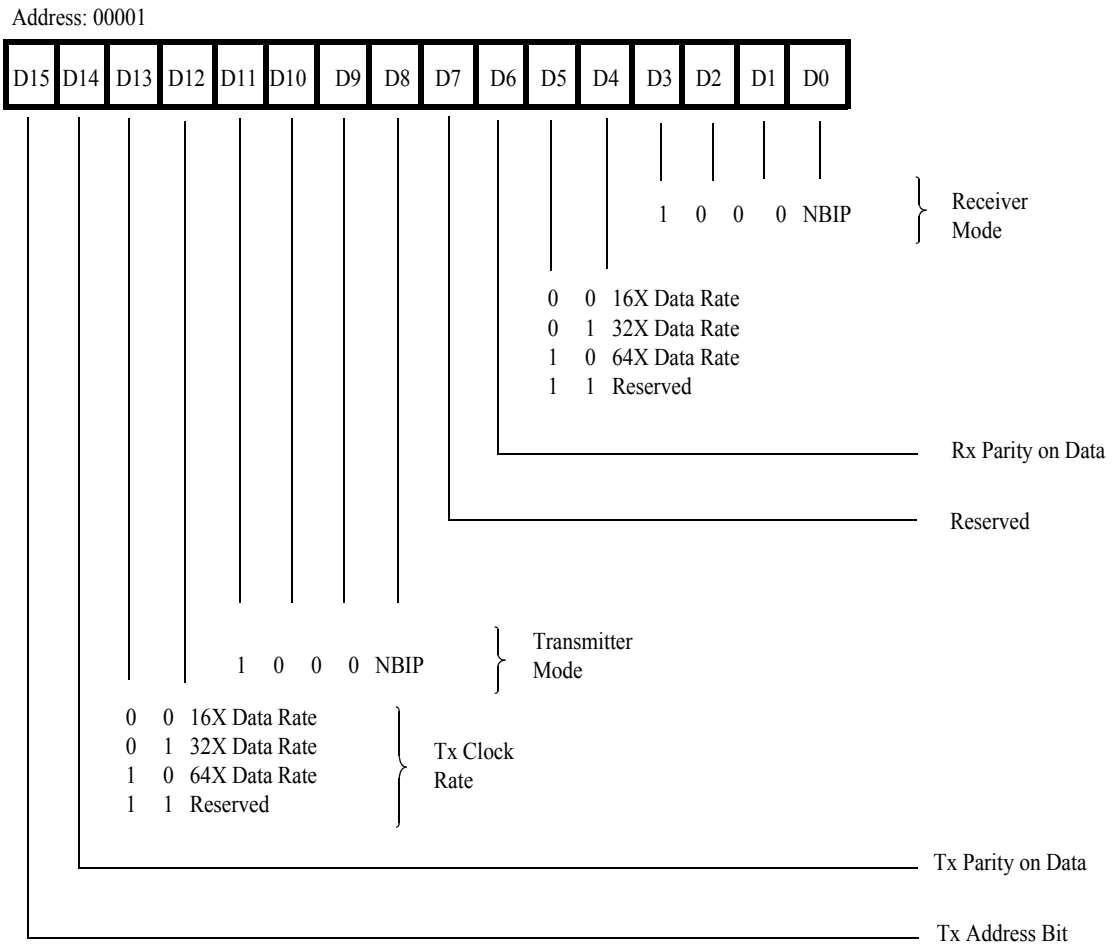


Figure 40. Channel Mode Register, NBIP Mode

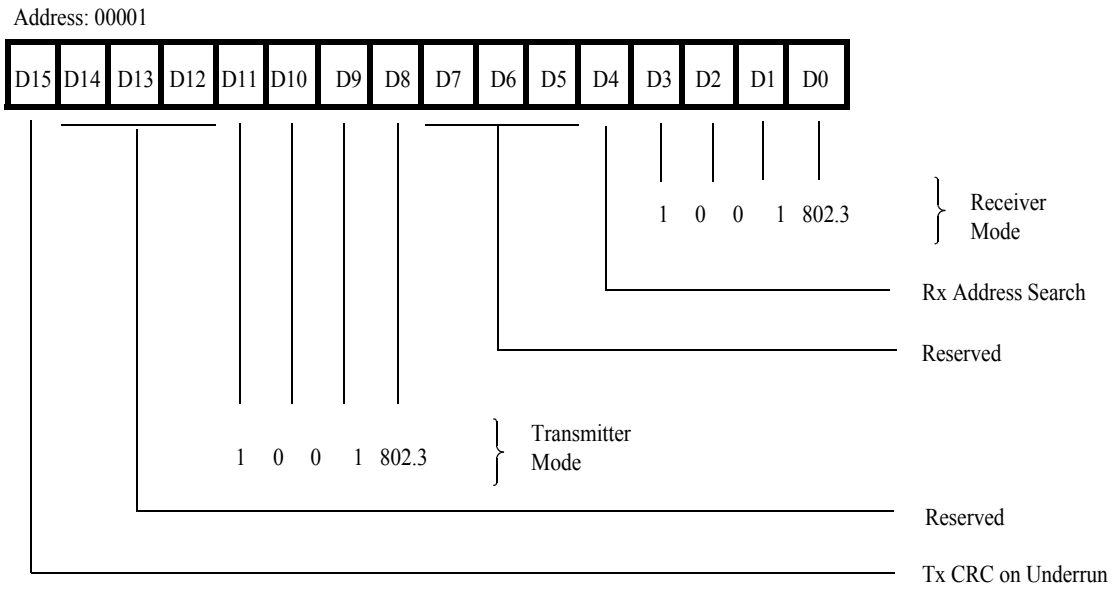


Figure 41. Channel Mode Register, 802.3 Mode

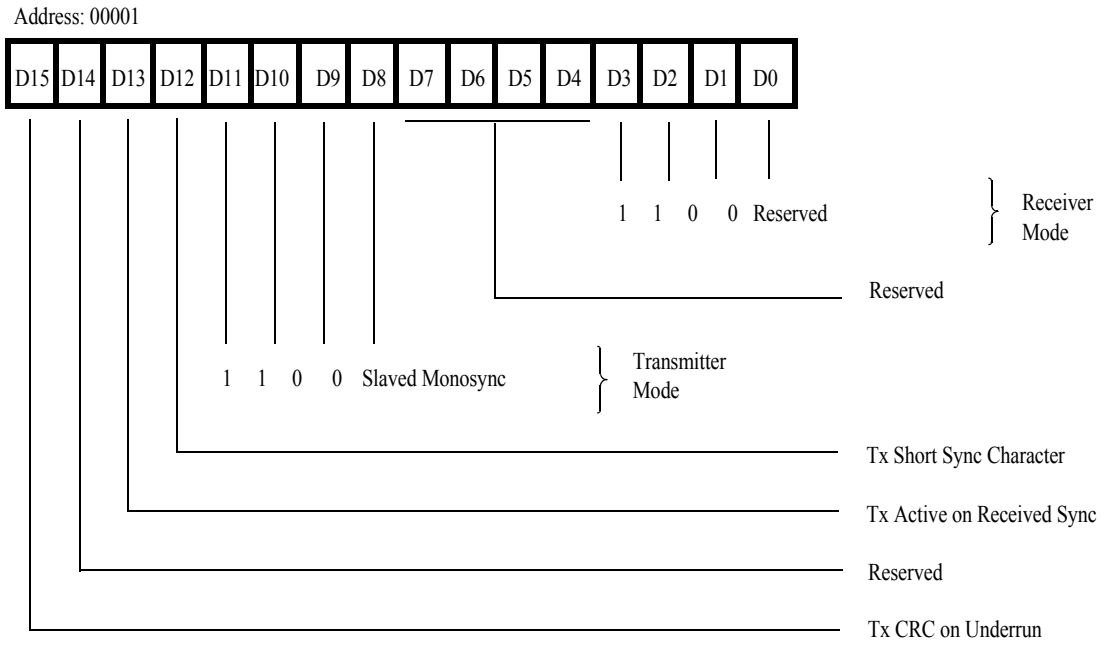


Figure 42. Channel Mode Register, Slaved Monosync Mode

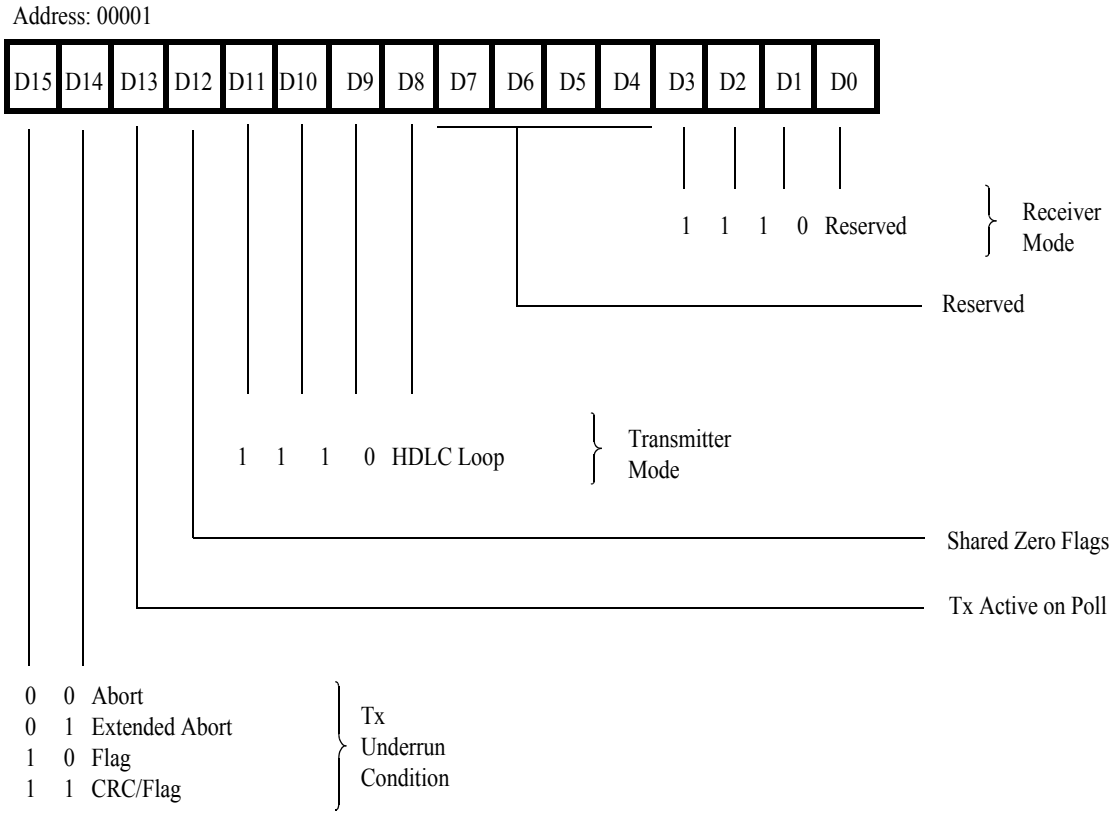


Figure 43. Channel Mode Register, HDLC Loop Mode

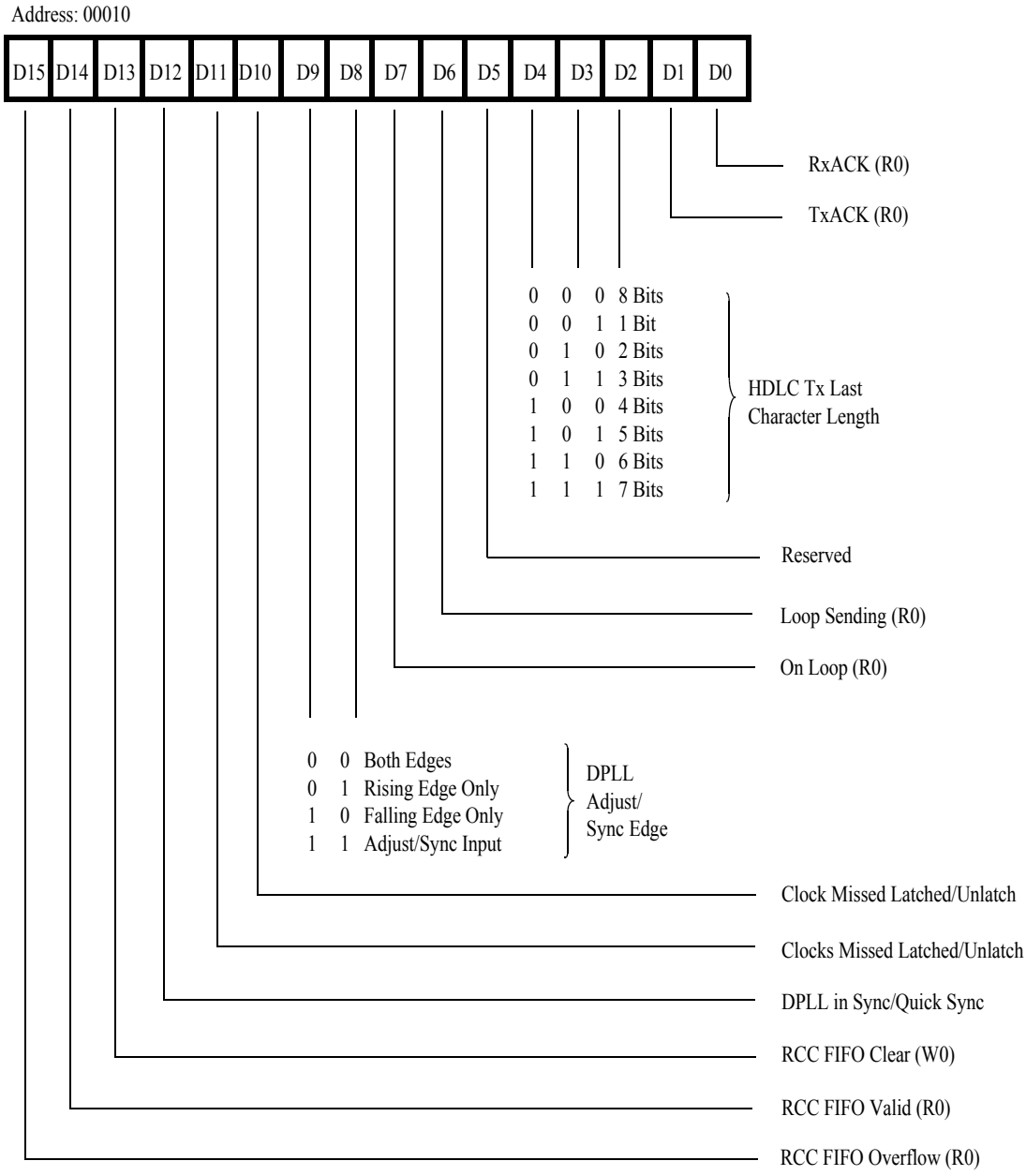


Figure 44. Channel Command/Status Register

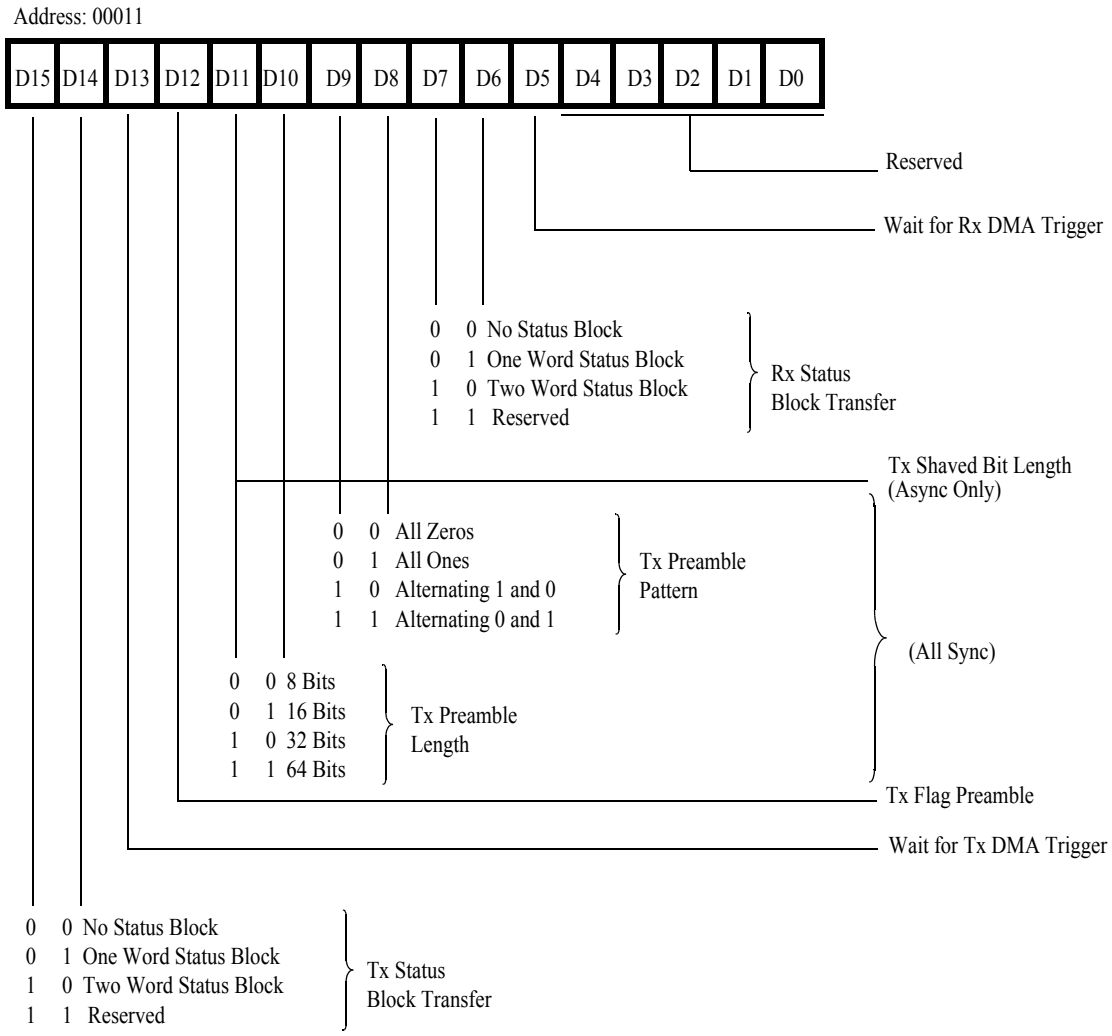


Figure 45. Channel Control Register

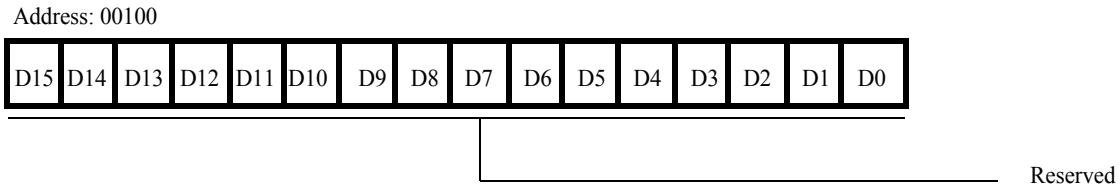


Figure 46. Primary Reserved Register

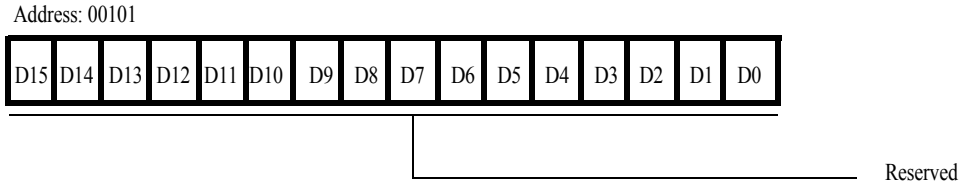


Figure 47. Secondary Reserved Register

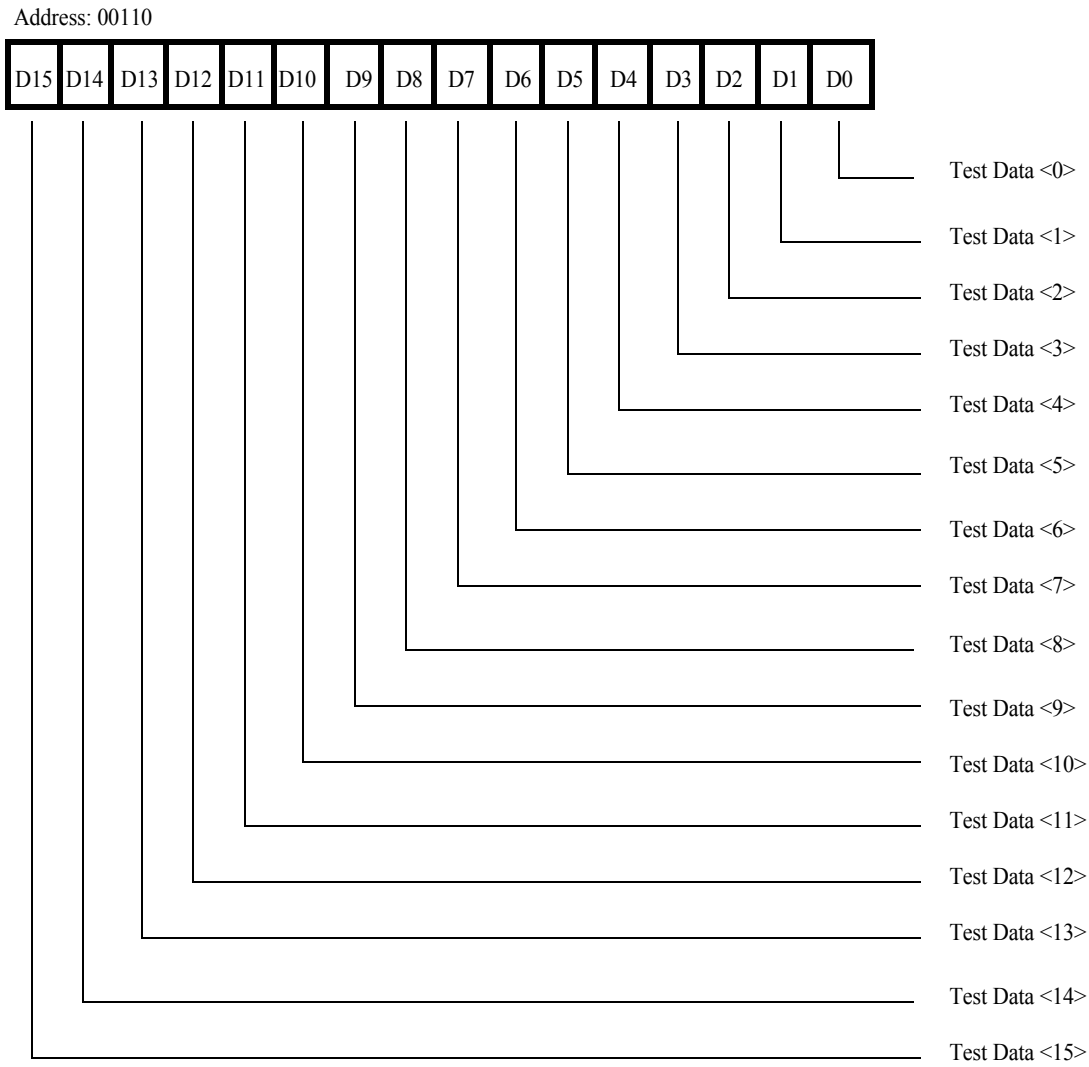


Figure 48. Test Mode Data Register

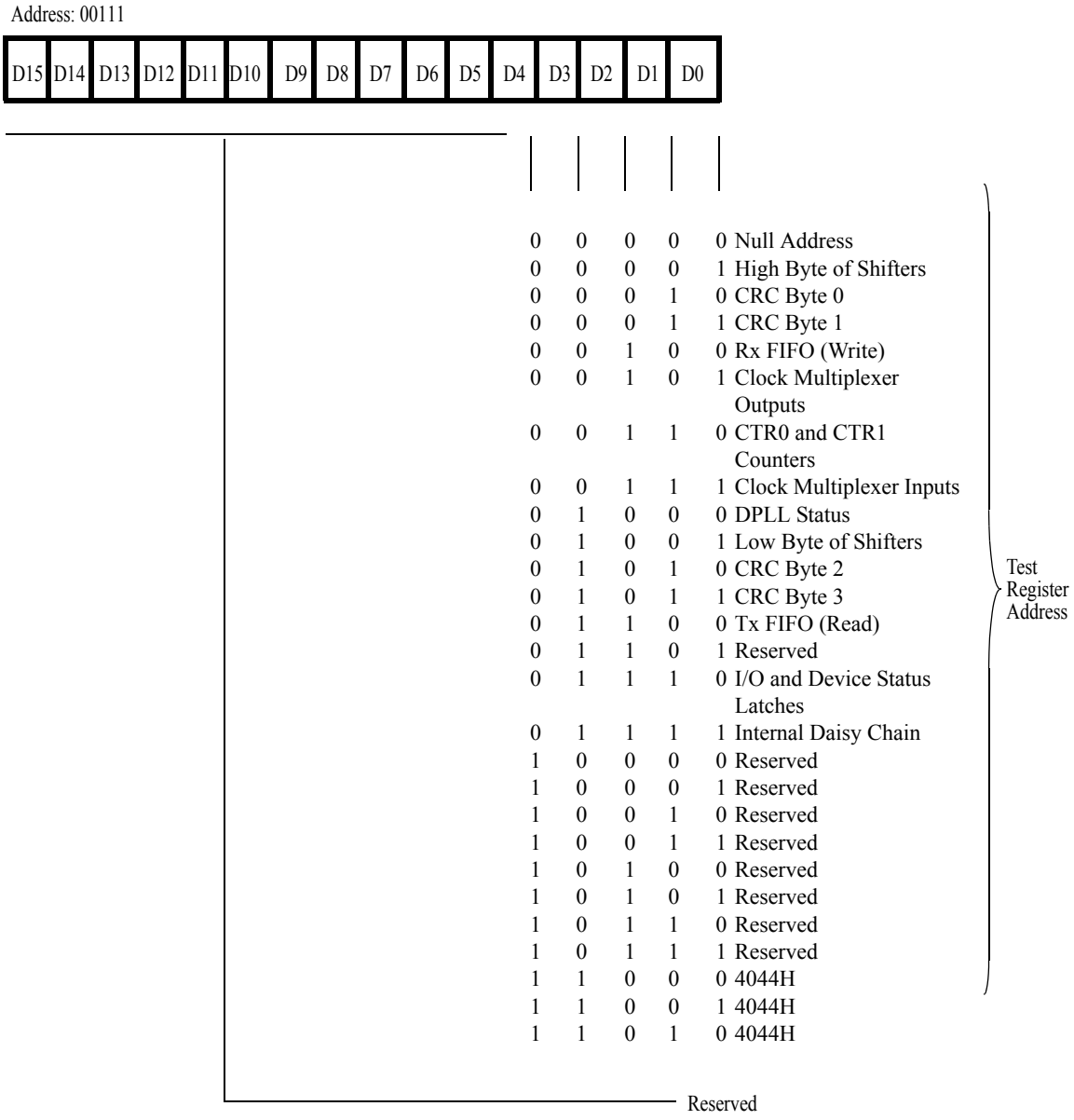


Figure 49. Test Mode Control Register

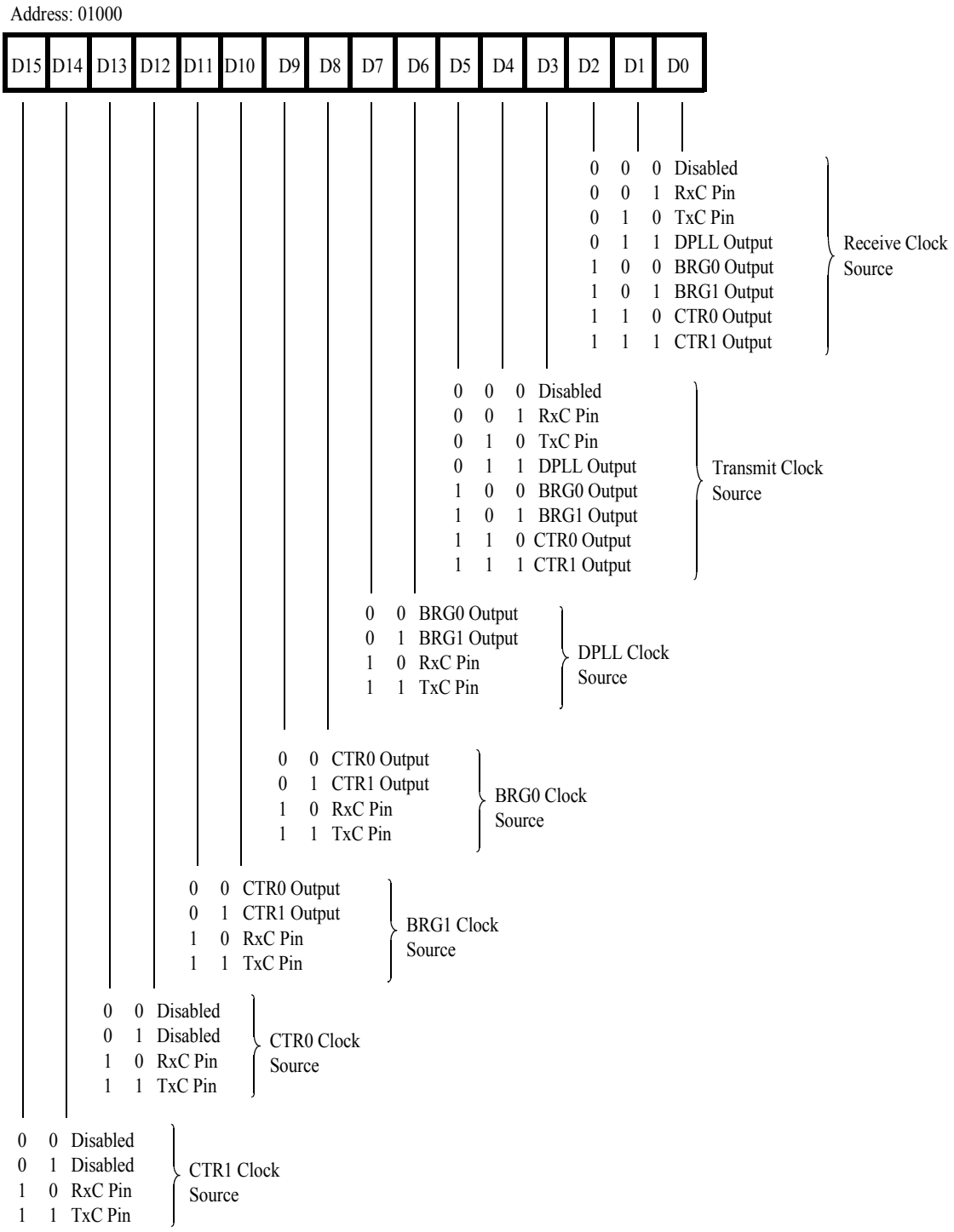


Figure 50. Clock Mode Control Register

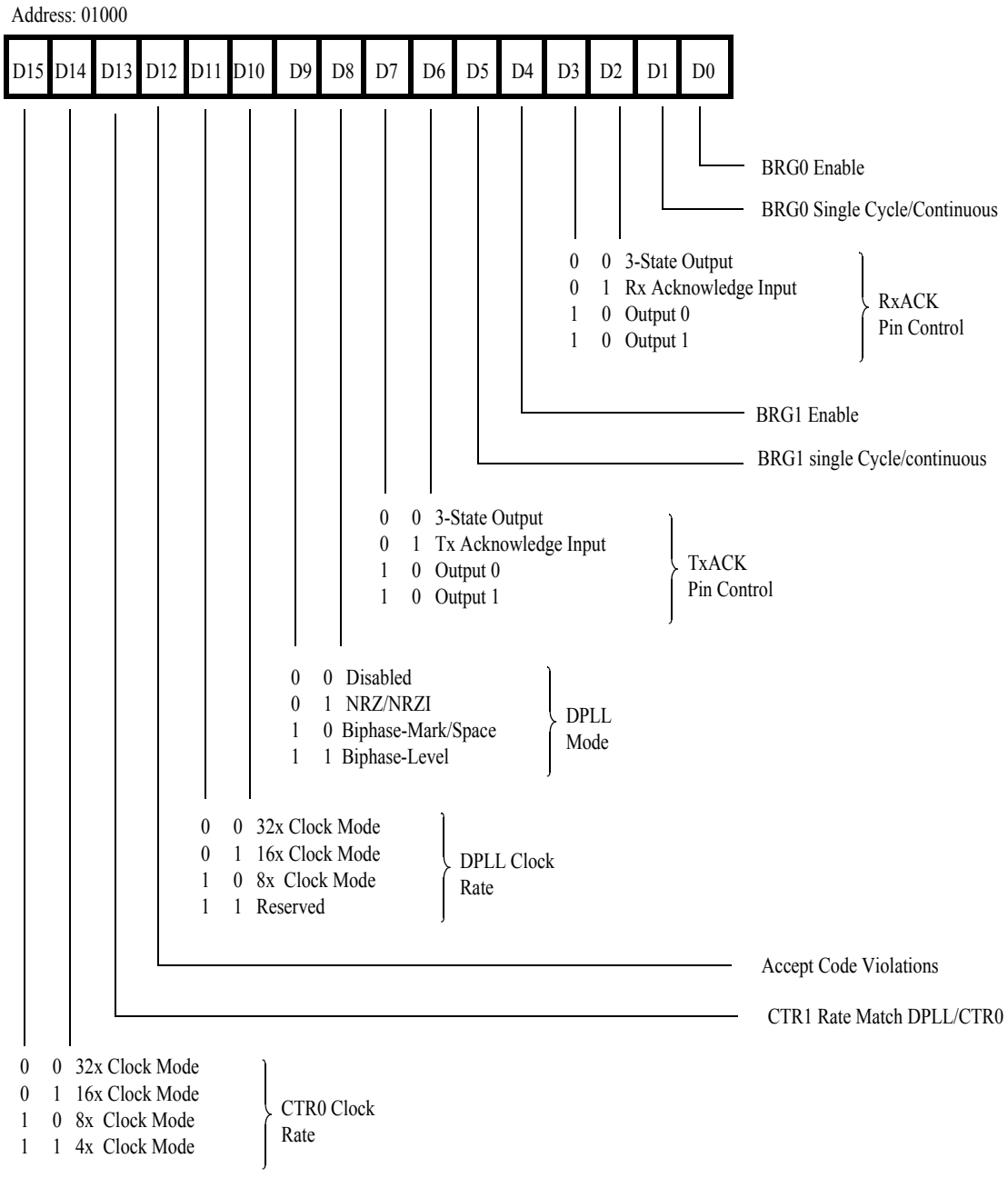


Figure 51. Hardware Configuration Register

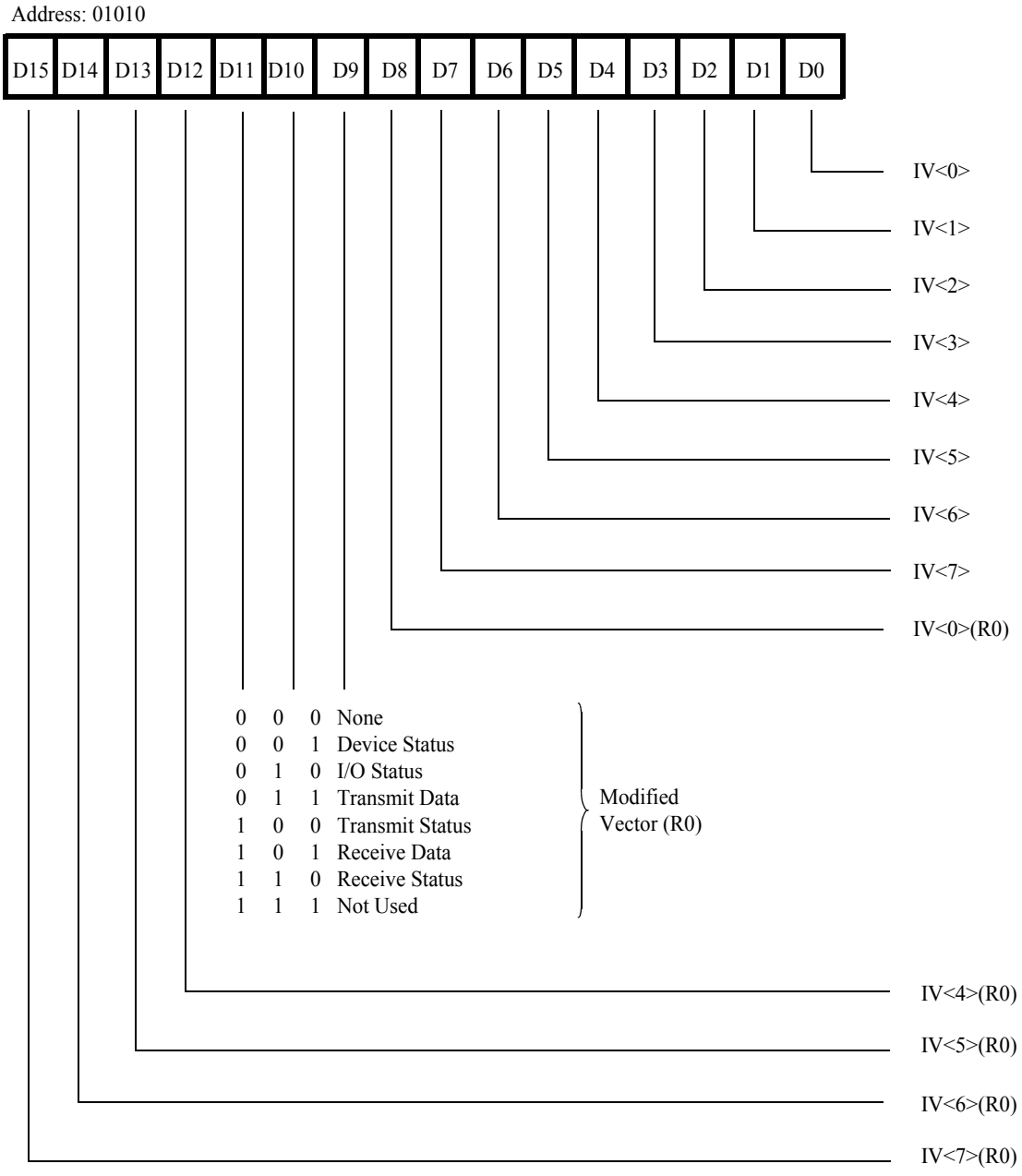


Figure 52. Interrupt Vector Register

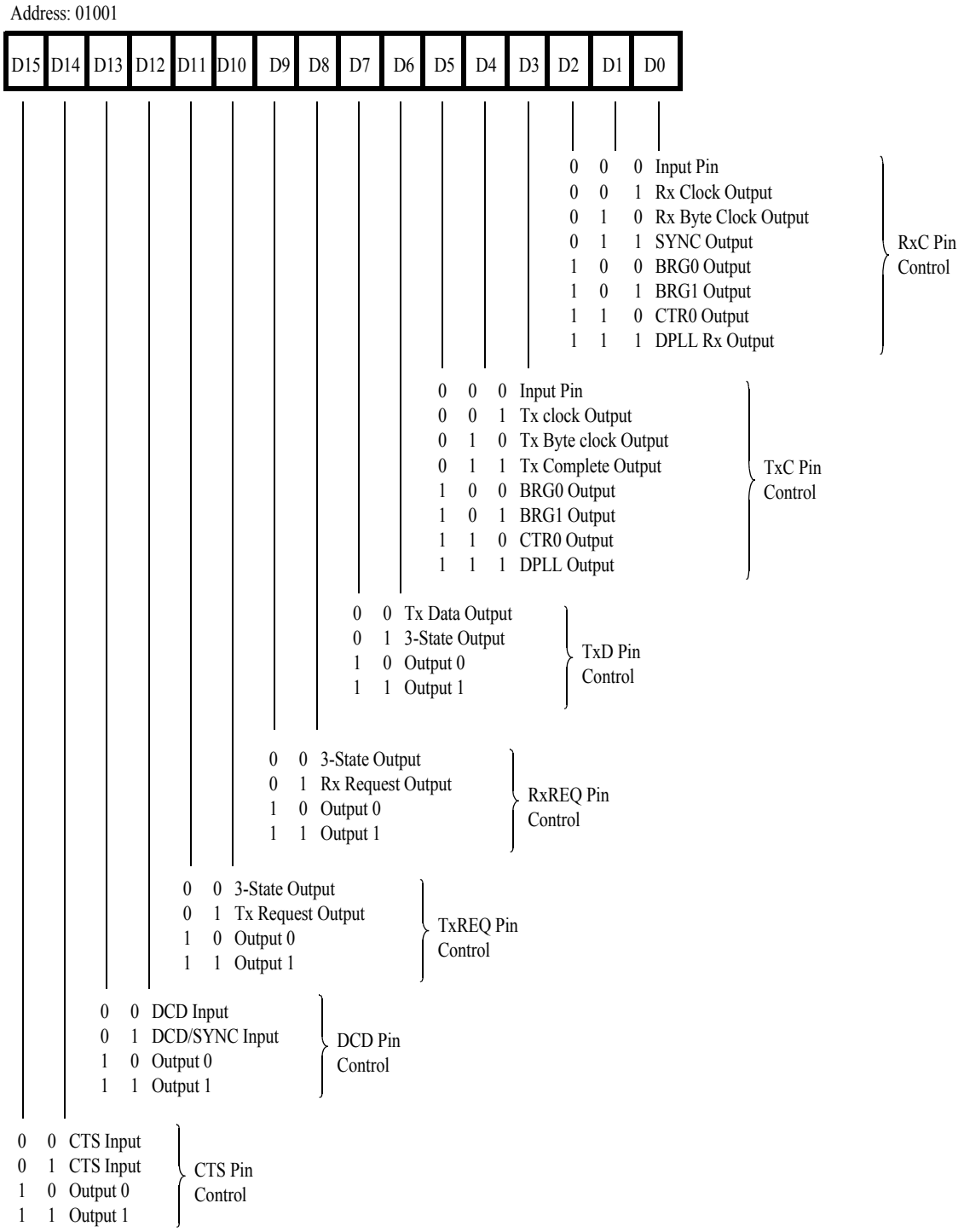


Figure 53. I/O Control Register

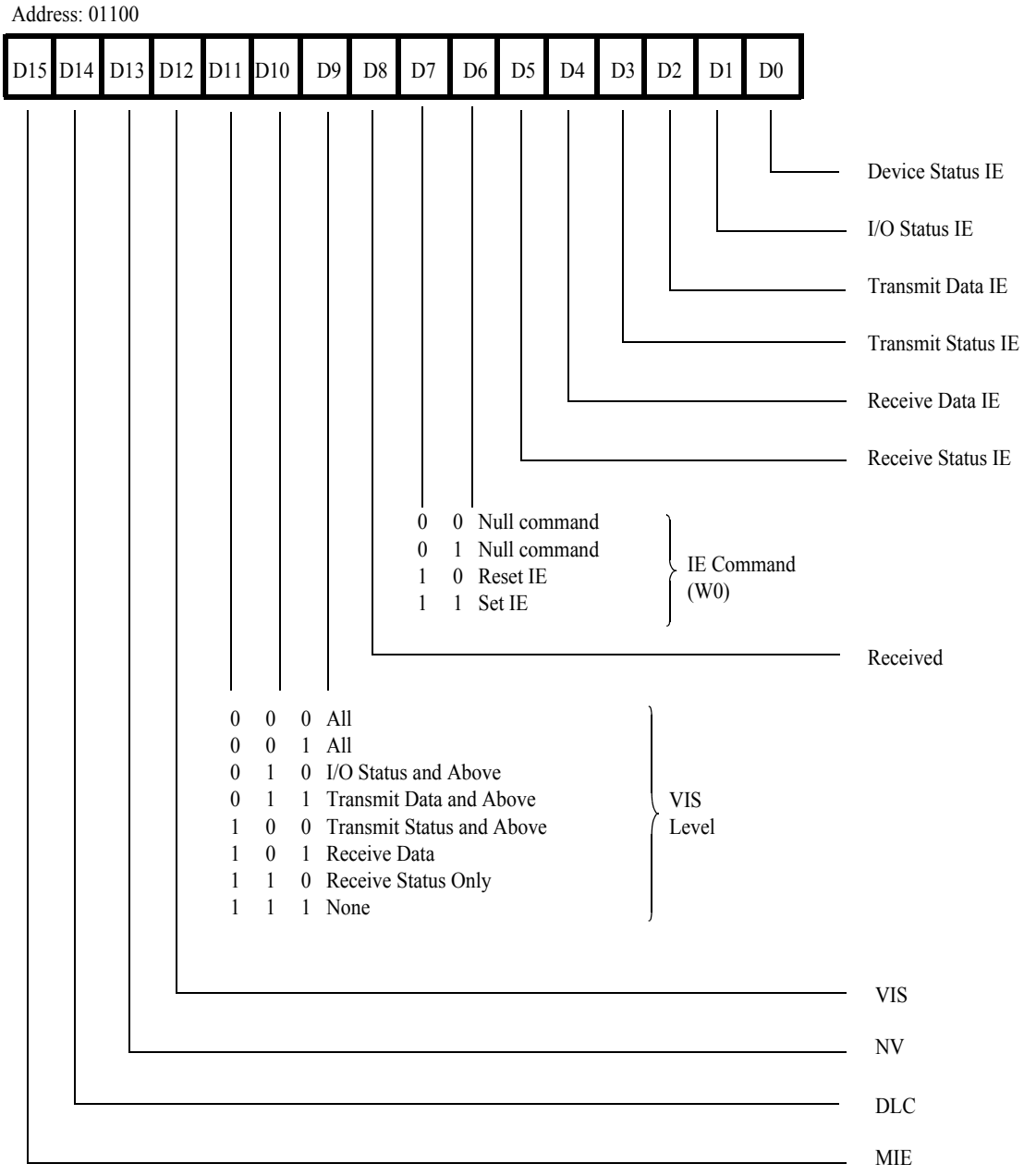


Figure 54. Interrupt Control Register

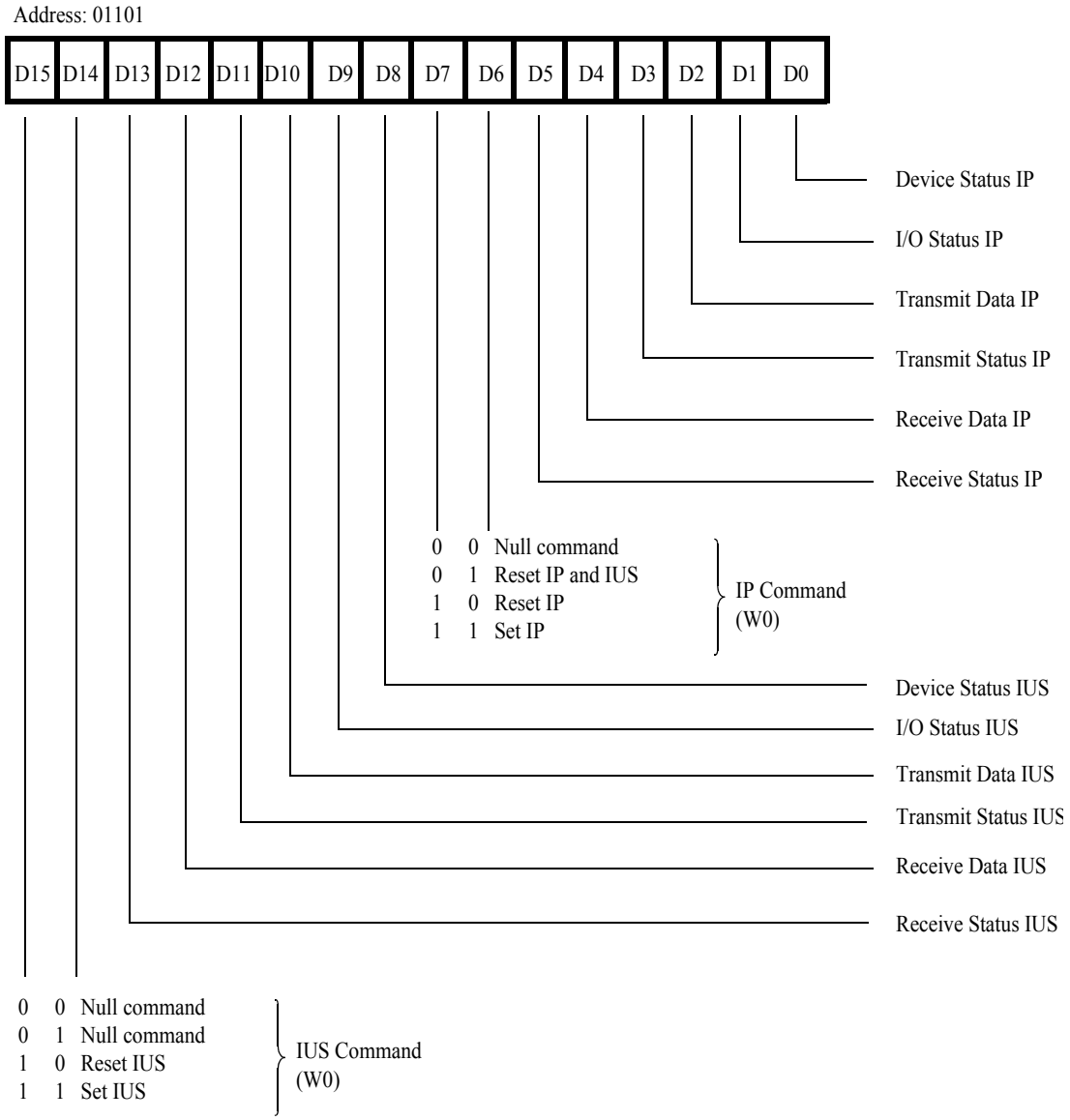


Figure 55. Daisy-Chain Control Register

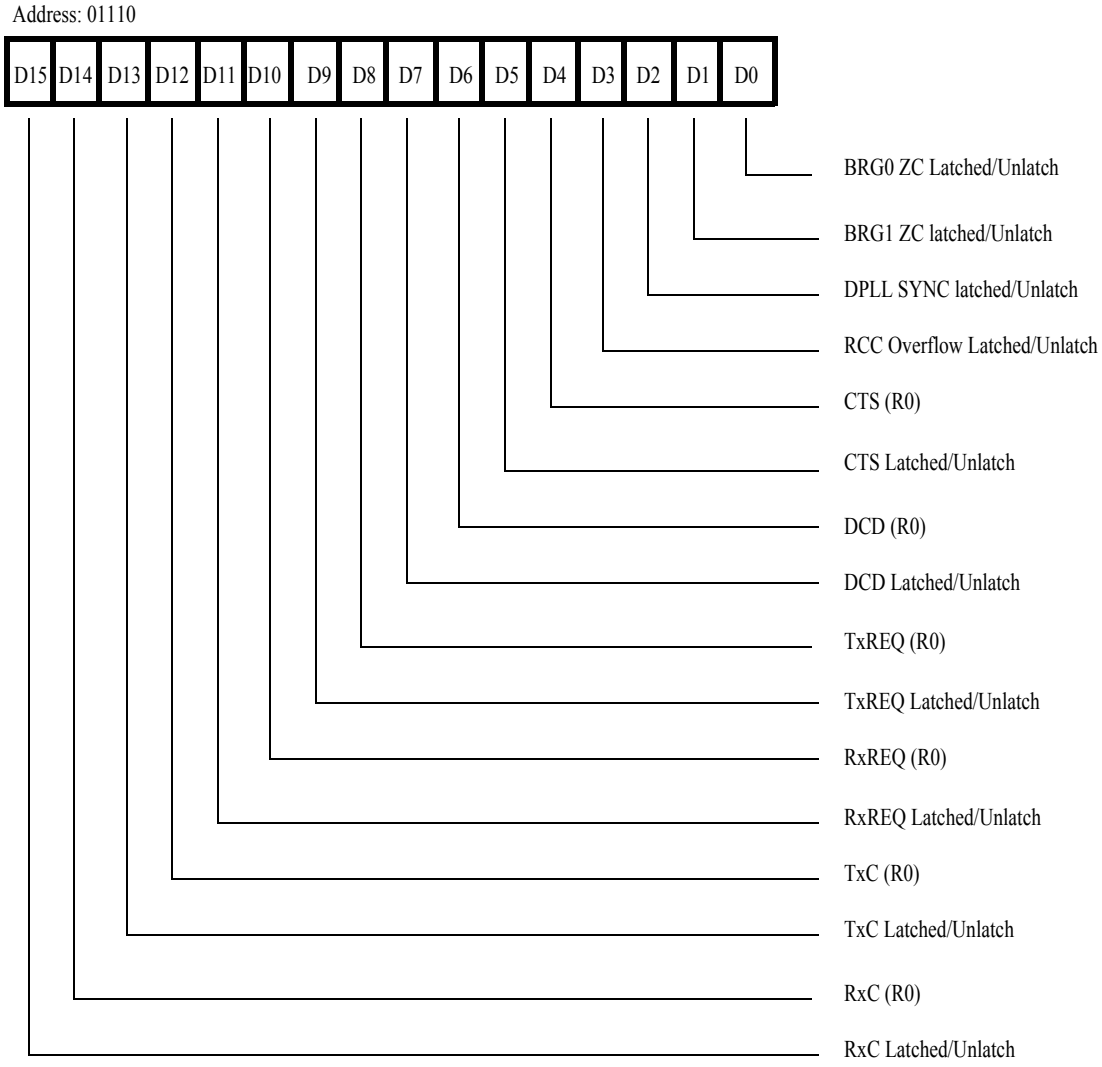


Figure 56. Miscellaneous Interrupt Status Register

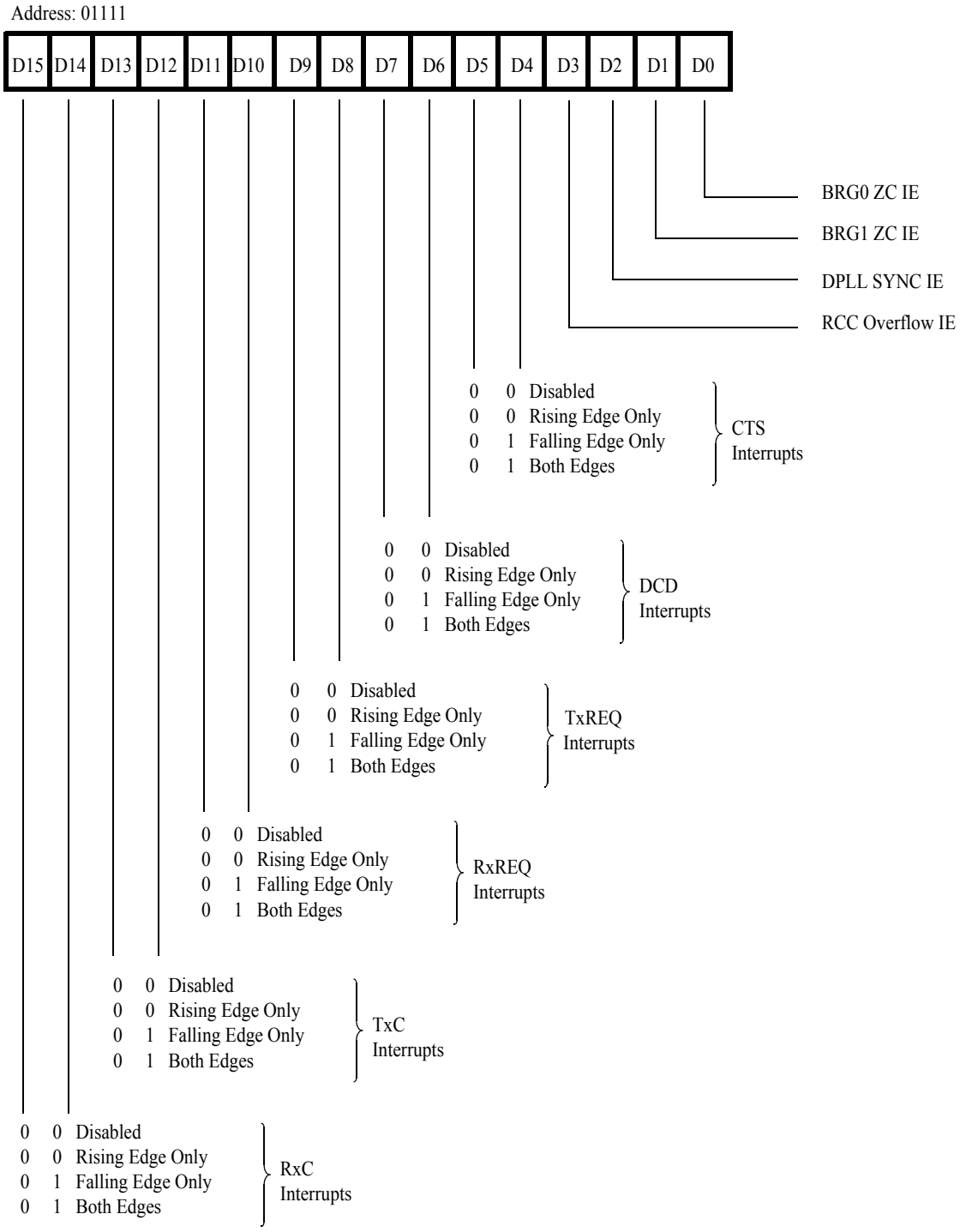


Figure 57. Status Interrupt Control Register

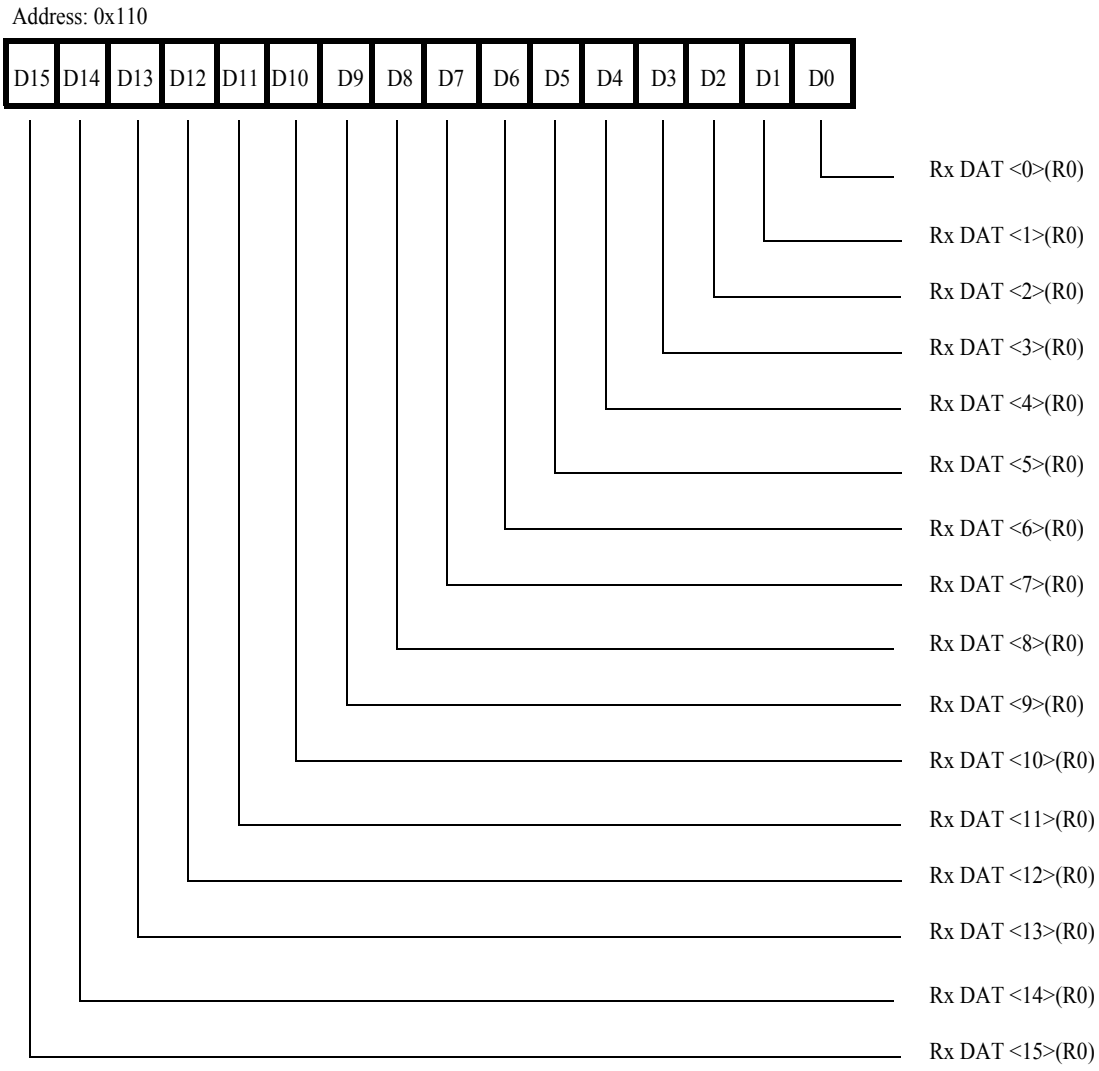


Figure 58. Receive Data Register

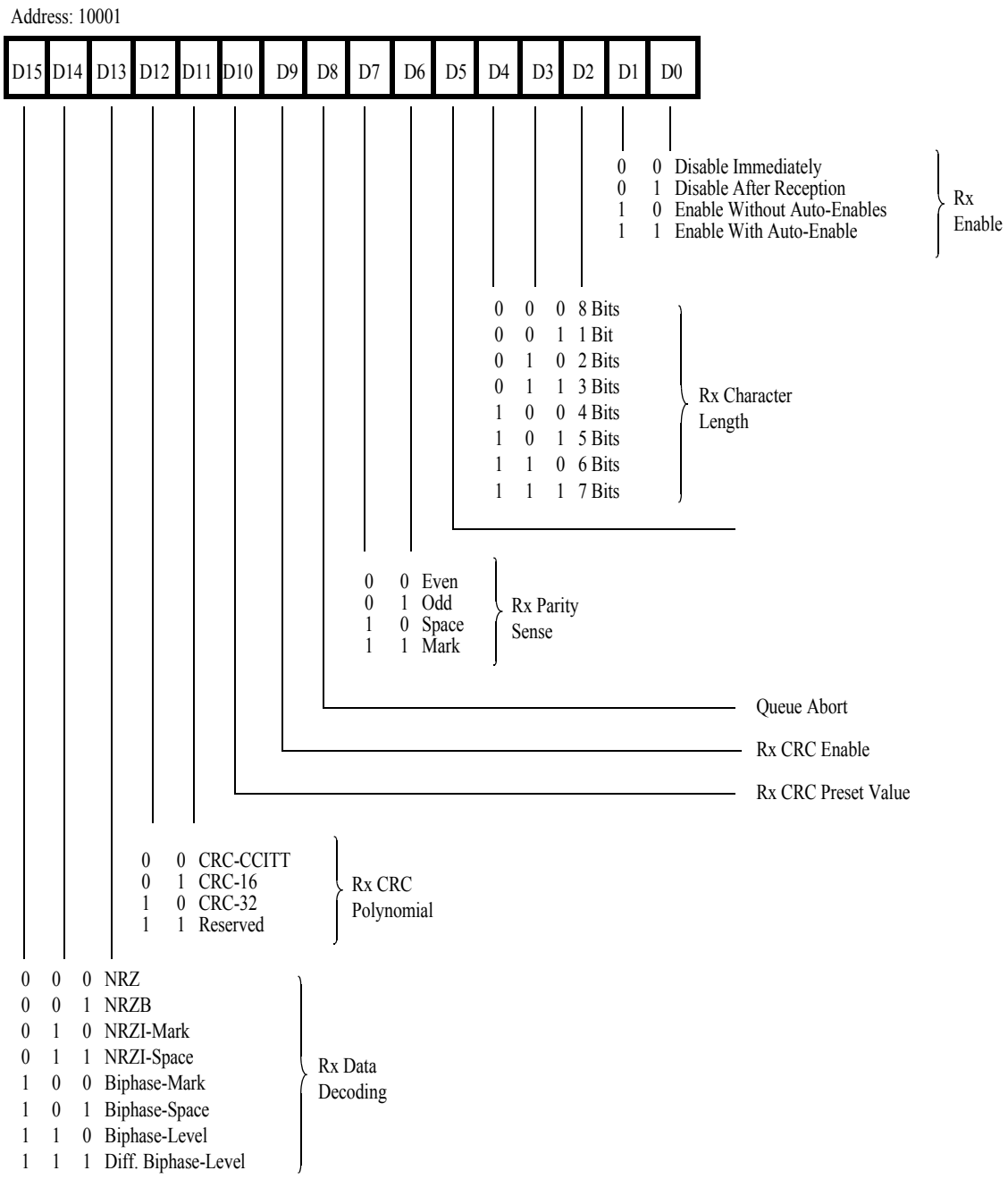


Figure 59. Receive Mode Register

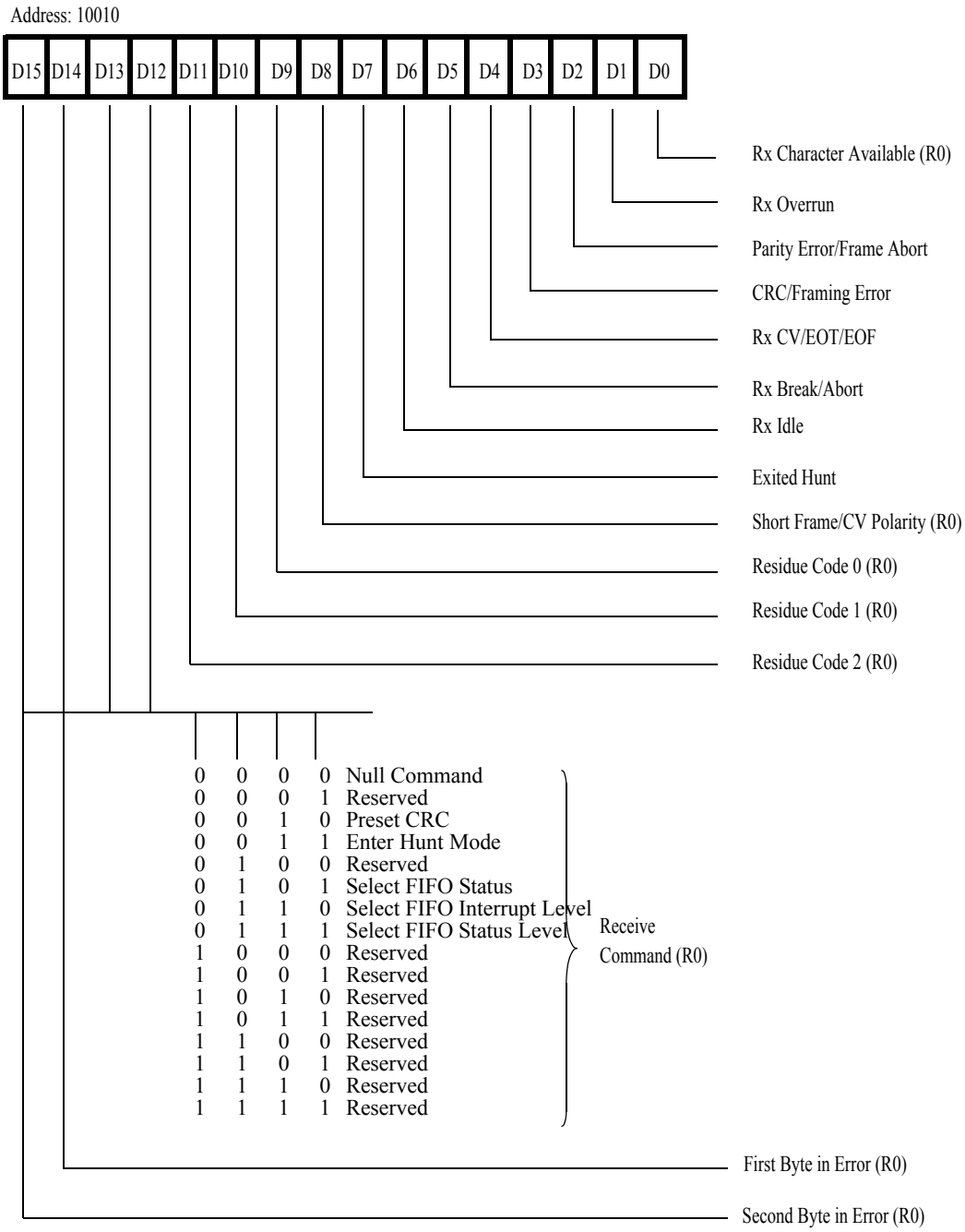


Figure 60. Receive Command Status Register

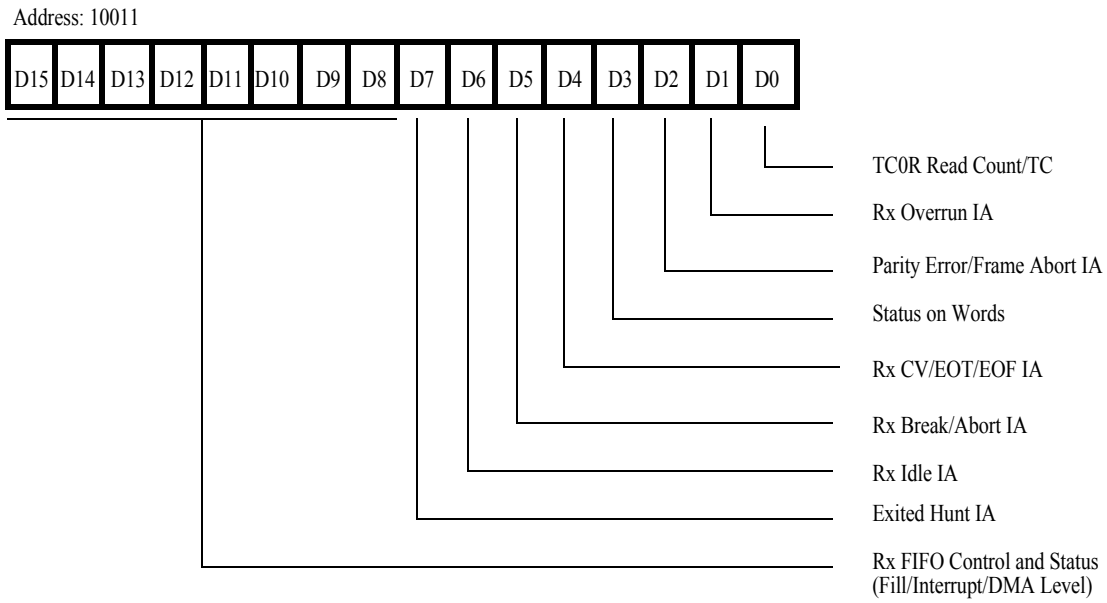


Figure 61. Receive Interrupt Control Register

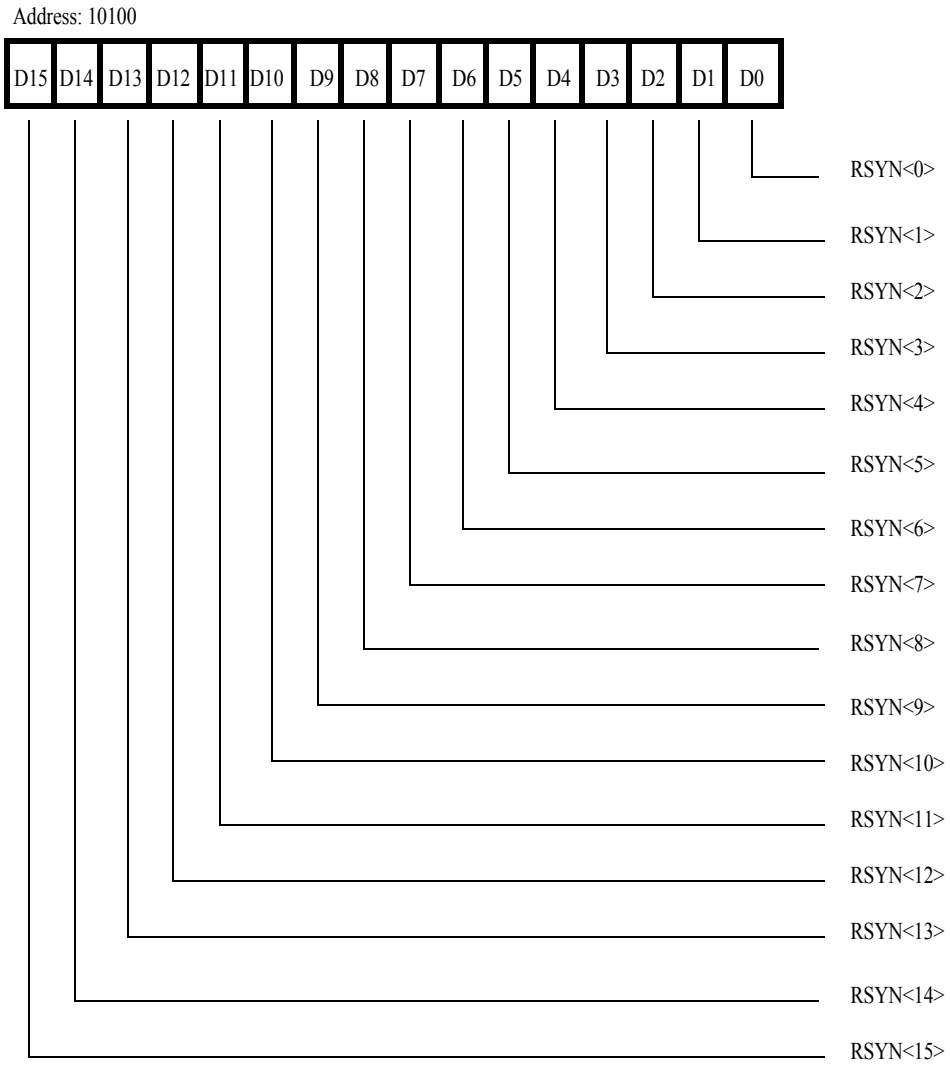


Figure 62. Receive Sync Register

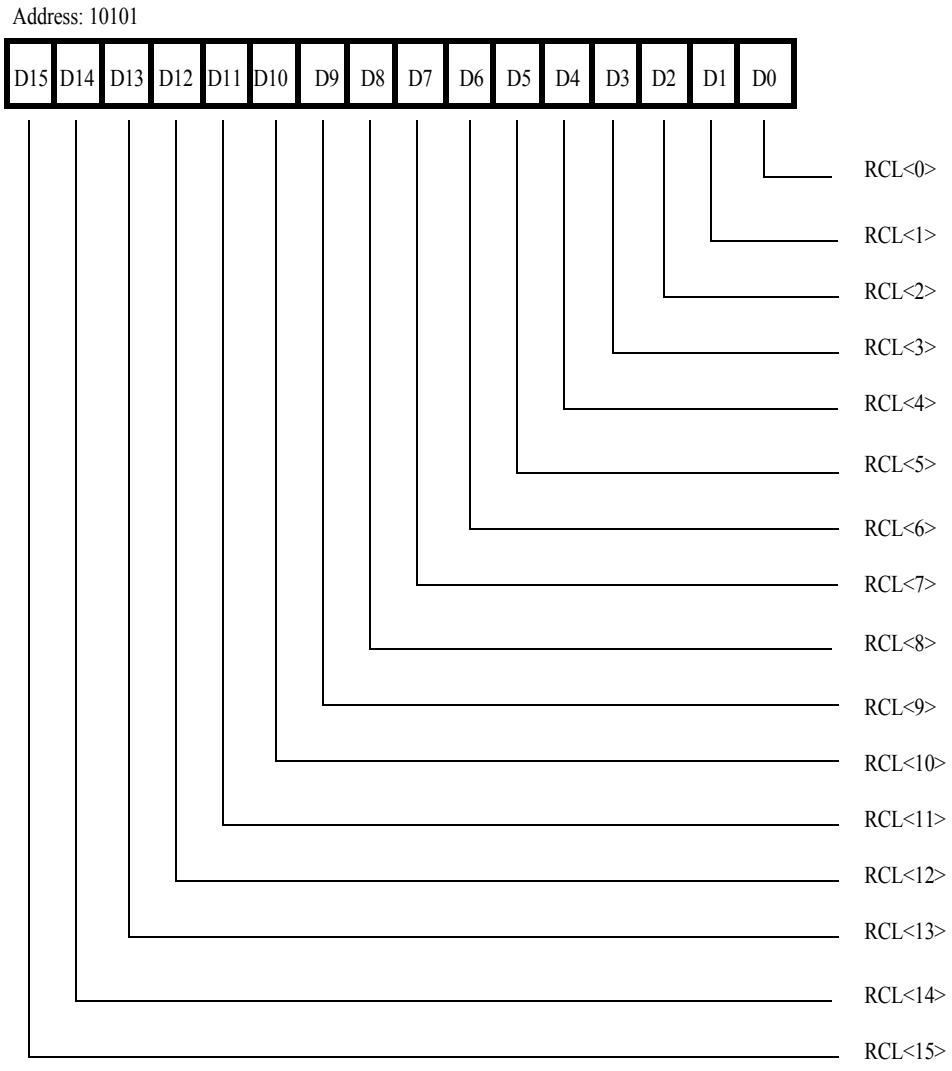


Figure 63. Receive Count Limit Register

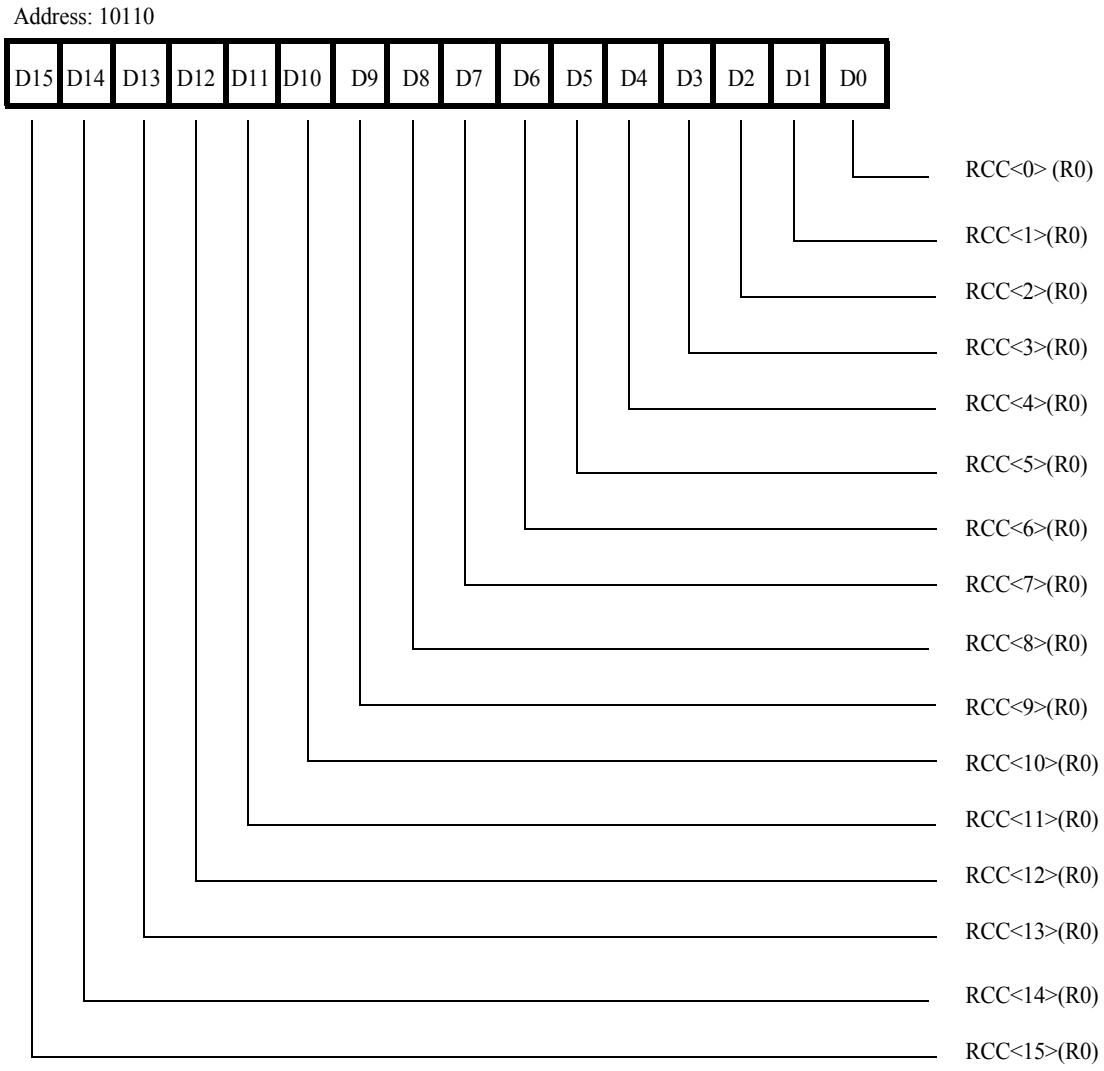


Figure 64. Receive Character Count Register

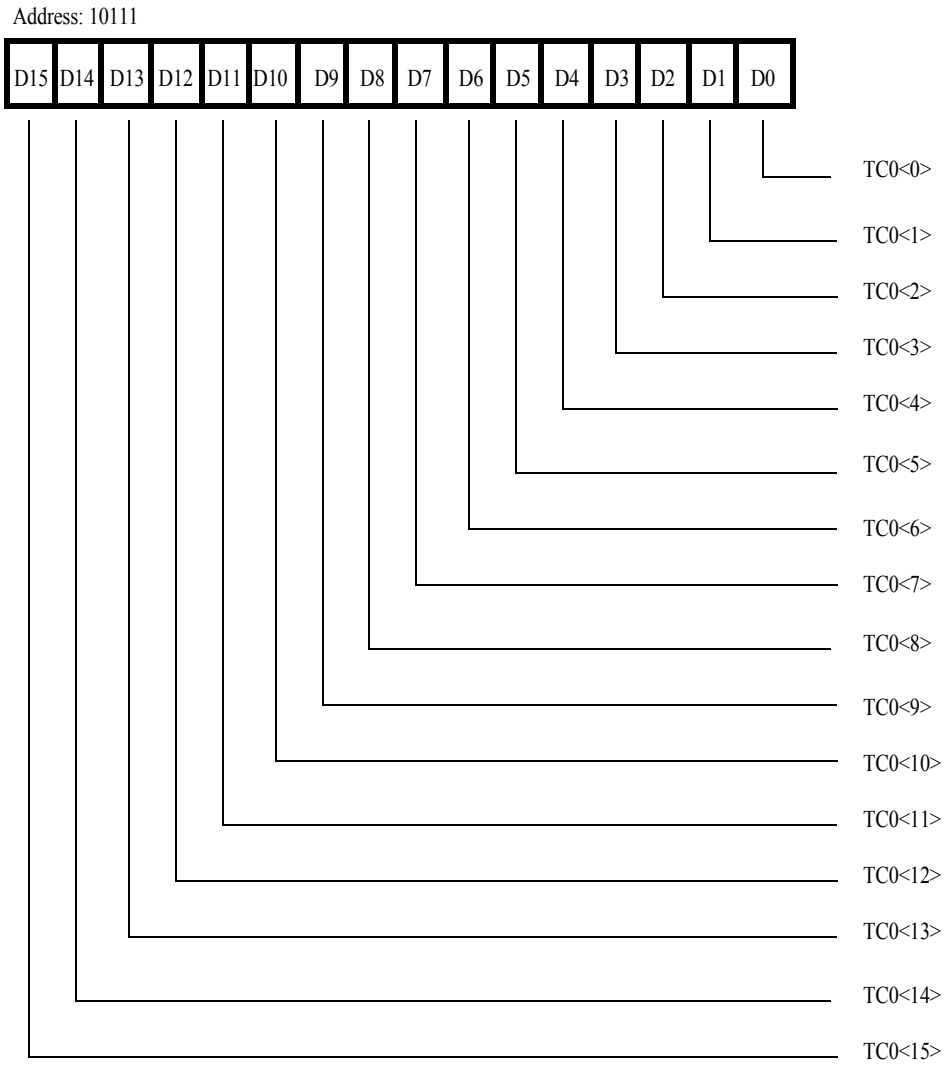


Figure 65. Time Constant 0 Register

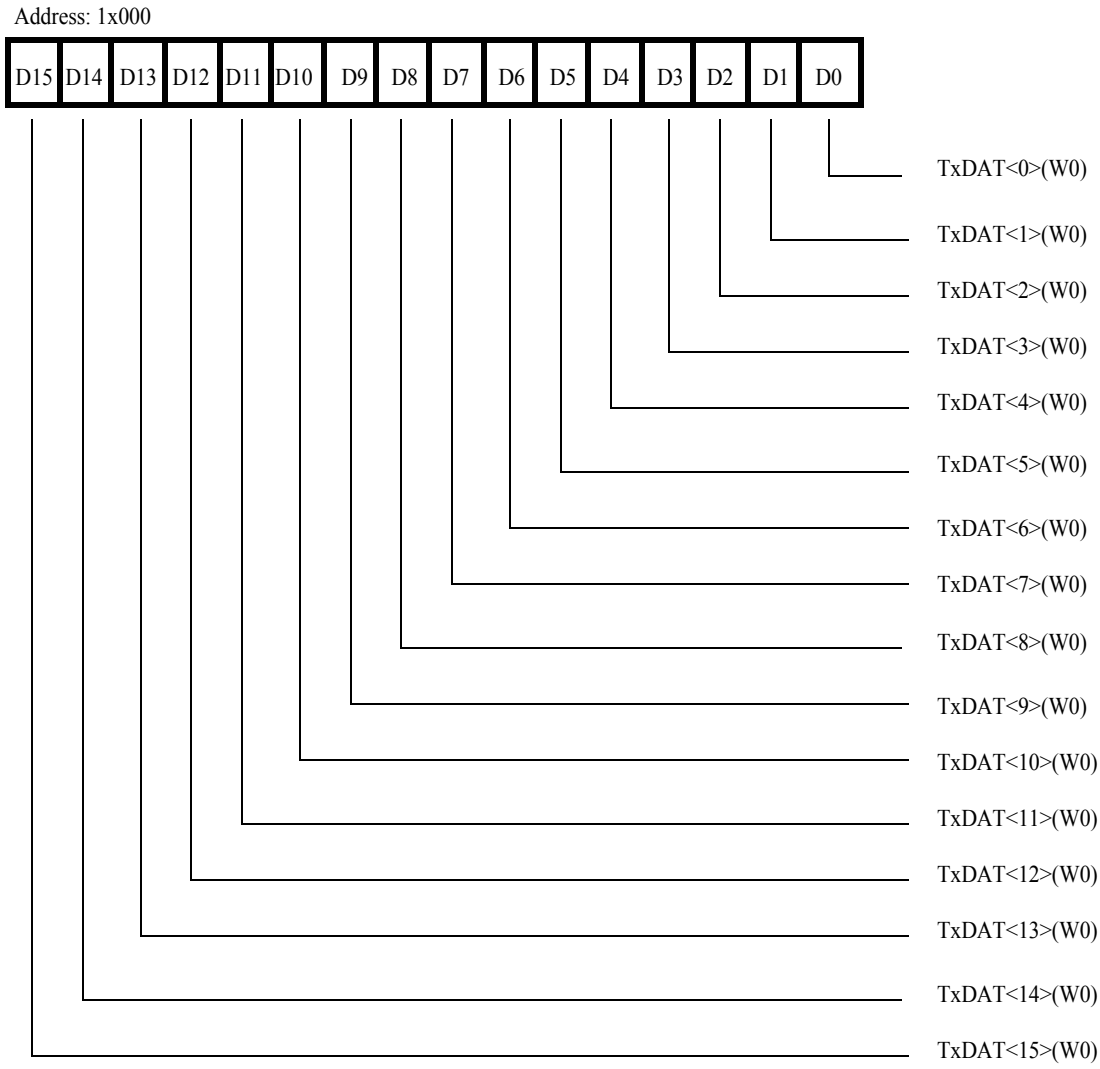


Figure 66. Transmit Data Register

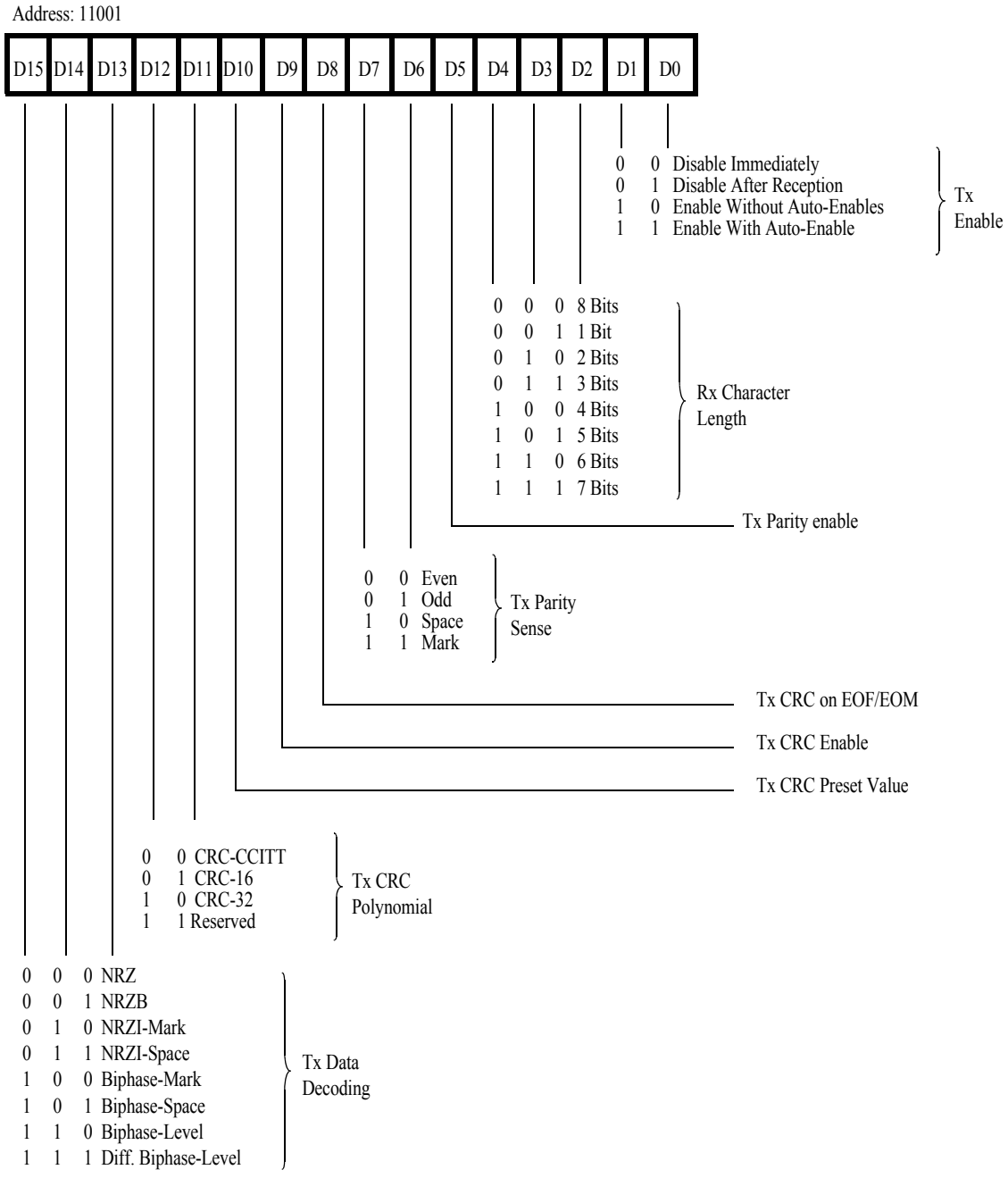


Figure 67. Transmit Mode Register

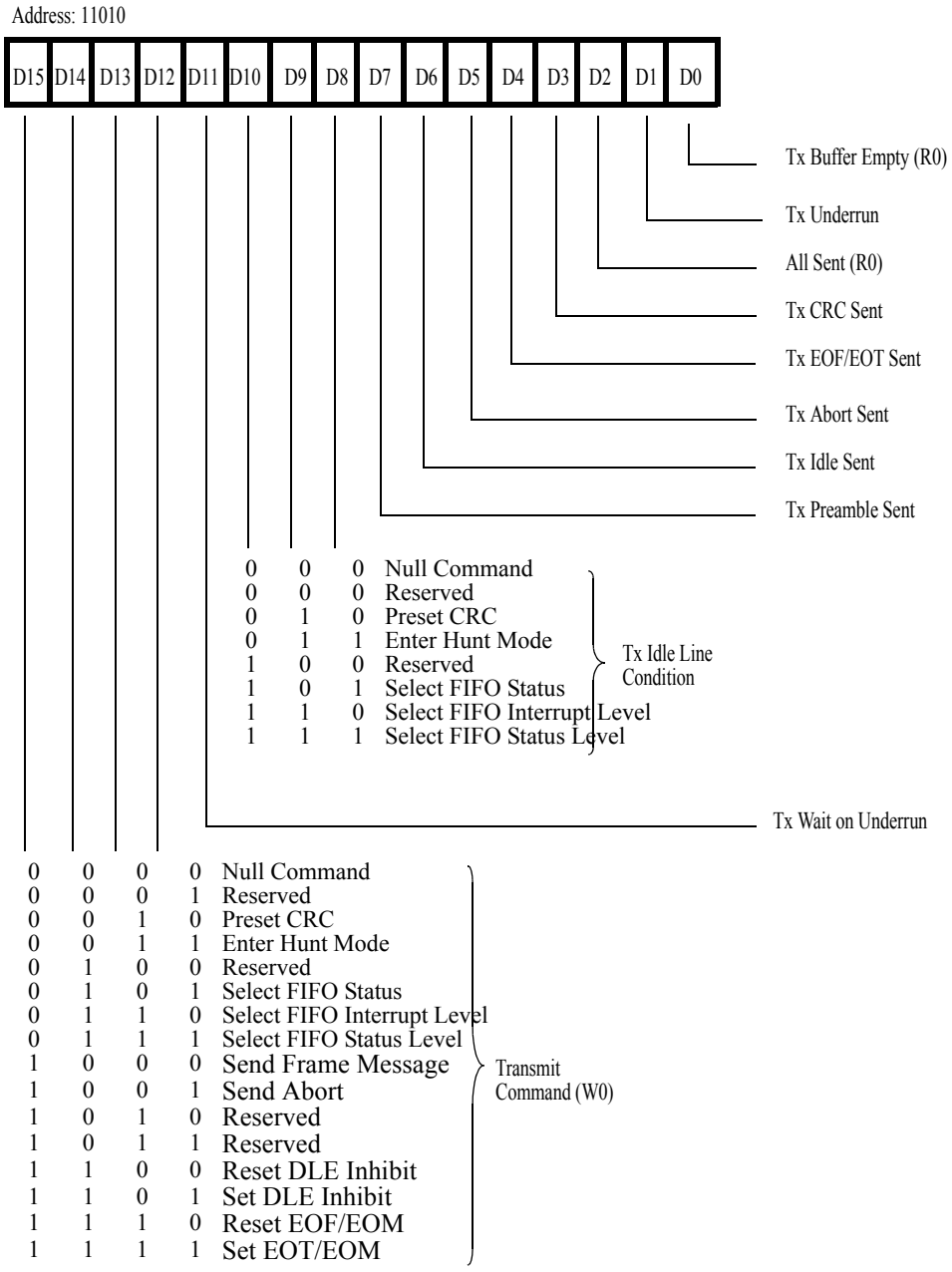


Figure 68. Transmit Command/Status Register

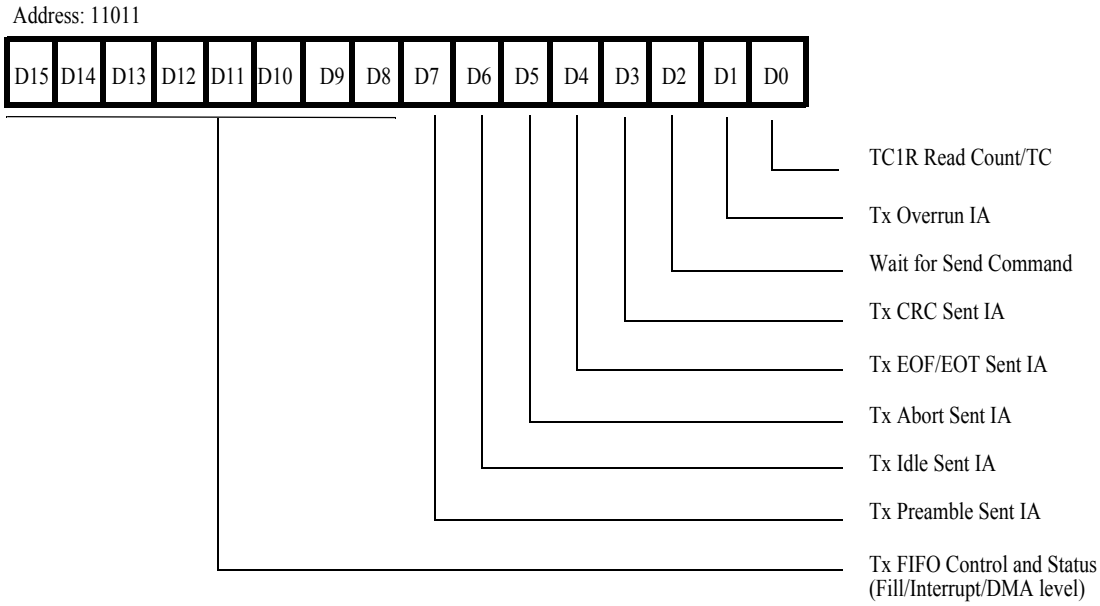


Figure 69. Transmit Interrupt Control Register

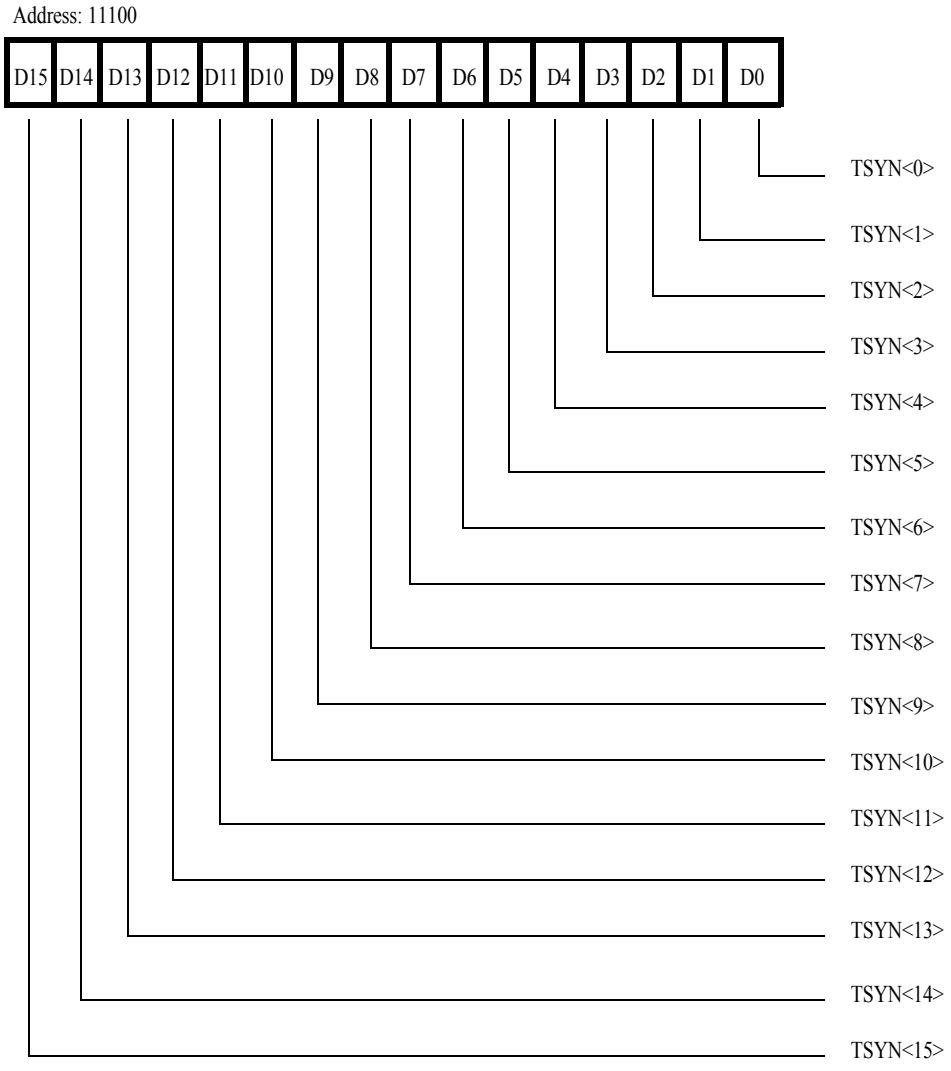


Figure 70. Transmit Sync Register

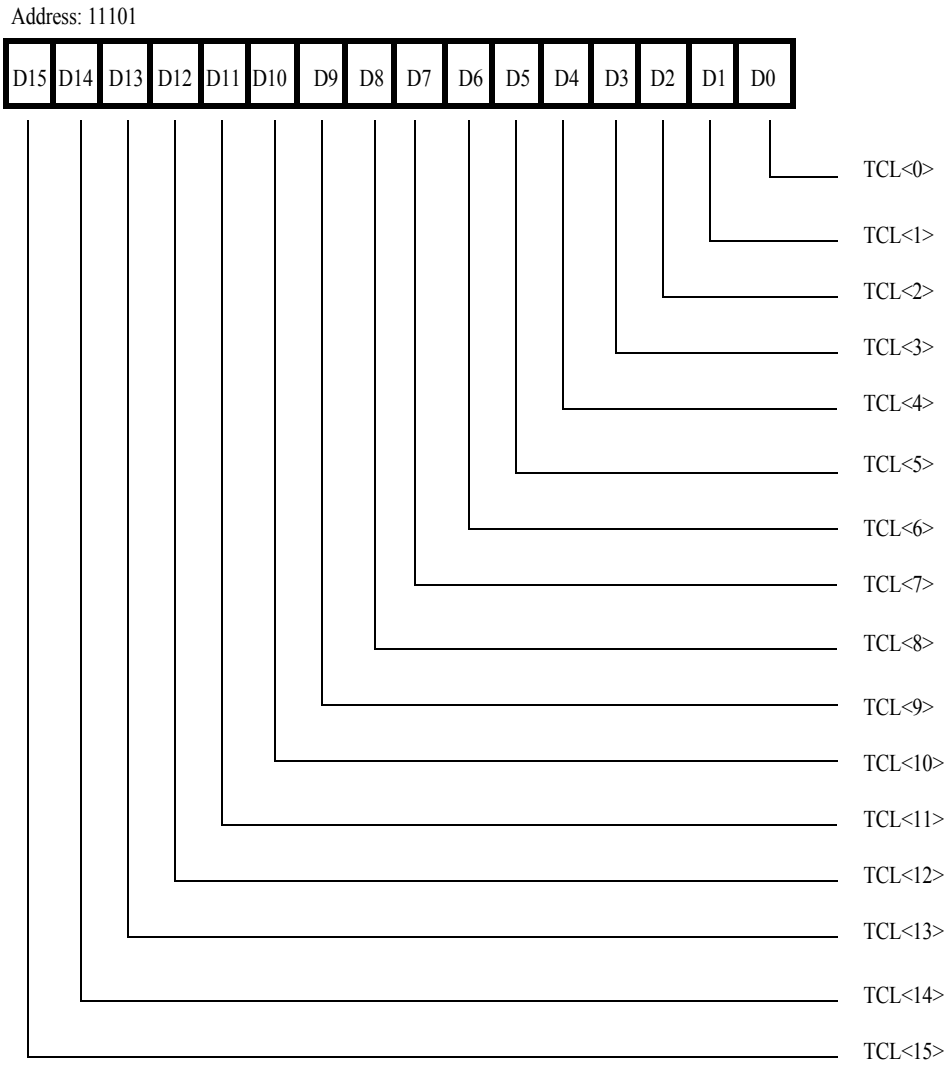


Figure 71. Transmit Count Limit Register

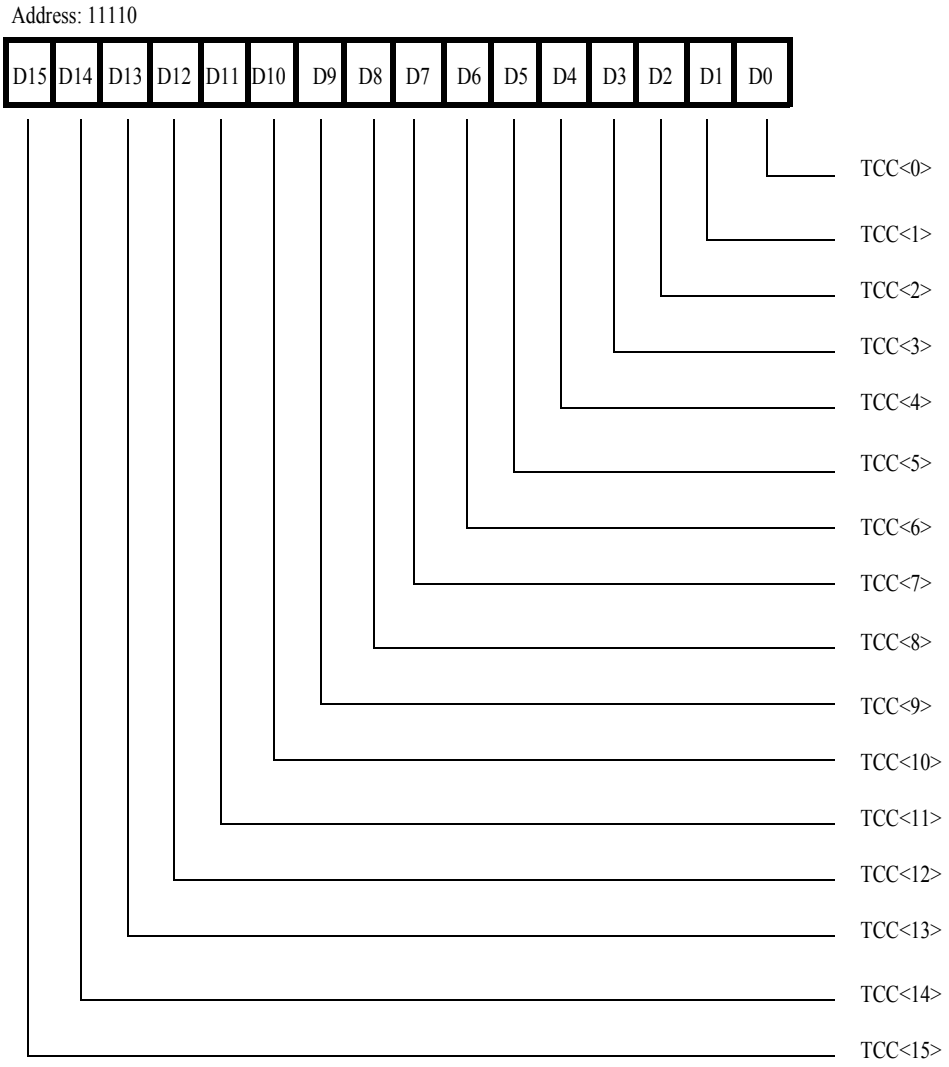


Figure 72. Transmit Character Count Register

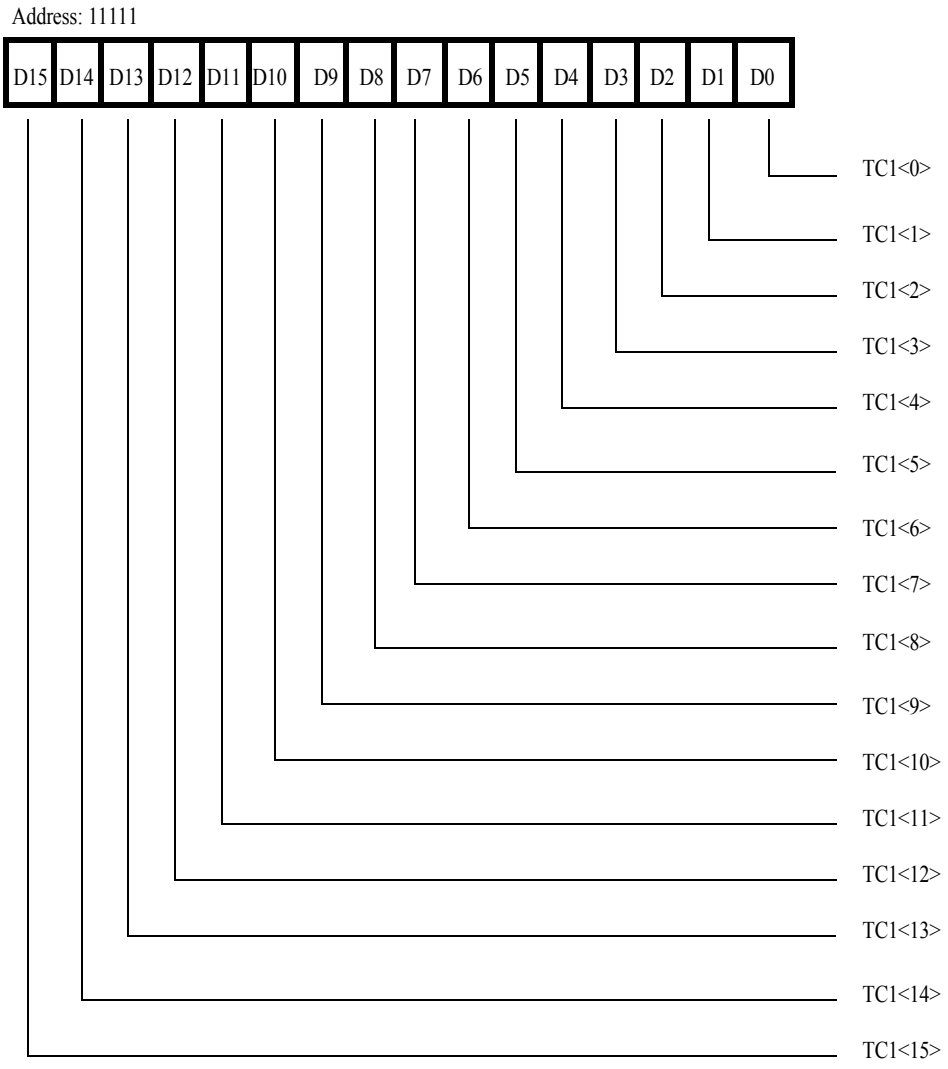
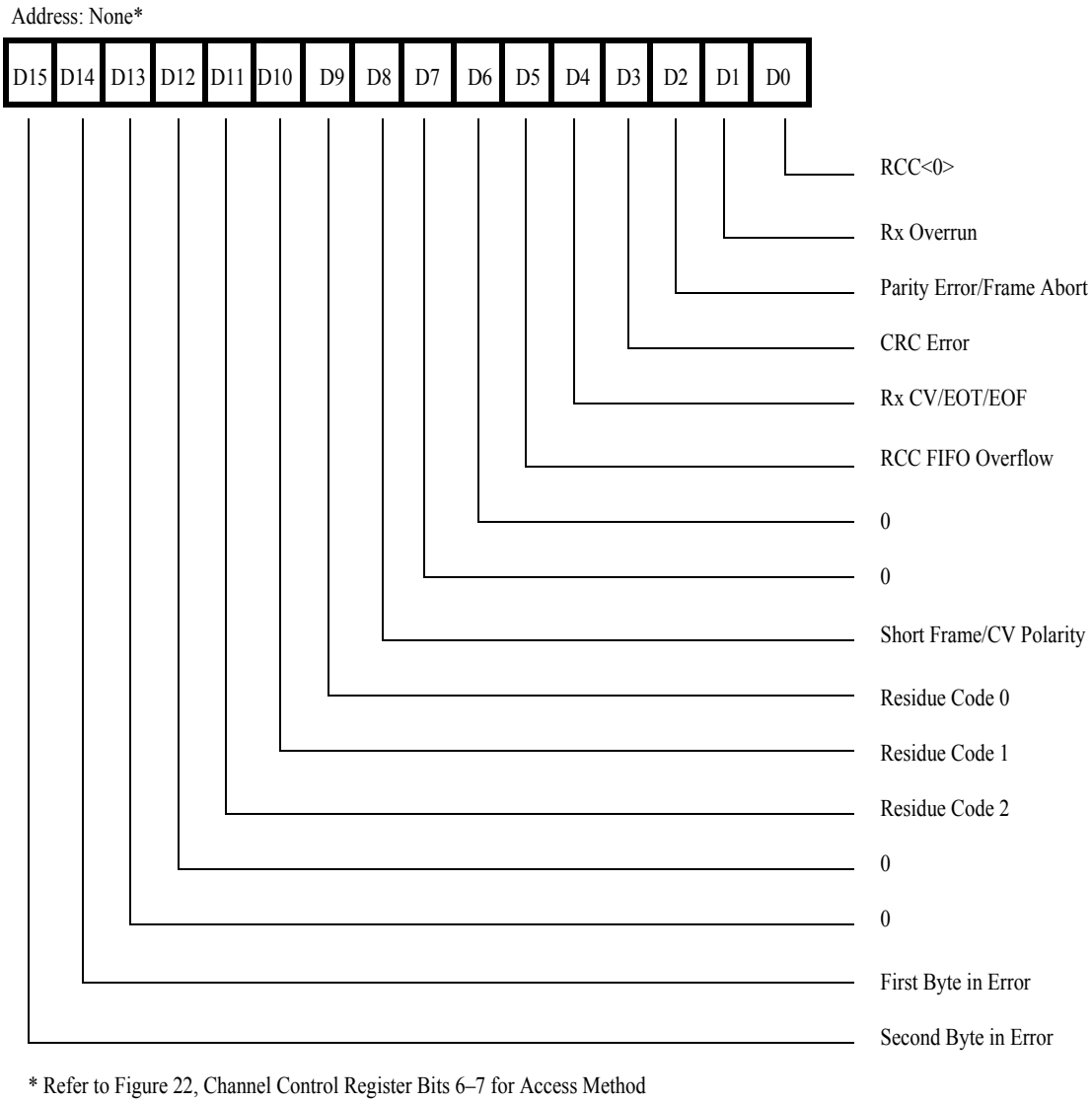
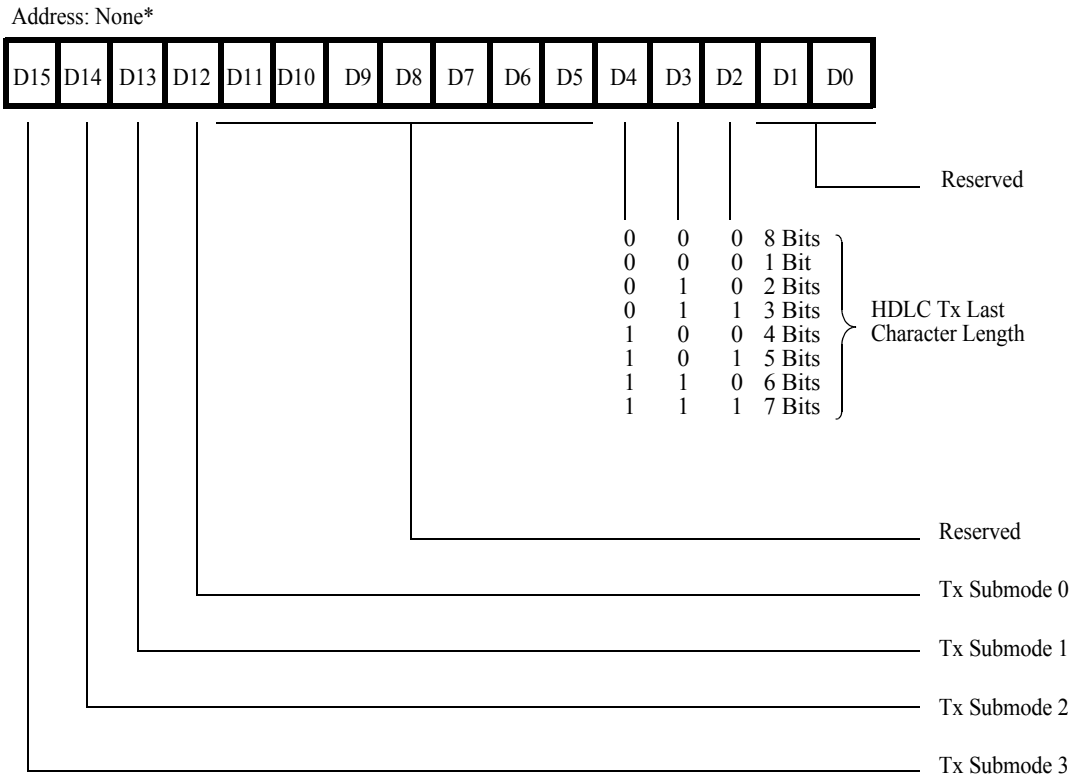


Figure 73. Time Constant 1 Register



* Refer to Figure 22, Channel Control Register Bits 6–7 for Access Method

Figure 74. Receive Status Block Register



* Refer to Figure 22, Channel Control Register Bits 6–7 for Access Method

Figure 75. Transmit Status Block Register

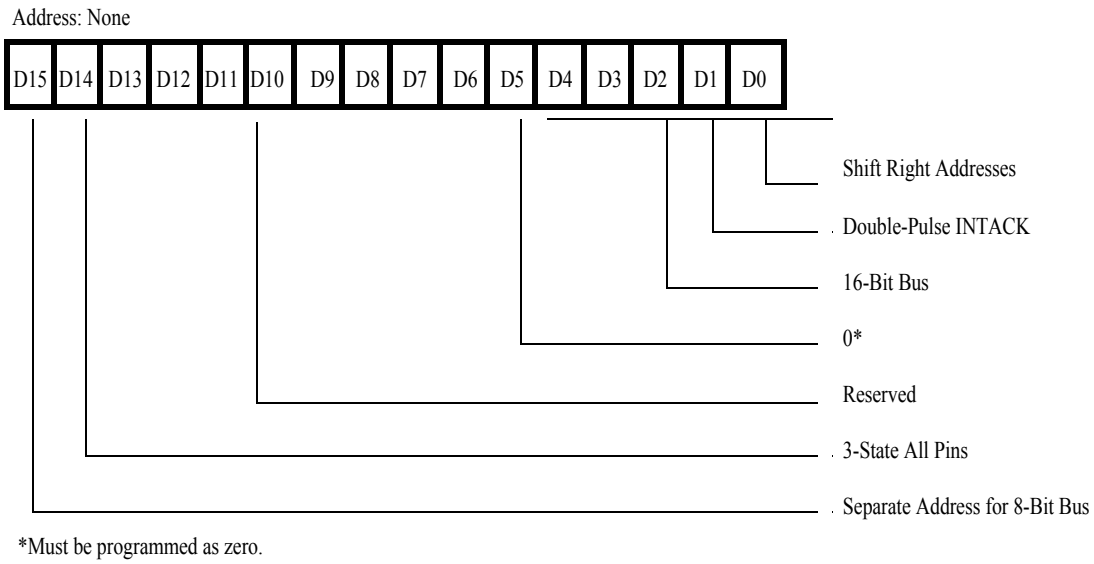
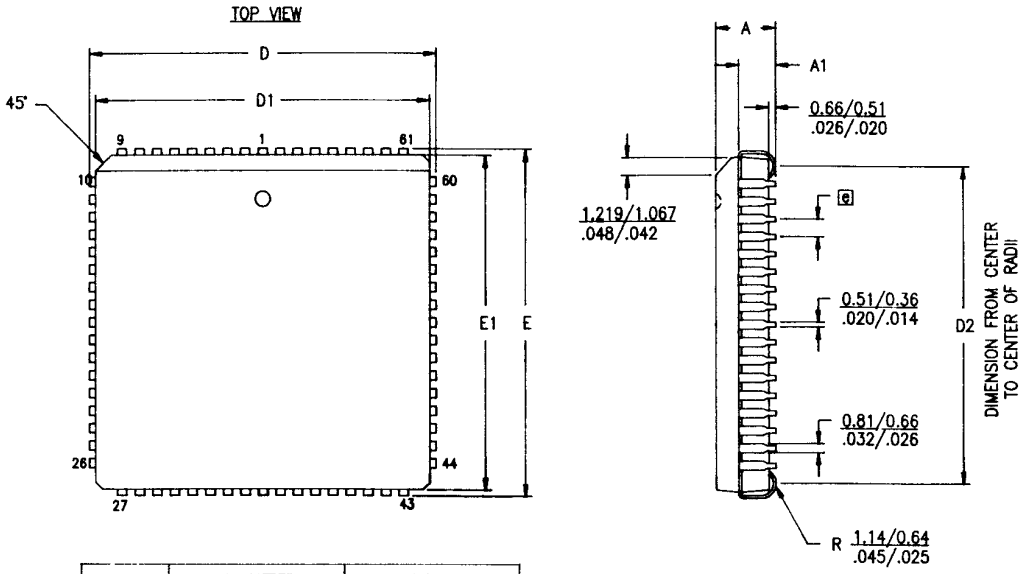


Figure 76. Bus Configuration Register

Packaging

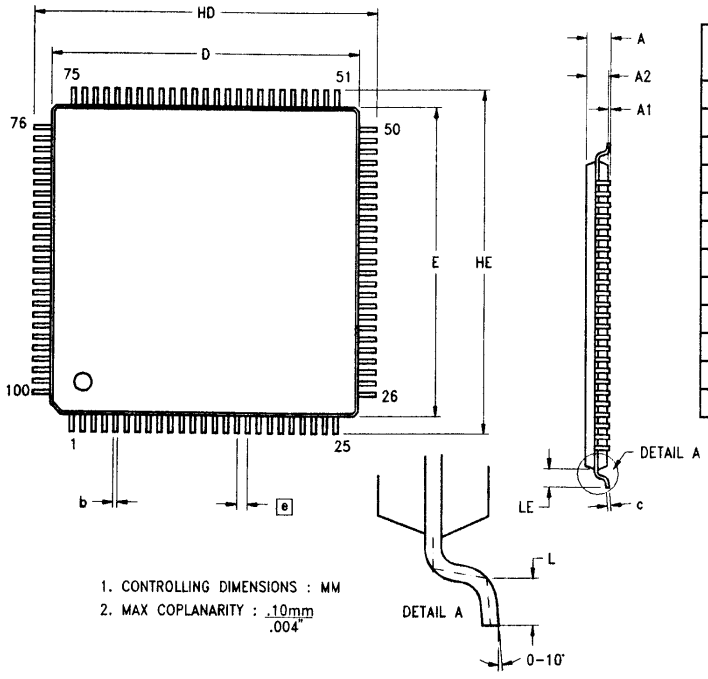
Figure 77 displays the 68-pin PLCC package diagram.



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.32	4.57	.170	.180
A1	2.43	2.92	.095	.115
D/E	25.02	25.40	.985	1.000
D1/E1	24.13	24.33	.950	.958
D2	22.86	23.62	.900	.930
e	1.27 TYP		.050 TYP	

Figure 77. 68-Pin PLCC Package Diagram

Figure 78 displays 100-pin VQFP package diagram



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	1.35	1.60	.053	.063
A1	0.05	0.20	.002	.008
A2	1.30	1.50	.051	.059
b	0.15	0.26	.006	.010
c	0.10	0.20	.004	.008
HD	15.85	16.15	.624	.636
D	13.90	14.10	.547	.555
HE	15.85	16.15	.624	.636
E	13.90	14.10	.547	.555
e	0.50 TYP		.0197 TYP	
L	0.35	0.65	.014	.026
LE	0.90	1.10	.035	.043

Figure 78. 100-Pin VQFP Package Diagram

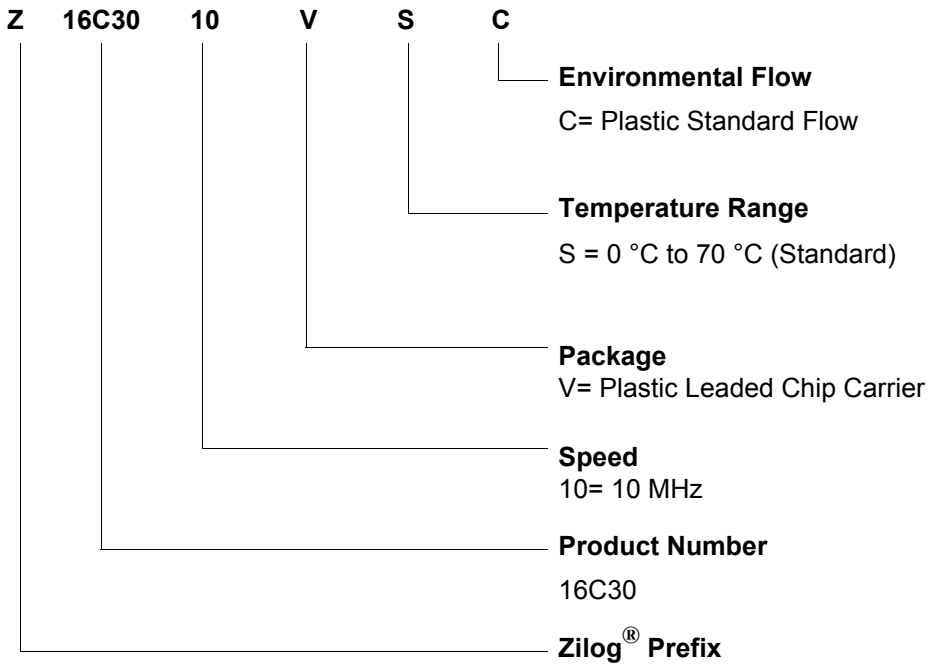
Ordering Information

Order the Z16C30 Series from Zilog®, using the following part numbers. For more information on ordering, consult your local Zilog sales office. The Zilog website (www.zilog.com) lists all regional offices and provides additional Z16C30 product information.

Z16C30 (10 MHz)	
68-Pin PLCC	Z16C3010VSC

For fast results, contact your local Zilog® sales office for assistance in ordering the part desired.

Codes



Customer Support

For answers to technical questions about the product, documentation, or any other issues with Zilog's offerings, please visit Zilog's Knowledge Base at http://www.zilog.com/kb.

For any comments, detail technical questions, or reporting problems, please visit Zilog's Technical Support at http://support.zilog.com.