# Z16C32

## IUSC™ INTEGRATED UNIVERSAL SERIAL CONTROLLER

## FEATURES

- Two Full-Capacity 20 MHz DMA Channels, Each with 32-Bit Addressing and 16-Bit Data Transfers.
- DMA Modes Include Single Buffer, Pipelined, Array-Chained and Linked-Array Chained.
- Ring Buffer Feature Supports Circular Queue of Buffers in Memory.
- Linked Frame Status Transfer Feature Writes Status Information for Received Frames and Reads Control Information for Transmit Frames to the DMA Channel's Array or Linked List to Significantly Simplify Processing Frame Status and Control Information.
- Programmable Throttling of DMA Bus Occupancy in Burst Mode with Bus Occupancy Time Limitation.
- 0 to 20 Mbit/sec, Full-Duplex Channel, with Two Baud Rate Generators and a Digital Phase-Locked Loop for Clock Recovery.
- 32-Byte Data FIFOs for Receiver and Transmitter
- Up to 12.5 MByte/sec (16-Bit) Data Bus Bandwidth
- Multiprotocol Operation Under Program Control with Independent Mode Selection for Receiver and Transmitter.
- Async Mode with One-to-Eight Bits/Character, 1/16 to Two Stop Bits/Character in 1/16 Bit Increments; 16x, 32x, or 64x Oversampling; Break Detect and Generation; Odd, Even, Mark, Space or No Parity and Framing Error Detection. Supports 9-Bit and MIL-STD-1553B Protocols.
- HDLC/SDLC Mode with 8-Bit Address Compare; Extended Address Field Option; 16- or 32-Bit CRC; Programmable Idle Line Condition; Optional Preamble Transmission and Loop Mode. Selectable Number of Flags Between Back-to-Back Frames.
- Byte Oriented Synchronous Mode with One-to-Eight Bits/Character; Programmable Sync and Idle Line Conditions; Optional Receive Sync Stripping; Optional Preamble Transmission; 16- or 32-Bit CRC; Transmit-to-Receive Slaving (for X.21).
- External Character Sync Mode for Receive
- 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.
- 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.
- Receive and Transmit Time Slot Assigners for ISDN, T1 and E1 (CEPT) Applications.
- 8-Bit General-Purpose Port with Transition Detection
- Low Power CMOS
- 68-Pin PLCC Package
- Electronic Programmer's Manual Support Tool and Software Drivers are Available.

## GENERAL DESCRIPTION

The Z16C32 IUSC™ (Integrated Universal Serial Controller) is a multiprotocol datacommunications device with on-chip dual-channel DMA. The integration of a high-speed serial communications channel with high-performance DMA facilitates higher data throughput than can be achieved with discrete serial/DMA chip combinations.

## GENERAL DESCRIPTION (Continued)

There are additional reasons for using the Z16C32 IUSC than just reduced chip count and board space economy. The DMA and serial channel intercommunication offers application benefits as well. For example, events such as the reception of the end of a HDLC frame is internally communicated from the serial controller to the DMA so that each frame can be written into a separate memory buffer. The buffer chaining capabilities, ring buffer support, automated frame status/control blocks, and buffer termination at the end of the frame combine to significantly reduce CPU overhead (Figure 1).

The IUSC is software configurable to satisfy a wide variety of serial communication applications. The 20 Mbit/second data rate and multiple protocol support make it ideal for applications in today's dynamic environment of changing specifications and increasing speed. The many programmable features allow the user to tune the device response to meet system requirements and adapt to future requirements. The IUSC contains a variety of sophisticated internal functions including two baud rate generators, a digital phase-locked loop, character counters, and 32-byte FIFOs for both the receiver and the transmitter.

The on-chip DMA channels allow high speed data transfers for both the receiver and the transmitter. The IUSC supports automatic status and control transfer through DMA and allows initialization of the serial controller under DMA control. Each DMA channel can do a 16-bit transfer in as little as three 50 ns clock cycles and can generate addresses compatible with 32-, 24- or 16-bit memory ranges. The DMA channels operate in any of four modes: single buffer, pipelined, array-chained, or linked-list. The array-chained and linked-list modes provide scatter-read and gather-write capabilities with minimal software intervention. To prevent the DMA from holding bus mastership too long, mastership time may be limited by counting the absolute number of clock cycles, the number of bus transactions, or both.

The CPU bus interface is designed for use with any conventional multiplexed or non-multiplexed bus from manufacturers of CISC and RISC processors including Intel, Motorola, and Zilog. The bus interface is configurable for 16-bit data, 8-bit data with separate address or 8-bit data without separate address to support multiplexed or non-multiplexed busses.

The IUSC handles asynchronous formats, synchronous bit-oriented formats such as HDLC and synchronous byte-oriented formats (e.g., BISYNC and DDCMP). This device supports virtually any serial data transfer application.

The IUSC can generate and check CRC in any synchronous mode. Complete access to the CRC value allows system software to resend or manipulate the CRC as needed in various applications. The IUSC also provides facilities for modem control signals. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

Interrupts are supported by a daisy-chain hierarchy within the serial channel and between the serial channel and the DMA. Separate interrupt vectors for each type of interrupt within the serial controller and the DMA facilitate fast discrimination of the interrupt source. The IUSC supports Pulsed, Double Pulsed, and Status Interrupt Acknowledge cycles.

Support tools are available to aid the designer in efficiently programming the IUSC. The Technical Manual describes in detail all the features and gives programming sequence hints. The Electronic Programmer's Manual, DC #8287-02, is an MS-DOS, disk-based programming initialization tool that can generate custom sequences. Also, Zilog offers assorted application notes and development boards to assist the designer in hardware and software development. Contact your nearest Zilog representative for additional information.

**Notes:**
All Signals with a preceding front slash, "/", are active Low, e.g.: B//W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

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

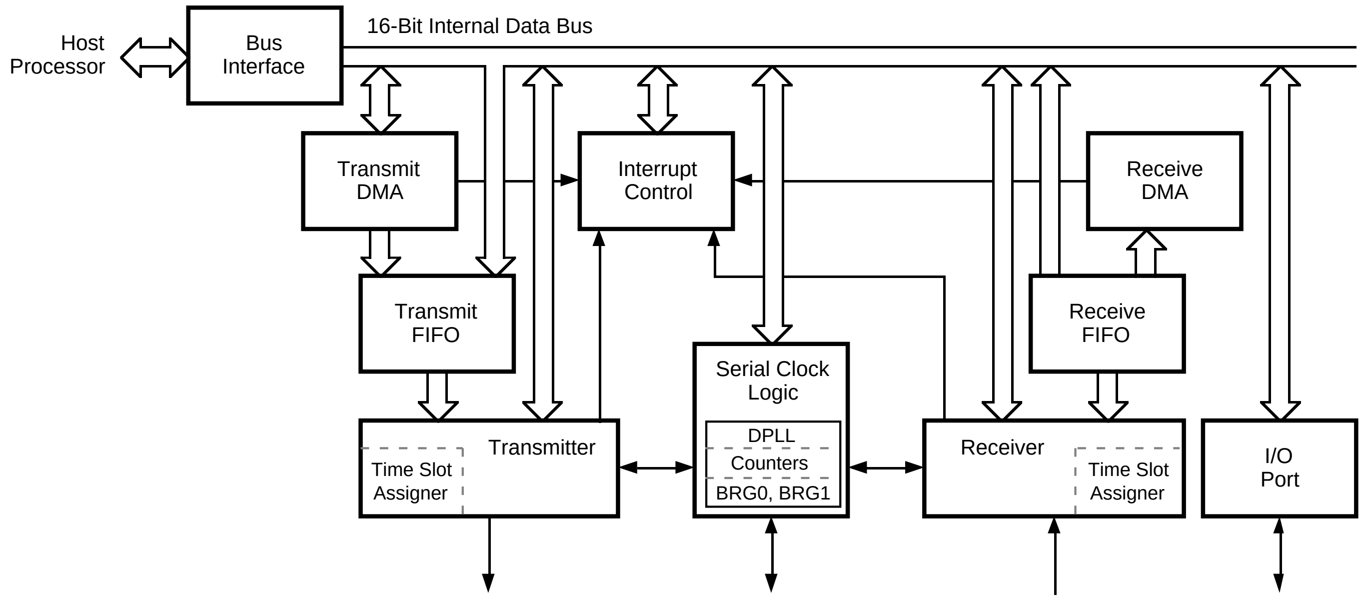



**Figure 1. Z16C32 IUSC Block Diagram**

## GENERAL DESCRIPTION (Continued)

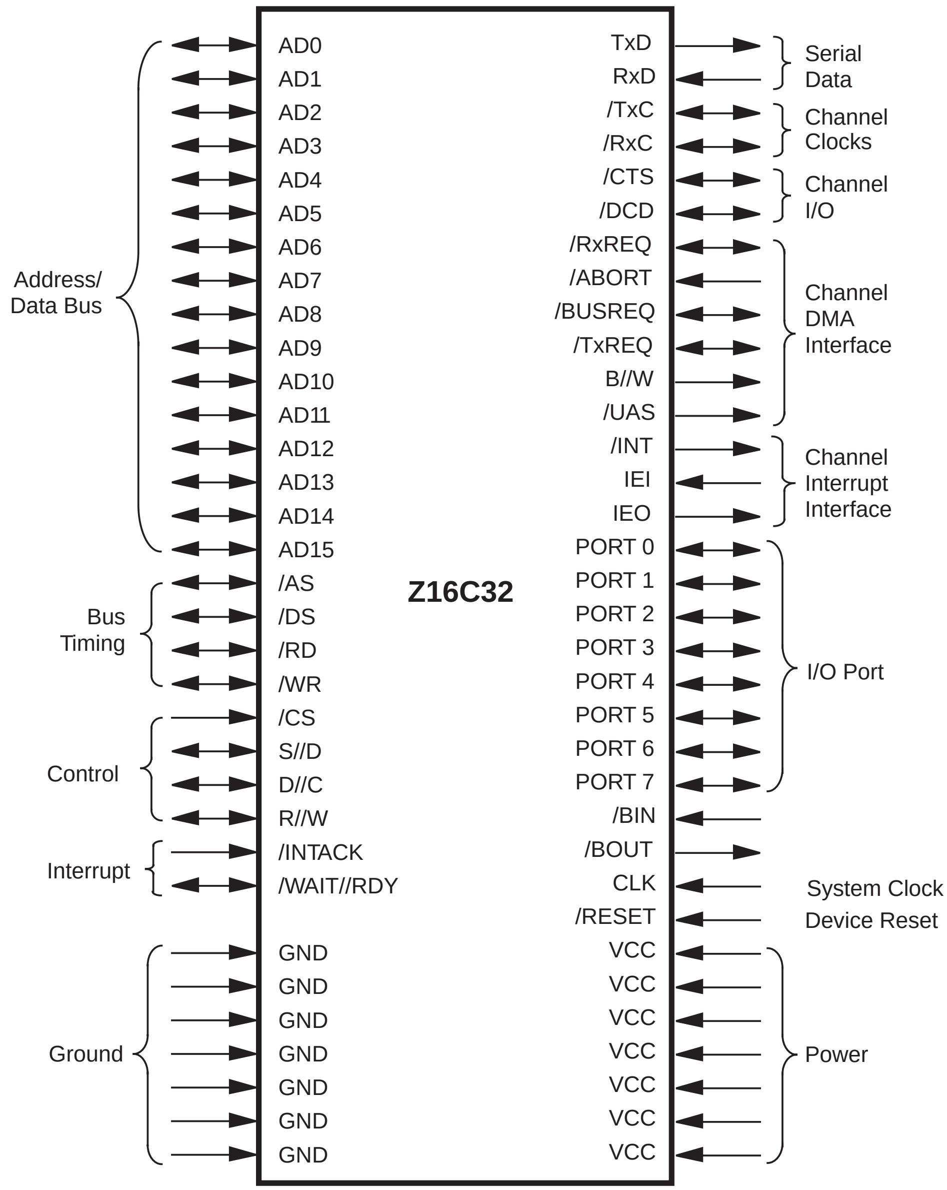



**Figure 2. Z16C32 Pin Functions**

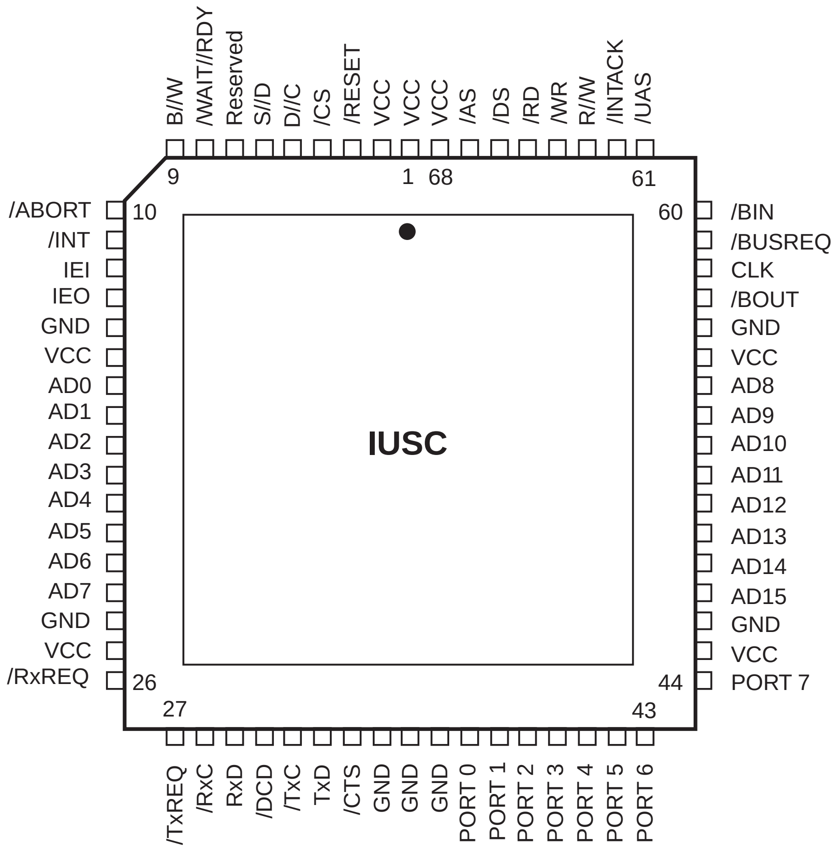



**Figure 3. Z16C32 68-Pin PLCC Pin Assignments**

## PIN DESCRIPTION

Figure 2 shows the logical pin groupings of the IUSC's pins, and Figure 3 shows the physical pin assignments.

Only one strobe pin (/DS, /RD, /WR or Pulsed INTACK) should ever be active at one time. Any unused input pin (if an input when the IUSC is bus master or slave) must be pulled up to its inactive state.

**/RESET** *Reset* (input, active Low). A Low on this line places the IUSC in a known, inactive state, and conditions it so that the data, from the next write operation that asserts the /CS pin, goes into the Bus Configuration Register (BCR) regardless of register addressing. /RESET should be driven Low as soon as possible during power-up, and as needed when restarting the overall system or the communications subsystem.

**CLK** *System Clock* (input). This signal is the timing reference for the DMA and bus interface logic. (The serial controller section is clocked by the selected sources of receive and transmit clocking.)

**AD15-0** *Address/Data Bus* (inputs/tri-state outputs). After Reset, these lines carry data between the controlling microprocessor and the IUSC, and may also carry multiplexed addresses of registers within the IUSC. Such operation, between the host processor and the IUSC, is often called slave mode. Once the software has set up the device and placed it into operation, these lines also carry multiplexed addresses and data between the IUSC and system memory; such operation is called master mode. AD15-0 can be used in a variety of ways based on whether the IUSC senses activity on /AS after Reset, and on the data written to the Bus Configuration Register (BCR).

**/CS** *Chip Select* (input, active Low). A Low on this line indicates that the controlling microprocessor's current bus cycle refers to a register in the IUSC. The IUSC ignores /CS when a Low on /INTACK indicates that the current bus operation is an interrupt acknowledge cycle. On a multiplexed bus the IUSC latches the state of this pin at rising edges on /AS; on a non-multiplexed bus, it latches /CS at leading/falling edges on /DS, /RD, or /WR.

## PIN DESCRIPTION (Continued)

**S//D** *Serial/DMA* (input/tri-state output, input High indicates serial). Cycles with /CS Low, and /INTACK and this pin both High, access registers in the serial controller section. Cycles with /INTACK High, and /CS and this pin both Low, access registers in the DMA controller section. The state of this line when the Bus Configuration Register is written determines wait vs acknowledge operation, as described in the text. On a multiplexed bus, the IUSC latches the state of this pin at rising edges on /AS; on a non-multiplexed bus, it latches the state at leading/falling edges on /DS, /RD, or /WR.

Software can program the IUSC so that when it is acting as a bus master, it drives this line High to indicate a DMA cycle for serial data and Low to indicate an "array" or "list" access. (Array/list accesses read the address and length of the next memory buffer.)

**D//C** *Data/Control* (input/tri-state output, input High indicates Data). A slave read cycle with /CS Low , and all three of /INTACK, S//D, and this pin High, fetches data from the serial controller's receive FIFO through the Receive Data Register (RDR). A slave write cycle with the same conditions writes data into the transmit FIFO through its Transmit Data Register (TDR). Slave cycles with /INTACK and S//D High, and /CS and this pin Low , read or write registers in the serial controller. On a multiplexed bus, the IUSC determines which register to access from the low-order AD lines at the rising edge of /AS; on a non-multiplexed bus it typically selects the register based on the Least Significant Bits of the serial controller's Channel Command/Address Register. On a multiplexed bus, the IUSC latches the state of this pin at rising edges on /AS; on a non-multiplexed bus it latches the state at leading/ falling edges on /DS, /RD, or /WR.

For slave cycles on a multiplexed bus, with /INTACK High and both /CS and S//D Low, the state of this line at the rising edge of /AS selects between the registers of the transmit DMA channel (Low ) and those of the receive DMA channel (High). On a non-multiplexed bus, with /INTACK High and /CS and S//D both Low, the IUSC can take the DMA channel selection from this line or from the DMA Command/Address Register.

Software can program the IUSC so that when it is acting as a bus master, it drives this line High to indicate a DMA cycle for the receiver and Low to indicate a cycle for the transmitter.

**/AS** *Address Strobe* (input/tri-state output, active Low). After a reset, the IUSC's bus interface logic monitors this signal to see if the host bus multiplexes addresses and data on AD15-0. If the logic sees activity on /AS before (or during) software writes to the Bus Configuration Register, then in subsequent slave cycles directed to the IUSC, it captures register selection from the AD lines, S//D, and C//D on rising edges of /AS.

When the IUSC takes control of the bus and operates as a master, it always uses the bus in a multiplexed fashion, driving /AS Low when it places the least significant 16 bits of an address on the AD15-0 lines. External devices can be used to de-multiplex the address and data, if this is necessary to match the characteristics of the host processor or host bus.

For a non-multiplexed bus, this pin should be pulled up to +5V using a resistor of about 10 kOhms. If a processor uses a non-multiplexed bus, yet has an output called Address Strobe (e.g., 680x0 devices), this pin should not be tied to the output.

**/UAS** *Upper Address Strobe* (tri-state output, active Low). When the IUSC takes control of the bus and operates as a master, it drives /UAS Low when it places the more significant 16 bits of an address on AD15-0. External memory and other slave devices (or de-multiplexing latches) should capture the MS address at each rising edge on this line.

**R//W** *Read/Write control* (input/tri-state output, Low signifies "write"). R//W and /DS indicate read and write cycles on the bus, for host processors/buses having this kind of signalling. When the IUSC has taken control of the bus and is operating in master mode, this pin is an output that remains valid throughout the Low time of /DS. In slave cycles, the IUSC samples R//W at each leading/falling edge on /DS.

**/DS** *Data Strobe* (input/tri-state output, active Low). R//W and /DS indicate read and write cycles on the bus, for host processors/buses having this kind of signalling. It is an output when the IUSC has taken control of the bus and is operating in master mode, otherwise, it is an input that is qualified by /CS Low or /INTACK Low. In master mode, the R//W line remains valid throughout the Low time of this line. In slave mode, the IUSC samples R//W at each leading/ falling edge on this line. For slave write cycles and master read cycles, the IUSC captures data at the rising (trailing) edge on this line. For slave read cycles the IUSC provides valid data on the AD lines within the specified access time after this line goes Low , and keeps the data valid until after the master releases this line to High. For master write cycles, the IUSC places valid data on the AD lines before it drives this signal to Low, and keeps the data valid until after it drives this line back to High.

**/RD** *Read Strobe* (input/tri-state output, active Low). This line indicates a read cycle on the bus, for host processors/ buses having this kind of signalling. It is an output when the IUSC has taken control of the bus and is operating in master mode, otherwise, it is an input that is qualified by /CS Low or /INTACK Low. For master read cycles, the IUSC captures data at the rising (trailing) edge of this line. For slave read cycles the IUSC provides valid data on the AD lines within the specified access time after this line goes Low, and keeps the data valid until after the master releases this line to High.

**/WR** *Write Strobe* (input/tri-state output, active Low). This line indicates write cycles on the bus, for host processors/ buses having this kind of signalling. It is an output when the IUSC has taken control of the bus and is operating in master mode, otherwise it is an input that is qualified by /CS Low. For slave write cycles, the IUSC captures write data at the rising (trailing) edge of this line. For master write cycles, the IUSC places valid data on the AD lines before it drives this signal to Low , and keeps the data valid until after it drives this line back to High.

**B//W** *Byte / Word Select* (tri-state output, High indicates 8-bit transfer). When the IUSC takes control of the bus and operates as a master, a High on this line indicates that a byte is to be transferred, and a Low indicates that 16 bits are to be transferred. The IUSC ignores this signal during slave cycles: it takes the byte/word distinction from an AD line at the rising edge of /AS, or from a bit in the serial or DMA Command/Address Register.

**/WAIT//RDY** *Wait, Ready, or Acknowledge handshaking* (input/tri-state output, active Low). This line is an input when the IUSC has taken control of the bus and is operating in master mode. For slave cycles, the IUSC activates this line as an output. In both directions, the line can carry wait or acknowledge signalling depending on the state of the S//D input during the initial BCR write. If S//D is High when the BCR is written, this line operates as a Ready/Wait line for Zilog and most Intel processors. In this mode, the IUSC will not complete a master cycle while this line is Low, and it asserts this line Low until it's ready to complete an interrupt acknowledge cycle; it never asserts this line when the host accesses one of the IUSC registers.

If S//D is Low when the BCR is written, this line operates thereafter as an Acknowledge line for Motorola and some Intel processors. In this mode, the IUSC will not complete a master cycle until this line is Low. It asserts this line Low for register read and write cycles, and when it is ready to complete an interrupt acknowledge cycle.

For slave cycles, this is a full time (totem pole) output. The board designer can combine this signal with similar signals from other slaves, by means of an external logic gate or a tri-state or open-collector driver.

**/INT** *Interrupt Request* (output, active Low). The IUSC drives this line Low when (1) its IEI pin is High, (2) one or more of its interrupt condition(s) is (are) enabled and pending, and (3) the Under Service flag is not set for its highest priority enabled/pending condition, nor for any higher-priority internal condition. Software can program whether the bus interface drives this pin in a totem-pole or an open-drain fashion.

**/INTACK** *Interrupt Acknowledge* (input, active Low). A Low on this line indicates that the host processor is performing an interrupt acknowledge cycle. In some systems, a Low on this line may further indicate that external logic has selected this IUSC as the device to be acknowledged, or as a potential device to be acknowledged. A field in the Bus Configuration Register selects whether this line carries a level-sensitive "status" signal that the IUSC should sample at the leading edge of /AS or /DS, or a single-pulse or double-pulse protocol. The IUSC responds to an interrupt acknowledge cycle in a variety of ways depending on this programming and the state of the /INT and IEI lines, as described in the text.

**IEI** *Interrupt Enable In* (input, active High). This signal and the IEO pin can be part of an interrupt-acknowledge daisy chain with other devices that may request interrupts. If IEI is High outside of an interrupt acknowledge cycle, one or more IUSC interrupt condition(s) is (are) enabled and pending, and the Under Service flag isn't set for the highest priority condition nor for any higher-priority one, then the IUSC requests an interrupt by driving its /INT pin Low. If the IEI pin is High during an interrupt acknowledge cycle, one or more IUSC interrupt condition(s) is (are) enabled and pending, and the Under Service flag isn't set for the highest priority condition nor for any higher-priority, then the IUSC keeps IEO Low and responds to the cycle.

**IEO** *Interrupt Enable Out* (output, active High). This signal and/or IEI can be part of an interrupt acknowledge daisy chain with other devices that may request interrupts. The IUSC drives its IEO pin Low whenever its IEI pin is Low, and/or if the Under Service flag is set for any condition. This IUSC drives this signal slightly differently during an interrupt acknowledge cycle, in that it also forces IEO Low if it is (has been) requesting an interrupt.

## PIN DESCRIPTION (Continued)

**/BUSREQ** *Bus Request* (output, active Low). The DMA controller section drives this line Low to request control of the host bus. /BUSREQ can be an open-drain or totem-pole output depending on a bit in the Bus Configuration Register. In open-drain mode the IUSC samples the pin as an input and only drives it Low after sampling it high.

**/BIN** *Bus Acknowledge In* (input, active Low). When the IUSC receives a falling edge on this input, it samples whether it has been driving (or has just begun to drive) /BUSREQ. If so, it keeps /BOUT High and takes control of the host bus. If not, it passes the bus grant by driving /BOUT Low. This signal can be used with /BOUT to form a bus-grant daisy chain for arbitration of bus control. Alternatively, it can be connected to a direct, positive grant from an external arbiter, and the /BOUT pin can be left unconnected.

**/BOUT** *Bus Acknowledge Out* (output, active Low). As noted above, this signal can be used with /BIN to form a bus-grant daisy chain for arbitration of bus control.

**/ABORT** *Abort Master Cycle* (input, active Low ). A Low on this line during a master cycle makes the currently active DMA channel terminate its activity and enter a disabled state. Note that /ABORT is only effective during a DMA cycle, so that the IUSC knows which channel should be aborted. Also note that external logic must set /WAIT//RDY to the right state for the cycle to complete, before /ABORT becomes effective.

**RxD** *Received Data* (input, positive logic). The serial input.

**TxD** *Transmit Data* (output, positive logic). The serial output.

**/RxC** *Receive Clock* (input or output). This signal can be used as a clock input for any of the functional blocks in the serial controller. Or, software can program the IUSC so that this pin is an output carrying any of several receiver or internal clock signals, a general-purpose input or output, or an interrupt input.

**/TxC** *Transmit Clock* (input or output). This signal can be used as a clock input for any of the functional blocks in the serial controller. Or, software can program the IUSC so that this pin is an output carrying any of several transmitter or internal clock signals, a general purpose input or output, or an interrupt input.

**/RxREQ** *Receive DMA Request* (input or output). In device testing or in applications not using the serial controller and DMA controller sections together in the usual way, this pin can carry an active Low DMA Request from the receive FIFO. On the IUSC this request is internally routed to the on-chip Receive DMA channel; it is more typical to use the RxREQ pin as a general-purpose output or as an interrupt input.

**/TxREQ** *Transmit DMA Request* (input or output). In device testing or in applications not using the serial controller and DMA controller sections together in the usual way, this pin can carry an active Low DMA Request from the transmit FIFO. On the IUSC this request is internally routed to the on-chip Transmit DMA channel, and it's more typical to use the RxREQ pin as a general-purpose output or as an interrupt input.

**/DCD** *Data Carrier Detect* (input or output, active Low). Software can program the IUSC so that this signal enables/disables the receiver. In addition, software can program the device to request interrupts in response to transitions on this line. The pin can also be used as a simple input or output.

**/CTS** *Clear to Send* (input or output, active Low). Software can program the IUSC so that this signal enables/disables the transmitter. In addition, software can program the device to request interrupts in response to transitions on this line. The pin can also be used as a simple input or output.

**PORT7/TxCOMPLT** *General-Purpose I/O or Transmit Complete* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or so that it carries a Transmit Complete signal from the Transmitter, that can control an external driver. The IUSC captures transitions on this pin in internal latches.

**PORT6/FSYNC** *General-Purpose I/O or Frame Sync* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or a Frame Sync input for the IUSC's Time Slot Assigner circuits. The IUSC captures transitions on this pin in internal latches.

**PORT5/RxSYNC** *General-Purpose I/O or Receive Sync* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or so that it carries a Receive Sync output from the Receiver. The IUSC captures transitions on this pin in internal latches.

**PORT4/TxTSA** *General-Purpose I/O or Transmit Time Slot Assigner Gate* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or so that it carries the Gate output of the Transmit Time Slot Assigner, that can enable an external TxD driver in time-slotted ISDN or Fractional T1 applications. The IUSC captures transitions on this pin in internal latches, as described in the text.

**PORT 3/RxTSA** *General-Purpose I/O or Receive Time Slot Assigner Gate* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or so that it carries the Gate output of the Receive Time Slot Assigner. The IUSC captures transitions on this pin in internal latches.

**PORT 2** *General-Purpose I/O* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output. The IUSC captures transitions on this pin in internal latches.

**PORT 1-0/CLK 1-0** *General-Purpose I/Os or Reference Clocks* (inputs or outputs). Software can program the IUSC so that either of these pins is a general-purpose input or output, or a reference clock that can be divided down to derive clocking for the Receiver and/or Transmitter. When one of these pins is a general-purpose I/O, the IUSC captures transitions on it in internal latches.

**$V_{CC}$, $V_{SS}$** *Power and Ground.* The inclusion of seven pins for each power rail ensures good signal integrity, prevents transients on outputs, and improves noise margins on inputs. The IUSC's internal power distribution network requires that all these pins be connected appropriately.

## ARCHITECTURE

The IUSC integrates a fast and efficient dual-channel DMA with a highly versatile serial communications controller. The functional capabilities of the IUSC are described from two different points of view; as a datacommunications device, it transmits and receives data in a wide variety of datacommunications protocols; as a microprocessor peripheral with two DMA channels that offer such features as four DMA transfer types, a flexible bus interface, and vectored interrupts. The architecture is described in three sections, DMA and Bus Interface Capabilities, Communication between the DMA and Serial Channel, and Serial Communication Capabilities. The structure of the IUSC is shown in Figure 1.

# DMA AND BUS INTERFACE CAPABILITIES

The IUSC's two versatile DMA channels combined with a flexible bus interface gives it the ability to meet a wide variety of application requirements. The time required to move data into and out of the transmitter and receiver is minimized by the IUSC's speed (20 MHz clock, three clock cycles per word, typical); two buffer-chaining modes with linked-frame status transfer; early buffer termination to keep received frames in separate memory buffers; and vectored interrupts. Some of the these features are briefly described below, however, the user should refer to the IUSC Technical Manual for additional information.

## DMA Modes

The IUSC contains two DMA channels, one for the transmitter and one for the receiver. Each channel supports a 32-bit address and a 16-bit byte count. The channels operate in one of four modes. In normal mode, the processor must reload the address and length at the end of each buffer. In Pipelined mode, the processor can load the address and length of the next buffer at any time during the DMA transfer to the first buffer. In Array-Chained mode the processor creates a table of address/length pairs in memory for automatic transfer by the channel. In Linked List mode the processor creates a linked list of address and length pairs in memory to be automatically transferred by the channel.

Single Buffer Mode is the most basic of the four data transfer types. The starting address of each memory buffer and the maximum number of characters to be transferred to or from memory are programmed into the IUSC registers. When the DMA is enabled, it transfers all data between system memory and the transmit and receive FIFOs.

Pipelined Mode is similar to Single Buffer Mode with the addition of an extra set of registers into which the processor can load to reload the DMA with the address and count of the next memory buffer. Therefore, when a buffer is complete, the IUSC is pre-programmed with the address and count of the next buffer so the DMA need not stop between each buffer as long as software stays one step ahead of memory buffer usage.

In Array Mode, one of the two chaining modes, software sets up a table of memory buffer information. The length of the array is only limited by the amount of system memory available for buffers. The IUSC is programmed with the location of the array of buffer addresses and sizes. This mode has the advantage that a burst of short frames is less likely to overrun the systems ability to keep up. The use of receive status block and transmit control block along with the early buffer termination feature simplifies the segmentation and reassembly of serial messages in memory buffers. When a DMA channel fetches a buffer count of zero, it stops and can create an End-Of-Array interrupt.

Linked List Mode is the most versatile of DMA modes. It has the Array Mode's ability to switch buffers rapidly without the requirement for the buffer information to be in a continuous table. Each link entry contains: The starting address to write or read the data; the size of the buffer; optional status or control information; and a pointer to the next link. Memory buffers can easily be added and removed from the list by changing the links in list entries.

## DMA Features

In Linked List Mode, the IUSC has a programmable feature to facilitate the use of buffers in a ring. When this feature is enabled, the DMA writes a zero back to the buffer length field of each array or list entry after it is read. Therefore, if a linked list wraps around on itself, a DMA channel will not reuse a buffer until software has processed the buffer, and indicated that its eligible for reuse by writing a nonzero value in the count field (fetching a count value of zero deactivates the DMA channel). This feature can also be used in array mode to track buffer use.

In both Bus Slave and Master Modes, the IUSC can read and write data words in either byte order. It supports the Little Endian convention used by many Intel microprocessors and the Big Endian convention used by many Motorola microprocessors. When the IUSC is bus master, it can be programmed to generate only the upper 16-bit address when required and, consequently, save a clock cycle on each transfer (three clocks per transfer instead of four). When using the IUSC on a 16-bit bus and the starting address of the message is on an odd address, the IUSC automatically reorients itself onto even word boundaries by first fetching a byte. This is especially valuable when retransmitting a frame with a different size header than was received. Two pins are available as status signals of the type of transfer in progress.

There are a variety of command and status registers to control and monitor the DMA channels. A DMA channel can be aborted with either the /ABORT pin or by software command. A pause command is also available to temporarily suspend transfers.

## Bus Interface & Utilization

The bus interface module stands between the external bus pins and an on-chip 16-bit data bus that interconnects the other functional modules. It includes several flexible bus interfacing options that are controlled by the contents of the Bus Configuration Register (BCR). The BCR is automatically the destination of the first write to the IUSC by the host processor after a reset.

The IUSC is compatible with both multiplexed and non-multiplexed bus interfaces and can transfer either 8 or 16 bits. It supports data transfers with /RD and /WR or R//W and /DS strobe pins and either format of byte ordering. The IUSC generates the Wait or Ready acknowledge handshaking used by Intel or Motorola microprocessors. Also, three styles of interrupt acknowledge signals are supported for automated return of an interrupt vector to any common microprocessor.

There are several options that control how the IUSC uses the bus. The /BIN and /BOUT pins are available to form a bus-grant daisy chain. The IUSC has several options on how it arbitrates requests for bus mastership between channels and how long it stays off the bus between requests. The priority of the two DMA channels is programmable and can alternate between requests to allow both channels equal access to the bus. Once one of the channels has mastership of the bus, control can be passed to the other channel if it is requesting or the IUSC can be forced off the bus. A programmable preempt feature selects whether the higher priority channel can take over control of the bus if it starts requesting control while the lower priority channel is using the bus.

The IUSC maximizes the use of its 32-byte FIFOs by holding /BUSREQ active until the transmit FIFO is full, the receive FIFO is empty, or both. The programmable dwell timers can be used to limit how long the IUSC holds bus mastership by counting either bus transfers, clock cycles or both. Therefore, the combination of programmable FIFO request levels, channel arbitration options, and programmable dwell timer features provide application software the flexibility to optimize the IUSCs bus occupancy to meet system throughput and bus response requirements.

## Interrupts

The interrupt subsystem of the IUSC derives from Zilog's experience in providing the most advanced interrupt capabilities in the microprocessor field. These capabilities are at their best when used with a Zilog microprocessor, but it is easy to interface the IUSC to work well with other microprocessors as well. Four pins are dedicated to create an interrupt daisy-chain hierarchy within the Serial Channel and between the Serial Channel and the DMA.

When an IUSC responds to an interrupt acknowledge from the CPU, it places an interrupt vector on the data bus. To speed interrupt response time, the IUSC modifies three bits in the vector to indicate which type of interrupt is being requested. Separate vectors are provided for the serial channel and DMA to easily discriminate the interrupt source.

The DMA has four interrupt sources each for the receive and transmit channels. Each interrupt source is independently enabled and there is a master enable for all DMA interrupts. The four interrupt sources are End Of Array/ End of Link, End Of Buffer, Hardware Abort, and Software Abort.

Each of the six types of interrupts in the serial portion IUSC (Receive Status, Receive Data, Transmit Status, Transmit Data, I/O Status and Device Status) has three bits associated with it: Interrupt Pending (IP), Interrupt-Under-Service (IUS) and Interrupt Enable (IE). If the IE bit for a given source is set, then that bit can source request interrupts. Note that individual sources within the six types also have their own interrupt arm bits. Finally, there is a Master Interrupt Enable (MIE) bit which globally enables or disables all interrupts from the serial channel.

The Interrupt (/INT), Interrupt Acknowledge (/INTACK), Interrupt Enable In (IEI) and Interrupt Enable Out (IEO) pins are provided to create an automated mechanism to place the vector on the bus among the highest priority pending interrupts from multiple devices. The device with the highest pending interrupt (/INT Low, IEI High) places a vector on the bus in response to an interrupt acknowledge cycle.

In the IUSC, the IP bit signals that an interrupt is pending. If an IUS bit is set, this interrupt is being serviced and all interrupt sources of lower priority are prevented from requesting interrupts. An IUS bit is set during an interrupt acknowledge cycle if there are no higher priority devices requesting interrupts.

## DMA AND BUS INTERFACE CAPABILITIES (Continued)

There are six sources of Receive Status interrupt. Each one is individually armed: Receiver exited hunt, received idle line, received break/abort, received code violation/end-of-transmission/end-of-message, parity error/abort 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 one is individually armed: Preamble sent, idle line sent, abort sent, end-of-frame/end-of-message 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 individual edge selection and arming for each pin. The pins that can be programmed to generate I/O Status interrupts are /RxC, /TxC, /RxREQ, /TxREQ, /DCD and /CTS. These interrupts are independent of the programmed function of the pins.

The Device Status interrupt has four individually enabled sources: Receive character counter underflow, DPLL sync acquired, BRG1 zero count and BRGO zero count. Refer to the IUSC Technical Manual for more details.

## COMMUNICATION BETWEEN THE DMA AND SERIAL CHANNELS

The IUSC's intra-chip communication between the DMA and serial communications controller gives it the power to achieve higher efficiency than is possible with a separate DMA controller. The Linked Frame Status Transfer feature writes the status and byte count of each received frame to memory as part of an array or linked list. This provides a simple and easy to use mechanism for storing the results of a received message without arbitrary restrictions on how quickly the host software must examine the results. Similarly, control information for transmit frames can be automatically read by the DMA from the array or link and transferred into registers in the serial logic.

In all modes, the DMA can accept a signal from the serial channel for early buffer termination. When the end of a message is received, the data is transferred to the buffer and the status is written to memory. The status is written after the data in single buffer and pipelined modes or to the array/link in array and linked-list modes if Linked-Frame Status Transfer is enabled. This early buffer termination is treated identically to the terminal count condition in the DMA. Therefore, the receipt of the end of a message is a seamless transition from one memory buffer to the next.

An example of using these intercommunication features using linked list mode is shown in Figure 4. This example shows the format of a ring of memory buffers with the linked frame status transfer and ring buffer features enabled. Any protocol that sets the "RxBound" bit (RCSR4 = 1), like HDLC or 802.3, is appropriate to this example. The linked list is shown in Figure 4 with three links for simplicity and may be as large as memory allows. The sixth word in each list entry is reserved and should not be used (it keeps the list entries on 32-bit boundaries). If the end of the buffer is reached and it is not the end of the frame, the IUSC writes zeros as the status and count. Also, if the transmit channel needs to start a new memory buffer other than at the beginning of a frame, the DMA ignores the transmit control block.

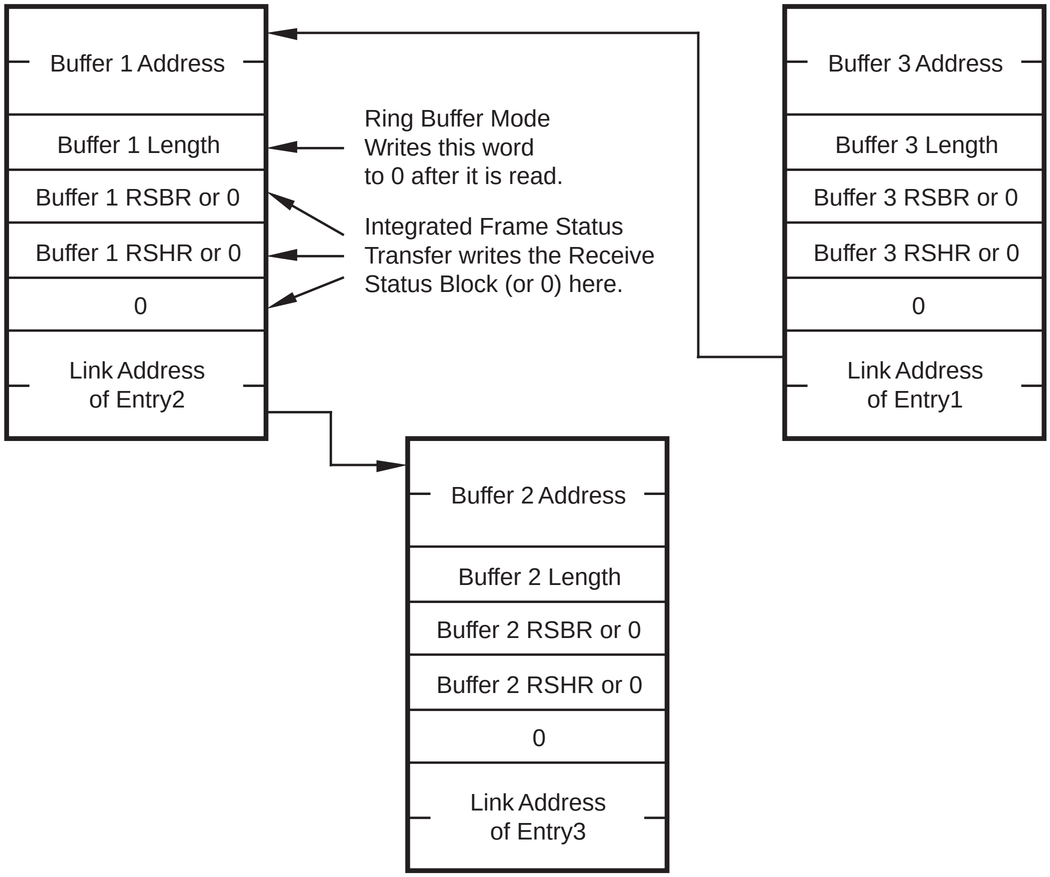



**Figure 4. Linked List Mode with Linked Frame Status Transfer and Ring Buffer Features**

Another method by which the DMA and serial channel work together is using the Transmit Character Counter to break a large block of data into a number of fixed length frames. For example, it is desired to transmit a large file which is located in several memory buffers as fixed length smaller frames. With the IUSC, the serial channel is programmed to send the end-of-frame sequence each time the set number of bytes is transmitted. Therefore, DMA transfers are not interrupted, nor is system response required to break the large file into frames.

The IUSC provides higher throughput than discrete serial and DMA chip solutions because discrete chips do not directly communicate with each other and, therefore, the status of one device must be read by the CPU and communicated to the other. This typically requires interrupts and the suspension of activity until status/control information is updated. This uses precious time and bus bandwidth, which can limit total throughput.

## DATA COMMUNICATIONS CAPABILITIES

The IUSC provides a full-duplex channel programmable for use in any common data communication protocol. The receiver and transmitter are completely independent and each is supported by a 32-byte deep FIFO and a 16-bit frame length counter. All modes allow optional even, odd, mark or space parity. Synchronous modes allow the choice of either of two 16-bit or a 32-bit CRC polynomials. Character length of up to 8 bits can be programmed for the receiver and transmitter independently. Error and status conditions are carried with the data in the receive FIFO to greatly reduce the CPU overhead required to send or receive a message, while key control parameters accompany transmit characters through the Tx FIFO. Interrupts can be individually armed 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 like receive character boundary, received sync, transmit character boundary and transmission complete can be sent to pins for use by external circuitry.

### Protocols

**Asynchronous Mode.** The receiver and transmitter 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 includes recovery logic following a framing error. The transmitter is capable of sending one, two, or anywhere in the range of 9/16th to two stop bits per character in 1/16 bit increments.

**Nine-Bit 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. In the transmitter, this bit is automatically inserted with the value that is FIFO'ed from the transmit data.

**Isochronous Mode.** Both transmitter and receiver operate on start-stop (async) data using a 1x clock. The transmitter sends 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 sends zero, one or two stop bits.

**HDLC Mode.** In this mode, the receiver recognizes flags, performs optional address matching, accommodates extended address fields, and performs zero deletion and CRC checking. The receiver is capable of receiving shared-zero flags, recognizes abort sequences and can receive arbitrary length frames. 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 automatically sends a 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 is programmed for the last character in the frame.

Frames terminated with an ABORT can be marked with a status bit on the preceding character in addition to the status interrupt that can be enabled. Abort is only detected in-frame and, therefore, eliminates false detection due to an idle line. The IUSC provides four choices (flag, all 1s, all 0s, or alternating 1s and 0s) of line preamble to condition the line before beginning data transmission. This feature is valuable to get the receiver DPLL in sync and as a flow control mechanism to slow down frame transmission without slowing down the clock or disabling the transmitter.

**HDLC Loop Mode.** This mode is available only in the transmitter and allows the IUSC to be used in an HDLC Loop configuration. In this mode, the receiver is programmed to operate in HDLC mode to allow the transmitter to echo received messages. Upon receipt of a particular bit pattern (actually a sequence of seven consecutive ones) the transmitter stops repeating data and inserts its own frame(s).

**802.3 Mode.** This mode implements the data format of IEEE 802.3 with a 16-bit address compare. In this mode, /DCD and /CTS are used to implement the carrier sense and collision detect interactions with the receiver and transmitter. Back-off timing must be provided externally.

**Monosync Mode.** In this mode, a single character is used for synchronization. The sync character can be either eight bits long or the same length as the data characters. The receiver can automatically strip sync characters from the received data stream. The transmitter is programmed to automatically send CRC on either an underrun or at the end of a programmed message length.

**Slaved Monosync Mode.** This mode is available only in the transmitter and allows the transmitter (operating just as though it were in monosync mode) to send data with its byte boundaries synchronized to those of the received data.

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

**Transparent Bisync 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 calculations without CPU intervention. The transmitter is programmed to send either SYN, DLE-SYN, CRC-SYN, or CRC-DLE-SYN upon underrun and automatically sends the closing DLE-SYN with optional CRC at the end of a programmed message length.

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

## Data Encoding

The IUSC is programmed to encode and decode the serial data in any of eight different ways (Figure 5). 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.** NRZB is inverted from NRZ.

**NRZI-Mark.** In NRZI-Mark, a 1 is represented by a transition at the beginning of a bit cell, i.e., 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 a bit cell, i.e., 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.

**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 are transitions at the beginning of the bit cell to set up the level required to make the correct center transition.

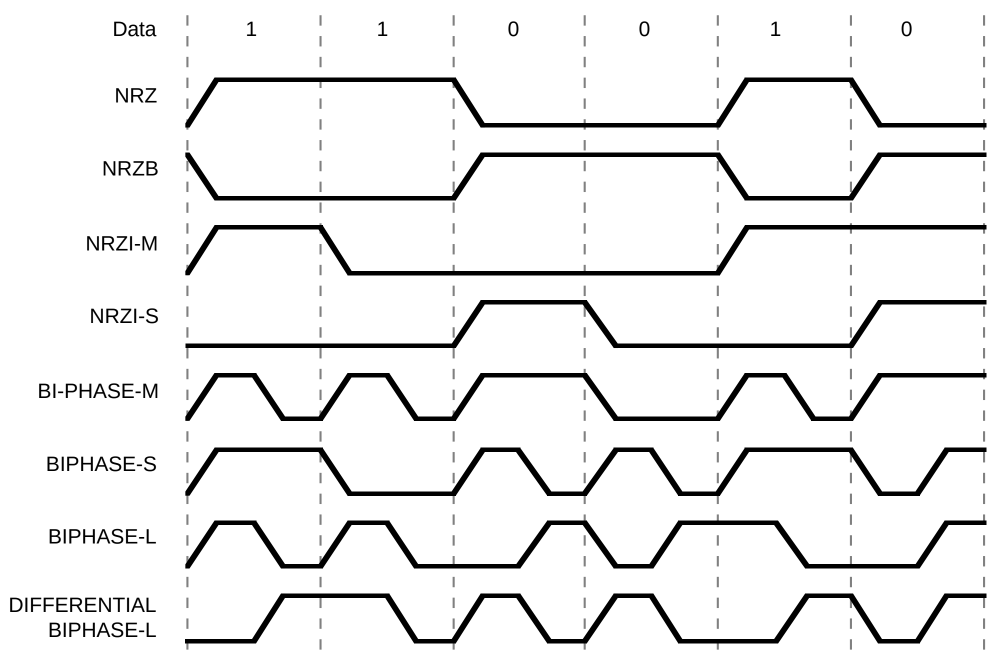



**Figure 5. Data Encoding**

## DATA COMMUNICATIONS CAPABILITIES (Continued)

### Character Counters

The IUSC contains separate 16-bit character counters for the receiver and transmitter. The receive character counter is set to a programmable starting value or automatically at the beginning of each received frame and can be reloaded under software control during a frame. The counter decrements with each receive character. At the end of the receive message the current value in the counter is automatically loaded into a four-deep FIFO. With the Receive Status Block (RSB) feature enabled, the counter value and the status (RCSR) can be automatically transferred to memory following the data. In array and linked list modes, the RSB can be transferred to the array or list entry for easy software access. This allows DMA transfer of data to proceed without CPU intervention at the end of a received frame, as the values in the FIFO allow the CPU to determine the status and length of each frame.

Similarly, the transmit character counter is loaded automatically at the beginning of each transmit frame and can be reloaded under software control during a frame. The counter is decremented with each write to the transmit FIFO. When the counter reaches zero, and that byte is sent, the transmitter automatically terminates the message in the appropriate fashion (usually by sending the CRC and the closing flag or sync character) without requiring CPU intervention. In linked list and array modes, the transmit character count and frame control word can be fetched from the linked list or array.

### Baud Rate Generators

The IUSC 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 cycle of its selected input clock, and the time constant can be 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 can be written at any time but the new value does 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 input to the Baud Rate Generator can be the /TxC pin, the /RxC pin, a PORT pin, or the output of either counter. The baud rate generator output frequency is related to the baud rate generator input clock frequency by the following formula:

Output frequency = Input frequency/time constant + 1.

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

The output of either Baud Rate Generator can be used as the transmit or receive clock, the reference clock input to the DPLL circuit, and/or can be output on the /RxC or /TxC pin.

### Digital Phase-Locked Loop

The IUSC contains a DPLL (Digital Phase-Locked Loop) 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 with the data stream, to construct a clock for the data. This clock can 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. While counting, the DPLL watches the incoming data stream for transitions. When a transition is detected, the DPLL may make 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

The IUSC contains two 5-bit counters, which are programmed to divide an input clock by 4, 8, 16 or 32. The outputs of these two counters are sent to the clock multiplexer. The counters can be used as prescalers for the Baud Rate Generators. They also provide a stable transmit clock from a common source when the DPLL is providing the receive clock. The PORT0 and PORT1 pins can be used as inputs to the counters.

## Clock Multiplexers

The clock multiplexer logic selects the receive and transmit clocks and optional outputs on the /RxC and/or /TxC pin(s). In the Z16C32, the PORT0 and PORT1 pins can be used directly as receive and transmit clocks, as well as being used as inputs to the counters.

## Time Slot Assigner

The IUSC is equipped with two Time Slot Assigners to support ISDN and Fractional T1 communications. There is one assigner for the receiver. Each time slot assigner selects one or more time slots within a frame, however, the selected time slots must be contiguous. The first selected time slot is programmable from slot 0 (the first slot) to slot 127 of the frame. The number of concatenated slots is programmable from 1 to 15 (total slots). The time of the first slot can be offset an integral number of clocks. This offset is a delay and is programmable from 0 (no offset) to 7 clocks in increments of one clock (one bit cell). This offset can be used to compensate for delays in frame sync detection logic.

## Test Modes

The IUSC 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 IUSC 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 IUSC.

## I/O Port

The Port pins are general-purpose I/O pins. They are used as additional modem control lines or other I/O functions. Each port bit is individually programmable as general-purpose input, as an output, or for a dedicated input or output function. This programming is done in the Port Control Register. Whether used as inputs or outputs, the port pins can be read at any time.

The dedicated functions of the port pins include Time Slot Assigner gate outputs, transmit complete output, clock inputs, receive sync output, or frame sync input.

The port pins capture edge transitions. Programming for the capture is done using the Port Latched/Unlatch command bits in the Port Status Register. Each port bit is individually controlled. The Latched/Unlatch bit is used as a status signal to indicate that a transition has occurred on the port pin and as a command to open the latches that capture this transition. Both rising edge and falling edge are detected. When a transition is detected, the latch closes, holding the post transition state of the input.

The Latched/Unlatch bit is held at 0 if no transitions occur on the port pin; this bit is set to a 1 when a rising edge or falling edge transition is detected, or immediately after the latch is opened if one or more transitions occurred while the latch was closed. Writing a 0 to the Latched/Unlatch bit has no effect on the latch. Writing a 1 to this bit resets the status bit and opens the latch. To use the port as an input without edge detection, a 1 would be written to the Latched/Unlatch bit to open the latch and then the Port Status Register would be read to obtain the current pin input status.

## PROGRAMMING

An Electronic Programmer's Manual (MS DOS based) and a Technical Manual are available to provide details about programming the IUSC. Also included are explanations and features of all registers in the IUSC.

The registers in the IUSC are programmed by the system to configure the channel. Before this can occur, the system must set up the bus interface by writing to the Bus Configuration Register (BCR). The BCR has no specific address and is only accessible after a hardware reset of the device. The first write to the IUSC, after a hardware reset, programs the BCR. From that time on other channel registers can be accessed. No specific address need be presented to the IUSC for the BCR write; the IUSC knows that the first write after a hardware reset is destined for the BCR.

In the multiplexed bus case, all registers are directly addressable through the address latched by /AS at the beginning of each bus cycle. The D//C pin is still used to directly access the receive and send data registers (RDR and TDR) with a multiplexed bus; if D//C is High, the address latched by /AS is ignored and an access of RDR or TDR is performed.

In the non-multiplexed bus case, the channel registers are accessed indirectly using the address pointer in the Channel Command/Address Register (CCAR). 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.

Two more points about the IUSC should be noted here. Channel Reset bit in the CCAR places the channel in the reset state. To exit this reset state either a word of all zeros is written to the CCAR (16-bit bus) or a byte of all zeros is written to the lower byte of the CCAR (8-bit bus). Secondly, after reset, the transmit and receive clocks are disabled. 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 Serial/DMA (S//D) pin is used to differentiate between the serial channel and the DMA registers. The DMA registers fall into three logic groupings; common registers that apply to both transmit and receive, transmit registers, and receive registers. The registers for DMA transmit functions and receive functions are symmetric and therefore, a single diagram is shown for each in the following pages. When addressing the DMA registers, the Data/Control (D//C) pin selects between the transmit and receive registers. For example, there is a DMA byte count register for transmit and receive (TBCR and RBCR) at address 10101 with S//D pin Low. The TBCR is selected with the D//C pin Low, and the RBCR is selected with the D//C pin High. The format of these two registers is shown in Figure 20.

The register addressing is shown in Table 2 and the table assumes that the BCR register bit 0 is set to 1. The A5-A1 column in the Table reflects the state of AD5-AD1, AD13-AD9, CCAR5-CCAR1 or DCAR5-DCAR1 as applicable. The bit assignments of the registers are shown in Figures 7 through 80. See the IUSC Technical Manual for details. The register addressing is shown in Table 2 and the bit assignments for the registers are shown in Figure 6.

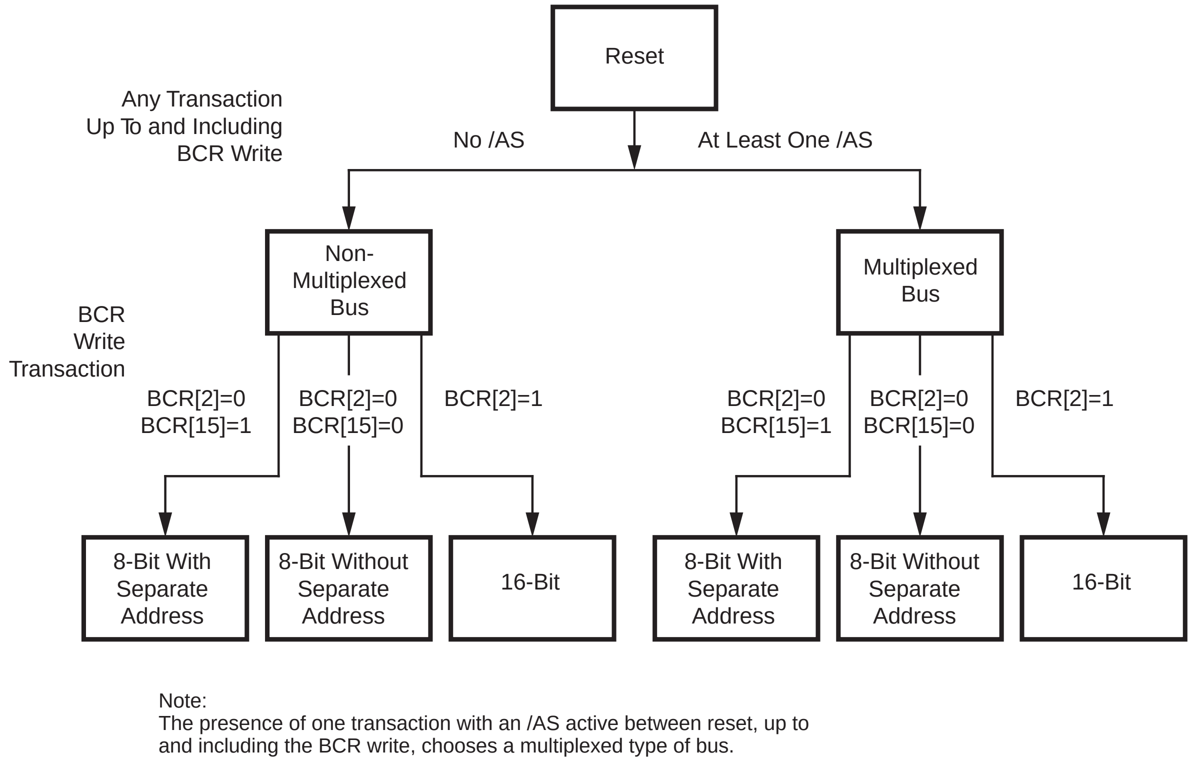



**Figure 6. BCR Reset Sequence and Bit Assignments**

## PROGRAMMING (Continued)

**Table 1. Register Address List**

| S//D | D//C | Address A5-A1 | Name | Description |
|---|---|---|---|---|
| 1 | 0 | 00000 | CCAR | Channel Command/Address Register |
| 1 | 0 | 00001 | CMR | Channel Mode Register |
| 1 | 0 | 00010 | CCSR | Channel Command/Status Register |
| 1 | 0 | 00011 | CCR | Channel Control Register |
| 1 | 0 | 00100 | PSR | Port Status Register |
| 1 | 0 | 00101 | PCR | Port Control Register |
| 1 | 0 | 00110 | TMDR | Test Mode Data Register |
| 1 | 0 | 00111 | TMCR | Test Mode Control Register |
| 1 | 0 | 01000 | CMCR | Clock Mode Control Register |
| 1 | 0 | 01001 | HCR | Hardware Configuration Register |
| 1 | 0 | 01010 | IVR | Interrupt Vector Register |
| 1 | 0 | 01011 | IOCR | I/O Control Register |
| 1 | 0 | 01100 | ICR | Interrupt Control Register |
| 1 | 0 | 01101 | DCCR | Daisy-Chain Control Register |
| 1 | 0 | 01110 | MISR | Misc. Interrupt Status Register |
| 1 | 0 | 01111 | SICR | Status Interrupt Control Register |
| 1 | 1 | XXXXX | RDR | Receive Data Register (Read Only) |
| 1 | 0 | 1X000 | RDR | Receive Data Register (Read Only) |
| 1 | 0 | 10001 | RMR | Receive Mode Register |
| 1 | 0 | 10010 | RCSR | Receive Command/Status Register |
| 1 | 0 | 10011 | RICR | Receive Interrupt Control Register |
| 1 | 0 | 10100 | RSR | Receive Sync Register |
| 1 | 0 | 10101 | RCLR | Receive Count Limit Register |
| 1 | 0 | 10110 | RCCR | Receive Character Count Register |
| 1 | 0 | 10111 | TC0R | Time Constant 0 Register |
| 1 | 1 | XXXXX | TDR | Transmit Data Register (Write Only) |
| 1 | 0 | 1X000 | TDR | Transmit Data Register (Write Only) |
| 1 | 0 | 11001 | TMR | Transmit Mode Register |
| 1 | 0 | 11010 | TCSR | Transmit Command/Status Register |
| 1 | 0 | 11011 | TICR | Transmit Interrupt Control Register |
| 1 | 0 | 11100 | TSR | Transmit Sync Register |
| 1 | 0 | 11101 | TCLR | Transmit Count Limit Register |
| 1 | 0 | 11110 | TCCR | Transmit Character Count Register |
| 1 | 0 | 11111 | TC1R | Time Constant 1 Register |

**Table 1. Register Address List** (Continued)

| S//D | D//C | Address A5-A1 | Name | Description |
|---|---|---|---|---|
| X | 0 | XXXXX | BCR | Bus Configuration Register |
| 0 | X | 00000 | DCAR | DMA Command/Address Register |
| 0 | 0 | 00001 | TDCMR | Transmit DMA Channel Mode Register |
| 0 | X | 00011 | DCR | DMA Control Register |
| 0 | X | 00100 | DACR | DMA Array Count Register |
| 0 | X | 01001 | BDCR | Burst Dwell Control Register |
| 0 | X | 01010 | DIVR | DMA Interrupt Vector Register |
| 0 | X | 01100 | DICR | DMA Interrupt Control Register |
| 0 | X | 01101 | CDIR | Clear DMA Interrupt Register |
| 0 | X | 01110 | SDIR | Set DMA Interrupt Register |
| 0 | 0 | 01111 | TDIAR | Transmit DMA Interrupt Arm |
| 0 | 0 | 10101 | TBCR | Transmit Byte Count Register |
| 0 | 0 | 10110 | TARL | Transmit Address Register (Lower) |
| 0 | 0 | 10111 | TARU | Transmit Address Register (Upper) |
| 0 | 0 | 11101 | NTBCR | Next Transmit Byte Count Register |
| 0 | 0 | 11110 | NTARL | Next Transmit Address Register (Lower) |
| 0 | 0 | 11111 | NTARU | Next Transmit Address Register (Upper) |
| 0 | 1 | 00001 | RDMR | Receive DMA Mode Register |
| 0 | 1 | 01111 | RDIAR | Receive DMA Interrupt Arm |
| 0 | 1 | 10101 | RBCR | Receive DMA Byte Count Register |
| 0 | 1 | 10110 | RARL | Receive Address Register (Lower) |
| 0 | 1 | 10111 | RARU | Receive Address Register (Upper) |
| 0 | 1 | 11101 | NRBCR | Next Receive Byte Count Register |
| 0 | 1 | 11110 | NRARL | Next Receive Address Register (Lower) |
| 0 | 1 | 11111 | NRARU | Next Receive Address Register (Upper) |

## REGISTER DESCRIPTION

This section describes the function of the various bits in the registers of the device. Throughout this section the following conventions are discussed:

Control bits are written and read by the CPU and are not modified by the device. Command bits are written by the CPU to initiate an action in the device and are read as zeros. Status bits are controlled by the device and are read to check device status. Any writes to status bits are ignored by the device. Command/Status bits are controlled by both the device and the CPU. They may be written and read by the CPU and may also be modified by the device.

Reserved bits are not used in this implementation of the device and may or may not be physically present in the device. Any reserved bits that are physically present are readable and writable, but reserved bits that are not present are always read as zeros. To ensure compatibility with future versions of the device, reserved bits should always be written with zeros. Reserved commands should not be used for the same reason.

First, the DMA registers unique to the IUSC are described in the following pages (Figures 7-16) and then the serial channel registers are described (Figures 17-80).

Address: 00000 (Shared) *

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

- D0: Upper//Lower Byte Select (WO)
- D1: Address 0 (WO)
- D2: Address 1 (WO)
- D3: Address 2 (WO)
- D4: Address 3 (WO)
- D5: Address 4 (WO)
- D6: Byte//Word Access (WO)
- D7: Pointer Channel Select (WO)
- D8: Master Bus Request Enable

Channel Select (WO)**

| D11 | D10 | D9 | |
|---|---|---|---|
| 0 | 0 | 0 | Tx Channel |
| 0 | 0 | 1 | Rx Channel |
| 0 | 1 | 0 | Reserved |
| 0 | 1 | 1 | Reserved |
| 1 | 0 | 0 | Reserved |
| 1 | 0 | 1 | Reserved |
| 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | Reserved |

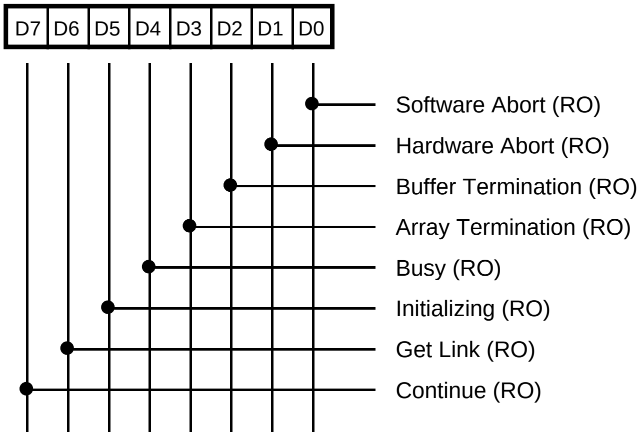
Channel Command

| D15 | D14 | D13 | D12 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Null Command |
| 0 | 0 | 0 | 1 | Reset This Channel |
| 0 | 0 | 1 | 0 | Start This Channel |
| 0 | 0 | 1 | 1 | Start/Continue This Channel |
| 0 | 1 | 0 | 0 | Pause This Channel |
| 0 | 1 | 0 | 1 | Abort This Channel |
| 0 | 1 | 1 | 0 | Reserved |
| 0 | 1 | 1 | 1 | Start/Init This Channel |
| 1 | 0 | 0 | 0 | Reset Highest IUS |
| 1 | 0 | 0 | 1 | Reset All Channels |
| 1 | 0 | 1 | 0 | Start All Channels |
| 1 | 0 | 1 | 1 | Start/Continue All Channels |
| 1 | 1 | 0 | 0 | Pause All Channels |
| 1 | 1 | 0 | 1 | Abort All Channels |
| 1 | 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | 1 | Start/Init All Channels |

Notes:
* (Shared) means, shared between DMA Channels

** (WO) means Write Only

**Figure 7. DMA Command/Address Register**




| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Command |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Reset |
| X | X | X | 1 | X | X | X | X | Start |
| 1 | X | X | 1 | X | X | X | X | Start/Continue (Buffered Mode) |
| X | X | X | 1 | X | X | X | X | Start/Continue (Other Than Buffered Mode) |
| X | X | X | 0 | X | X | X | X | Pause |
| X | X | X | 0 | X | X | X | 1 | Abort |
| X | X | 1 | 1 | X | X | X | X | Start/Init (Array or Linked Array Modes) |
| X | X | X | 1 | X | X | X | X | Start/Init (Normal or Buffered Modes) |

Notes:
X = Previous Value
(RO) = Read Only

**Figure 8. Affect of Commands on Status Bits**

## CONTROL REGISTERS (Continued)

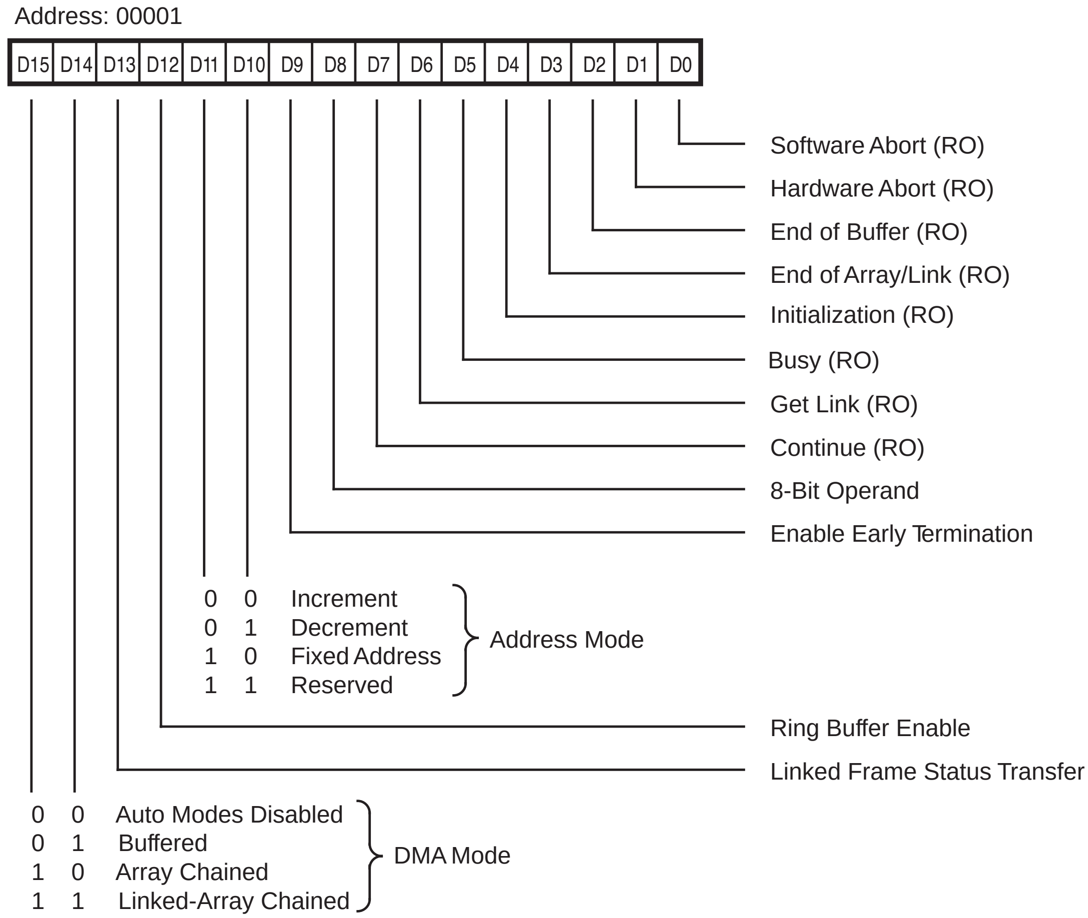
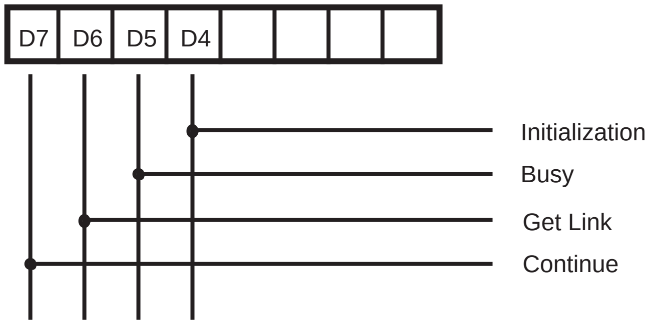



**Figure 9. Tx/Rx DMA Mode Register (TDMR) (RDMR)**




| D7 | D6 | D5 | D4 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Channel Disabled |
| 0 | 0 | 1 | 0 | Channel Enabled |
| X | X | X | 1 | Not Possible |
| X | 1 | X | X | Not Possible |
| 1 | X | X | X | Not Possible |

Normal Mode

| D7 | D6 | D5 | D4 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Channel Disabled, Base Registers Invalid |
| 0 | 0 | 1 | 0 | Channel Enabled, Base Registers Invalid |
| 1 | 0 | 0 | 0 | Channel Disabled, Base Registers Valid |
| 1 | 0 | 1 | 0 | Channel Enabled, Base Registers Valid |
| X | X | X | 1 | Not Possible |
| X | 1 | X | X | Not Possible |

Buffered Mode

| D7 | D6 | D5 | D4 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Channel Disabled, Data Transfer Phase |
| 0 | 0 | 1 | 0 | Channel Enabled, Data Transfer Phase |
| 0 | 0 | 0 | 1 | Channel Disabled, Array Transfer Phase |
| 0 | 0 | 1 | 1 | Channel Enabled, Array Transfer Phase |
| X | 1 | X | X | Not Possible |
| 1 | X | X | X | Not Possible |

Array-Chained Mode

| D7 | D6 | D5 | D4 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Channel Disabled, Data Transfer Phase |
| 0 | 0 | 1 | 0 | Channel Enabled, Data Transfer Phase |
| 0 | 0 | 0 | 1 | Channel Disabled, Array Transfer Phase |
| 0 | 0 | 1 | 1 | Channel Enabled, Array Transfer Phase |
| 0 | 1 | X | 0 | Not Possible |
| 0 | 1 | 0 | 1 | Channel Disabled, Link Transfer Phase |
| 0 | 1 | 1 | 1 | Channel Enabled, Link Transfer Phase |
| 1 | X | X | X | Not Possible |

Linked Array-Chained Mode

**Figure 10. Status Bit Combinations**

## CONTROL REGISTERS (Continued)

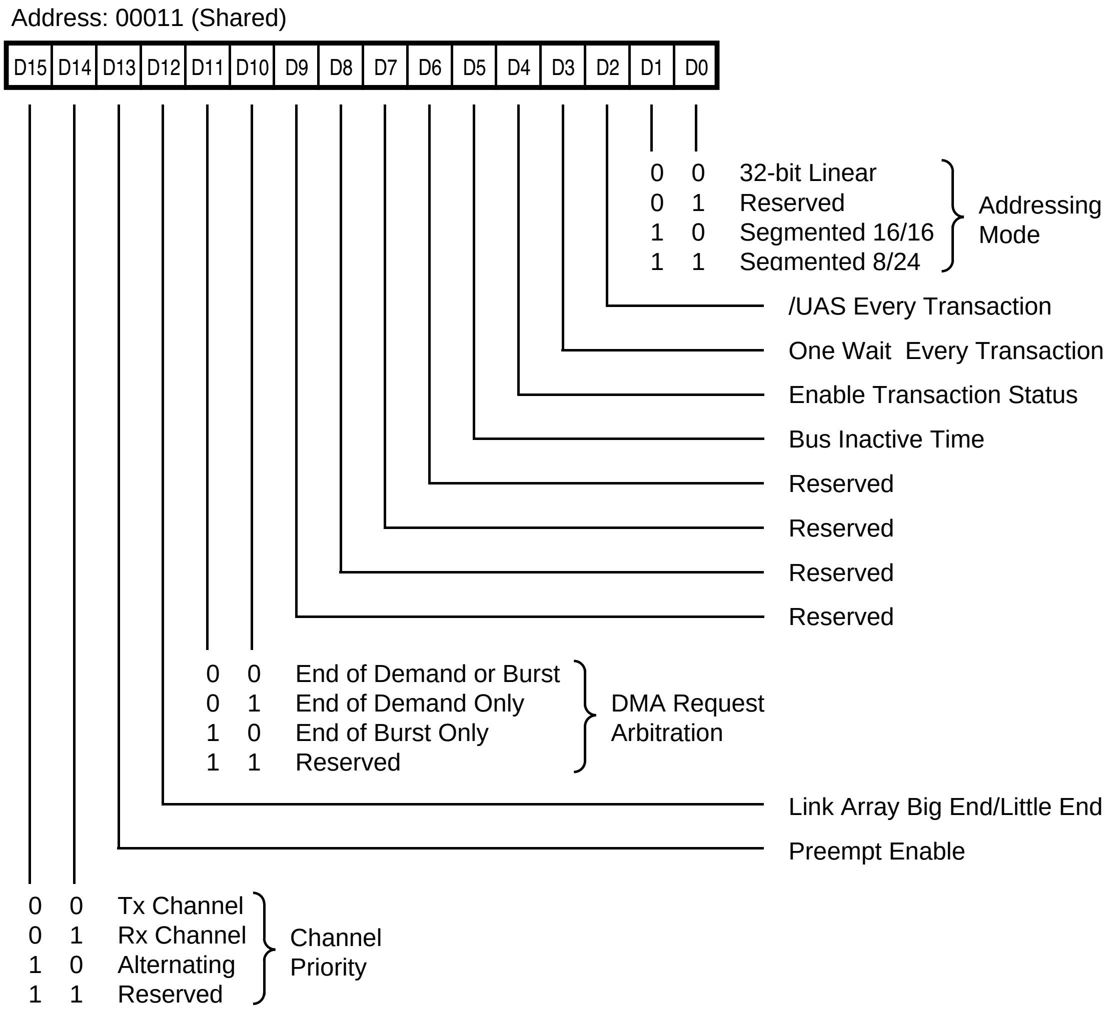



**Figure 11. DMA Control Register (DCR)**

Big End Array (16-Bit bus)

| | AD15 | | | | | | | | | | | | | | | AD0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Address n | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Address n+2 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |

Little End Array (16-Bit bus)

| | AD15 | | | | | | | | | | | | | | | AD0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Address n | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| Address n+2 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |

Big End Array (8-Bit bus)

| | AD7 | | | | | | | AD0 |
|---|---|---|---|---|---|---|---|---|
| Address n | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| Address n+1 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Address n+2 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 |
| Address n+3 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |

Little End Array (8-Bit bus)

| | AD7 | | | | | | | AD0 |
|---|---|---|---|---|---|---|---|---|
| Address n | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| Address n+1 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 |
| Address n+2 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Address n+3 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |

**Figure 12. Array-Chained Bit Ordering**

**Note:**
Bit 12 in DCR is used to control the byte ordering of addresses and counts stored in memory in the Array and Linked Array Modes. The above figure shows the two cases for both bus bandwidths.

## CONTROL REGISTERS (Continued)

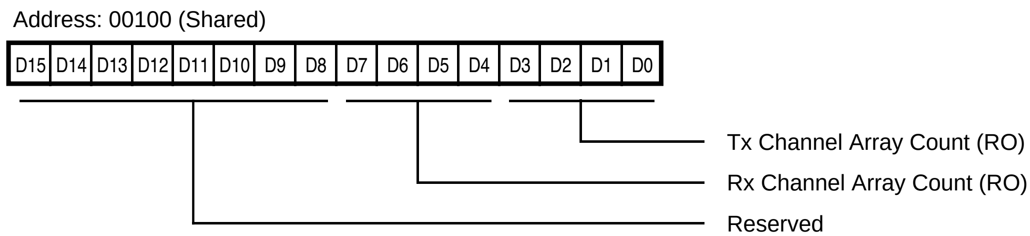



**Figure 13a. DMA Array Count Register (DACR)**

Array-Chained:

| Channel Array Count 3 | Channel Array Count 2 | Channel Array Count 1 | Channel Array Count 0 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Array Operation Not Started |
| 0 | 0 | 0 | 1 | Fetched Last Byte of Array |
| 0 | 0 | 1 | 0 | Fetched Fifth Byte of Array |
| 0 | 0 | 1 | 1 | Fetched Fourth Byte of Array |
| 0 | 1 | 0 | 0 | Fetched Third Byte of Array |
| 0 | 1 | 0 | 1 | Fetched Second Byte of Array |
| 0 | 1 | 1 | 0 | Fetched First Byte of Array |
| 0 | 1 | 1 | 1 | Fetched Last Byte of Buffer |

Linked Array-Chained:

| Getlink (in DCMR) | Channel Array Count 3 | Channel Array Count 2 | Channel Array Count 1 | Channel Array Count 0 | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | Array Operation Not Started |
| 0 | 0 | 0 | 0 | 1 | Fetched Last Byte of Array |
| 0 | 0 | 0 | 1 | 0 | Fetched Fifth Byte of Array |
| 0 | 0 | 0 | 1 | 1 | Fetched Fourth Byte of Array |
| 0 | 0 | 1 | 0 | 0 | Fetched Third Byte of Array |
| 0 | 0 | 1 | 0 | 1 | Fetched Second Byte of Array |
| 0 | 0 | 1 | 1 | 0 | Fetched First Byte of Array |
| 0 | 0 | 1 | 1 | 1 | Fetched Last Byte of Linked Address |
| 1 | 0 | 0 | 0 | 0 | Invalid |
| 1 | 0 | 0 | 0 | 1 | Invalid |
| 1 | 0 | 0 | 1 | 0 | Invalid |
| 1 | 0 | 0 | 1 | 1 | Invalid |
| 1 | 0 | 1 | 0 | 0 | Fetched Third Byte of Linked Address |
| 1 | 0 | 1 | 0 | 1 | Fetched Second Byte of Linked Address |
| 1 | 0 | 1 | 1 | 0 | Fetched First Byte of Linked Address |
| 1 | 0 | 1 | 1 | 1 | Fetched Last Byte of Buffer |

**Figure 13b. Channel Array Count Bit Combinations**

**Note:** See the Z16C32 Technical Manual for the appropriate table with Linked Status Transfer feature enabled.

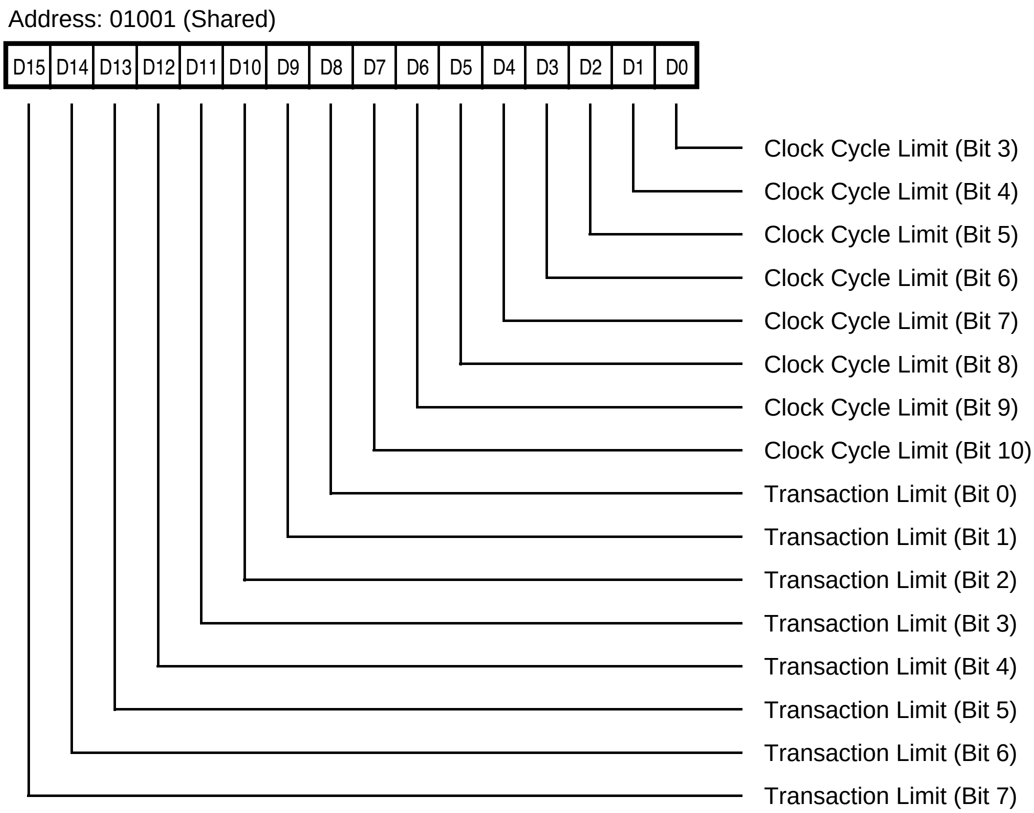



**Figure 14. Burst Dwell Control Register (BDCR)**

**Notes:**
BDCR Controls the amount of time that DMA may remain bus master.

Bits 15 through 8 are used to select a limit for the number of DMA transfers on the Bus while the DMA is bus master. This limit is a binary number, a value of zero disables the transaction limit function.

Bits 7 through 0 are used to select a limit for the number of clock cycles that the DMA may remain on the bus as bus master.

Bus transaction will always complete, even if the clock cycle limit is exceeded during the bus cycle, and even if the cycle is extended by external hardware signalling through /WAIT//RDY.

## CONTROL REGISTERS (Continued)

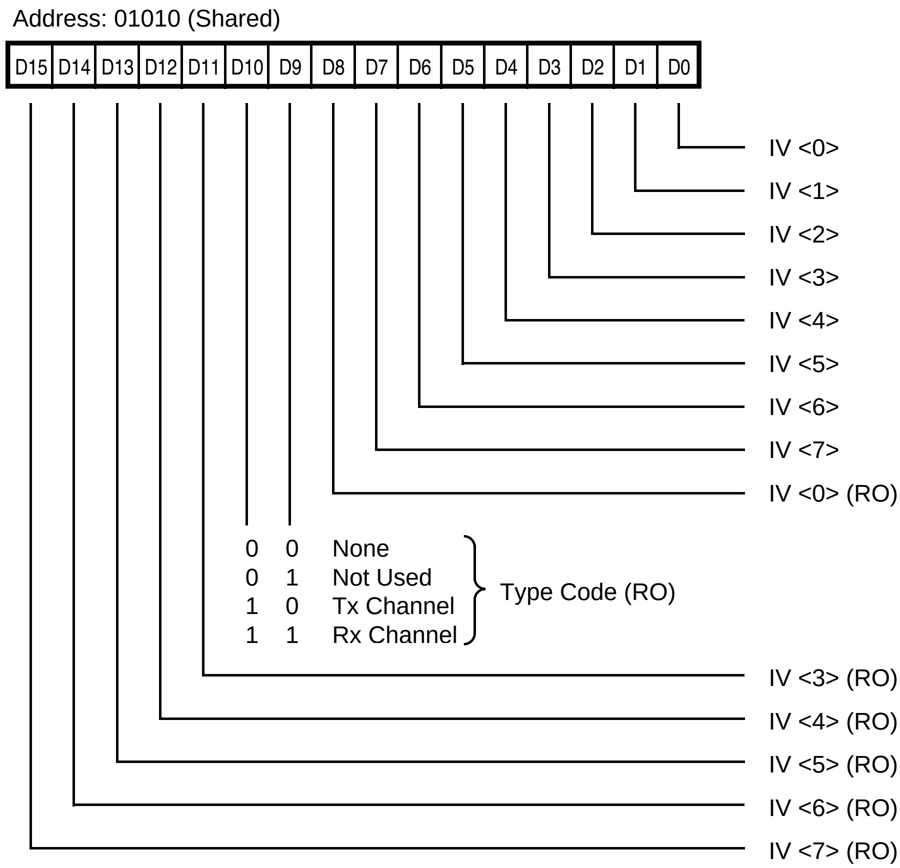



**Figure 15. DMA Interrupt Vector Register (DIVR)**

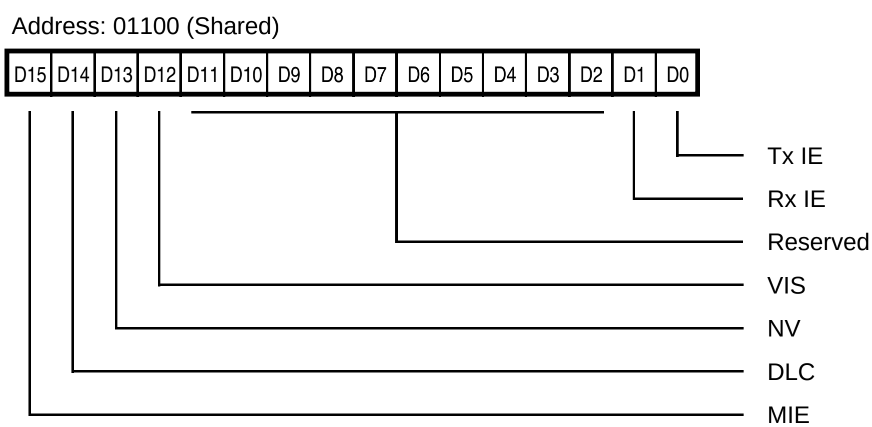



**Figure 16. DMA Interrupt Control Register (DICR)**

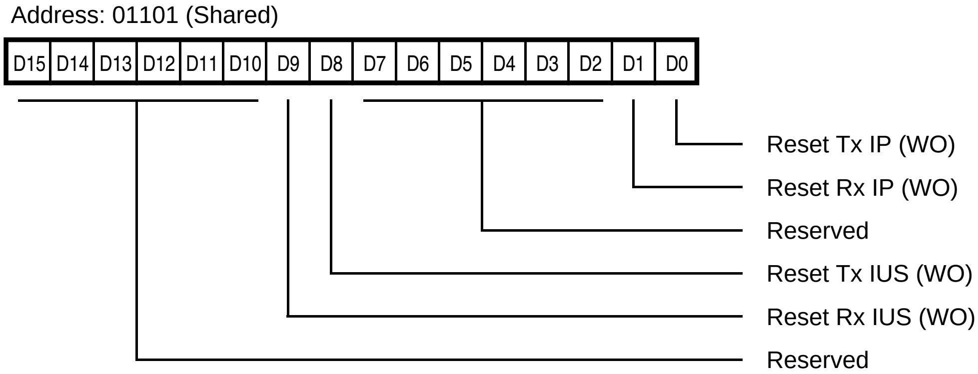



**Figure 17. Clear DMA Interrupt Register (CDIR)**

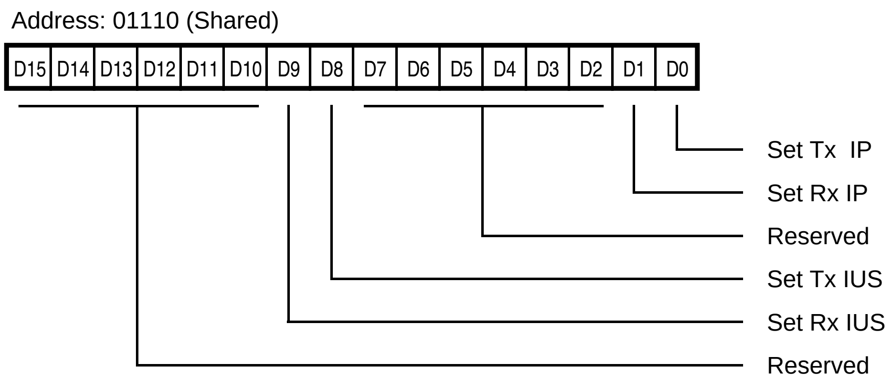



**Figure 18. Set DMA Interrupt Register (SDIR)**

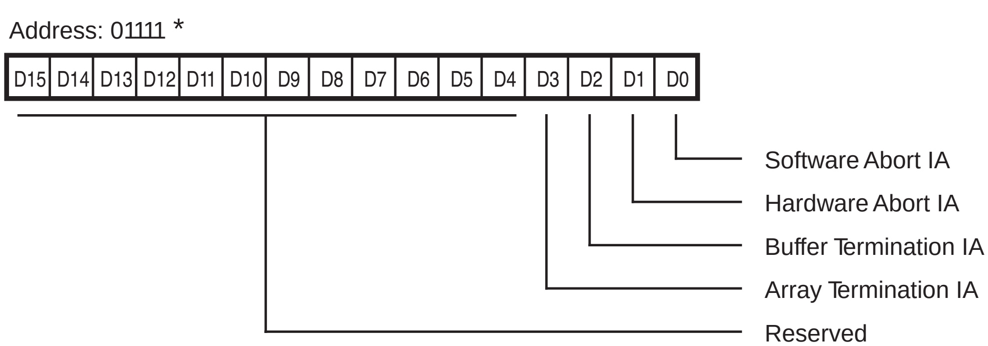



**Figure 19. Tx/Rx DMA Interrupt Arm (TDIAR)/(RDIAR)**

**Notes:**
* The format of this register is the same for the receiver and transmitter. The transmit register is accessed by addressing it with the D//C pin Low (0). The receive register is accessed by addressing it with the D//C pin High (1). This applies to Figures 19 through 25.

## CONTROL REGISTERS (Continued)

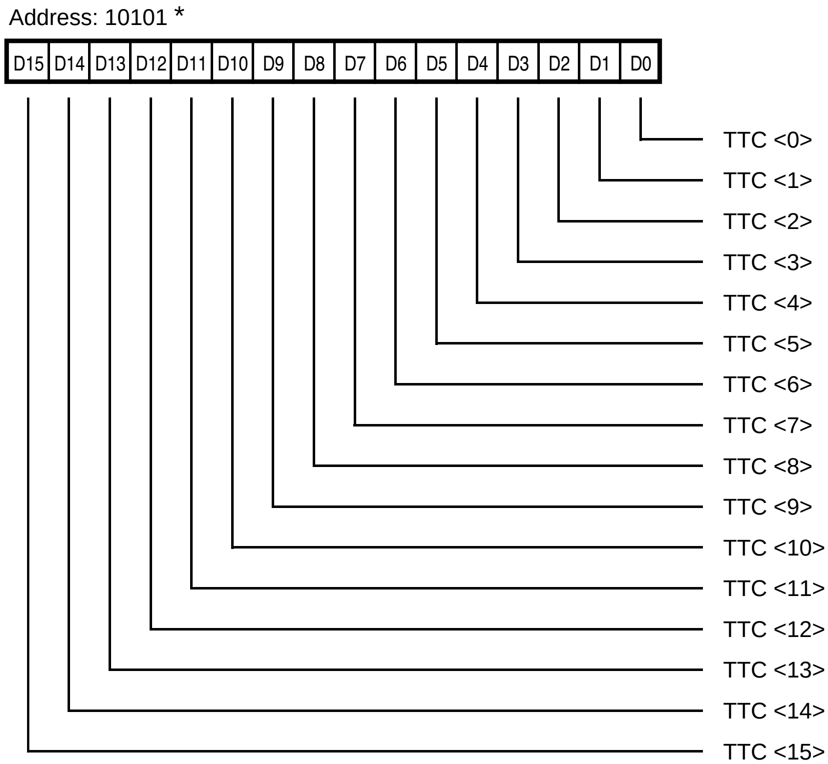



**Figure 20. Tx/Rx Byte Count Register (TBCR)/(RBCR)**

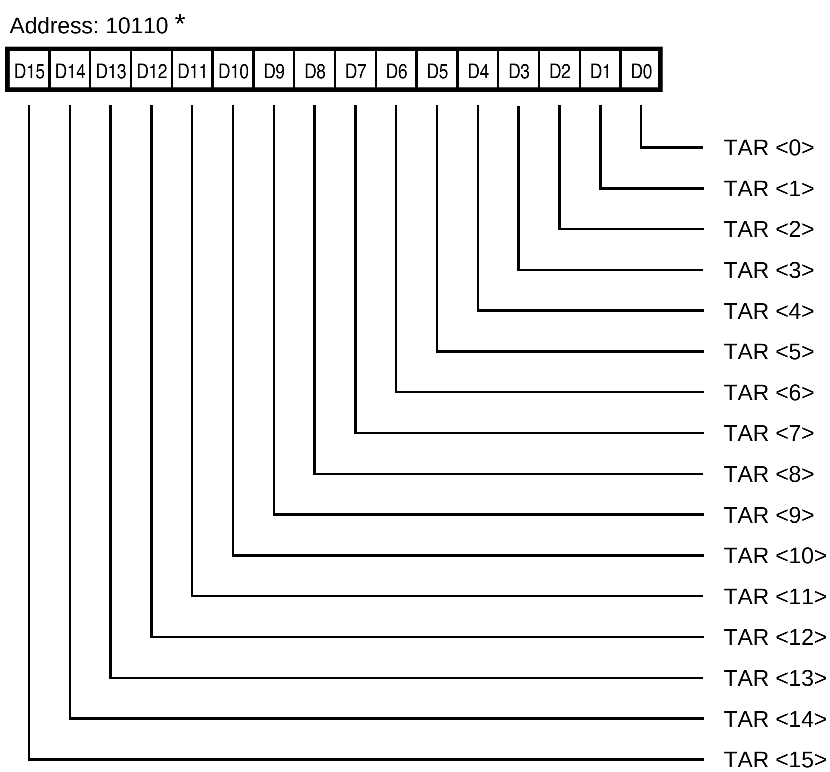



**Figure 21. Tx/Rx Address Register (lower) (TARL)/(RARL)**

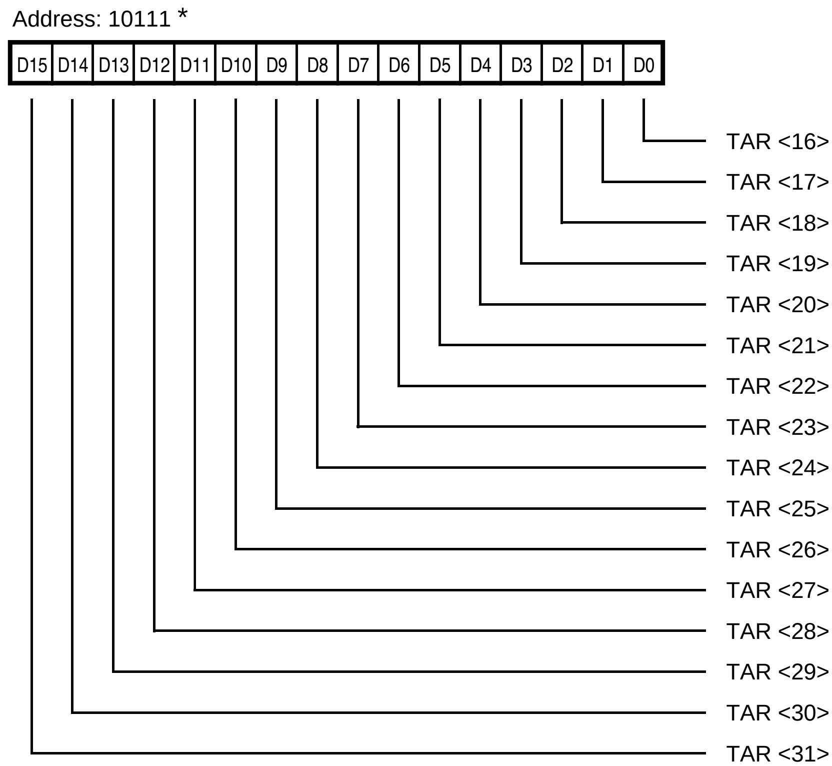



**Figure 22. Tx/Rx Address Register (Upper) (TARU)/(RARU)**

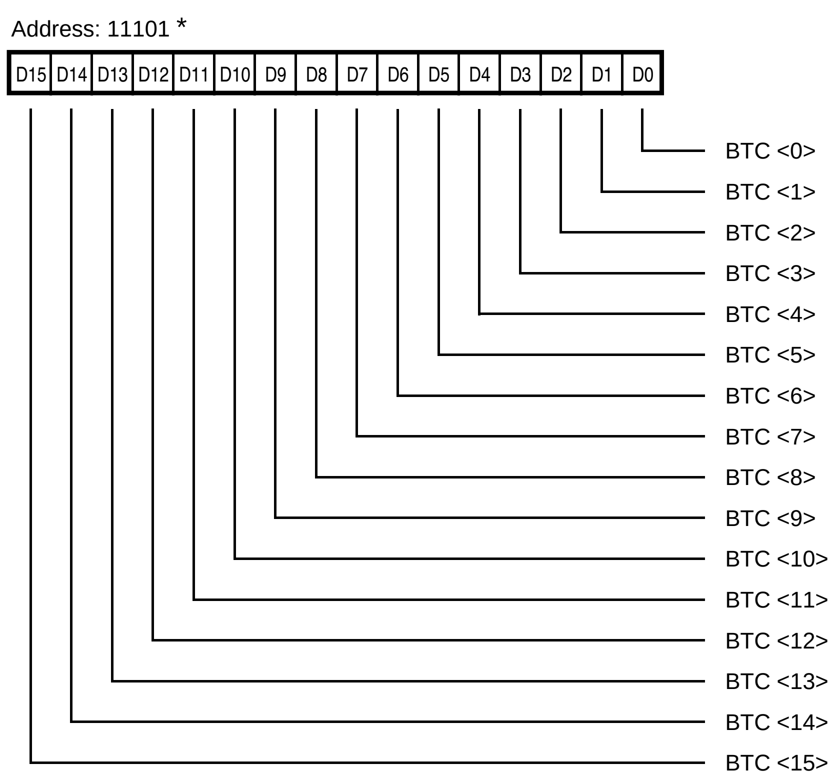



**Figure 23. Next Tx/Rx Byte Counter Register (NTBCR)/(RTBCR)**

## **CONTROL REGISTERS** (Continued)

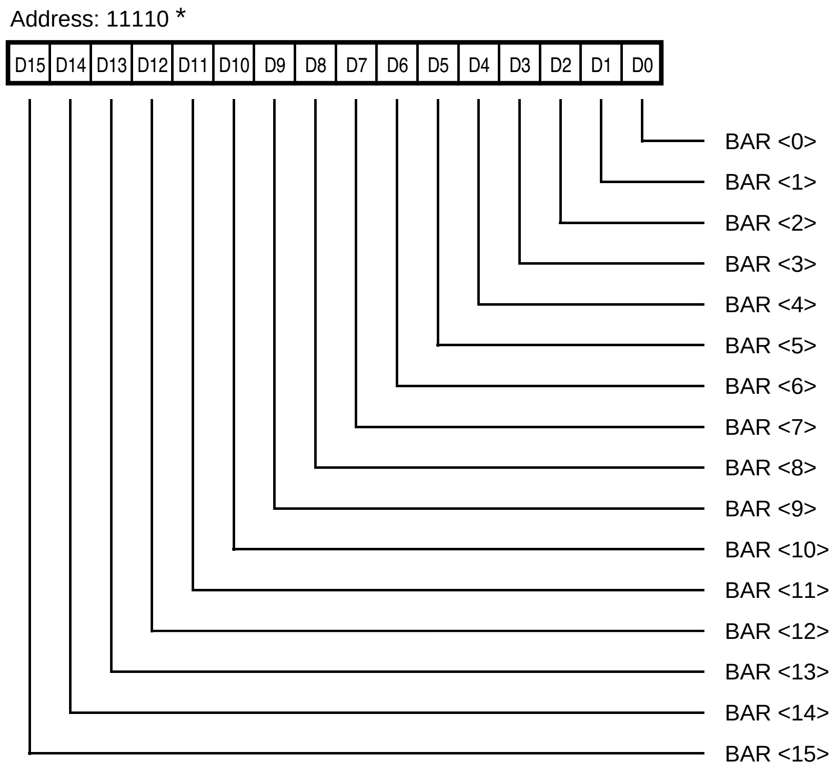



**Figure 24. Next Tx/Rx Address Register (Lower) (NTARL)/(RTARL)**

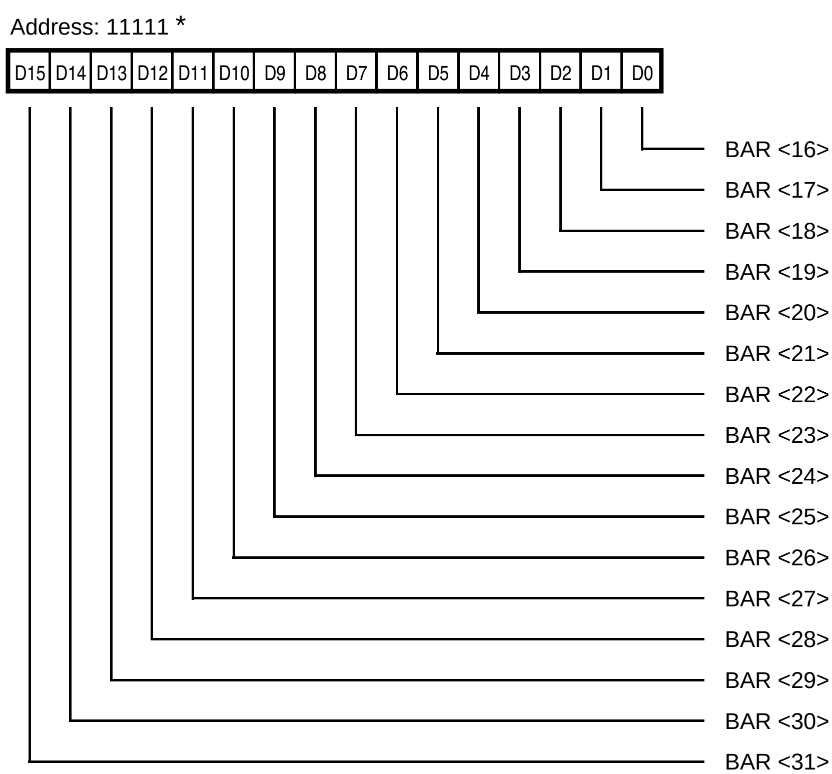



**Figure 25. Next Tx/Rx Address Register (Upper) (NTARU)/(RTARU)**

| | | AD15 | AD0 |
|---|---|---|---|
| Base Address | Buffer #1 | AD<31-24> | AD<23-16> |
| Base Address + 2 | | AD<15-8> | AD<7-0> |
| Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| Base Address + 6 | Buffer #2 | AD<31-24> | AD<23-16> |
| Base Address + 8 | | AD<15-8> | AD<7-0> |
| Base Address + 10 | | CNT<15-8> | CNT<7-0> |
| Base Address + 12 | Buffer #3 | AD<31-24> | AD<23-16> |
| Base Address + 14 | | AD<15-8> | AD<7-0> |
| Base Address + 16 | | CNT<15-8> | CNT<7-0> |

| | | | |
|---|---|---|---|
| Last Base Address | Dummy | Ignored | Ignored |
| Last Base Address + 2 | | Ignored | Ignored |
| Last Base Address + 4 | | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |
| Base Address Register After Termination | | | |

**Figure 26a. Array-Chained, 16-Bit Bus, Big End Array**

**Note:**
The addition of frame status/control information in the array with Linked Frame Status Transfer Enabled is similar for Big and Little End Array. See Figure 26b.

## CONTROL REGISTERS (Continued)

| | | AD15 | AD0 |
|---|---|---|---|
| Base Address | Buffer #1 | AD<31-24> | AD<23-16> |
| Base Address + 2 | | AD<15-8> | AD<7-0> |
| Base Address + 4 | | CNT <15-8> | CNT <7-0> |
| Base Address + 6 | | RSB/TCB <15-8> | RSB/TCB <7-0> |
| Base Address + 8 | | RCHR/TCLR <15-8> | RCHR/TCLR <7-0> |
| Base Address + 10 | | 0 | 0 |
| Base Address + 12 | Buffer #2 | AD<31-24> | AD<23-16> |
| Base Address + 14 | | AD<15-8> | AD<7-0> |
| Base Address + 16 | | CNT <15-8 > | CNT <7-0> |
| Base Address + 18 | | RSB/TCB <15-8> | RSB/TCB <7-0> |
| Base Address + 20 | | RCHR/TCLR <15-8> | RCHR/TCLR <7-0> |
| Base Address + 22 | | 0 | 0 |
| Base Address + 24 | Buffer #3 | AD<31-24> | AD<23-16> |
| Base Address + 26 | | AD<15-8> | AD<7-0> |
| Base Address + 28 | | CNT <15-8> | CNT <7-0> |
| Base Address + 30 | | RSB/TCB <15-8> | RSB/TCB <7-0> |
| Base Address + 32 | | RCHR/TCLR <15-8> | RCHR/TCLR <7-0> |
| Base Address + 34 | | 0 | 0 |

| | | | |
|---|---|---|---|
| Last Base Address | Dummy | Ignored | Ignored |
| Last Base Address + 2 | | Ignored | Ignored |
| Last Base Address + 4 | | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |
| Base Address Register After Termination | | | |

**Figure 26b. Array-Chained, 16-Bit Bus, Big End Array Linked Frame Status Transfer Enabled**

| | | AD15 | AD0 |
|---|---|---|---|
| Base Address | Buffer #1 | AD<15-8> | AD<7-0> |
| Base Address + 2 | | AD<31-24> | AD<23-16> |
| Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| Base Address + 6 | Buffer #2 | AD<15-8> | AD<7-0> |
| Base Address + 8 | | AD<31-24> | AD<23-16> |
| Base Address + 10 | | CNT<15-8> | CNT<7-0> |
| Base Address + 12 | Buffer #3 | AD<15-8> | AD<7-0> |
| Base Address + 14 | | AD<31-24> | AD<23-16> |
| Base Address + 16 | | CNT<15-8> | CNT<7-0> |

| | | | |
|---|---|---|---|
| Last Base Address | Dummy | Ignored | Ignored |
| Last Base Address + 2 | | Ignored | Ignored |
| Last Base Address + 4 | | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |
| Base Address Register After Termination | | | |

**Figure 27. Array-Chained, 16-Bit Bus, Little End Array**

## CONTROL REGISTERS (Continued)

| | | AD7 … AD0 |
|---|---|---|
| Base Address | Buffer #1 | AD<31-24> |
| Base Address + 1 | | AD<23-16> |
| Base Address + 2 | | AD<15-8> |
| Base Address + 3 | | AD<7-0> |
| Base Address + 4 | | CNT<15-8> |
| Base Address + 5 | | CNT<7-0> |
| Base Address + 6 | Buffer #2 | AD<31-24> |
| Base Address + 7 | | AD<23-16> |
| Base Address + 8 | | AD<15-8> |
| Base Address + 9 | | AD<7-0> |
| Base Address + 10 | | CNT<15-8> |
| Base Address + 11 | | CNT<7-0> |

| | | |
|---|---|---|
| Last Base Address | Dummy | Ignored |
| Last Base Address + 1 | | Ignored |
| Last Base Address + 2 | | Ignored |
| Last Base Address + 3 | | Ignored |
| Last Base Address + 4 | | 0 0 0 0 0 0 0 0 |
| Last Base Address + 5 | | 0 0 0 0 0 0 0 0 |
| Base Address Register After Termination | | |

**Figure 28a. Array-Chained, 8-Bit Bus, Big End Array**

**Note:**
The addition of frame status/control information in the array with Linked Frame Status Transfer Enabled is similar for Big and Little End Array. See Figure 28b.

| | | AD7 AD0 |
|---|---|---|
| Base Address | Buffer #1 | AD<31-24> |
| Base Address + 1 | | AD<23-16> |
| Base Address + 2 | | AD<15-8> |
| Base Address + 3 | | AD<7-0> |
| Base Address + 4 | | CNT<15-8> |
| Base Address + 5 | | CNT<7-0> |
| Base Address + 6 | | RSB/TCB <15-8> |
| Base Address + 7 | | RSB/TCB <7-0> |
| Base Address + 8 | | RCHR/TCLR <15-8> |
| Base Address + 9 | | RSHR/TCLR <7-0> |
| Base Address + 10 | | 0 |
| Base Address + 11 | | 0 |
| Base Address + 12 | Buffer #2 | AD<31-24> |
| Base Address + 13 | | AD<23-16> |
| Base Address + 14 | | AD<15-8> |
| Base Address + 15 | | AD<7-0> |
| Base Address + 16 | | CNT<15-8> |
| Base Address + 17 | | CNT<7-0> |
| Base Address + 18 | | RSB/TCB <15-8> |
| Base Address + 19 | | RSB/TCB <7-0> |
| Base Address + 20 | | RCHR/TCLR <15-8> |
| Base Address + 21 | | RSHR/TCLR <7-0> |
| Base Address + 22 | | 0 |
| Base Address + 23 | | 0 |
| Last Base Address | Dummy | Ignored |
| Last Base Address + 1 | | Ignored |
| Last Base Address + 2 | | Ignored |
| Last Base Address + 3 | | Ignored |
| Last Base Address + 4 | | 0 0 0 0 0 0 0 0 |
| Last Base Address + 5 | | 0 0 0 0 0 0 0 0 |
| Base Address Register After Termination | | |

**Figure 28b. Array-Chained, 8-Bit Bus, Big End Array, Linked Frame Status Transfer Enabled**

## CONTROL REGISTERS (Continued)

| | | AD7 … AD0 |
|---|---|---|
| Base Address | Buffer #1 | AD<7-0> |
| Base Address + 1 | | AD<15-8> |
| Base Address + 2 | | AD<23-16> |
| Base Address + 3 | | AD<31-24> |
| Base Address + 4 | | CNT<7-0> |
| Base Address + 5 | | CNT<15-8> |
| Base Address + 6 | Buffer #2 | AD<7-0> |
| Base Address + 7 | | AD<15-8> |
| Base Address + 8 | | AD<23-16> |
| Base Address + 9 | | AD<31-24> |
| Base Address + 10 | | CNT<7-0> |
| Base Address + 11 | | CNT<15-8> |

| | | |
|---|---|---|
| Last Base Address | Dummy | Ignored |
| Last Base Address + 1 | | Ignored |
| Last Base Address + 2 | | Ignored |
| Last Base Address + 3 | | Ignored |
| Last Base Address + 4 | | 0 0 0 0 0 0 0 0 |
| Last Base Address + 5 | | 0 0 0 0 0 0 0 0 |
| Base Address Register After Termination | | |

**Figure 29. Array-Chained, 8-Bit Bus, Little End Array**

| | | AD15 | AD0 |
|---|---|---|---|
| Base Address | Buffer #1 | AD<31-24> | AD<23-16> |
| Base Address + 2 | | AD<15-8> | AD<7-0> |
| Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| Base Address + 6 | Base #2 | AD<31-24> | AD<23-16> |
| Base Address + 8 | | AD<15-8> | AD<7-0> |
| #2 Base Address | Buffer #2 | AD<31-24> | AD<23-16> |
| #2 Base Address + 2 | | AD<15-8> | AD<7-0> |
| #2 Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| #2 Base Address + 6 | Base #3 | AD<31-24> | AD<23-16> |
| #2 Base Address + 8 | | AD<15-8> | AD<7-0> |
| #3 Base Address | Buffer #3 | AD<31-24> | AD<23-16> |
| #3 Base Address + 2 | | AD<15-8> | AD<7-0> |
| #3 Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| #n - 1 Base Address + 6 | Base #n | AD<31-24> | AD<23-16> |
| #n - 1 Base Address + 8 | | AD<15-8> | AD<7-0> |
| #n Base Address | Buffer #n | Ignored | Ignored |
| #n Base Address + 2 | | Ignored | Ignored |
| #n Base Address + 4 | | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |

**Figure 30a. Linked Array-Chained, 16-Bit Bus, Big End Array**

**Note:**
The addition of frame status/control information in the array with Linked Frame Status Transfer Enabled is similar for Big and Little End Array. See Figure 30b.

## CONTROL REGISTERS (Continued)

| | | AD15 | AD0 |
|---|---|---|---|
| Base Address | Buffer #1 | AD<31-24> | AD<23-16> |
| Base Address + 2 | | AD <15-8> | AD<7-0> |
| Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| Base Address + 6 | | RSB/TCB <15-8> | RSB/TCB <7-0> |
| Base Address + 8 | | RCHR/TCLR <15-8> | RCHR/TCLR <7-0> |
| Base Address + 10 | | 0 | 0 |
| Base Address + 12 | Base #2 | AD<31-24> | AD<23-16> |
| Base Address + 14 | | AD <15-8> | AD<7-0> |
| #2 Base Address | Buffer #2 | AD<31-24> | AD<23-16> |
| #2 Base Address + 2 | | AD <15-8> | AD<7-0> |
| #2 Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| #2 Base Address + 6 | | RSB/TCB <15-8> | RSB/TCB <7-0> |
| #2 Base Address + 8 | | RCHR/TCLR <15-8> | RCHR/TCLR <7-0> |
| #2 Base Address + 10 | | 0 | 0 |
| #2 Base Address + 12 | Base #3 | AD<31-24> | AD<23-16> |
| #2 Base Address + 14 | | AD <15-8> | AD<7-0> |
| #3 Base Address | Buffer #3 | AD<31-24> | AD<23-16> |
| #3 Base Address + 2 | | AD <15-8> | AD<7-0> |
| #3 Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| #3 Base Address + 6 | | RSB/TCB <15-8> | RSB/TCB <7-0> |
| #3 Base Address + 8 | | RCHR/TCLR <15-8> | RCHR/TCLR <7-0> |
| #3 Base Address + 10 | | 0 | 0 |

**Figure 30b. Linked Array-Chained, 16-Bit Bus, Big End Array**

| | | AD15 | AD0 |
|---|---|---|---|
| Base Address | Buffer #1 | AD<15-8> | AD<7-0> |
| Base Address + 2 | | AD<31-24> | AD<23-16> |
| Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| Base Address + 6 | Base #2 | AD<15-8> | AD<7-0> |
| Base Address + 8 | | AD<31-24> | AD<23-16> |
| #2 Base Address | Buffer #2 | AD<15-8> | AD<7-0> |
| #2 Base Address + 2 | | AD<31-24> | AD<23-16> |
| #2 Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| #2 Base Address + 6 | Base #3 | AD<15-8> | AD<7-0> |
| #2 Base Address + 8 | | AD<31-24> | AD<23-16> |
| #3 Base Address | Buffer #3 | AD<15-8> | AD<7-0> |
| #3 Base Address + 2 | | AD<31-24> | AD<23-16> |
| #3 Base Address + 4 | | CNT<15-8> | CNT<7-0> |
| #n - 1 Base Address + 6 | Base #n | AD<15-8> | AD<7-0> |
| #n - 1 Base Address + 8 | | AD<31-24> | AD<23-16> |
| #n Base Address | Buffer #n | Ignored | Ignored |
| #n Base Address + 2 | | Ignored | Ignored |
| #n Base Address + 4 | | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |

**Figure 31. Linked Array-Chained, 16-Bit Bus, Little End Array**

## CONTROL REGISTERS (Continued)

| Address | | AD7 AD0 |
|---|---|---|
| Base Address | Buffer #1 | AD<7-0> |
| Base Address + 1 | | AD<15-8> |
| Base Address + 2 | | AD<23-16> |
| Base Address + 3 | | AD<31-24> |
| Base Address + 4 | | CNT<7-0> |
| Base Address + 5 | | CNT<15-8> |
| Base Address + 6 | Base #2 | AD<7-0> |
| Base Address + 7 | | AD<15-8> |
| Base Address + 8 | | AD<23-16> |
| Base Address + 9 | | AD<31-24> |
| #2 Base Address | Buffer #2 | AD<7-0> |
| #2 Base Address + 1 | | AD<15-8> |
| #2 Base Address + 2 | | AD<23-16> |
| #2 Base Address + 3 | | AD<31-24> |
| #2 Base Address + 4 | | CNT<7-0> |
| #2 Base Address + 5 | | CNT<15-8> |
| #2 Base Address + 6 | Base #3 | AD<7-0> |
| #2 Base Address + 7 | | AD<15-8> |
| #2 Base Address + 8 | | AD<23-16> |
| #2 Base Address + 9 | | AD<31-24> |

**Figure 32a. Linked Array-Chained, 8-Bit Bus, Big End Array**

**Note:**
The addition of frame status/control information in the array with Linked Frame Status Transfer Enabled is similar to Big End Array. See Figure 32b.

| Address | | AD7 … AD0 |
|---|---|---|
| Base Address | Buffer #1 | AD<31-24> |
| Base Address + 1 | | AD<23-16> |
| Base Address + 2 | | AD<15-8> |
| Base Address + 3 | | AD<7-0> |
| Base Address + 4 | | CNT<15-8> |
| Base Address + 5 | | CNT<7-0> |
| Base Address + 6 | | RSB/TCB <15-8> |
| Base Address + 7 | | RSB/TCB <7-0> |
| Base Address + 8 | | RCHR/TCLR <15-8> |
| Base Address + 9 | | RCHR/TCLR <7-0> |
| Base Address + 10 | | 0 |
| Base Address + 11 | | 0 |
| Base Address + 12 | Base #2 | AD<31-24> |
| Base Address + 13 | | AD<23-16> |
| Base Address + 14 | | AD<15-8> |
| Base Address + 15 | | AD<7-0> |
| #2 Base Address | Buffer #2 | AD<31-24> |
| #2 Base Address + 1 | | AD<23-16> |
| #2 Base Address + 2 | | AD<15-8> |
| #2 Base Address + 3 | | AD<7-0> |
| #2 Base Address + 4 | | CNT<15-8> |
| #2 Base Address + 5 | | CNT<7-0> |
| #2 Base Address + 6 | | RSB/TCB <15-8> |
| #2 Base Address + 7 | | RSB/TCB <7-0> |
| #2 Base Address + 8 | | RCHR/TCLR <15-8> |
| #2 Base Address + 9 | | RCHR/TCLR <7-0> |
| #2 Base Address + 10 | | 0 |
| #2 Base Address + 11 | | 0 |

**Figure 32b. Linked Frame Status Transfer Enables**

## CONTROL REGISTERS (Continued)

| | | AD7 AD0 |
|---|---|---|
| Base Address | Buffer #1 | AD<31-24> |
| Base Address + 1 | | AD<23-16> |
| Base Address + 2 | | AD<15-8> |
| Base Address + 3 | | AD<7-0> |
| Base Address + 4 | | CNT<15-8> |
| Base Address + 5 | | CNT<7-0> |
| Base Address + 6 | Base #2 | AD<31-24> |
| Base Address + 7 | | AD<23-16> |
| Base Address + 8 | | AD<15-8> |
| Base Address + 9 | | AD<7-0> |
| #2 Base Address | Buffer #2 | AD<31-24> |
| #2 Base Address + 1 | | AD<23-16> |
| #2 Base Address + 2 | | AD<15-8> |
| #2 Base Address + 3 | | AD<7-0> |
| #2 Base Address + 4 | | CNT<15-8> |
| #2 Base Address + 5 | | CNT<7-0> |
| #2 Base Address + 6 | Base #3 | AD<31-24> |
| #2 Base Address + 7 | | AD<23-16> |
| #2 Base Address + 8 | | AD<15-8> |
| #2 Base Address + 9 | | AD<7-0> |

**Figure 33. Linked Array-Chained 8-Bit Bus, Little End Array**

Address: 00000

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

- D0: Upper//Lower Byte Select (WO)
- D1: Address 0 (WO)
- D2: Address 1 (WO)
- D3: Address 2 (WO)
- D4: Address 3 (WO)
- D5: Address 4 (WO)
- D6: Byte//Word Access (WO)
- D7: Channel Load DMA (WO)

Rx/Tx Mode (D9–D8):

| D9 | D8 | |
|---|---|---|
| 0 | 0 | Normal Operation |
| 0 | 1 | Auto Echo |
| 1 | 0 | External Local Loopback |
| 1 | 1 | Internal Local Loopback |

- D10: Rx/Tx Reset

Channel Command (WO) (D15–D11):

| D15 | D14 | D13 | D12 | D11 | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | Null Command |
| 0 | 0 | 0 | 0 | 1 | Reserved |
| 0 | 0 | 0 | 1 | 0 | Reset Highest IUS |
| 0 | 0 | 0 | 1 | 1 | Reserved |
| 0 | 0 | 1 | 0 | 0 | Trigger Channel Load DMA |
| 0 | 0 | 1 | 0 | 1 | Trigger Rx DMA |
| 0 | 0 | 1 | 1 | 0 | Trigger Tx DMA |
| 0 | 0 | 1 | 1 | 1 | Trigger Rx & Tx DMA |
| 0 | 1 | 0 | 0 | 0 | Reserved |
| 0 | 1 | 0 | 0 | 1 | Rx FIFO Purge |
| 0 | 1 | 0 | 1 | 0 | Tx FIFO Purge |
| 0 | 1 | 0 | 1 | 1 | Rx & Tx FIFO Purge |
| 0 | 1 | 1 | 0 | 0 | Reserved |
| 0 | 1 | 1 | 0 | 1 | Load Rx Character Count |
| 0 | 1 | 1 | 1 | 0 | Load Tx Character Count |
| 0 | 1 | 1 | 1 | 1 | Load Rx & Tx Character Count |
| 1 | 0 | 0 | 0 | 0 | Reserved |
| 1 | 0 | 0 | 0 | 1 | Load TC0 |
| 1 | 0 | 0 | 1 | 0 | Load TC1 |
| 1 | 0 | 0 | 1 | 1 | Load TC0 & TC1 |
| 1 | 0 | 1 | 0 | 0 | Select Serial Data LSB First * |
| 1 | 0 | 1 | 0 | 1 | Select Serial Data MSB First |
| 1 | 0 | 1 | 1 | 0 | Select Straight Memory Data * |
| 1 | 0 | 1 | 1 | 1 | Select Swapped Memory Data |
| 1 | 1 | 0 | 0 | 0 | Reserved |
| 1 | 1 | 0 | 0 | 1 | Reserved |
| 1 | 1 | 0 | 1 | 0 | Reserved |
| 1 | 1 | 0 | 1 | 1 | Reserved |
| 1 | 1 | 1 | 0 | 0 | Reserved |
| 1 | 1 | 1 | 0 | 1 | Reserved |
| 1 | 1 | 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | 1 | 1 | Reserved |

* Selected Upon Reset

**Figure 34. Channel Command/Address Register (CCAR)**

## CONTROL REGISTERS (Continued)

Address: 00001

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Receiver Mode

| D3 | D2 | D1 | D0 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Asynchronous |
| 0 | 0 | 0 | 1 | External Synchronous |
| 0 | 0 | 1 | 0 | Isochronous |
| 0 | 0 | 1 | 1 | Asynchronous with CV |
| 0 | 1 | 0 | 0 | Monosync |
| 0 | 1 | 0 | 1 | Bisync |
| 0 | 1 | 1 | 0 | HDLC |
| 0 | 1 | 1 | 1 | Transparent Bisync |
| 1 | 0 | 0 | 0 | NBIP |
| 1 | 0 | 0 | 1 | 802.3 |
| 1 | 0 | 1 | 0 | Reserved |
| 1 | 0 | 1 | 1 | Reserved |
| 1 | 1 | 0 | 0 | Reserved |
| 1 | 1 | 0 | 1 | Reserved |
| 1 | 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | 1 | Reserved |

- D4: Rx Submode 0
- D5: Rx Submode 1
- D6: Rx Submode 2
- D7: Rx Submode 3

Transmitter Mode

| D11 | D10 | D9 | D8 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Asynchronous |
| 0 | 0 | 0 | 1 | Reserved |
| 0 | 0 | 1 | 0 | Isochronous |
| 0 | 0 | 1 | 1 | Asynchronous with CV |
| 0 | 1 | 0 | 0 | Monosync |
| 0 | 1 | 0 | 1 | Bisync |
| 0 | 1 | 1 | 0 | HDLC |
| 0 | 1 | 1 | 1 | Transparent Bisync |
| 1 | 0 | 0 | 0 | NBIP |
| 1 | 0 | 0 | 1 | 802.3 |
| 1 | 0 | 1 | 0 | Reserved |
| 1 | 0 | 1 | 1 | Reserved |
| 1 | 1 | 0 | 0 | Slaved Monosync |
| 1 | 1 | 0 | 1 | Reserved |
| 1 | 1 | 1 | 0 | HDLC Loop |
| 1 | 1 | 1 | 1 | Reserved |

- D12: Tx Submode 0
- D13: Tx Submode 1
- D14: Tx Submode 2
- D15: Tx Submode 3

**Figure 35. Channel Mode Register (CMR)**

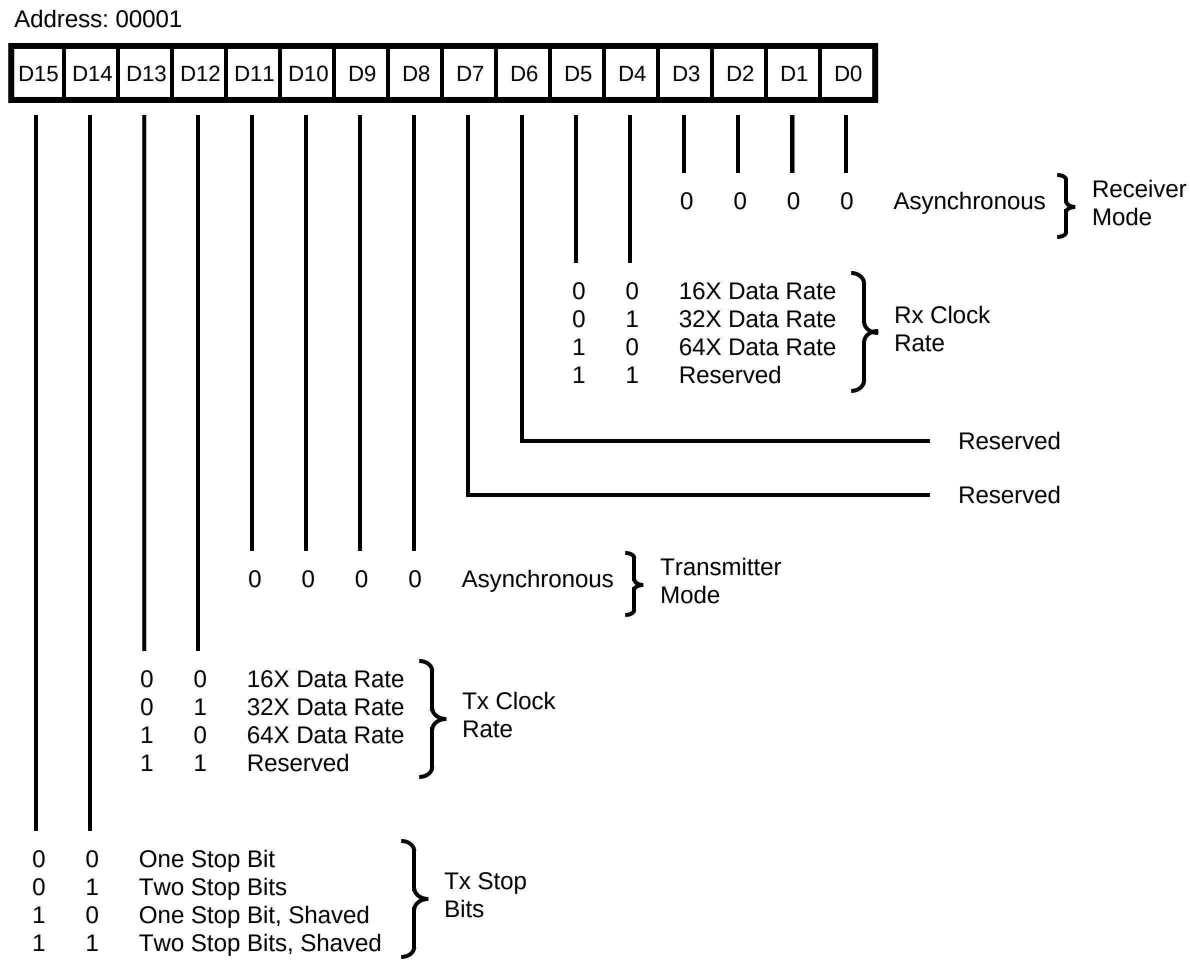



**Figure 36. Channel Mode Register, Asynchronous Mode**

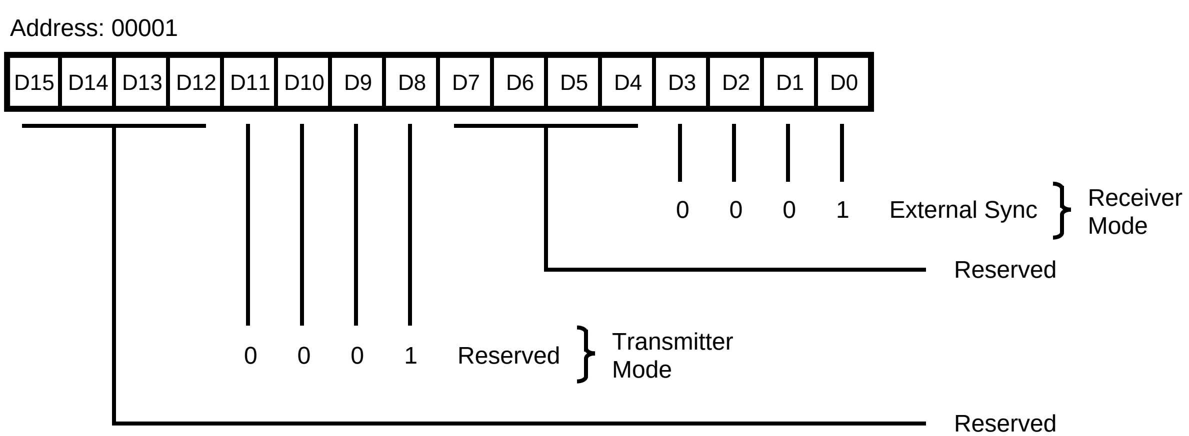



**Figure 37. Channel Mode Register, External Sync Mode**

## CONTROL REGISTERS (Continued)

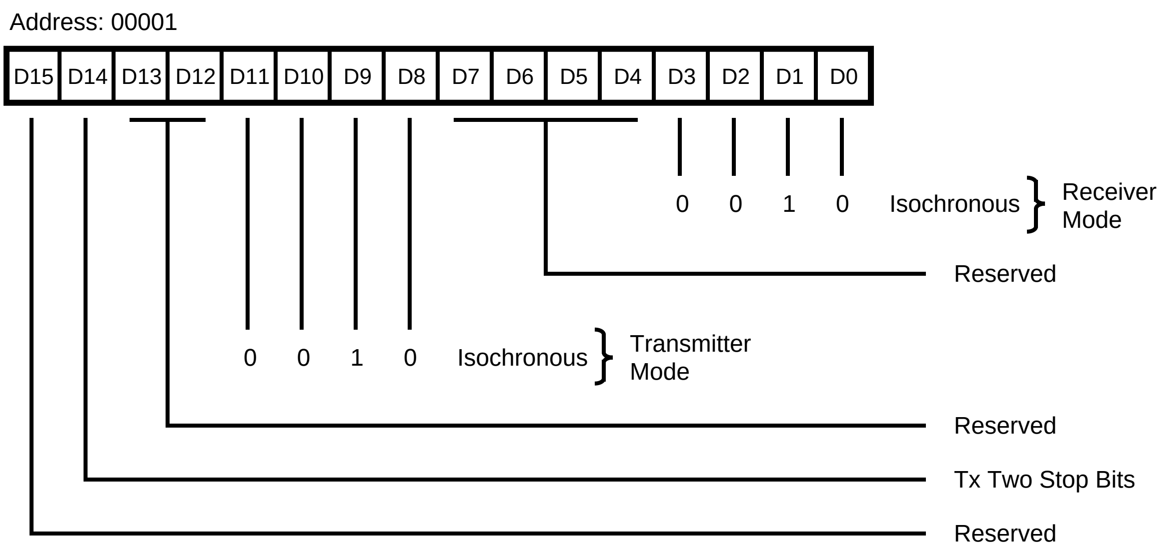



**Figure 38. Channel Mode Register, Isochronous Mode**

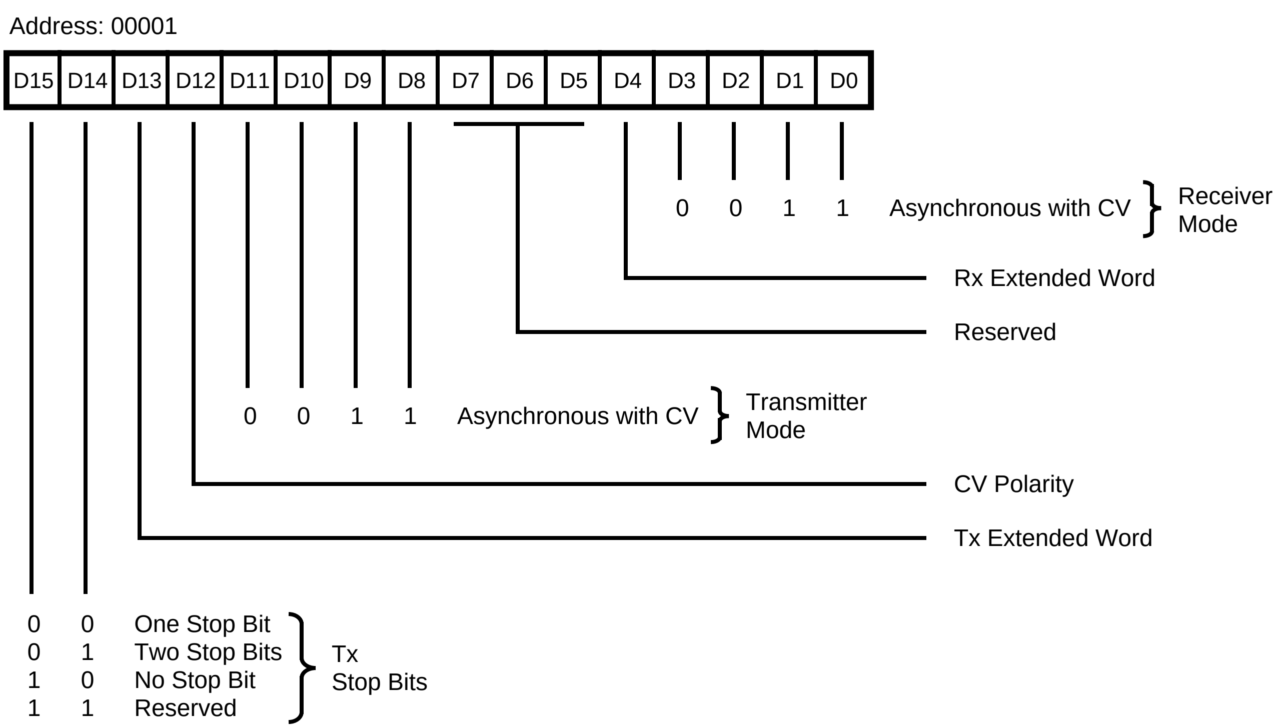



**Figure 39. Channel Mode Register, Asynchronous Mode with Code Violation (MIL STD 1553)**

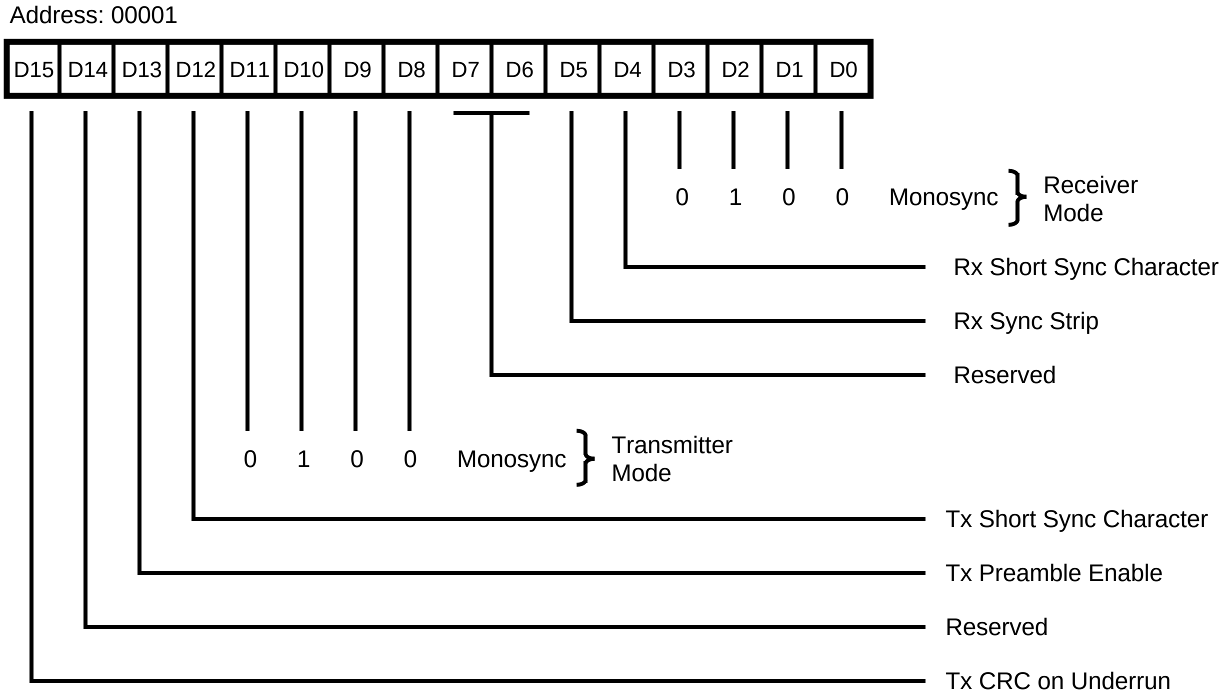



**Figure 40. Channel Mode Register, Monosync Mode**

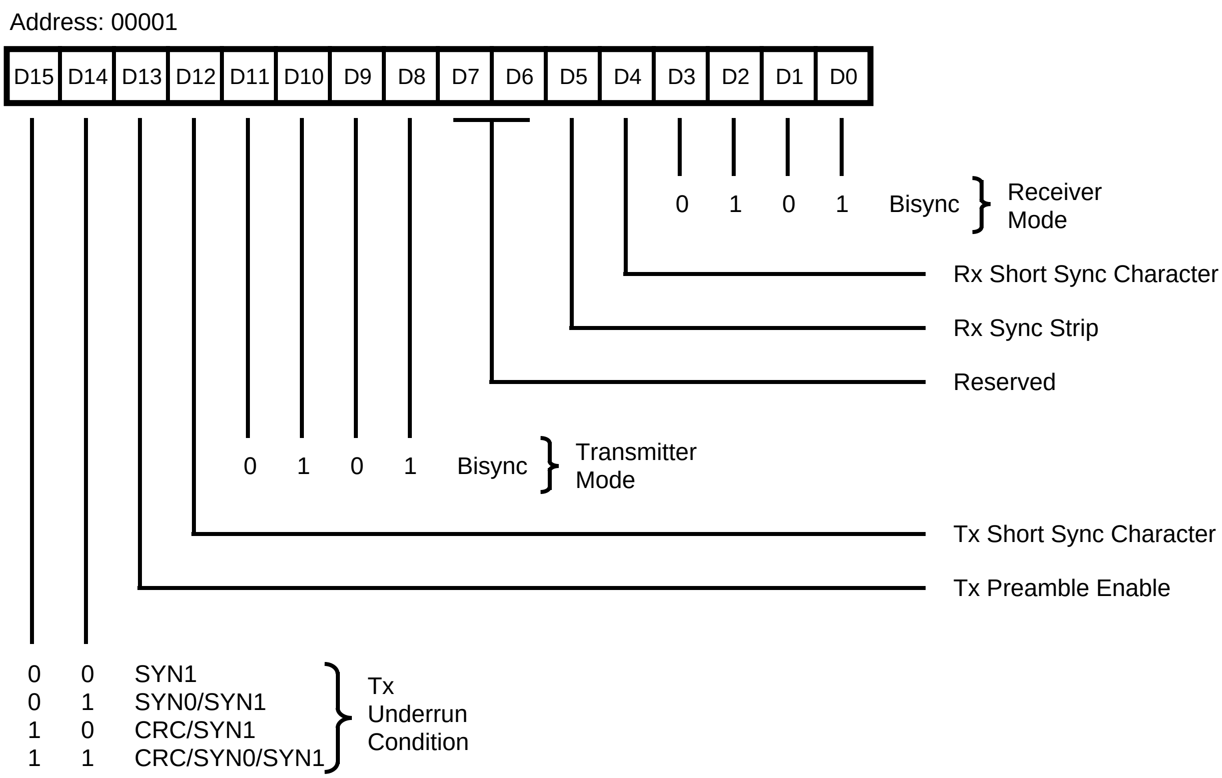



**Figure 41. Channel Mode Register, Bisync Mode**

## CONTROL REGISTERS (Continued)

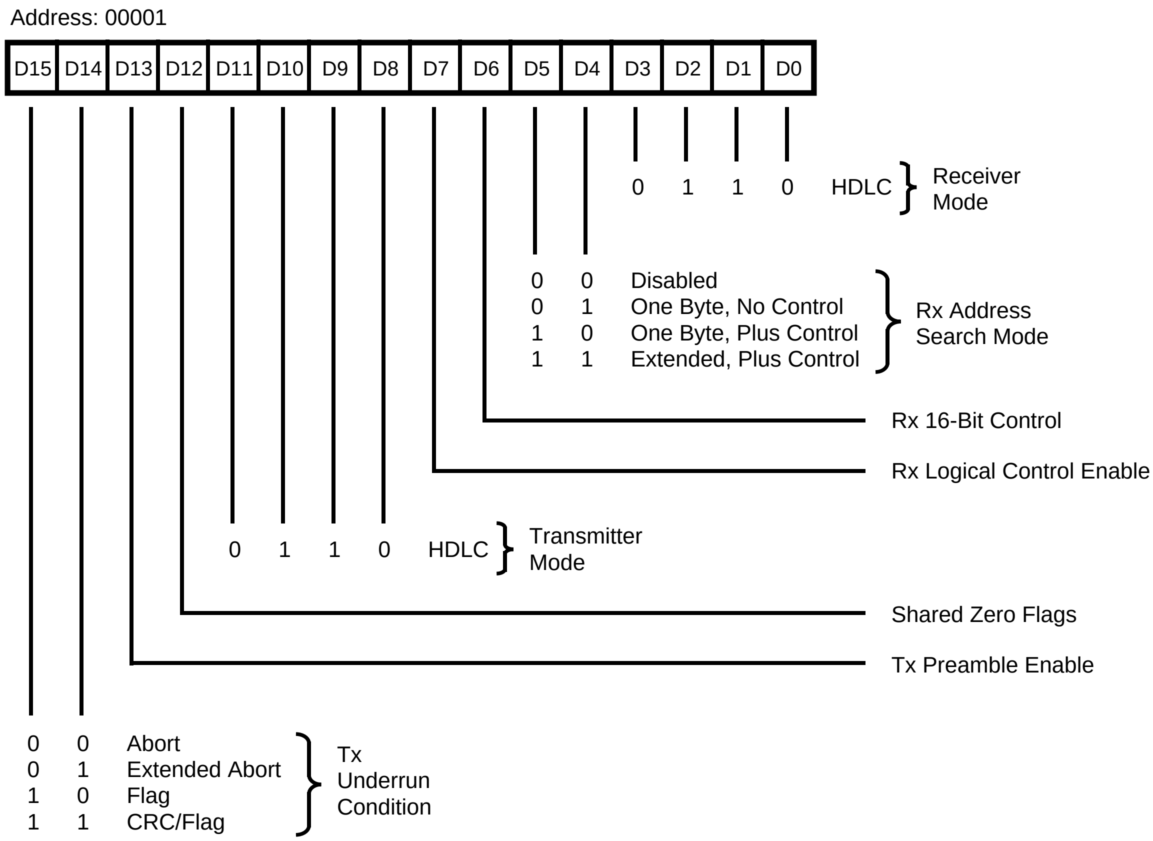



**Figure 42. Channel Mode Register, HDLC Mode**

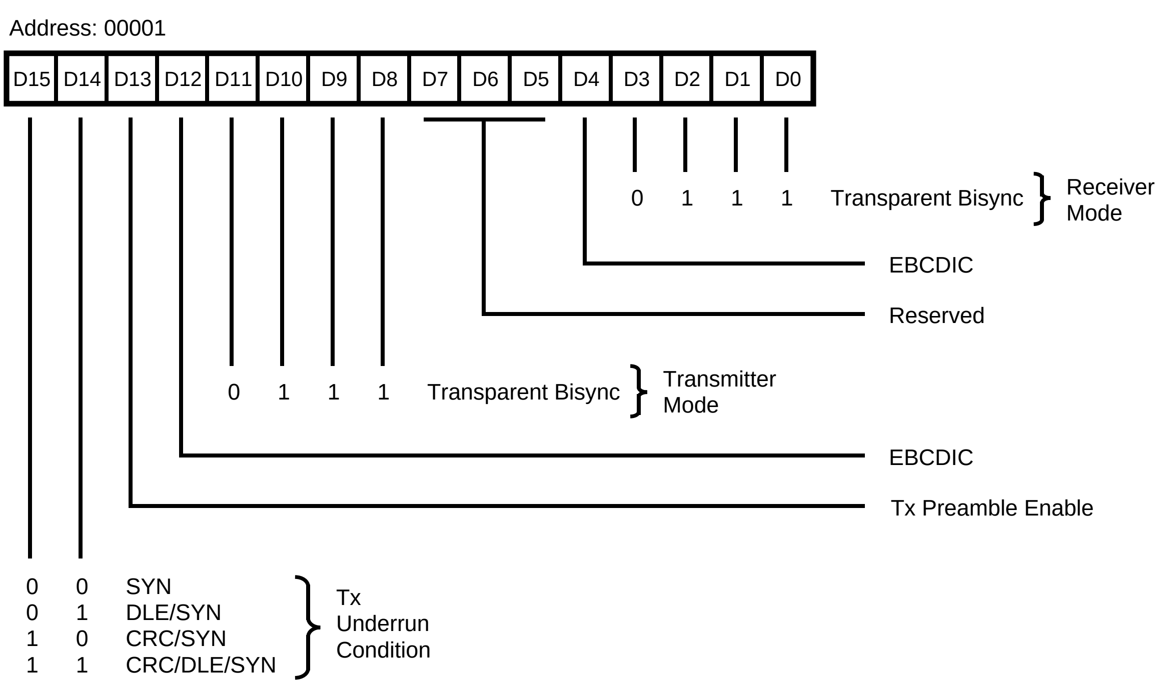



**Figure 43. Channel Mode Register, Transparent Bisync Mode**

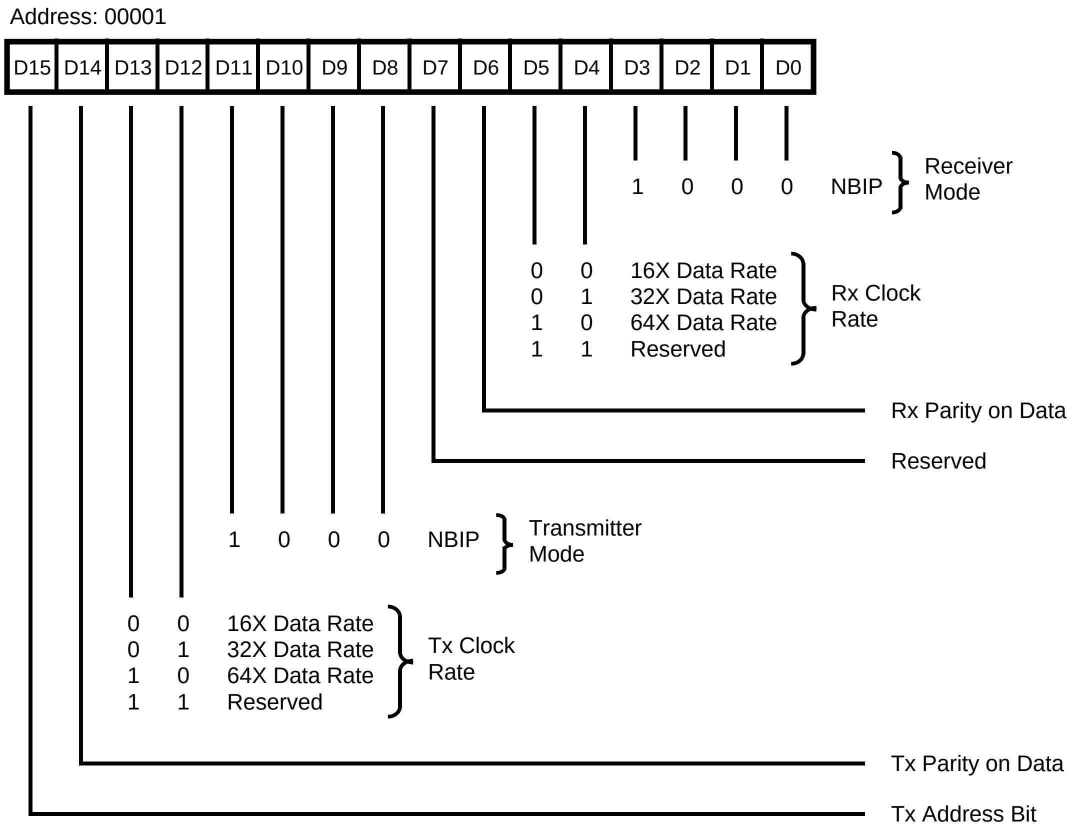



**Figure 44. Channel Mode Register, NBIP Mode**

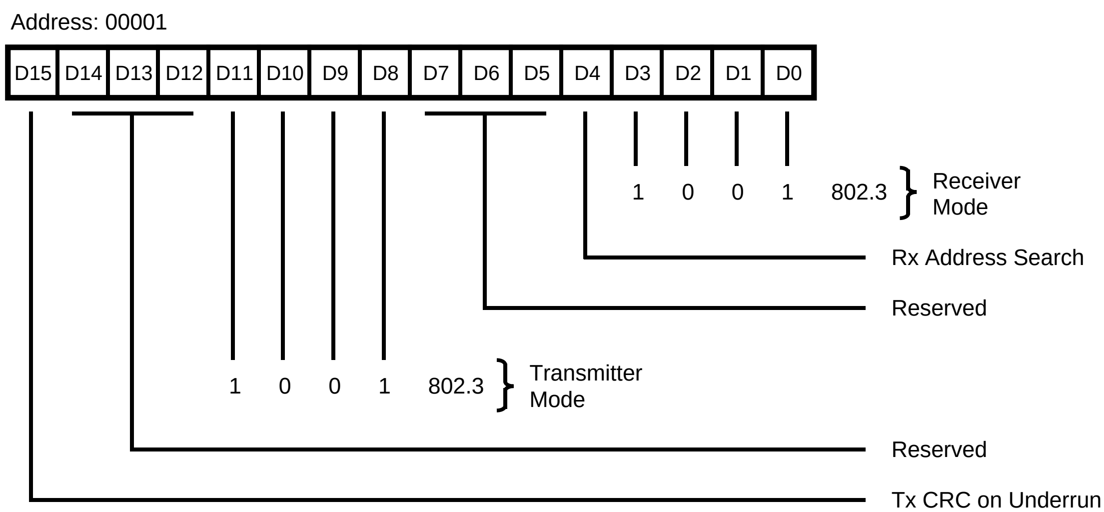



**Figure 45. Channel Mode Register, 802.3 Mode**

## CONTROL REGISTERS (Continued)

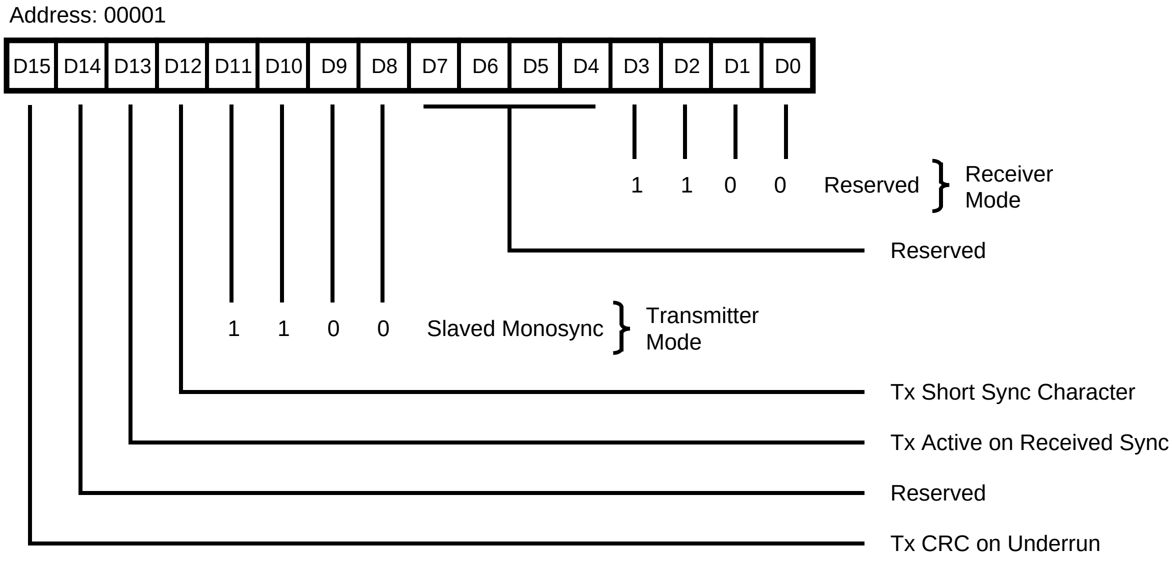



**Figure 46. Channel Mode Register, Slaved Monosync Mode**

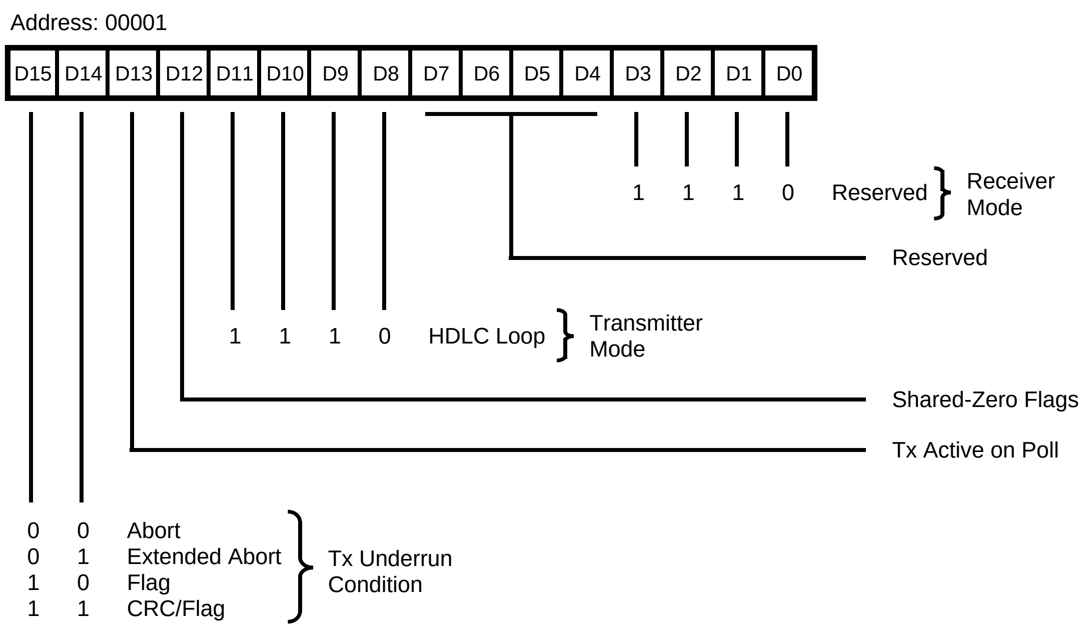



**Figure 47. Channel Mode Register, HDLC Loop Mode**

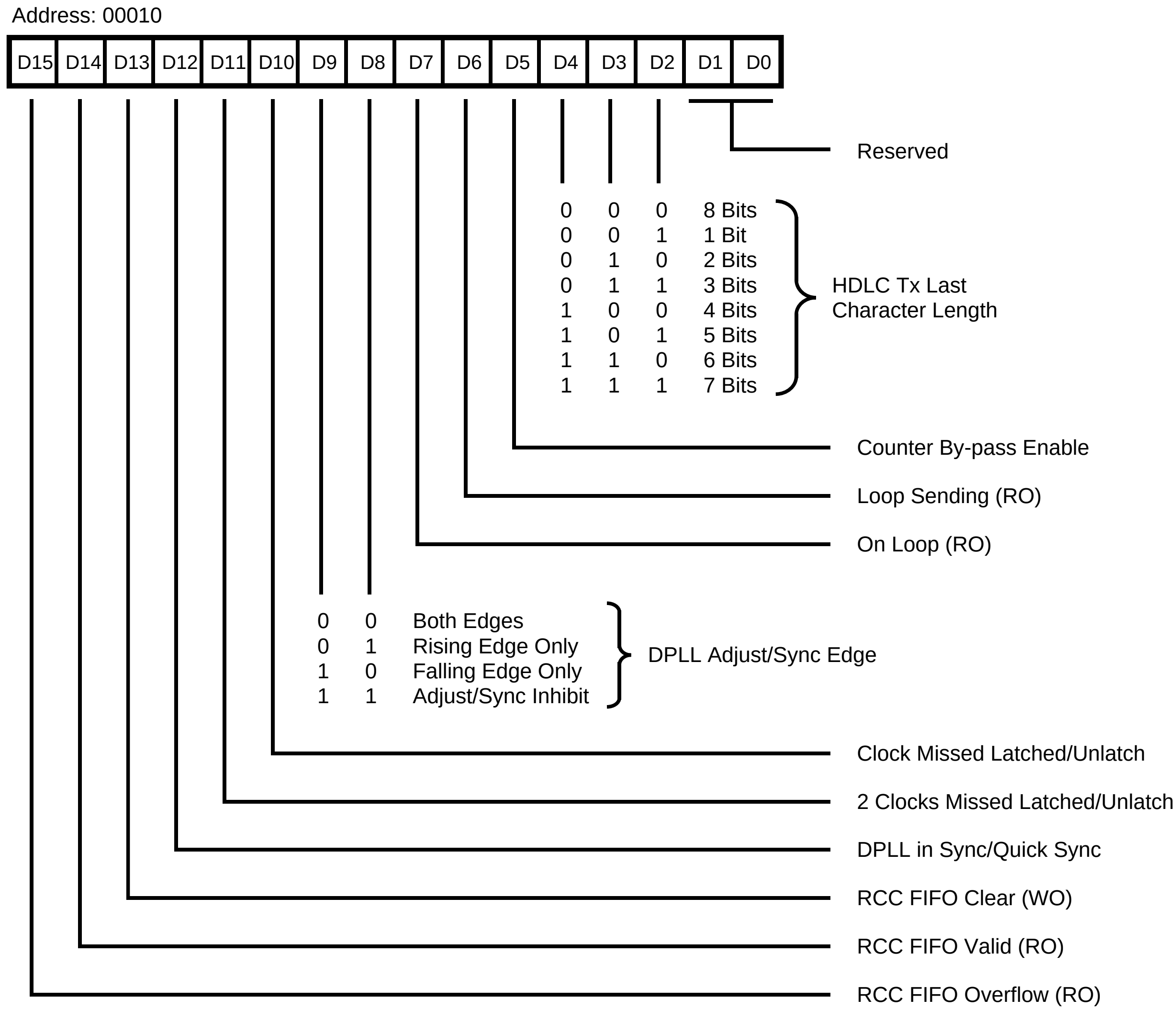



**Figure 48. Channel Command/Status Register (CCSR)**

## CONTROL REGISTERS (Continued)

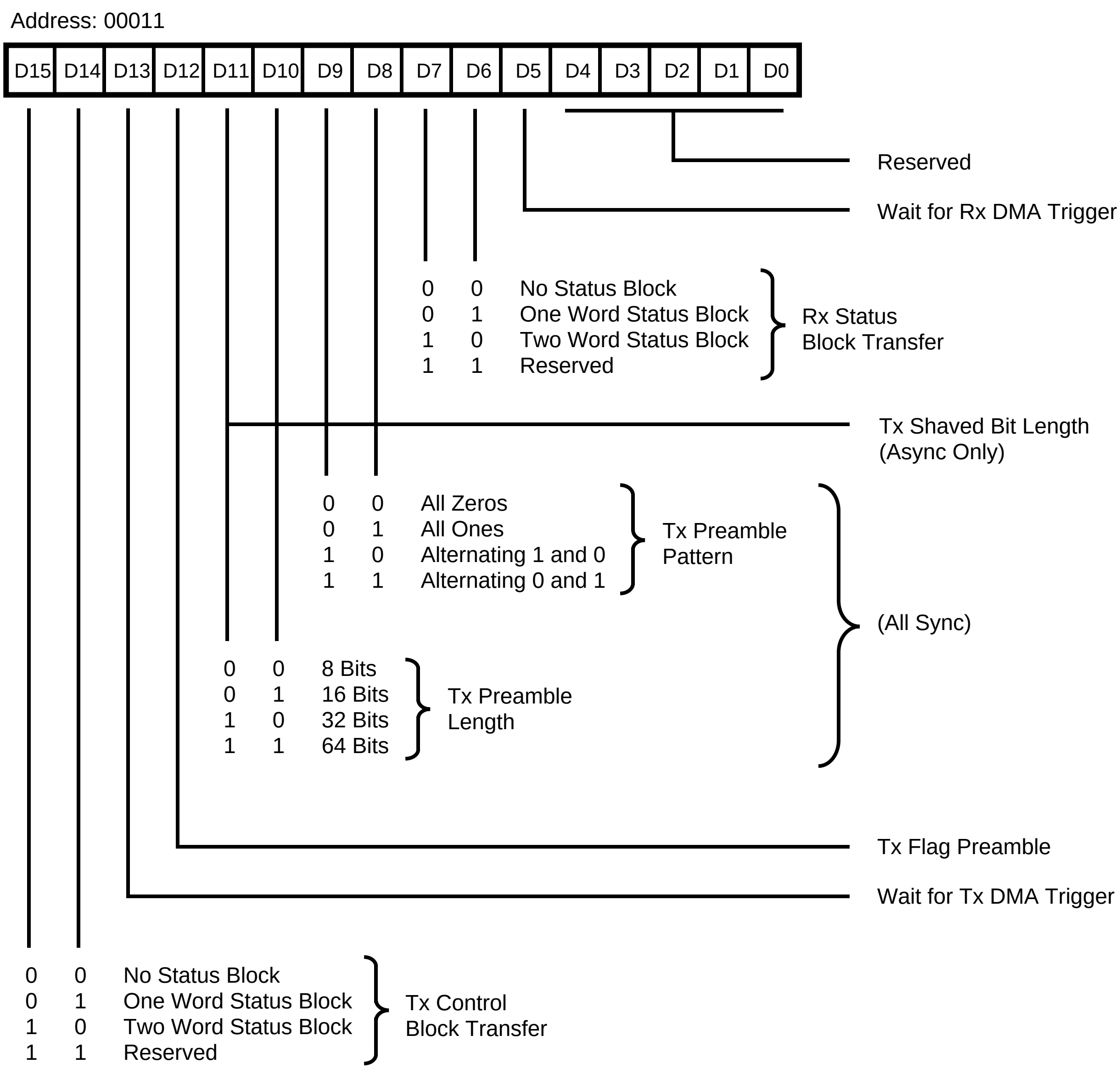



**Figure 49. Channel Control Register (CCR)**

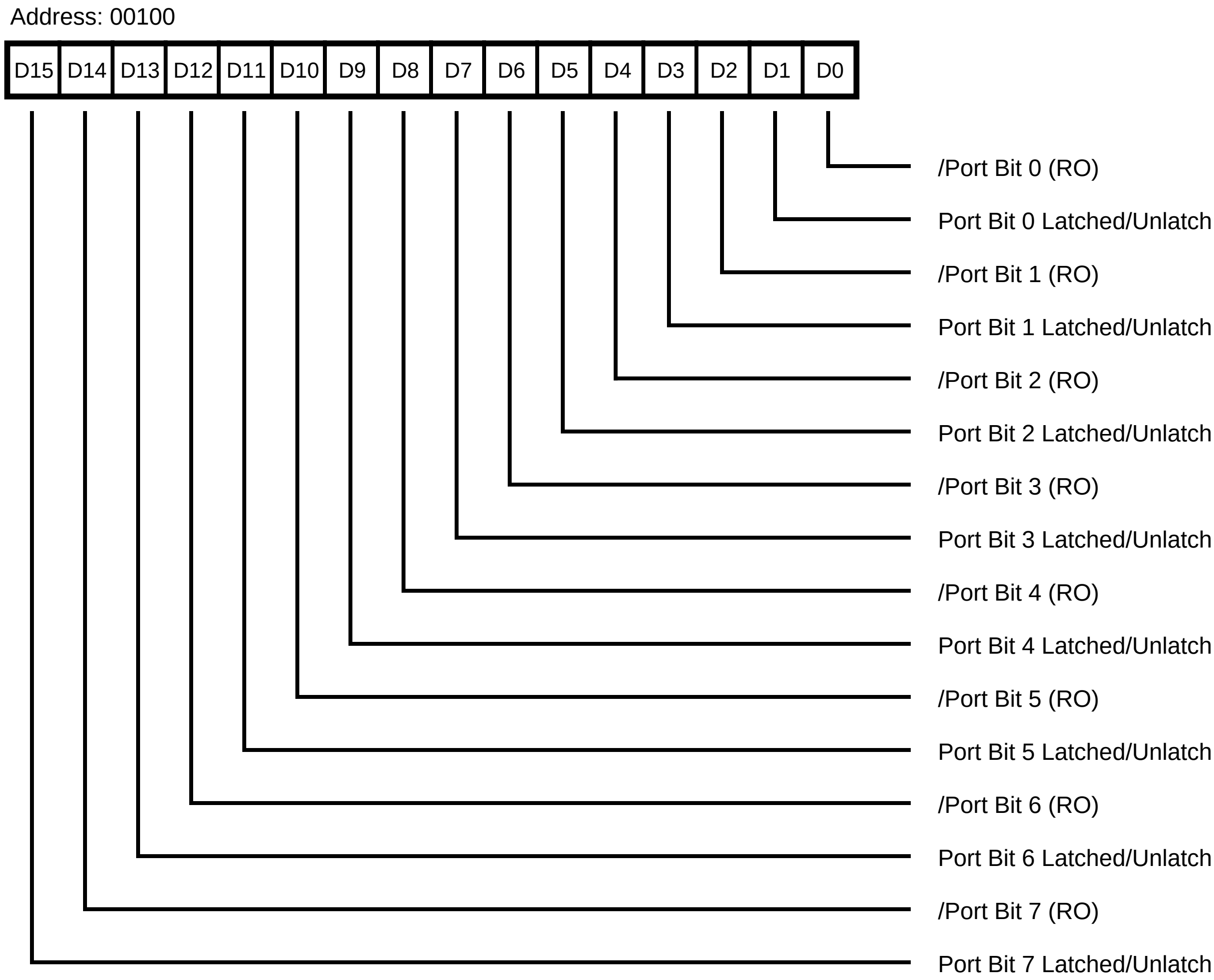



**Figure 50. Port Status Register (PSR)**

## CONTROL REGISTERS (Continued)

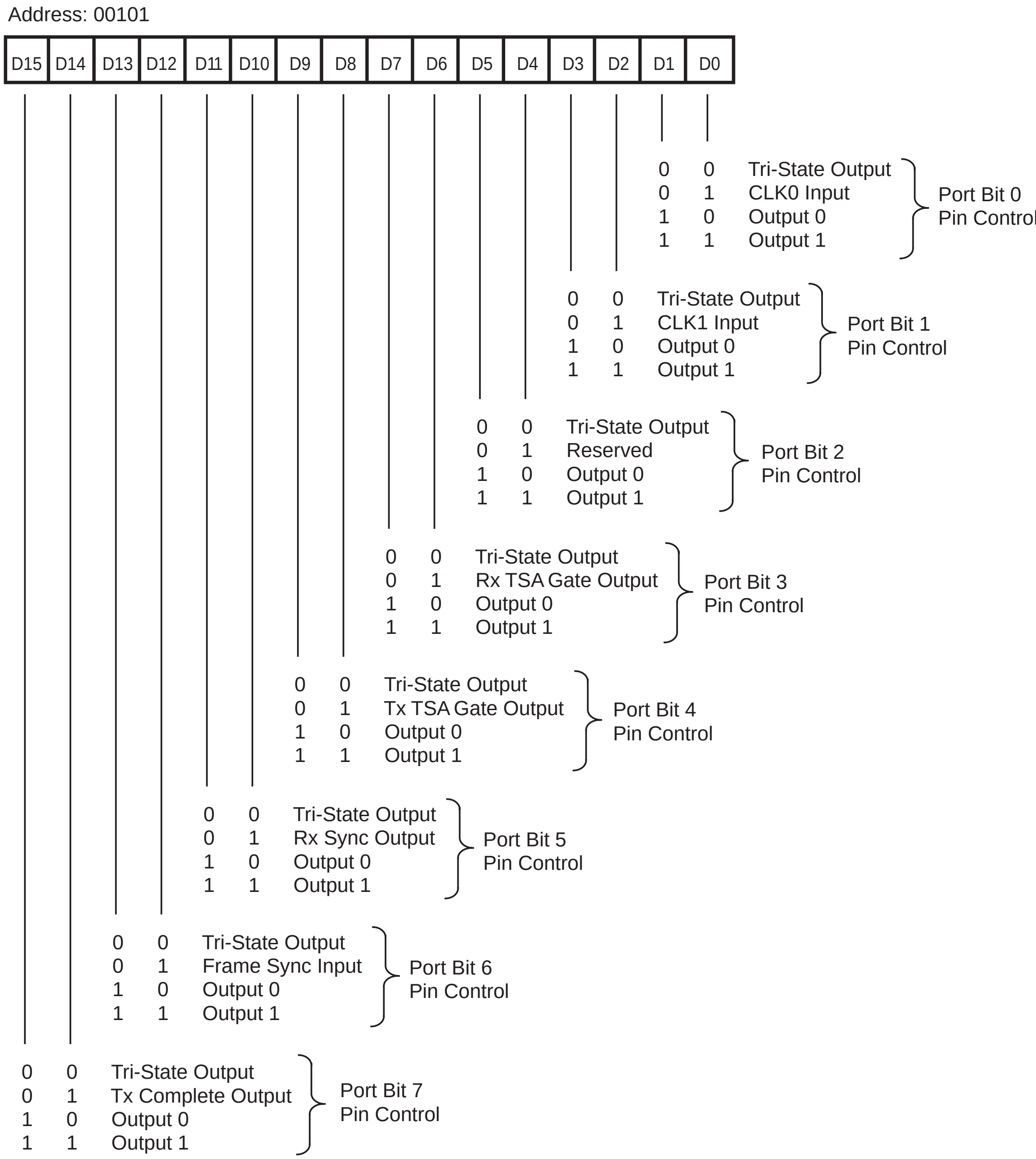



**Figure 51. Port Control Register (PCR)**

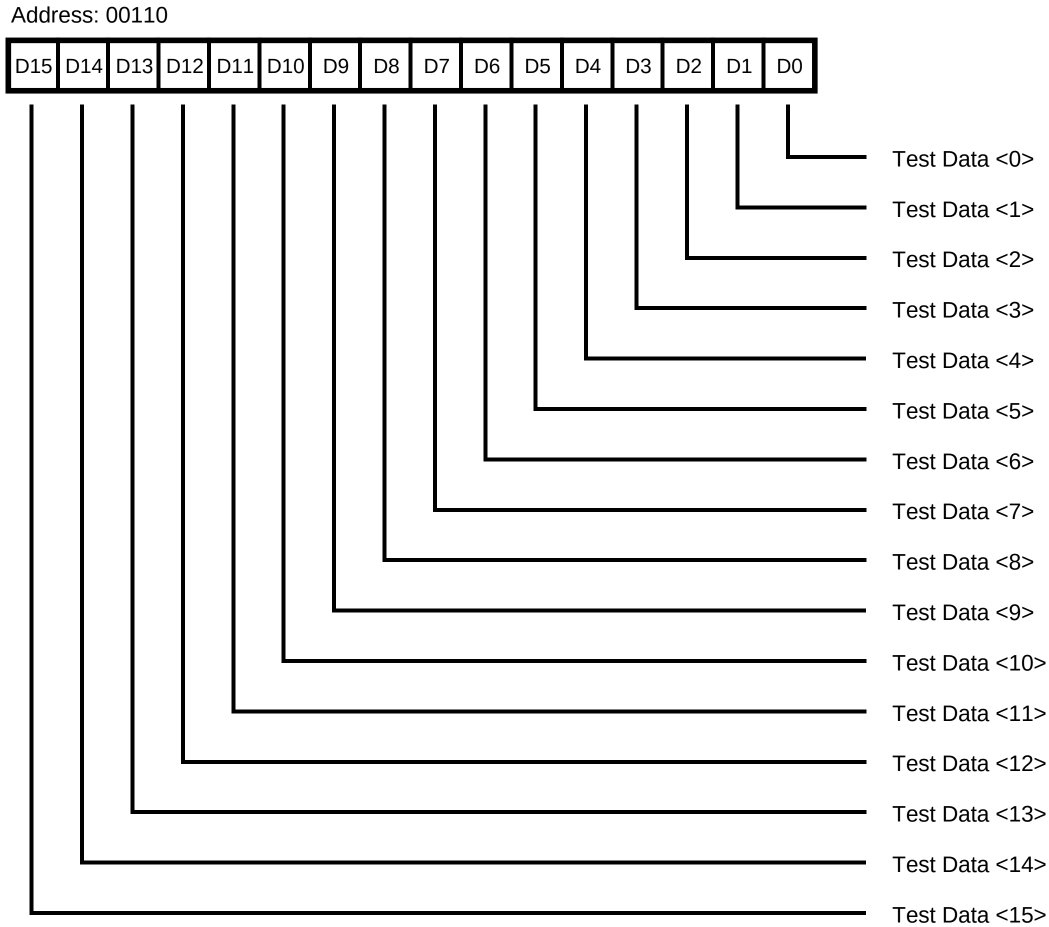



**Figure 52. Test Mode Data Register (TMDR)**

## CONTROL REGISTERS (Continued)

Address: 00111

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Test Register Address

| D4 | D3 | D2 | D1 | D0 | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | Null Address |
| 0 | 0 | 0 | 0 | 1 | High Byte of Shifters |
| 0 | 0 | 0 | 1 | 0 | CRC Byte 0 |
| 0 | 0 | 0 | 1 | 1 | CRC Byte 1 |
| 0 | 0 | 1 | 0 | 0 | Rx FIFO (Write) |
| 0 | 0 | 1 | 0 | 1 | Clock Multiplexer Outputs |
| 0 | 0 | 1 | 1 | 0 | CTR0 and CTR1 Counters |
| 0 | 0 | 1 | 1 | 1 | Clock Multiplexer Inputs |
| 0 | 1 | 0 | 0 | 0 | DPLL State |
| 0 | 1 | 0 | 0 | 1 | Low Byte of Shifters |
| 0 | 1 | 0 | 1 | 0 | CRC Byte 2 |
| 0 | 1 | 0 | 1 | 1 | CRC Byte 3 |
| 0 | 1 | 1 | 0 | 0 | Tx FIFO (Read) |
| 0 | 1 | 1 | 0 | 1 | Reserved |
| 0 | 1 | 1 | 1 | 0 | I/O and Device Status Latches |
| 0 | 1 | 1 | 1 | 1 | Internal Daisy Chain |
| 1 | 0 | 0 | 0 | 0 | Reserved |
| 1 | 0 | 0 | 0 | 1 | Reserved |
| 1 | 0 | 0 | 1 | 0 | Reserved |
| 1 | 0 | 0 | 1 | 1 | Reserved |
| 1 | 0 | 1 | 0 | 0 | Reserved |
| 1 | 0 | 1 | 0 | 1 | Reserved |
| 1 | 0 | 1 | 1 | 0 | Rx Count Holding Register |
| 1 | 0 | 1 | 1 | 1 | Reserved |
| 1 | 1 | 0 | 0 | 0 | Reserved |
| 1 | 1 | 0 | 0 | 1 | Reserved |
| 1 | 1 | 0 | 1 | 0 | Reserved |
| 1 | 1 | 0 | 1 | 1 | 4453H |
| 1 | 1 | 1 | 0 | 0 | Reserved |
| 1 | 1 | 1 | 0 | 1 | Reserved |
| 1 | 1 | 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | 1 | 1 | 4453H |

D15–D5: Reserved

**Figure 53. Test Mode Control Register (TMCR)**

**Note:**
When software writes the value 1F to the LS byte of the Test Mode Control Register (TMCR), and then reads the Test Mode Data Register (TMDR), current versions of the Z16C32 will return hex 4453. Future revisions, if any, will return other values.

Address: 01000

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Receive Clock Source**

| D2 | D1 | D0 | |
|---|---|---|---|
| 0 | 0 | 0 | Disabled |
| 0 | 0 | 1 | /RxC Pin |
| 0 | 1 | 0 | /TxC Pin |
| 0 | 1 | 1 | DPLL Output |
| 1 | 0 | 0 | BRG0 Output |
| 1 | 0 | 1 | BRG1 Output |
| 1 | 1 | 0 | CTR0 Output |
| 1 | 1 | 1 | CTR1 Output |

**Transmit Clock Source**

| D5 | D4 | D3 | |
|---|---|---|---|
| 0 | 0 | 0 | Disabled |
| 0 | 0 | 1 | /RxC Pin |
| 0 | 1 | 0 | /TxC Pin |
| 0 | 1 | 1 | DPLL Output |
| 1 | 0 | 0 | BRG0 Output |
| 1 | 0 | 1 | BRG1 Output |
| 1 | 1 | 0 | CTR0 Output |
| 1 | 1 | 1 | CTR1 Output |

**DPLL Clock Source**

| D7 | D6 | |
|---|---|---|
| 0 | 0 | BRG0 Output |
| 0 | 1 | BRG1 Output |
| 1 | 0 | /RxC Pin |
| 1 | 1 | /TxC Pin |

**BRG0 Clock Source**

| D9 | D8 | |
|---|---|---|
| 0 | 0 | CTR0 Output |
| 0 | 1 | CTR1 Output |
| 1 | 0 | /RxC Pin |
| 1 | 1 | /TxC Pin |

**BRG1 Clock Source**

| D11 | D10 | |
|---|---|---|
| 0 | 0 | CTR0 Output |
| 0 | 1 | CTR1 Output |
| 1 | 0 | /RxC Pin |
| 1 | 1 | /TxC Pin |

**CTR0 Clock Source**

| D13 | D12 | |
|---|---|---|
| 0 | 0 | Disabled |
| 0 | 1 | Port0 Pin |
| 1 | 0 | /RxC Pin |
| 1 | 1 | /TxC Pin |

**CTR1 Clock Source**

| D15 | D14 | |
|---|---|---|
| 0 | 0 | Disabled |
| 0 | 1 | Port1 Pin |
| 1 | 0 | /RxC Pin |
| 1 | 1 | /TxC Pin |

**Figure 54. Clock Mode Control Register (CMCR)**

## CONTROL REGISTERS (Continued)

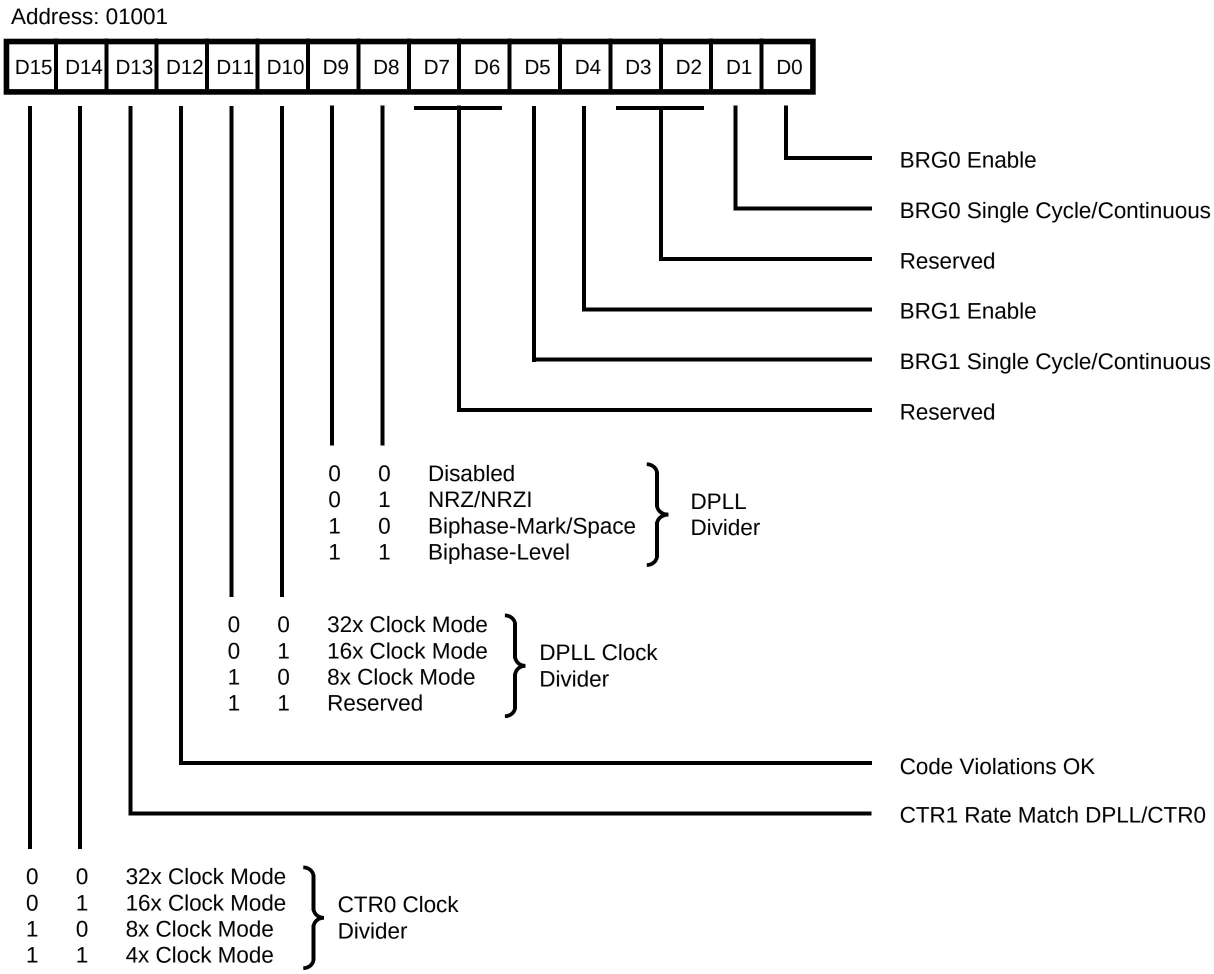



**Figure 55. Hardware Configuration Register (HCR)**

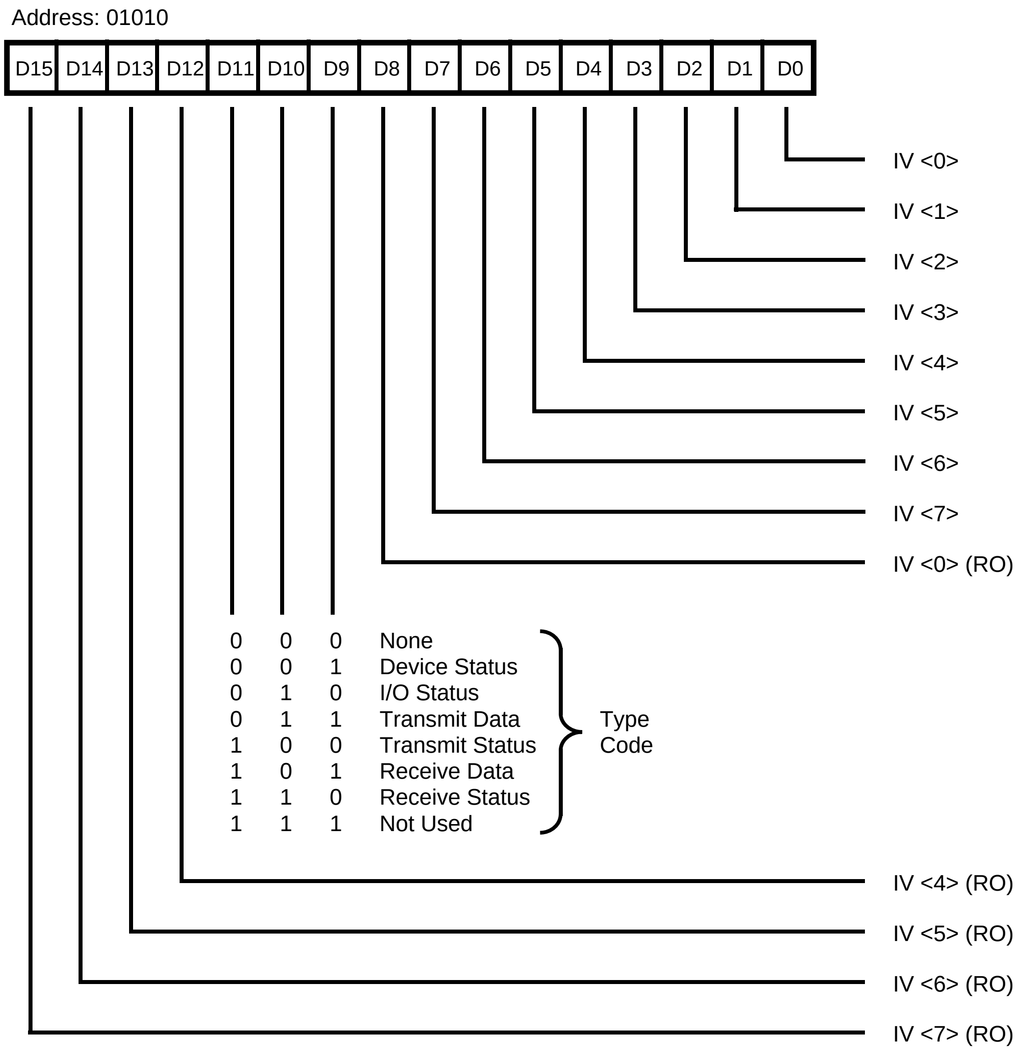



**Figure 56. Interrupt Vector Register (IVR)**

## CONTROL REGISTERS (Continued)

Address: 01011

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

/RxC Pin Control

| D2 | D1 | D0 | |
|---|---|---|---|
| 0 | 0 | 0 | Input Pin |
| 0 | 0 | 1 | Rx Clock Output |
| 0 | 1 | 0 | Rx Byte Clock Output |
| 0 | 1 | 1 | SYNC Output |
| 1 | 0 | 0 | BRG0 Output |
| 1 | 0 | 1 | BRG1 Output |
| 1 | 1 | 0 | CTR0 Output |
| 1 | 1 | 1 | DPLL Rx Output |

/TxC Pin Control

| D5 | D4 | D3 | |
|---|---|---|---|
| 0 | 0 | 0 | Input PIn |
| 0 | 0 | 1 | Tx Clock Output |
| 0 | 1 | 0 | Tx Byte Clock Output |
| 0 | 1 | 1 | Tx Complete Output |
| 1 | 0 | 0 | BRG0 Output |
| 1 | 0 | 1 | BRG1 Output |
| 1 | 1 | 0 | CTR1 Output |
| 1 | 1 | 1 | DPLL Tx Output |

TxD Pin Control

| D7 | D6 | |
|---|---|---|
| 0 | 0 | Tx Data Output |
| 0 | 1 | 3-State Output |
| 1 | 0 | Output 0 |
| 1 | 1 | Output 1 |

/RxREQ Pin Control

| D9 | D8 | |
|---|---|---|
| 0 | 0 | 3-State Output |
| 0 | 1 | Rx Request Output |
| 1 | 0 | Output 0 |
| 1 | 1 | Output 1 |

/TxREQ Pin Control

| D11 | D10 | |
|---|---|---|
| 0 | 0 | 3-State Output |
| 0 | 1 | Tx Request Output |
| 1 | 0 | Output 0 |
| 1 | 1 | Output 1 |

/DCD Pin Control

| D13 | D12 | |
|---|---|---|
| 0 | 0 | /DCD Input |
| 0 | 1 | /DCD//SYNC Input |
| 1 | 0 | Output 0 |
| 1 | 1 | Output 1 |

/CTS Pin Control

| D15 | D14 | |
|---|---|---|
| 0 | 0 | /CTS Input |
| 0 | 1 | /CTS Input |
| 1 | 0 | Output 0 |
| 1 | 1 | Output 1 |

**Figure 57. I/O Control Register (IOCR)**

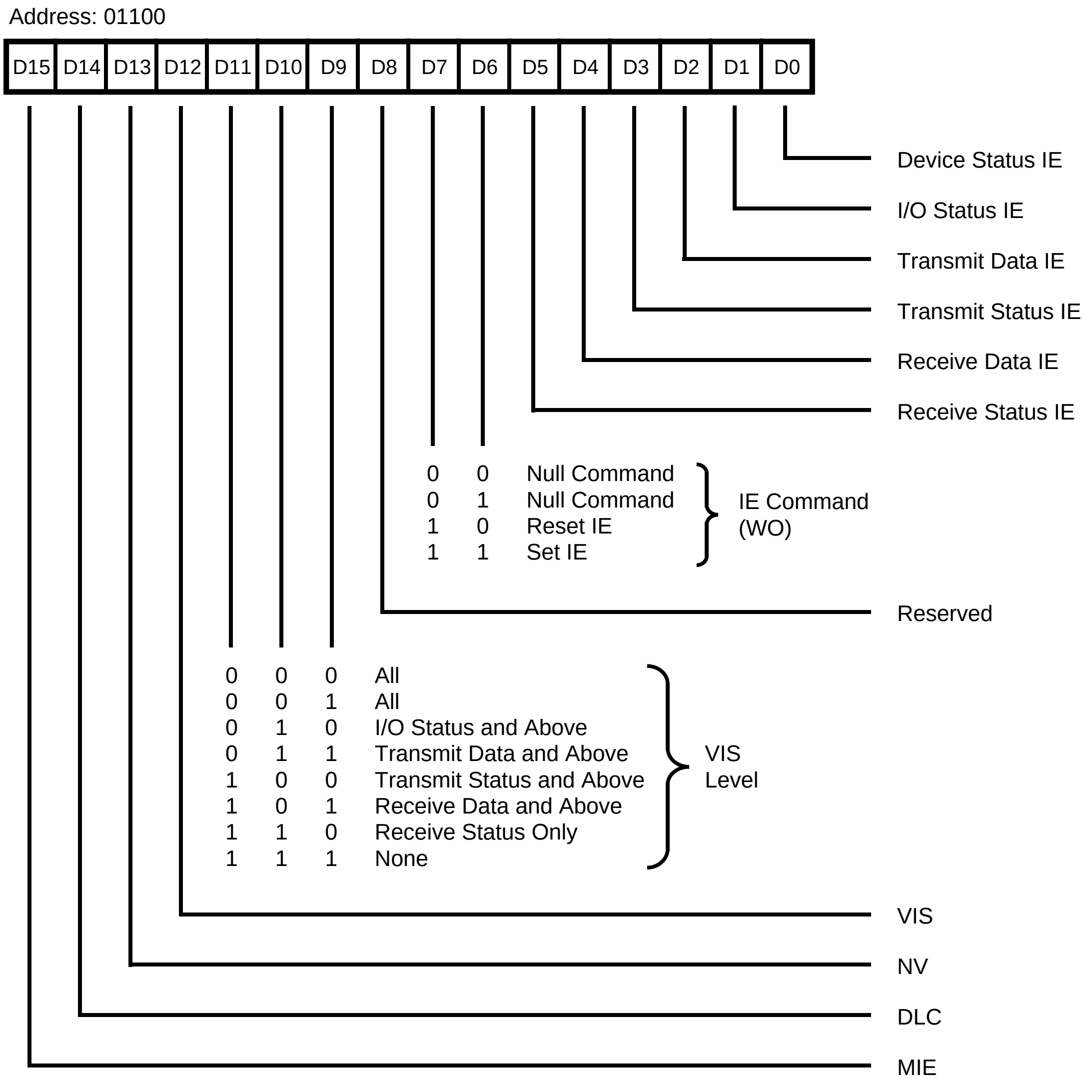



**Figure 58. Interrupt Control Register (ICR)**

## CONTROL REGISTERS (Continued)

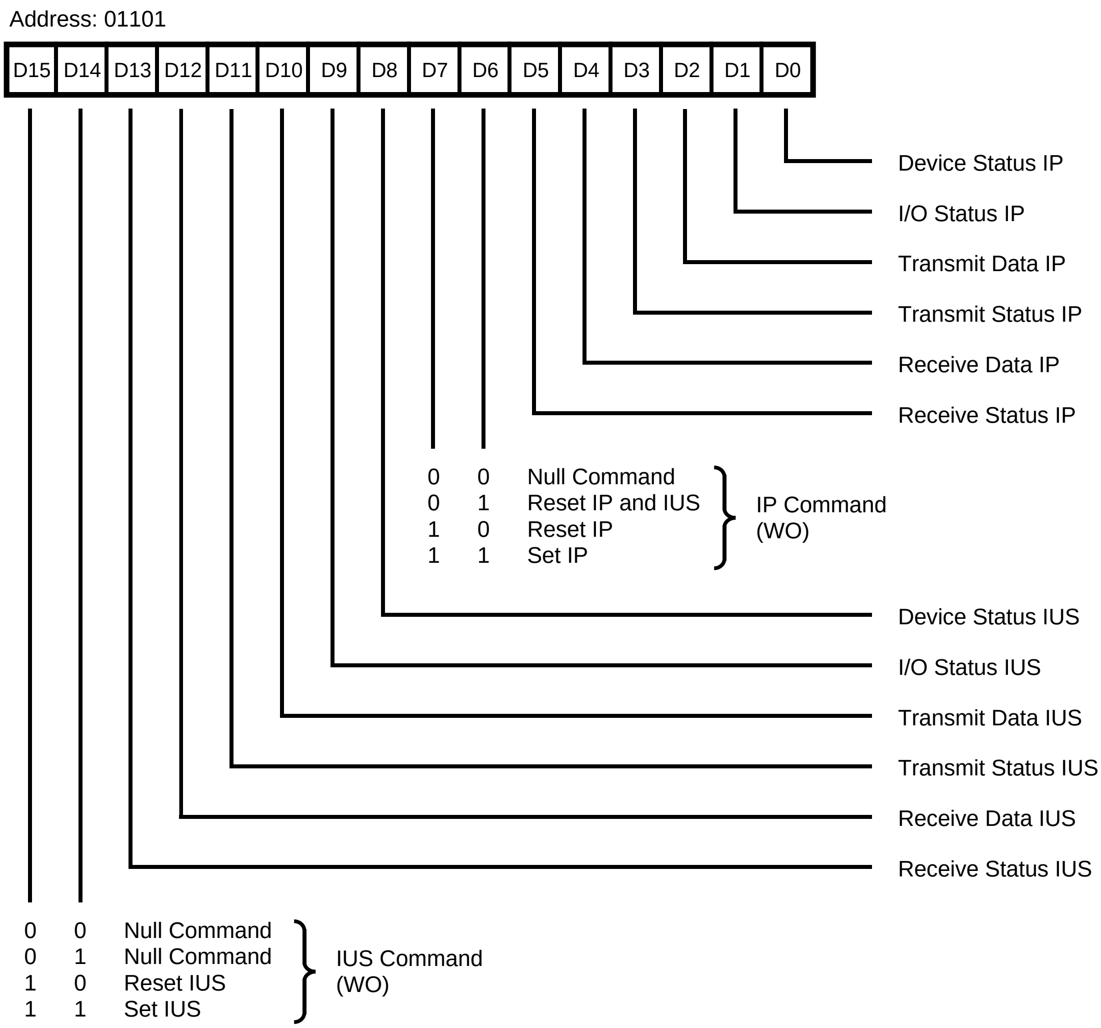



**Figure 59. Daisy-Chain Control Register (DCCR)**

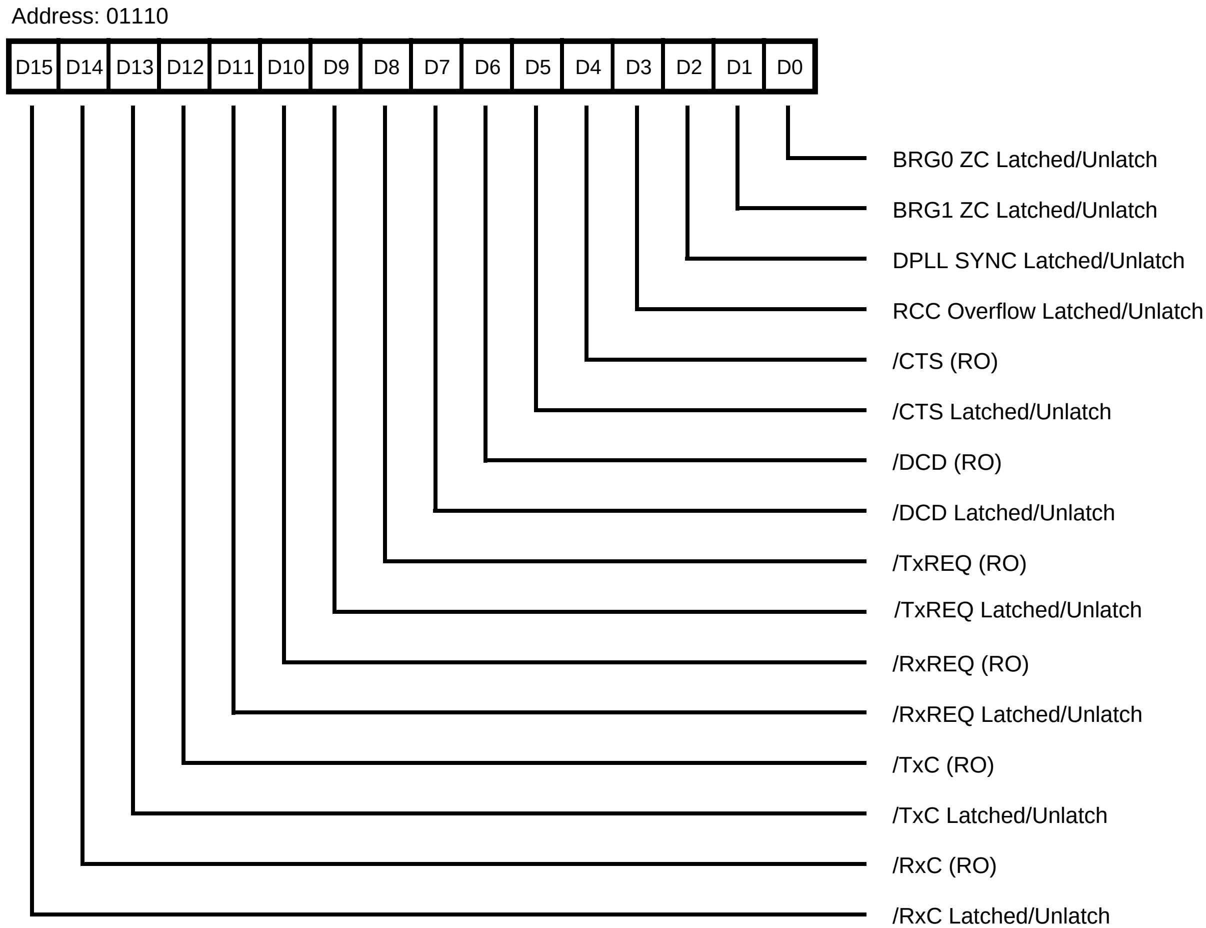



**Figure 60. Miscellaneous Interrupt Status Register (MISR)**

## CONTROL REGISTERS (Continued)

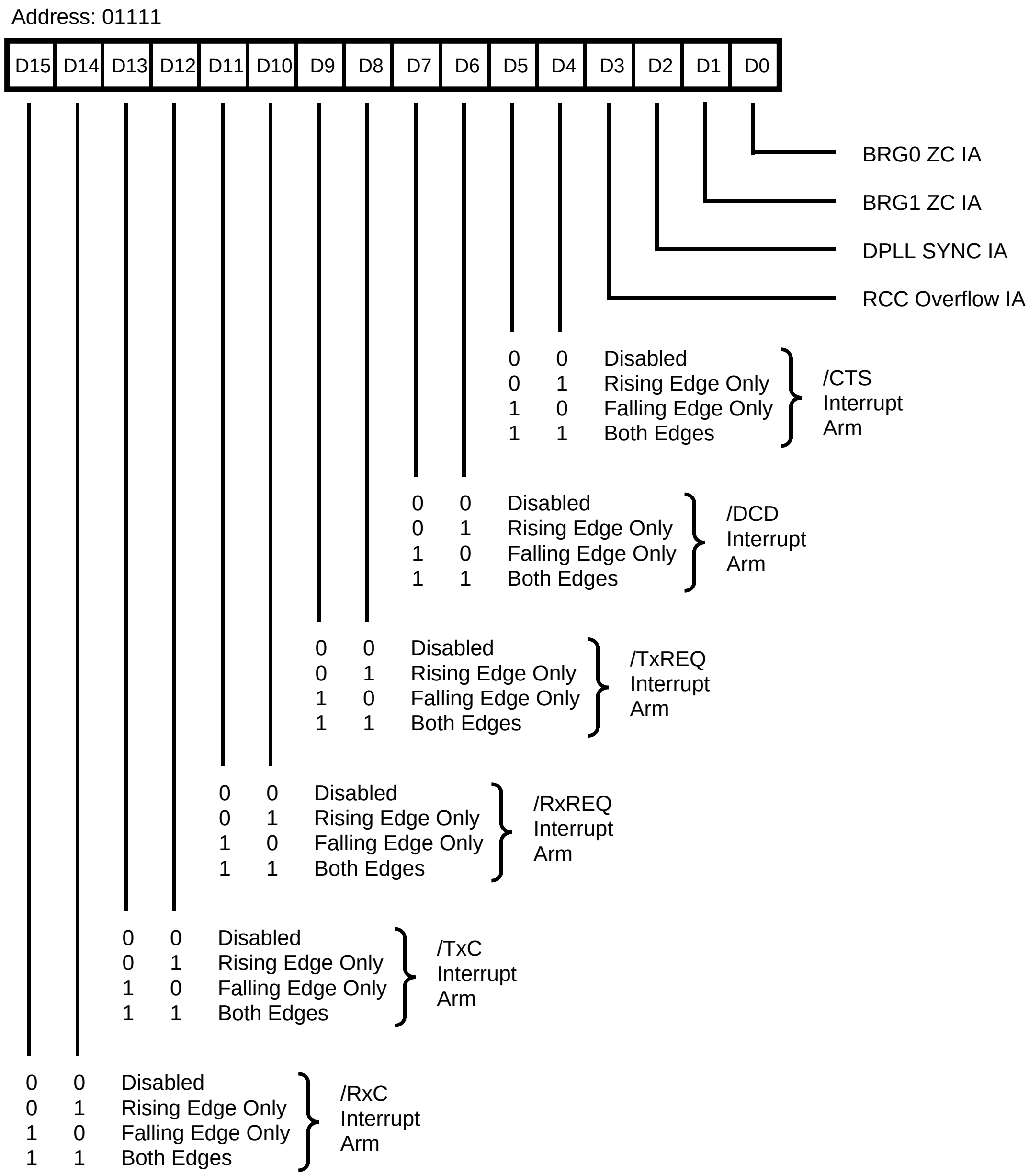



**Figure 61. Status Interrupt Control Register (SICR)**

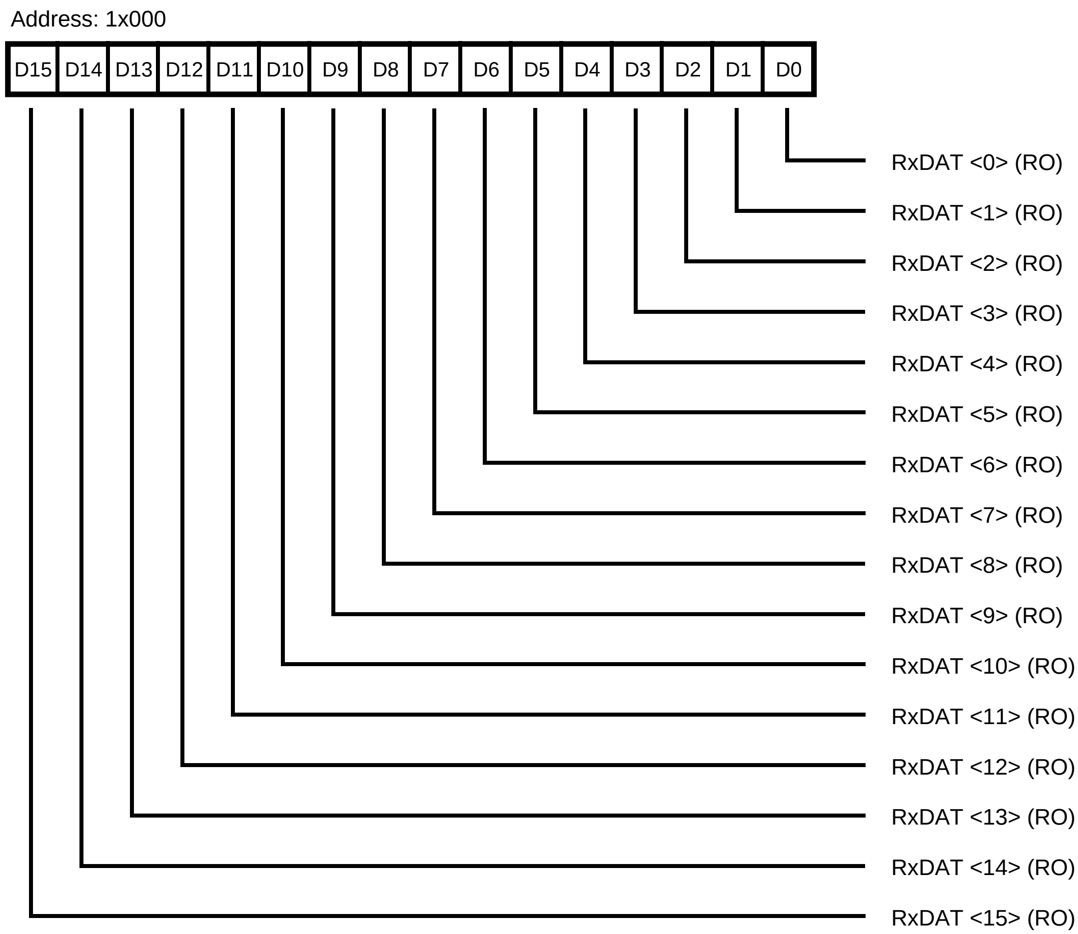



**Figure 62. Receive Data Register (RDR)**

## CONTROL REGISTERS (Continued)

Address: 10001

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Rx Enable (D1, D0)**

| D1 | D0 | |
|---|---|---|
| 0 | 0 | Disable Immediately |
| 0 | 1 | Disable After Reception |
| 1 | 0 | Enable Without Auto-Enables |
| 1 | 1 | Enable With Auto-Enables |

**Rx Character Length (D4, D3, D2)**

| D4 | D3 | D2 | |
|---|---|---|---|
| 0 | 0 | 0 | 8 Bits |
| 0 | 0 | 1 | 1 Bit |
| 0 | 1 | 0 | 2 Bits |
| 0 | 1 | 1 | 3 Bits |
| 1 | 0 | 0 | 4 Bits |
| 1 | 0 | 1 | 5 Bits |
| 1 | 1 | 0 | 6 Bits |
| 1 | 1 | 1 | 7 Bits |

D5: Rx Parity Enable

**Rx Parity Type (D7, D6)**

| D7 | D6 | |
|---|---|---|
| 0 | 0 | Even |
| 0 | 1 | Odd |
| 1 | 0 | Space |
| 1 | 1 | Mark |

D8: Abort Frame Status

D9: Rx CRC Enable

D10: Rx CRC Preset Value

**Rx CRC Polynomial (D12, D11)**

| D12 | D11 | |
|---|---|---|
| 0 | 0 | CRC-CCITT |
| 0 | 1 | CRC-16 |
| 1 | 0 | CRC-32 |
| 1 | 1 | Reserved |

**Rx Data Decoding (D15, D14, D13)**

| D15 | D14 | D13 | |
|---|---|---|---|
| 0 | 0 | 0 | NRZ |
| 0 | 0 | 1 | NRZB |
| 0 | 1 | 0 | NRZI-Mark |
| 0 | 1 | 1 | NRZI-Space |
| 1 | 0 | 0 | Biphase-Mark |
| 1 | 0 | 1 | Biphase-Space |
| 1 | 1 | 0 | Biphase-Level |
| 1 | 1 | 1 | Diff. Biphase-Level |

**Figure 63. Receive Mode Register (RMR)**

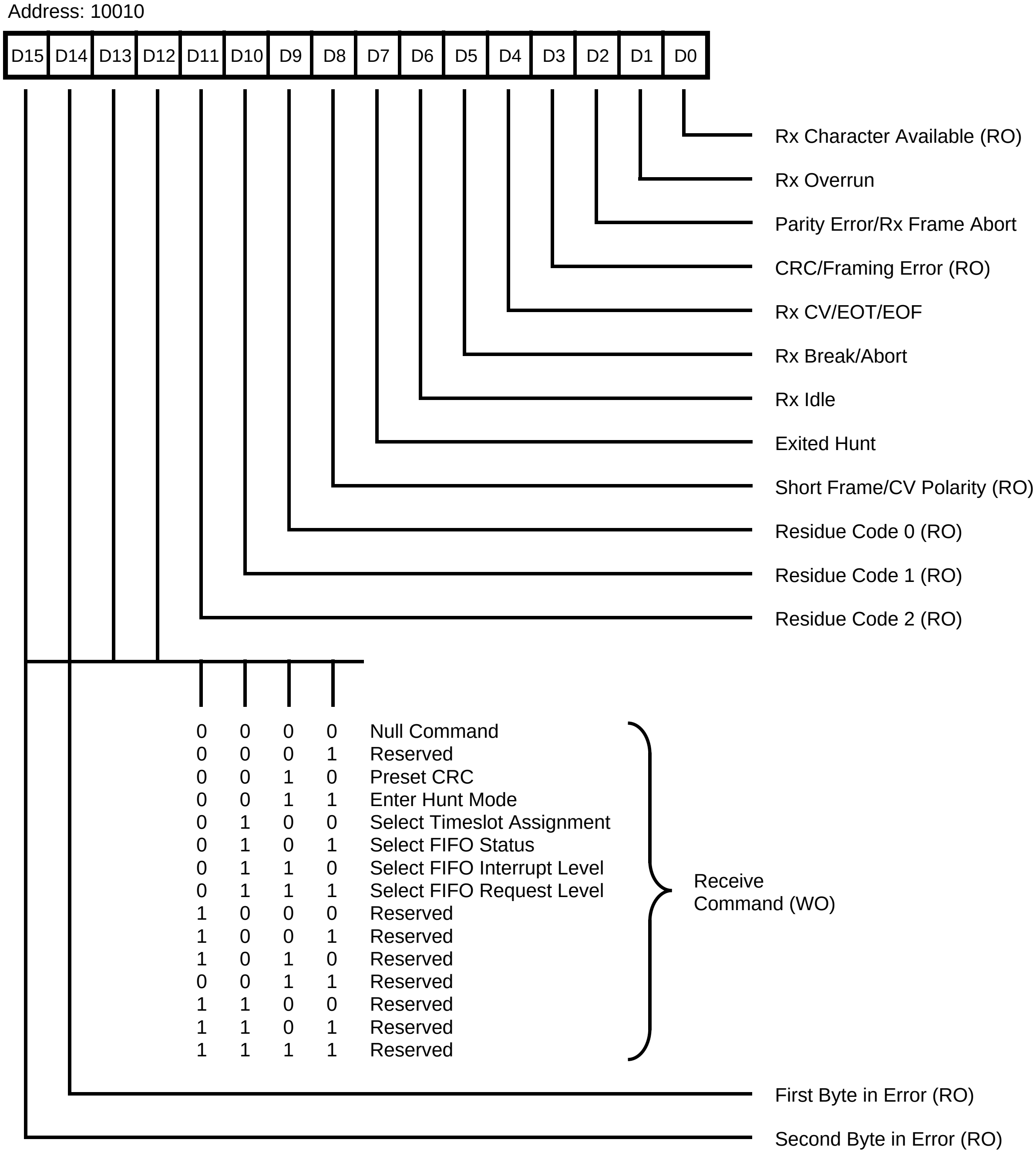



**Figure 64. Receive Command Status Register (RCSR)**

## CONTROL REGISTERS (Continued)

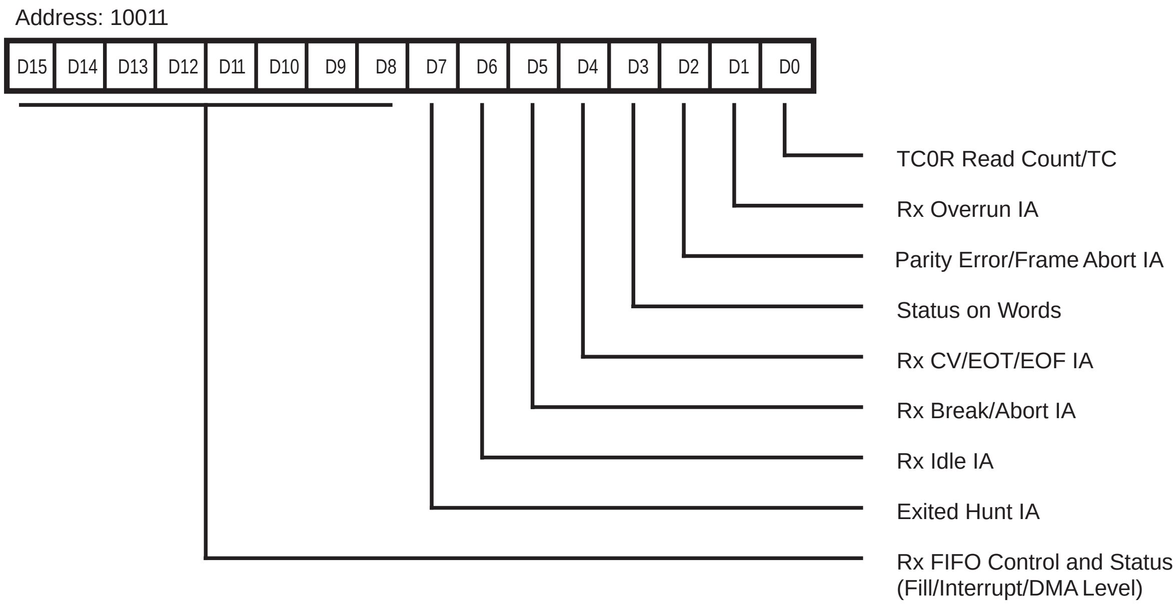



**Figure 65a. Receive Interrupt Control Register (RICR)**

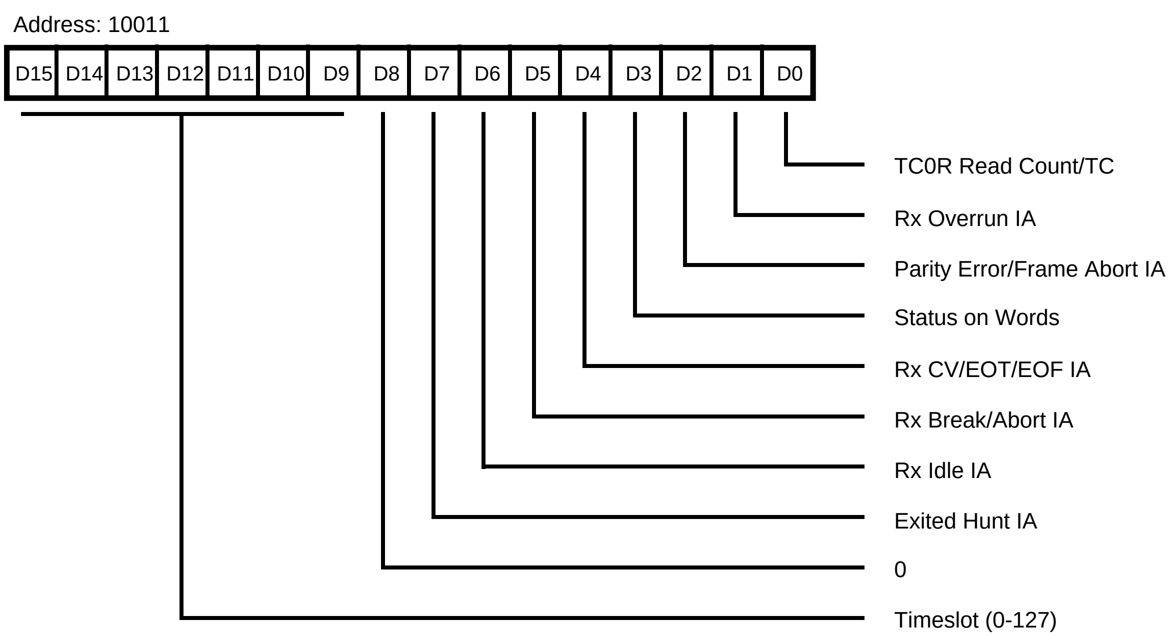



**Figure 65b. Receive Interrupt Control Register (RICR)**

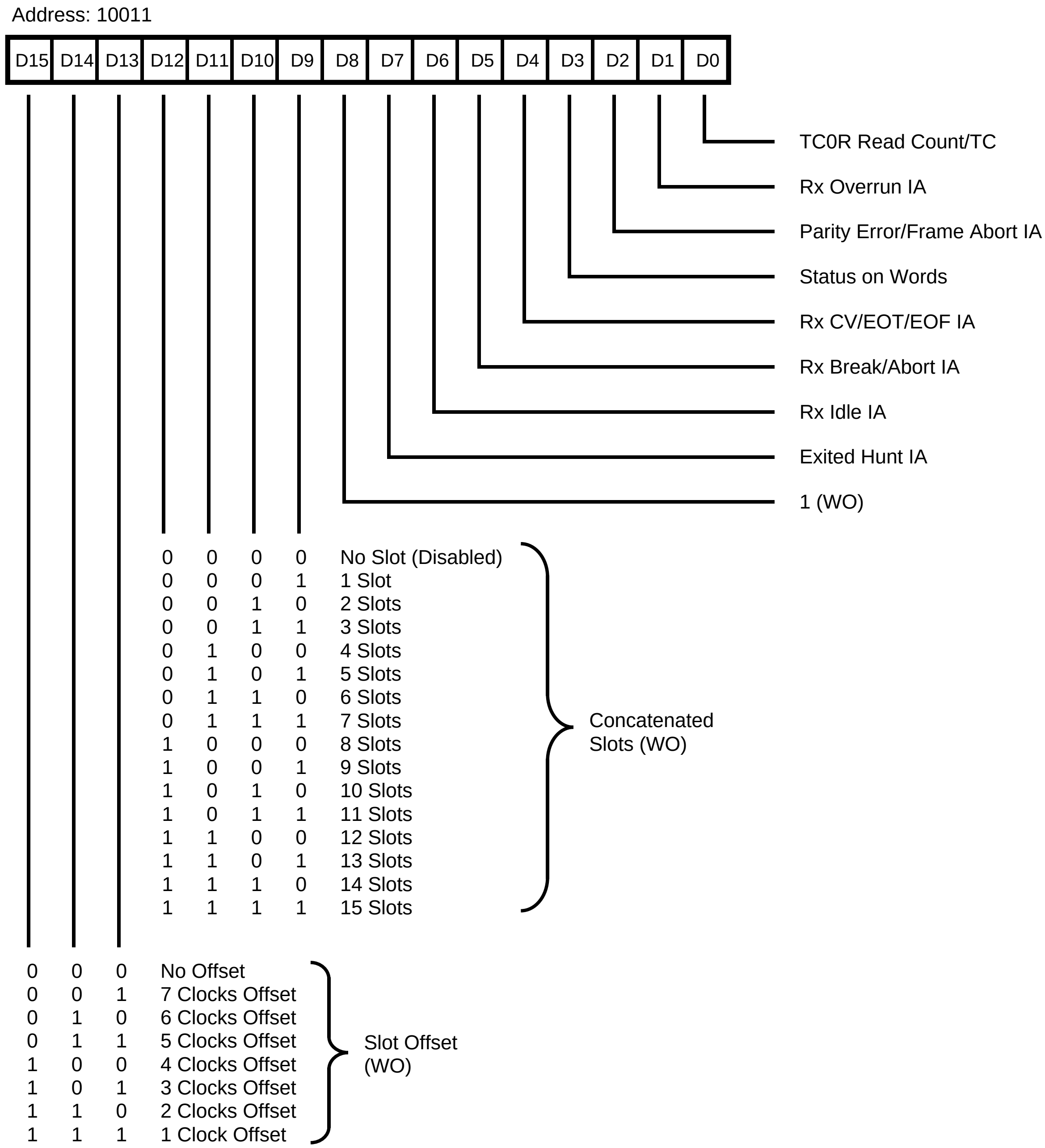



**Figure 65c. Receive Interrupt Control Register (RICR)**

## CONTROL REGISTERS (Continued)

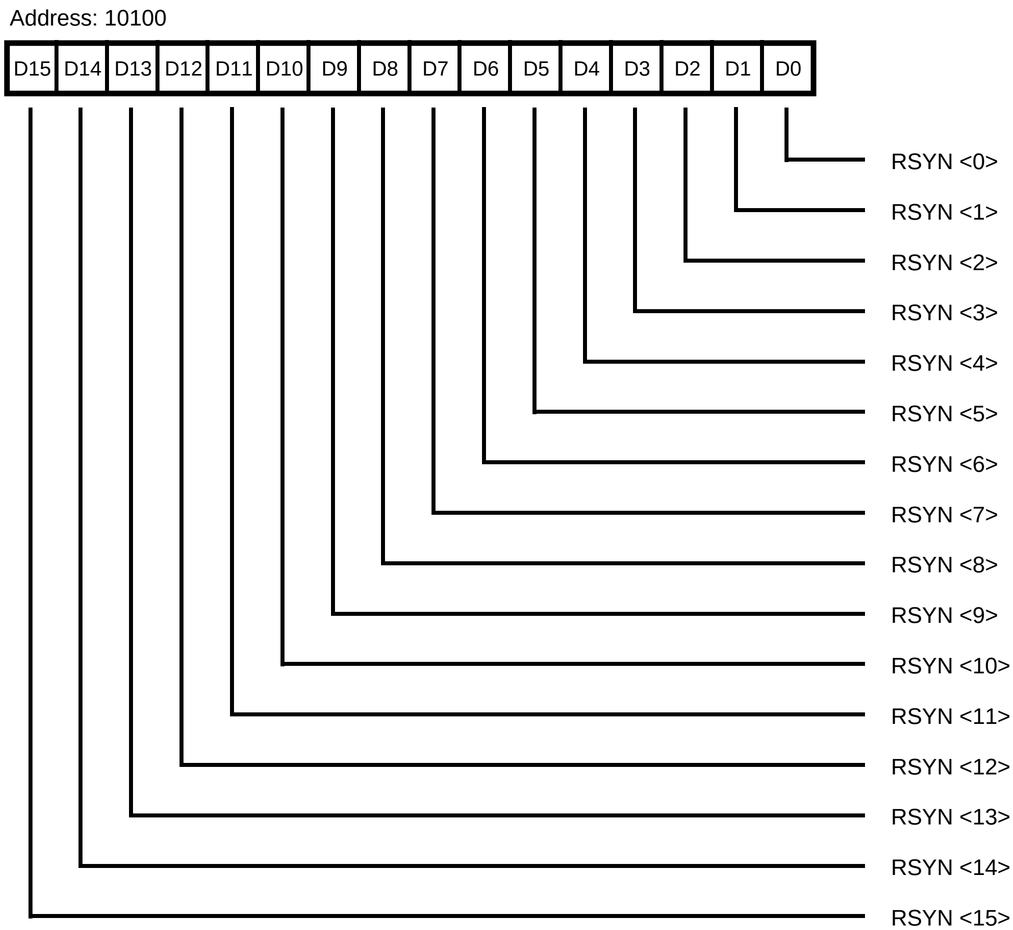



**Figure 66. Receive Sync Register (RSR)**

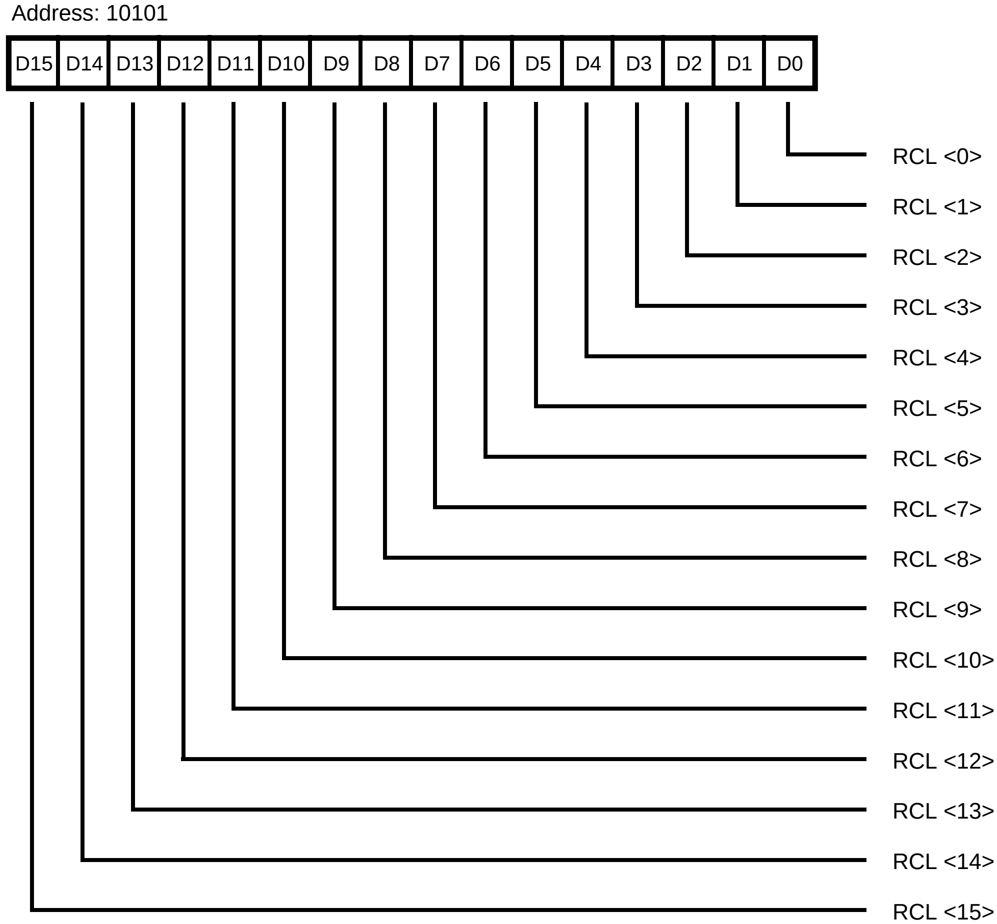



**Figure 67. Receive Count Limit Register (RCLR)**

## CONTROL REGISTERS (Continued)

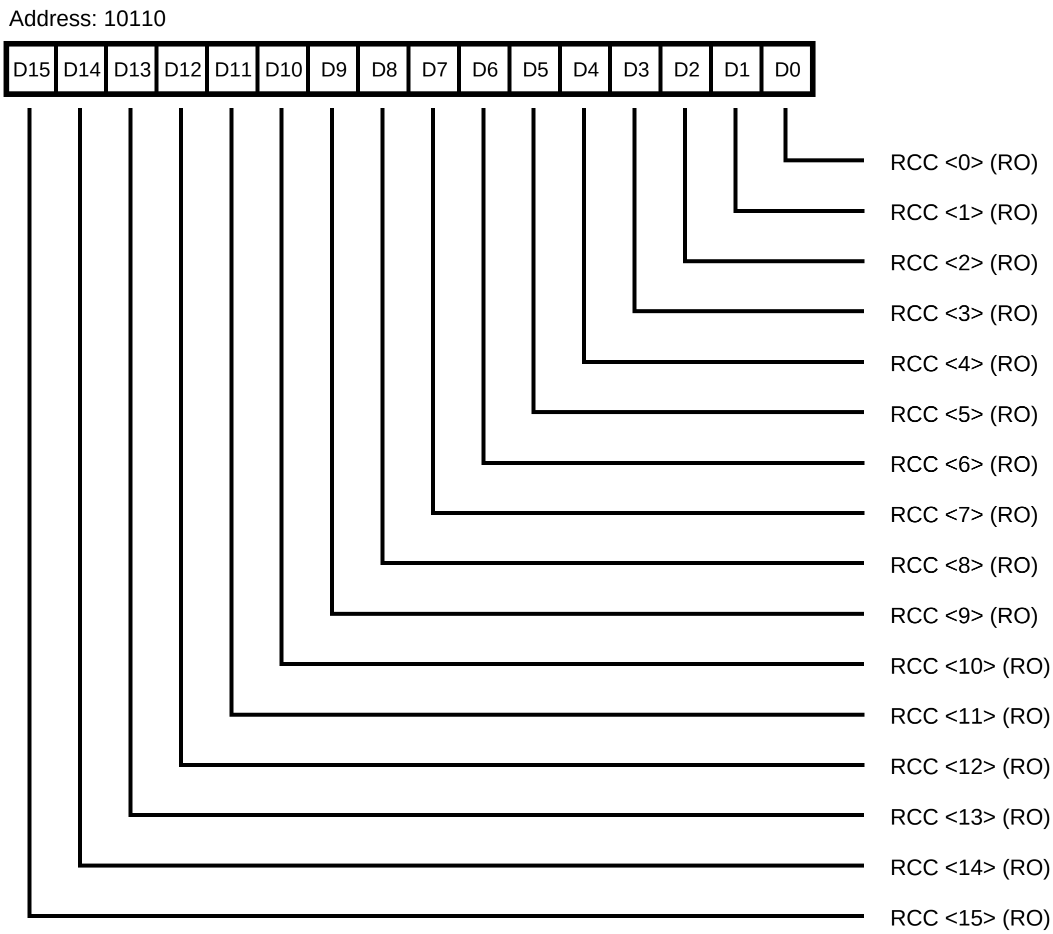



**Figure 68. Receive Character Count Register (RCCR)**

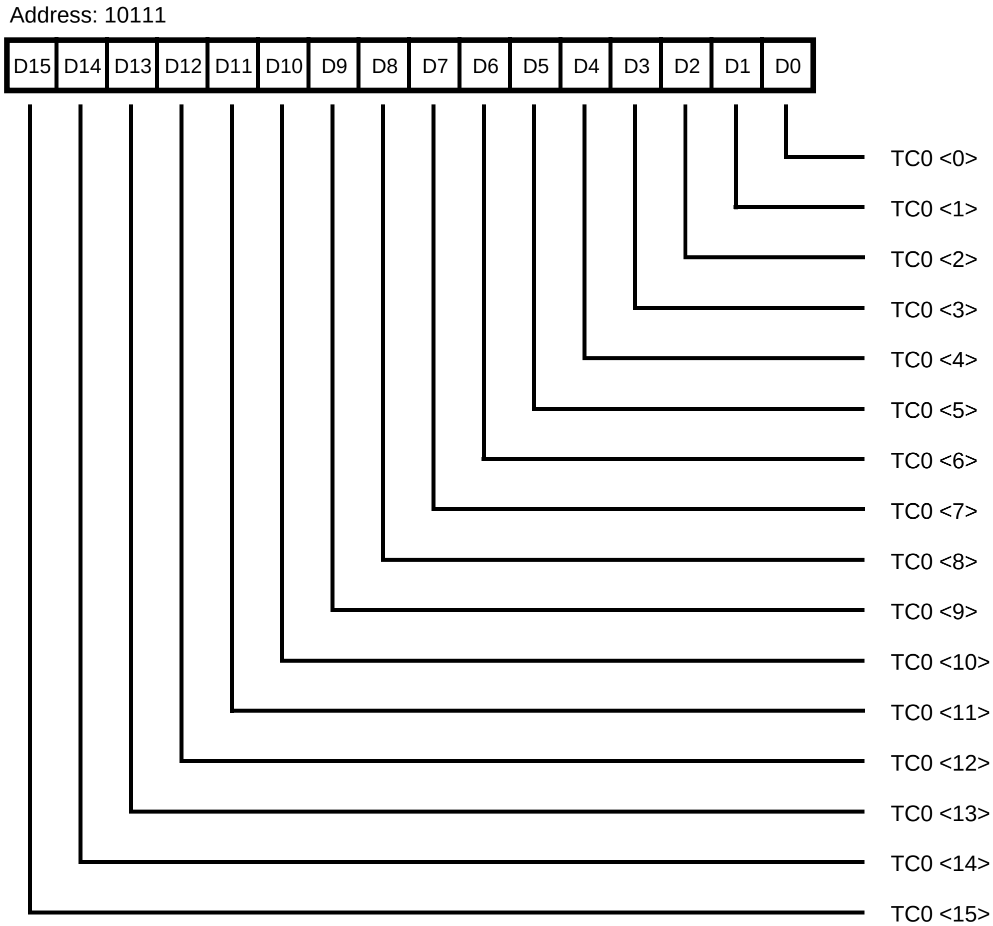



**Figure 69. Time Constant 0 Register (TC0R)**

## CONTROL REGISTERS (Continued)

Address: 1x000

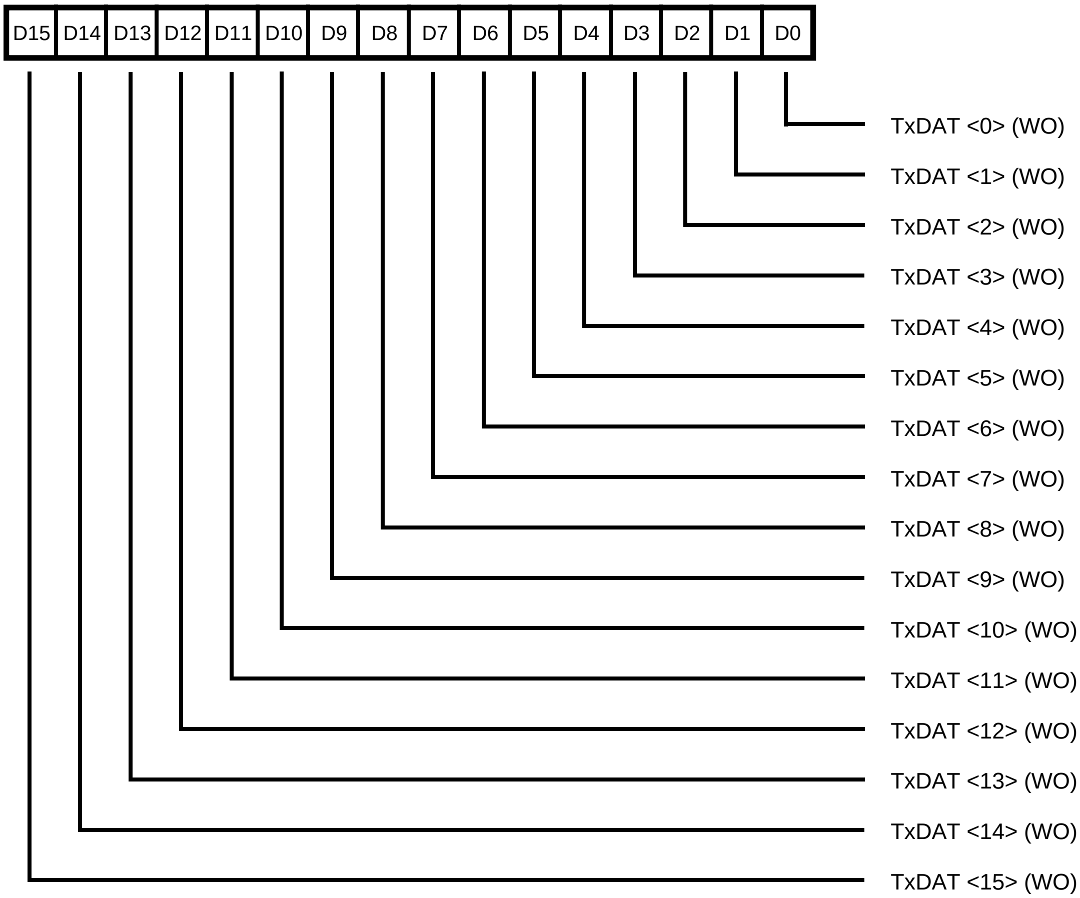



**Figure 70. Transmit Data Register (TDR)**

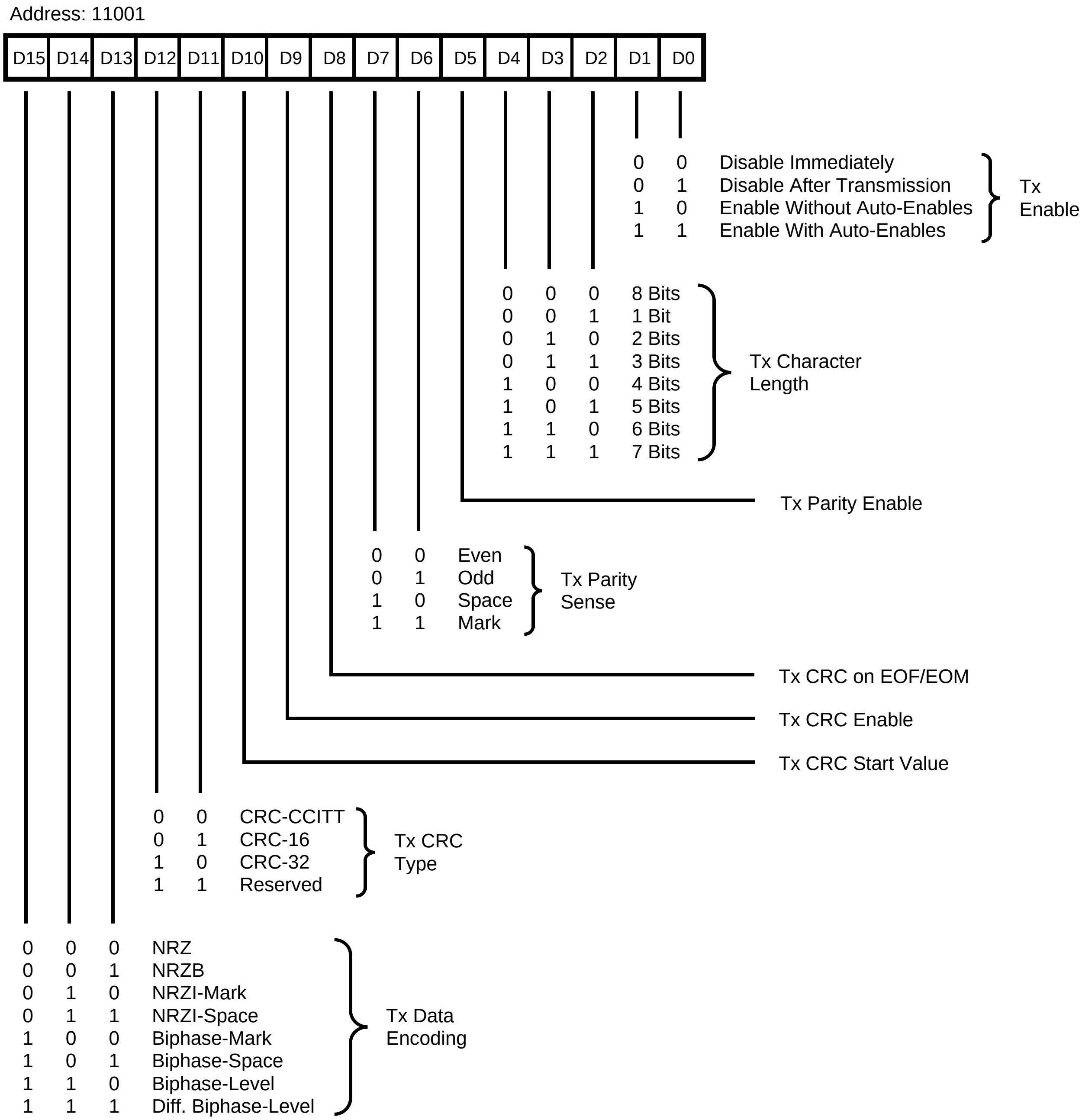



**Figure 71. Transmit Mode Register (TMR)**

## CONTROL REGISTERS (Continued)

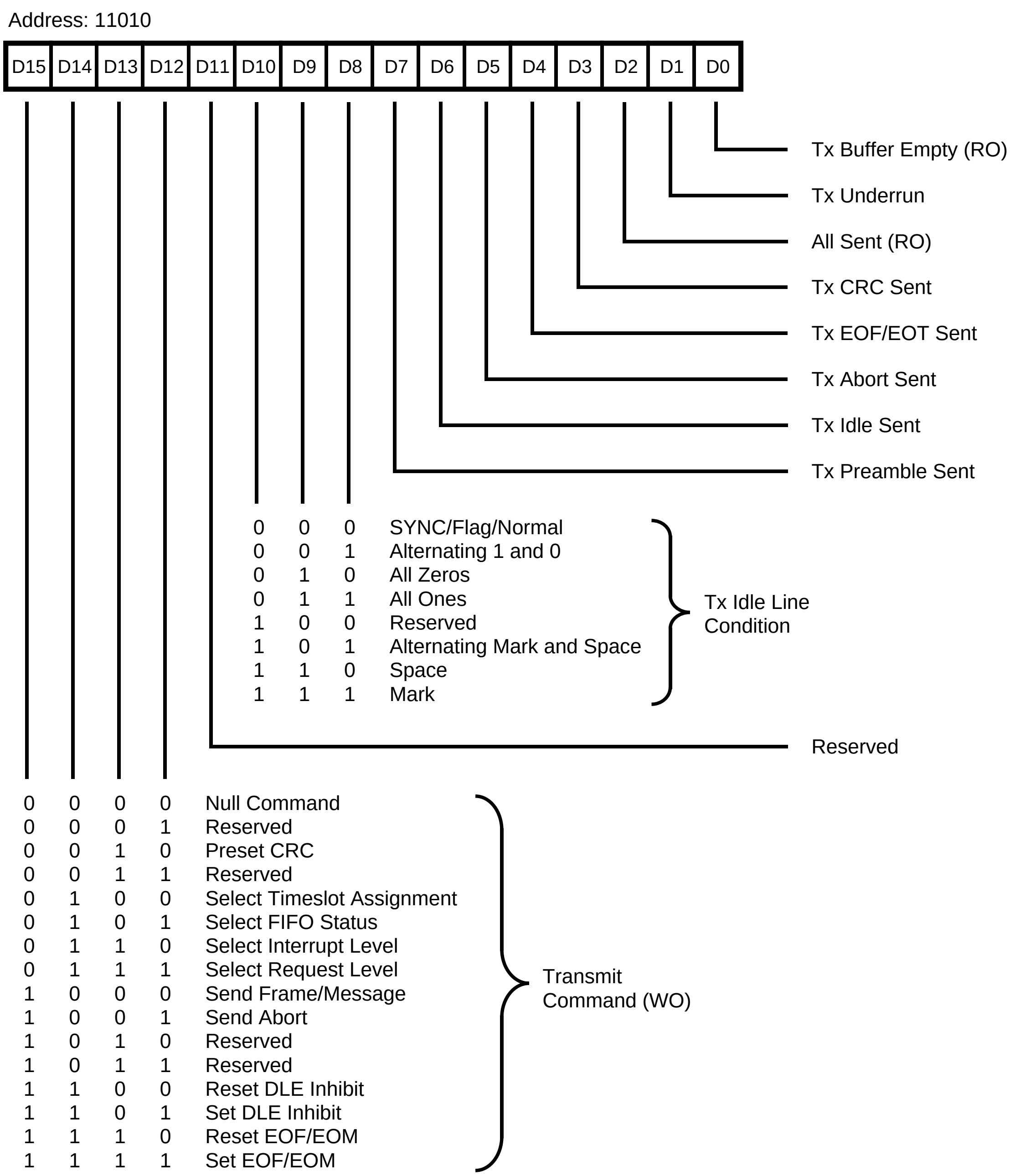



**Figure 72. Transmit Command/Status Register (TCSR)**

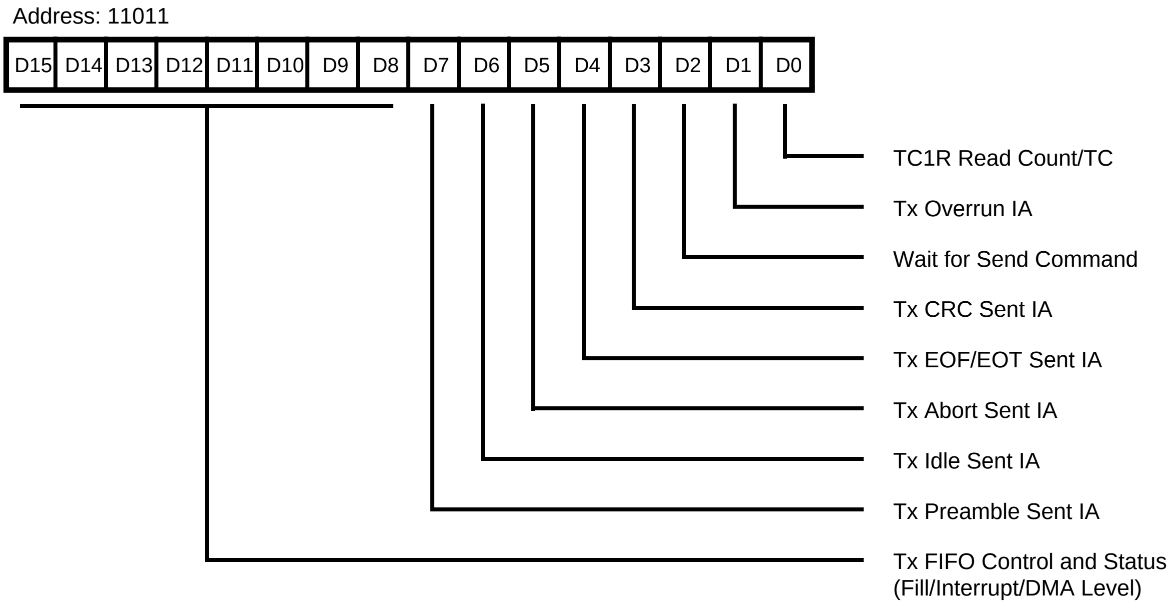



**Figure 73a. Transmit Interrupt Control Register (TICR)**

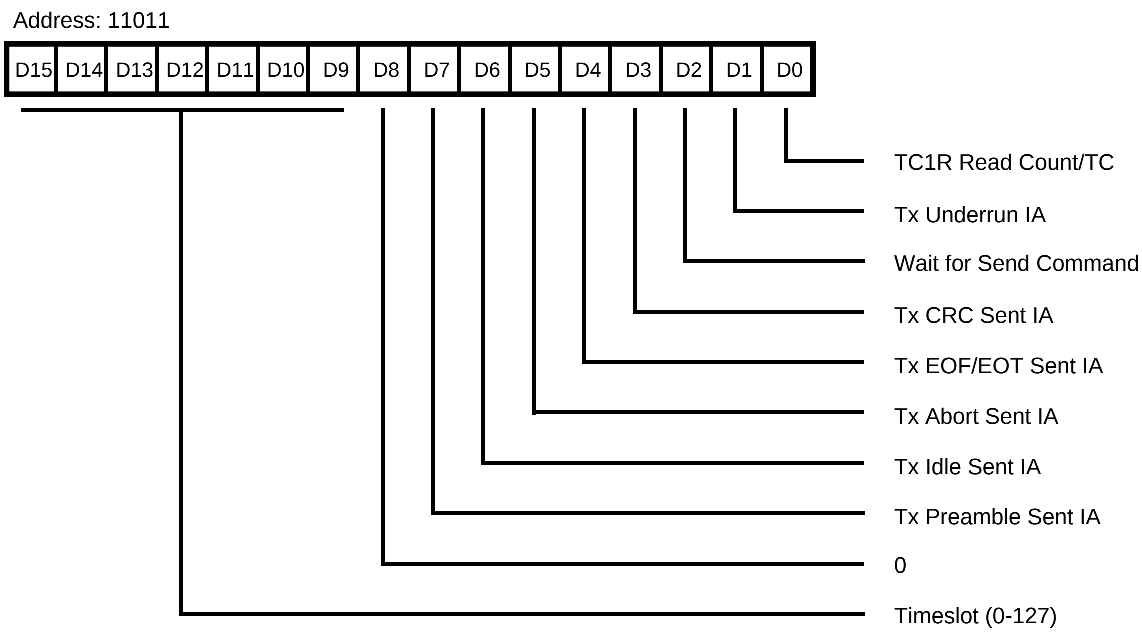



**Figure 73b. Transmit Interrupt Control Register (TICR)**

## CONTROL REGISTERS (Continued)

Address: 11011

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| Bit | Function |
|---|---|
| D0 | TC1R Read Count/TC |
| D1 | Tx Underrun IA |
| D2 | Wait for Send Command |
| D3 | Tx CRC Sent IA |
| D4 | Tx EOF/EOT Sent IA |
| D5 | Tx Abort Sent IA |
| D6 | Tx Idle Sent IA |
| D7 | Tx Preamble Sent IA |
| D8 | 1 (WO) |

Concatenated Slots (WO)

| D12 | D11 | D10 | D9 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | No Slot (Disabled) |
| 0 | 0 | 0 | 1 | 1 Slot |
| 0 | 0 | 1 | 0 | 2 Slots |
| 0 | 0 | 1 | 1 | 3 Slots |
| 0 | 1 | 0 | 0 | 4 Slots |
| 0 | 1 | 0 | 1 | 5 Slots |
| 0 | 1 | 1 | 0 | 6 Slots |
| 0 | 1 | 1 | 1 | 7 Slots |
| 1 | 0 | 0 | 0 | 8 Slots |
| 1 | 0 | 0 | 1 | 9 Slots |
| 1 | 0 | 1 | 0 | 10 Slots |
| 1 | 0 | 1 | 1 | 11 Slots |
| 1 | 1 | 0 | 0 | 12 Slots |
| 1 | 1 | 0 | 1 | 13 Slots |
| 1 | 1 | 1 | 0 | 14 Slots |
| 1 | 1 | 1 | 1 | 15 Slots |

Slot Offset (WO)

| D15 | D14 | D13 | |
|---|---|---|---|
| 0 | 0 | 0 | No Offset |
| 0 | 0 | 1 | 7 Clocks Offset |
| 0 | 1 | 0 | 6 Clocks Offset |
| 0 | 1 | 1 | 5 Clocks Offset |
| 1 | 0 | 0 | 4 Clocks Offset |
| 1 | 0 | 1 | 3 Clocks Offset |
| 1 | 1 | 0 | 2 Clocks Offset |
| 1 | 1 | 1 | 1 Clock Offset |

**Figure 73c. Transmit Interrupt Control Register (TICR)**

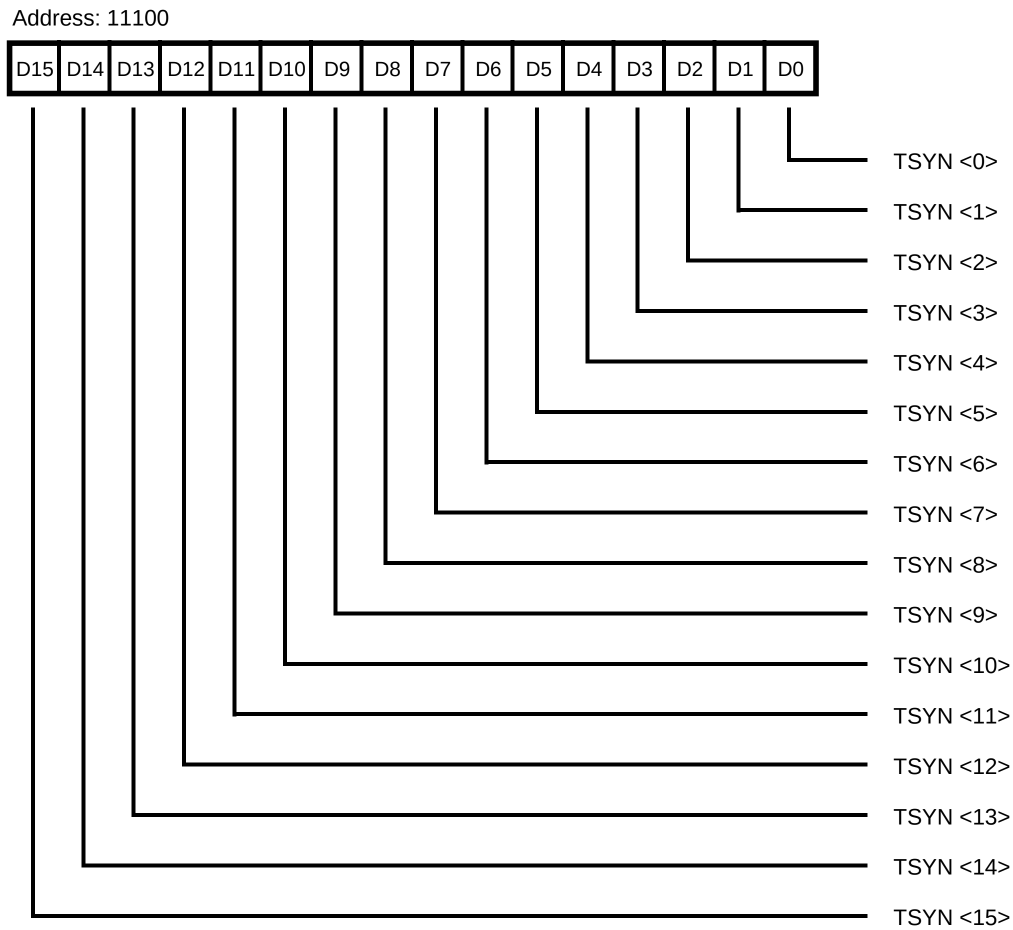



**Figure 74. Transmit Sync Register (TSR)**

## CONTROL REGISTERS (Continued)

Address: 11101

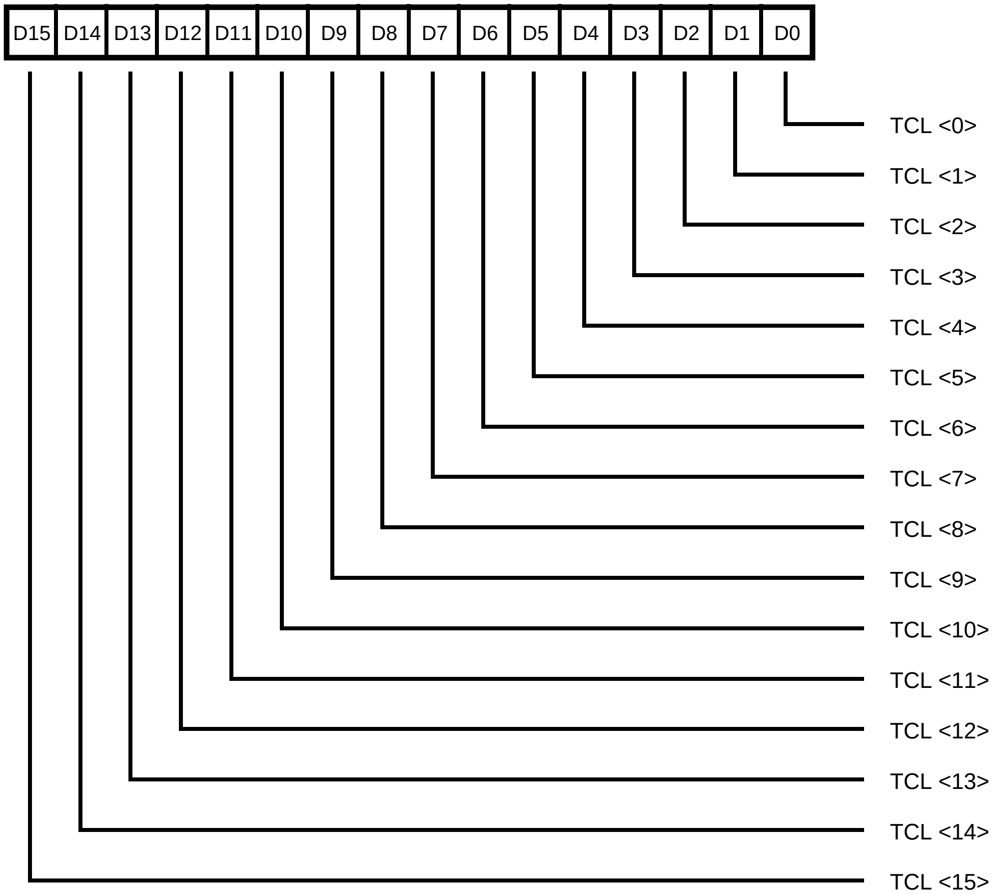



**Figure 75. Transmit Count Limit Register (TCLR)**

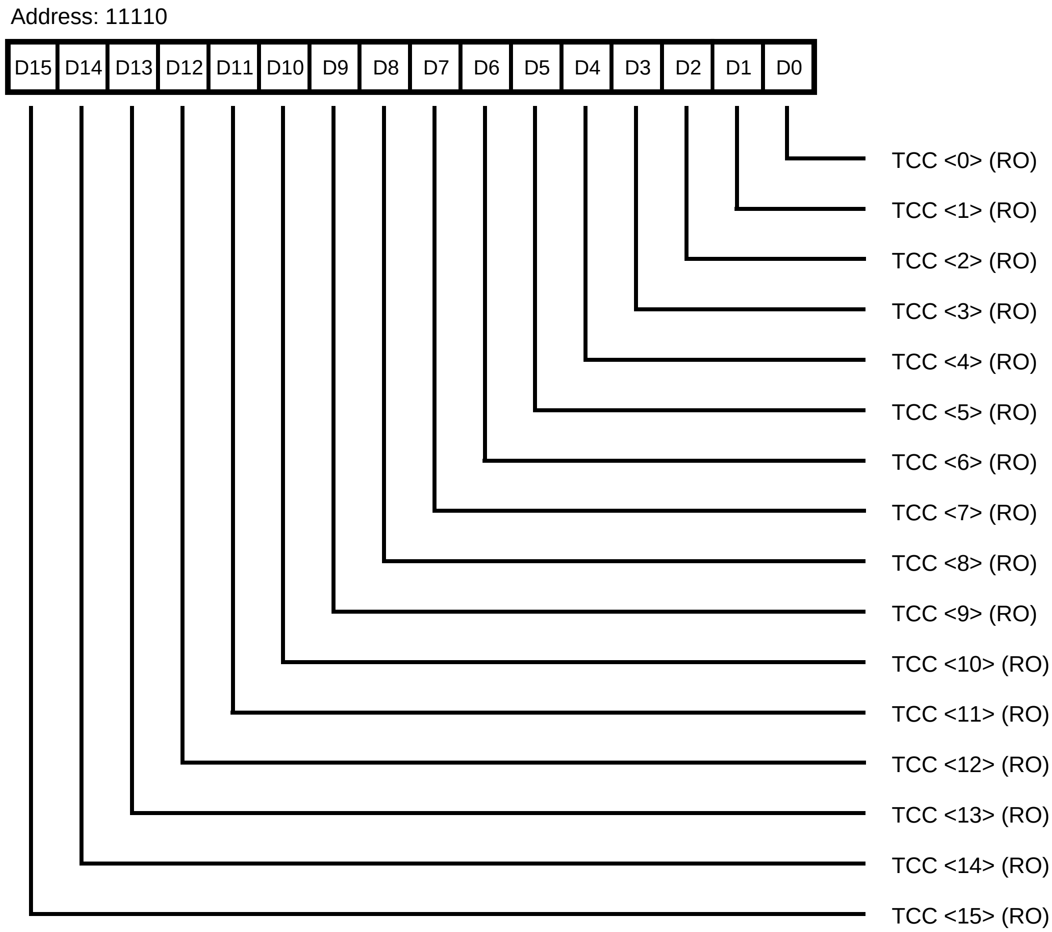



**Figure 76. Transmit Character Count Register (TCCR)**

## CONTROL REGISTERS (Continued)

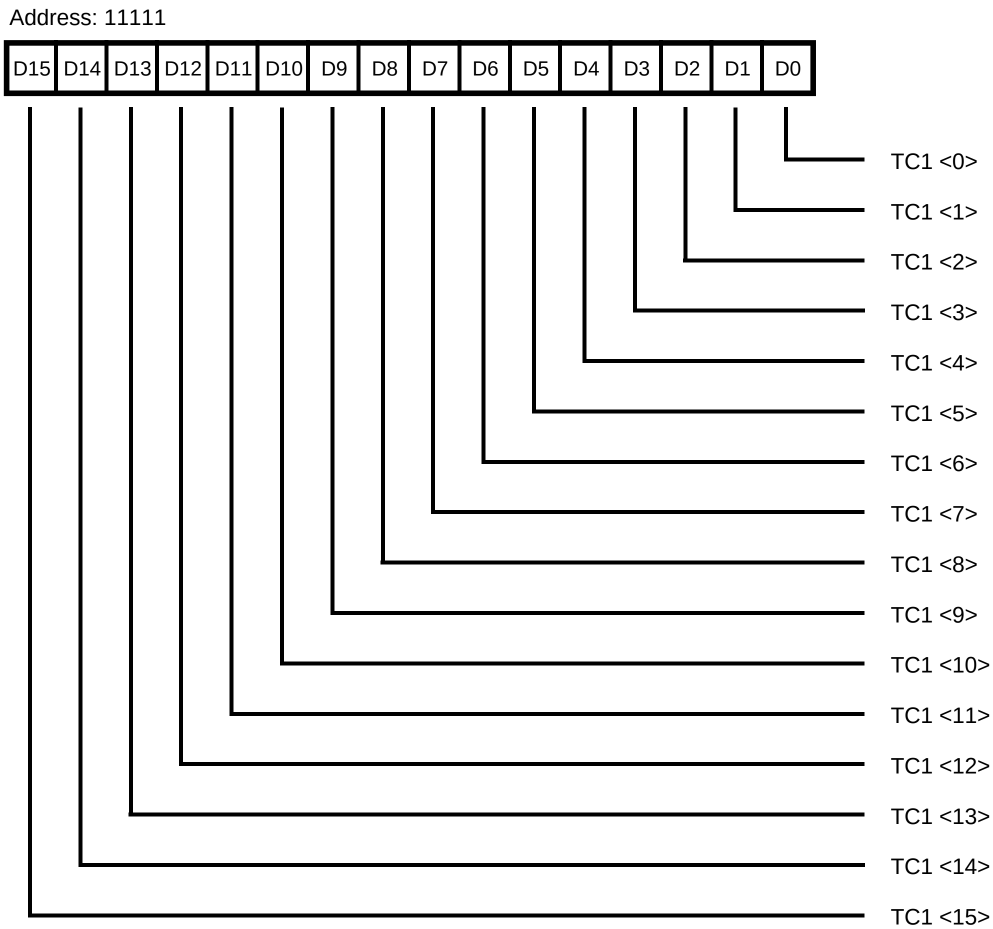
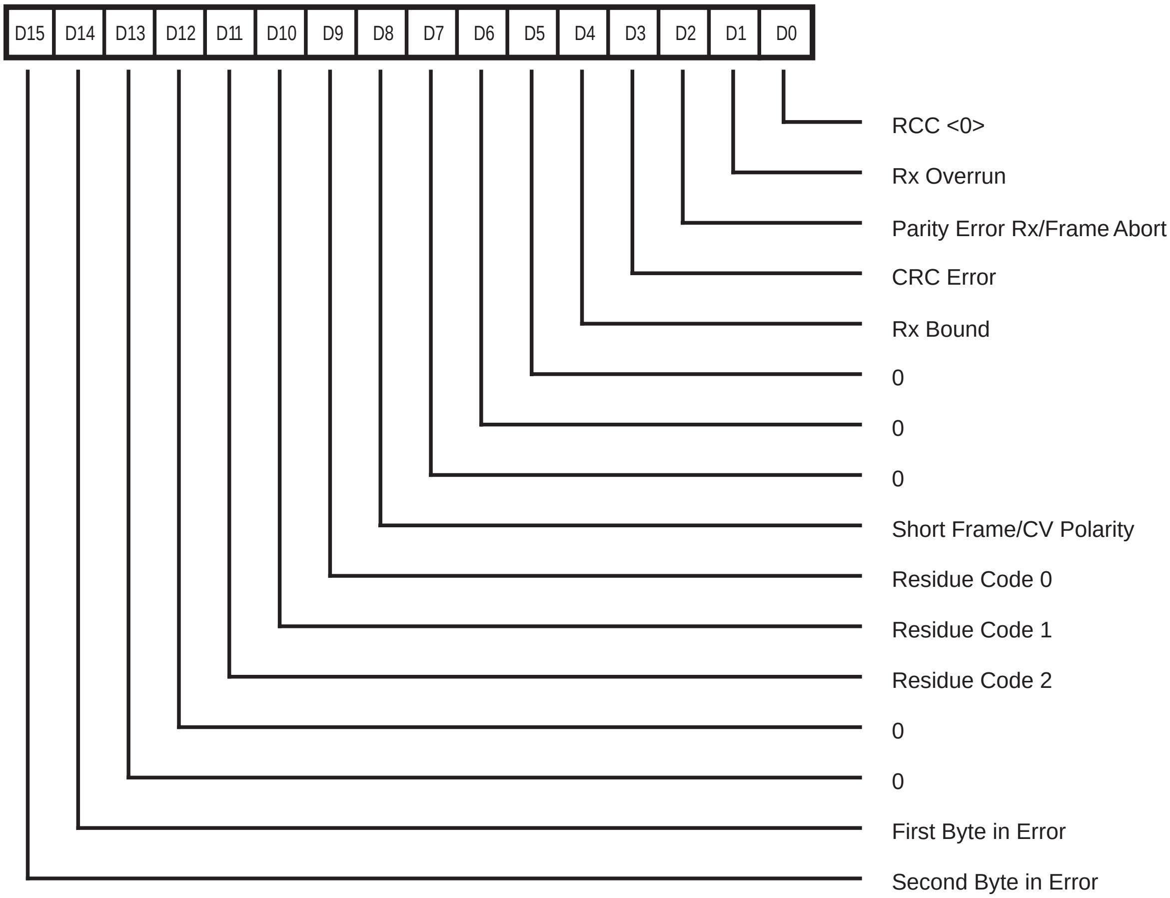



**Figure 77. Time Constant 1 Register (TC1R)**

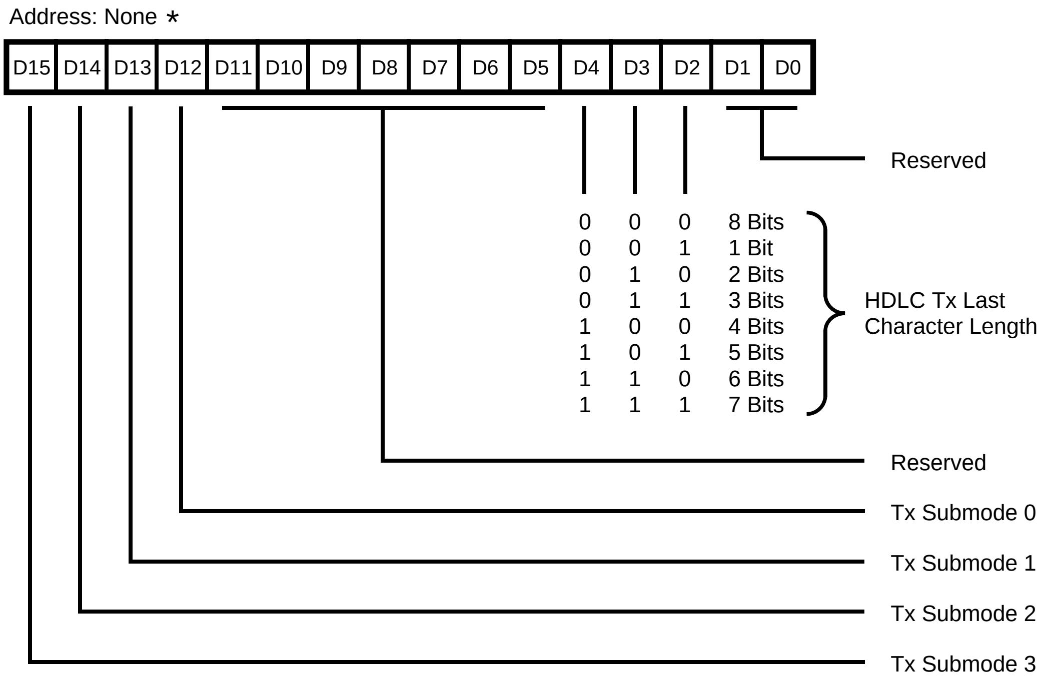
Address: None *




* Refer to Figure 22 (Channel Control Register)
Bits 6-7 for Access Method

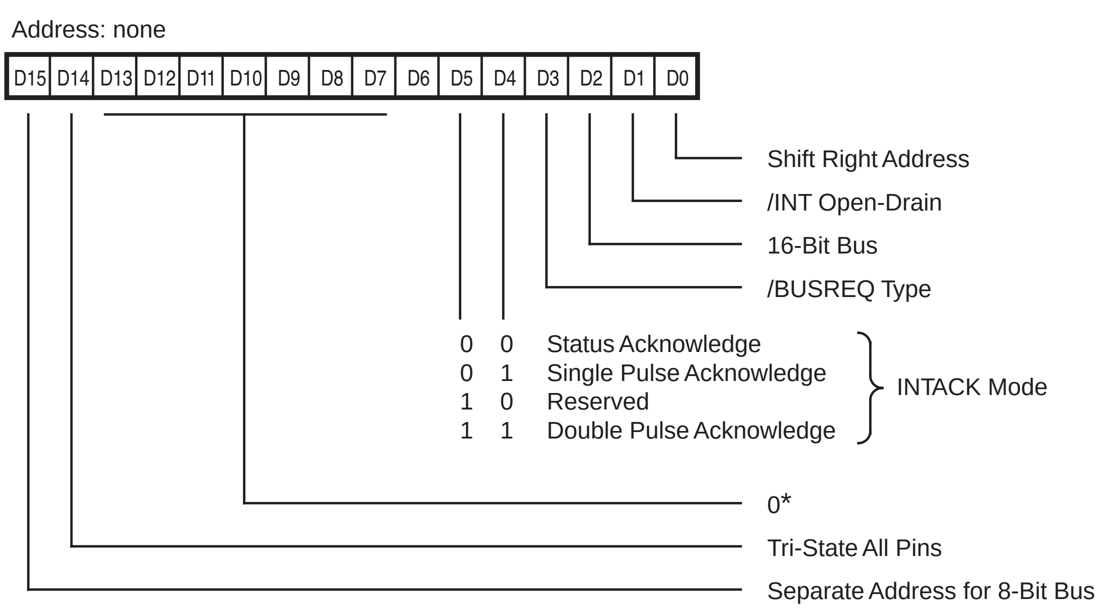
**Figure 78. Receive Status Block Register (RSBR)**

## CONTROL REGISTERS (Continued)




* Refer to Figure 22 (Channel Control Register) Bits15-14 for Access Method

**Figure 79. Transmit Status Block Register (TSBR)**




* Must be programmed as 0.

**Figure 80. Bus Configuration Register (BCR)**

## 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 |

**Notes:**
* Voltage on all pins with respect to GND.
† See Ordering Information.

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 Ground. Positive current flows into the referenced pin (Figure 81). Standard conditions are as follows:

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

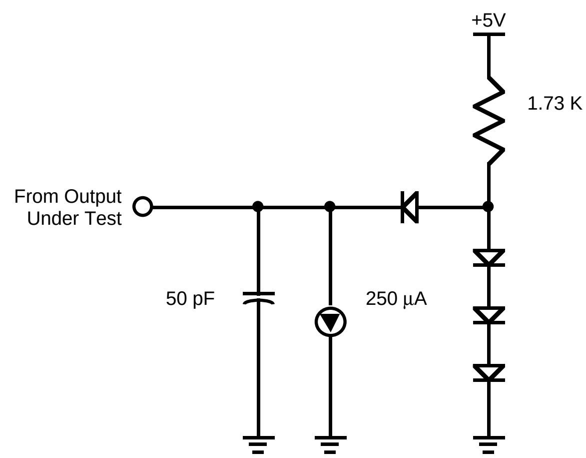



**Figure 81. Standard Test Load**

## CAPACITANCE

| Symbol | Parameter | Min | Max | Unit | Condition |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | | 10 | pF | * |
| $C_{OUT}$ | Output Capacitance | | 15 | pF | * |
| $C_{I/O}$ | Bidirectional Capacitance | | 20 | pF | |

**Notes:**
F = 1 MHz, over specified temperature range.
* Unmeasured pins returned to Ground.

## MISCELLANEOUS

Transistor Count - 100,000

## TEMPERATURE RANGE

Standard = 0°C to 70°C

## IUSC TIMING

The IUSC interface timing is similar to that found on a static RAM, except that it is much more flexible. Up to four separate timing strobe signals are present on the interface: /DS, /RD, /WR and /INTACK. Only one of these timing strobes is active at any time. Should the external logic activate more than one of these strobes at the same time, the IUSC will enter a pre-reset state. This state 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. IUSC Timing diagrams are shown from Figure 82 through Figure 106.

## 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 < $V_{IN}$ < +2.4V |
| $I_{OL}$ | Output Leakage | | | +10.00 | µA | 0.4 < $V_{OUT}$ < +2.4V |
| $I_{CC1}$ | $V_{CC}$ Supply Current | | 7 | 50 | mA | $V_{CC}$ = 5V $V_{IH}$ = 4.8V $V_{IL}$ = 0.2V |

**Note:**
$V_{CC}$ = 5V ± 10% unless otherwise specified, over specified temperature range.

## AC CHARACTERISTICS

Timing Diagrams (Figures 82-104)

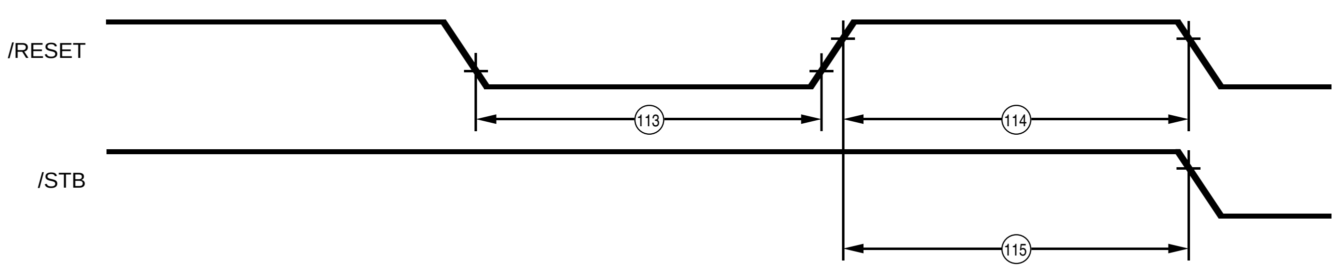



**Figure 82. Reset Timing**

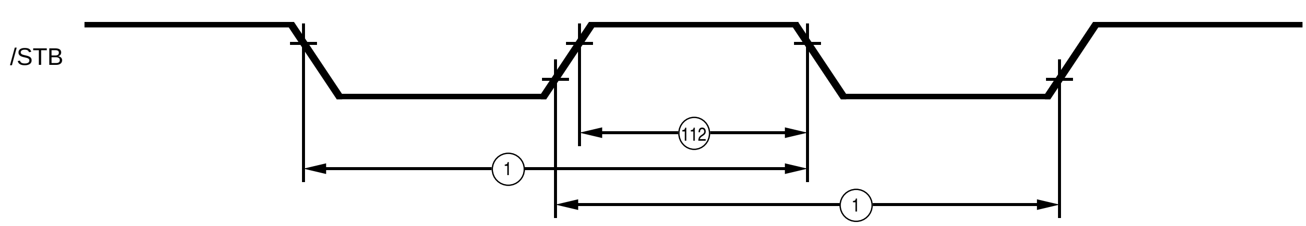



**Figure 83. Bus Cycle Timing**

**Note:**
/STB is any of the following: /DS, /RD, /WR or Pulsed /INTACK.

# AC CHARACTERISTICS
## Timing Diagrams (Continued)

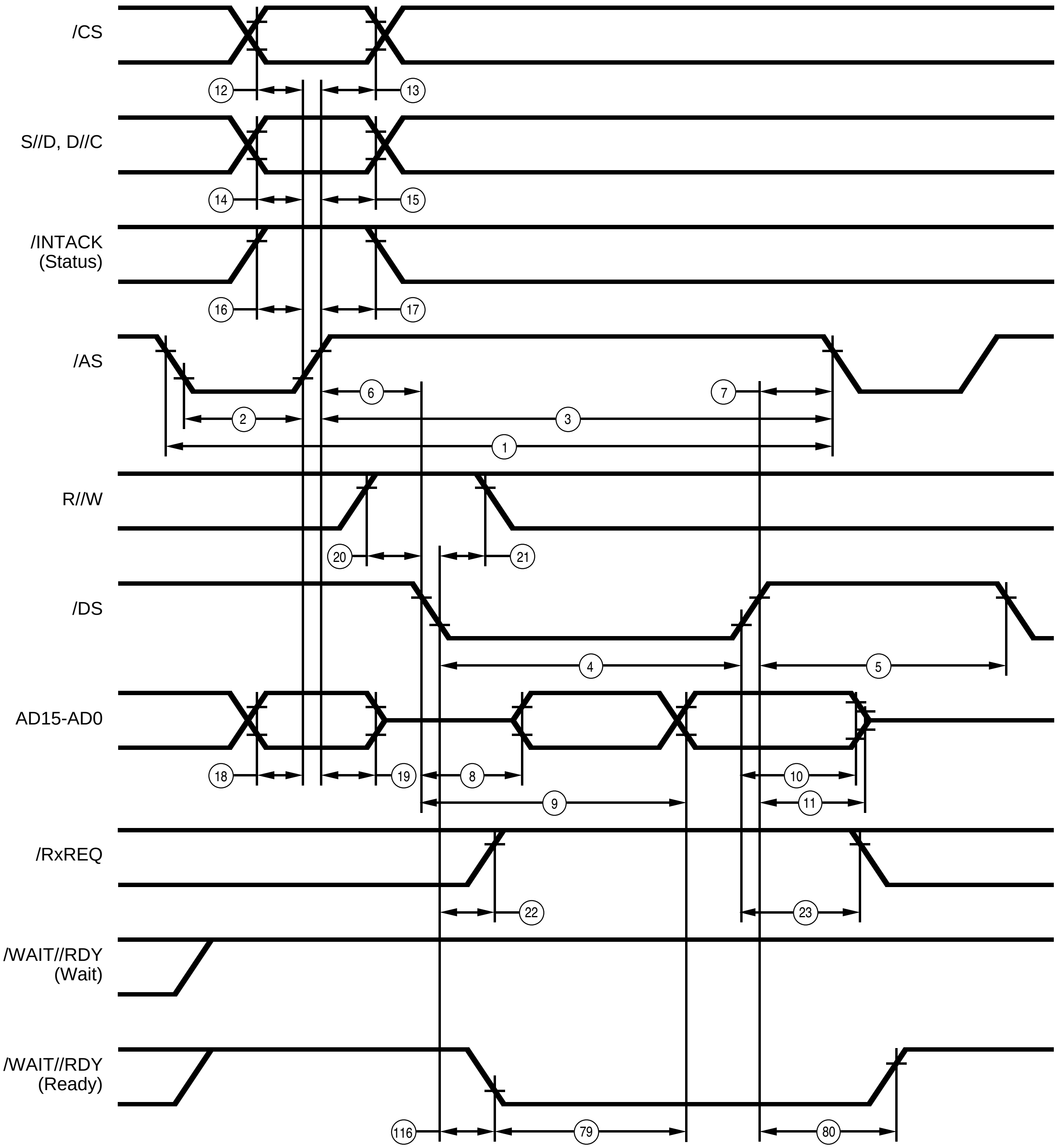



**Figure 84. Multiplexed /DS Read Cycle**

## AC CHARACTERISTICS
Timing Diagrams (Continued)

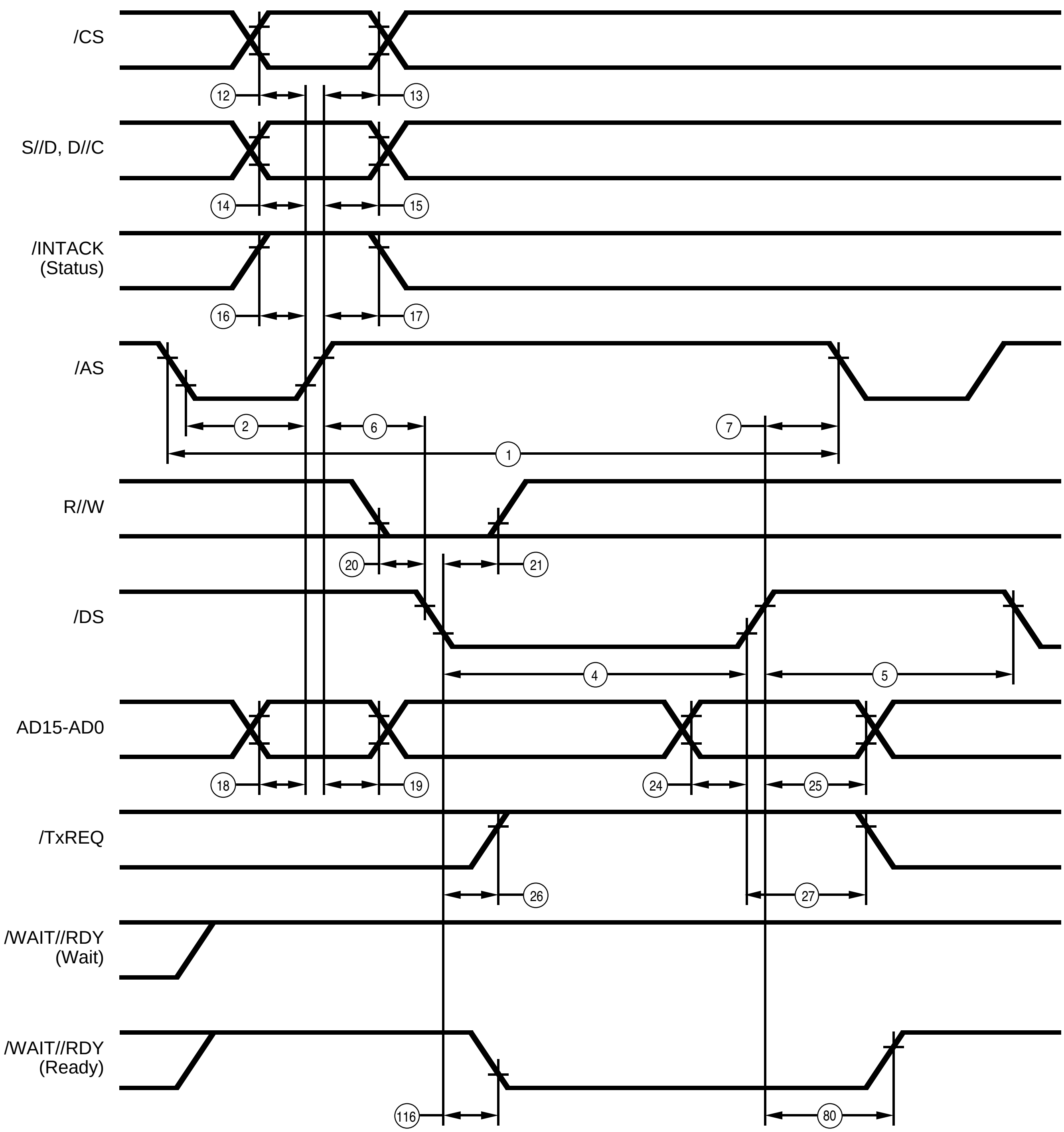



**Figure 85. Multiplexed /DS Write Cycle**

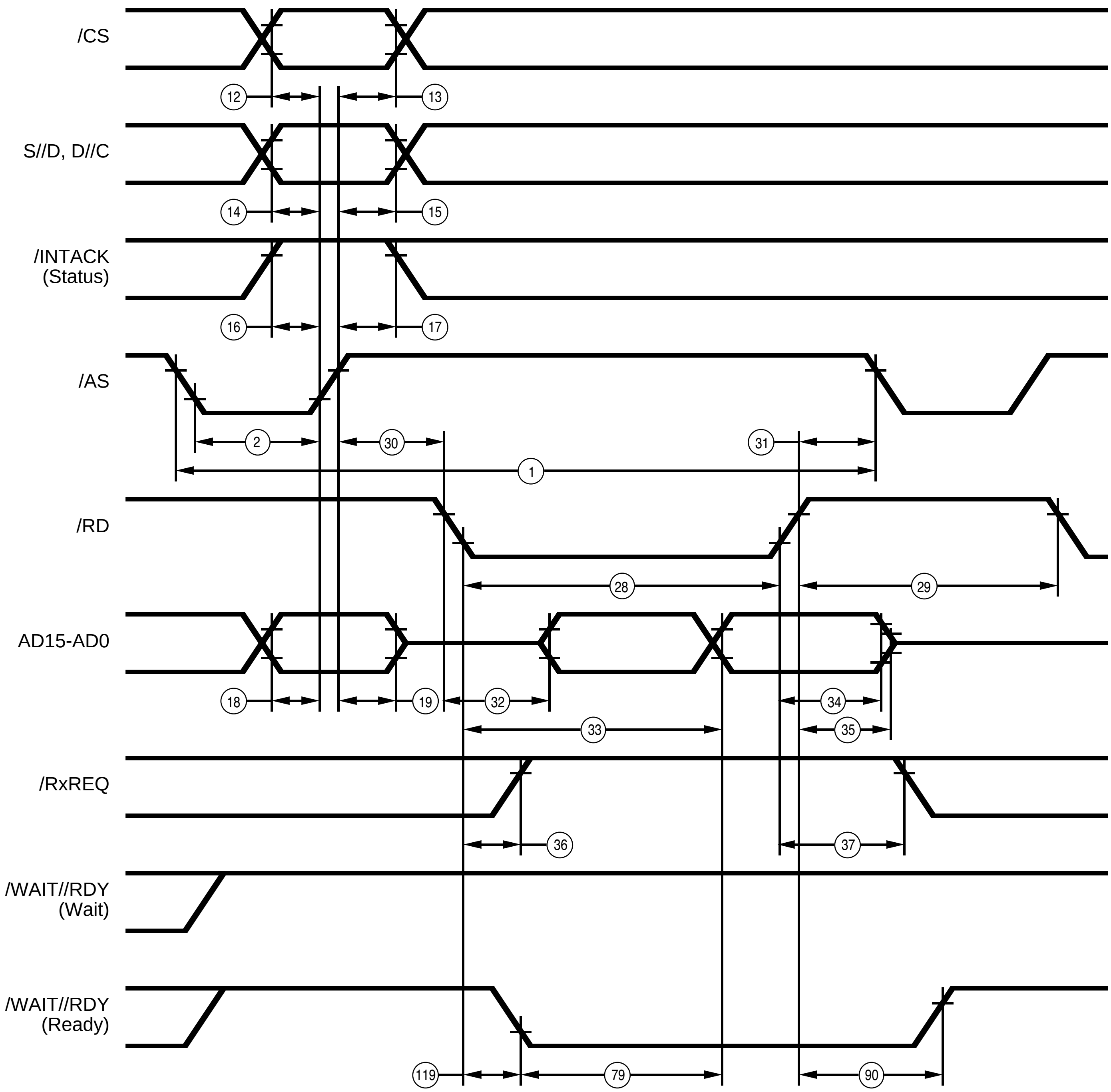



**Figure 86. Multiplexed /RD Read Cycle**

## AC CHARACTERISTICS
Timing Diagrams (Continued)

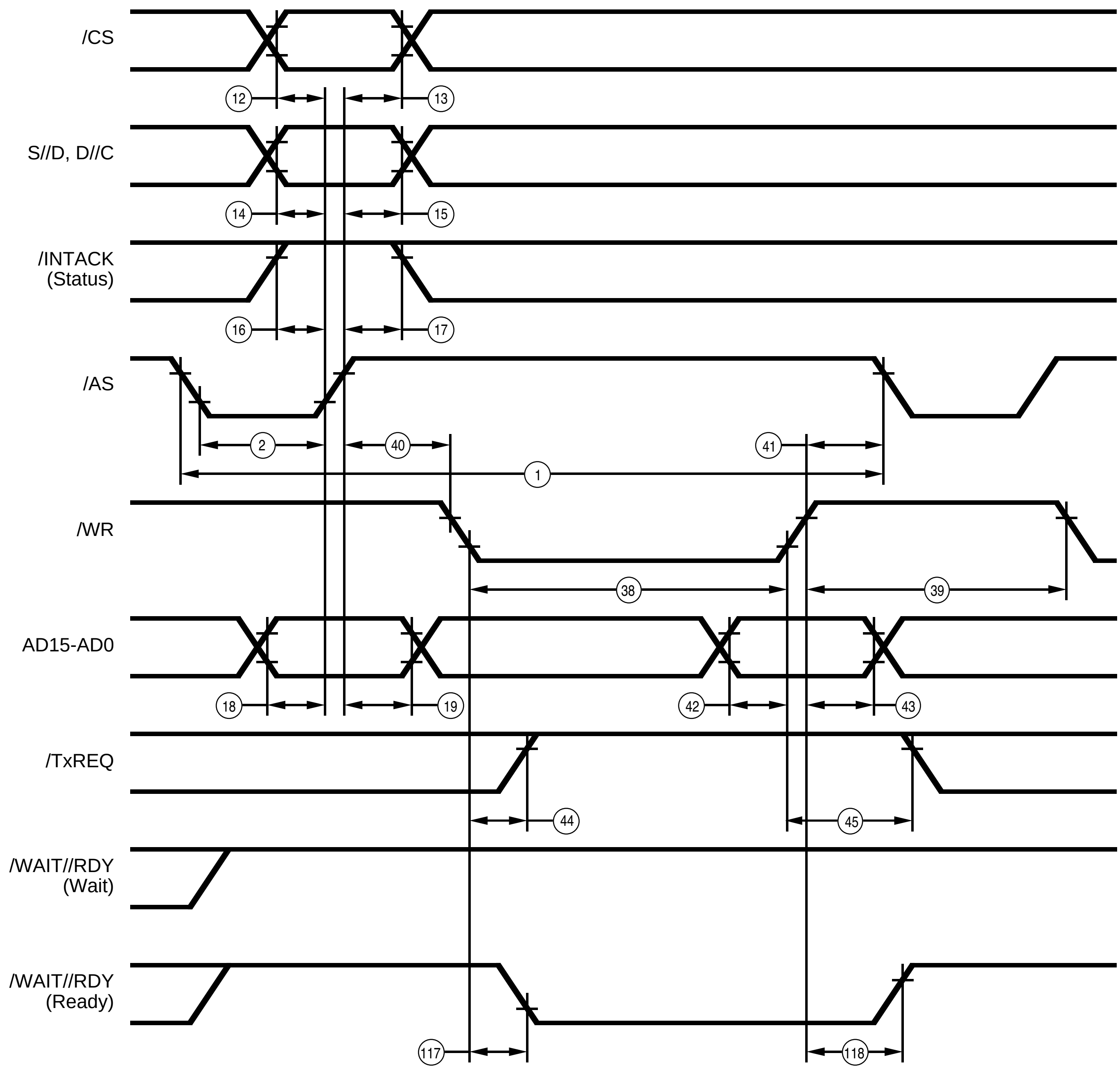



**Figure 87. Multiplexed /WR Write Cycle**

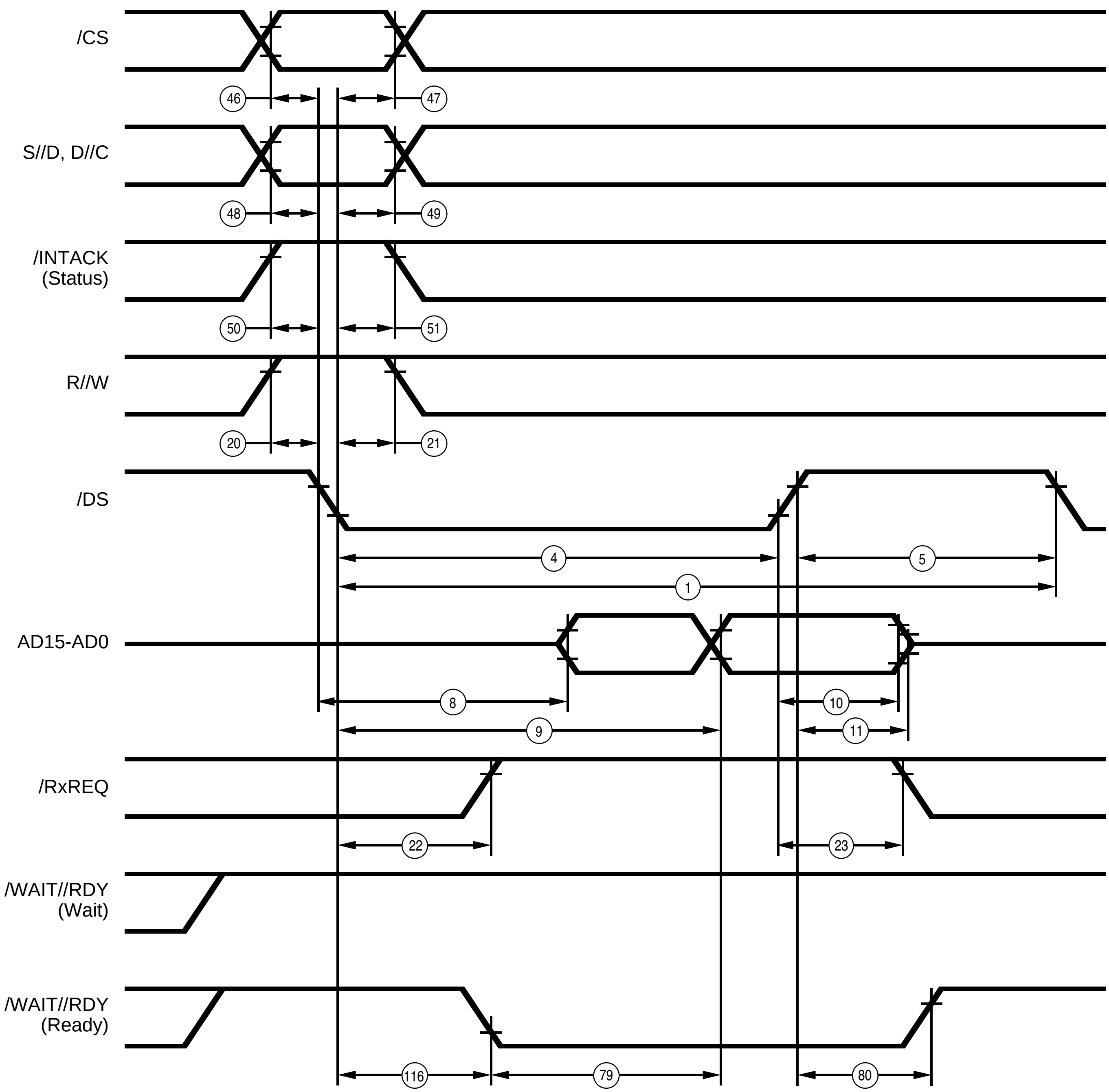



**Figure 88. Non-Multiplexed /DS Read Cycle**

## AC CHARACTERISTICS
### Timing Diagrams (Continued)

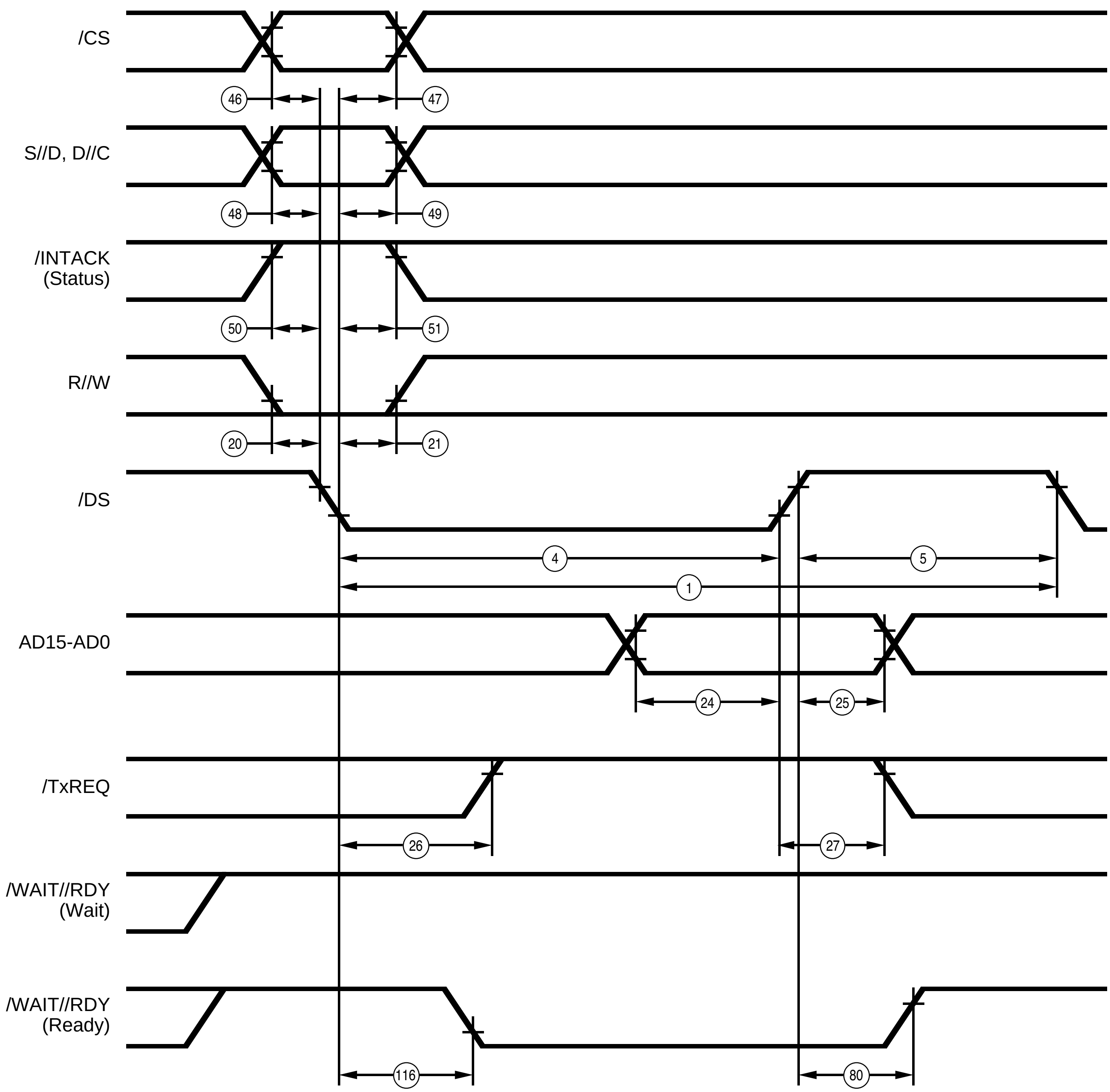



**Figure 89. Non-Multiplexed /DS Write Cycle**

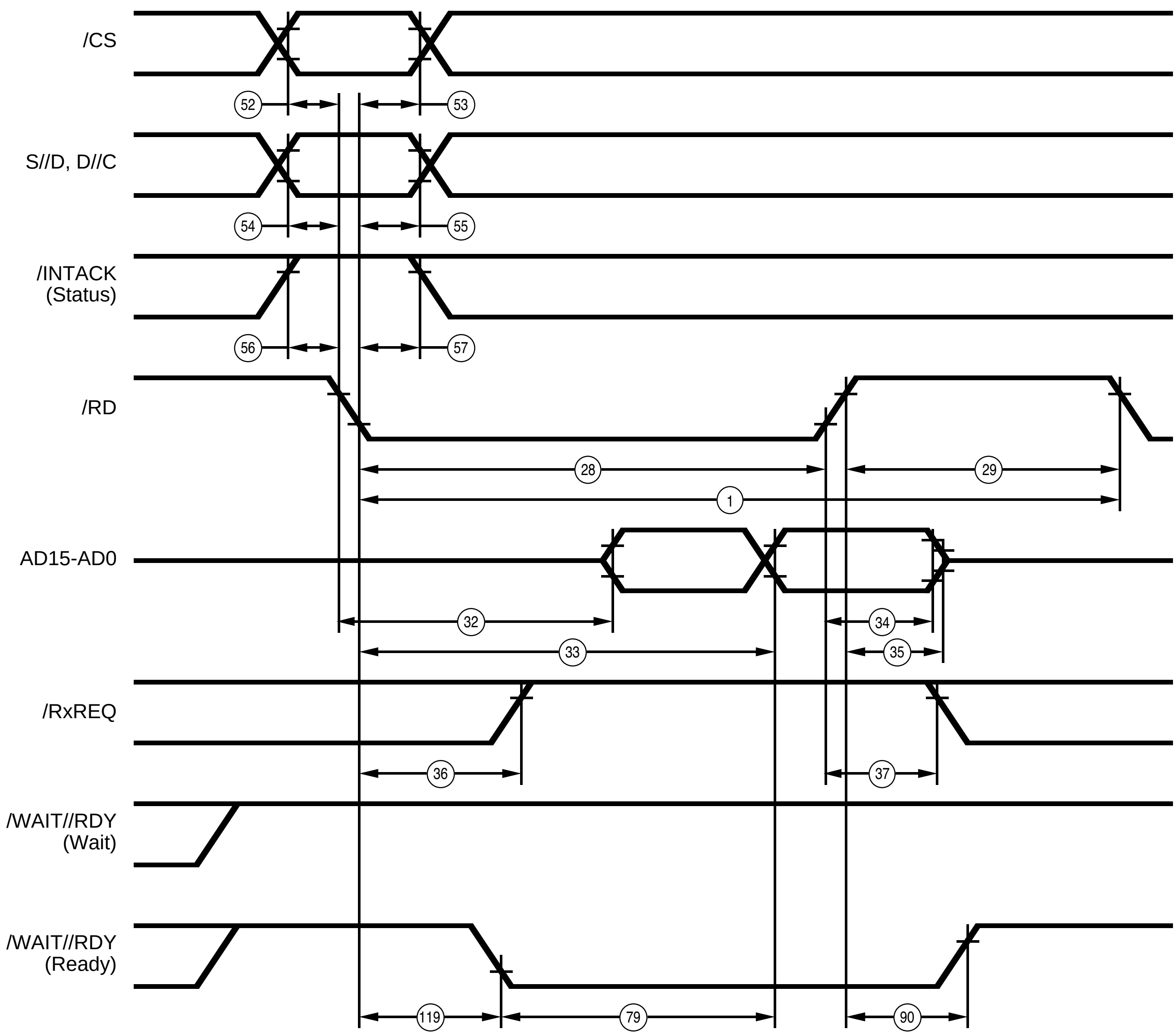



**Figure 90. Non-Multiplexed /RD Read Cycle**

## AC CHARACTERISTICS
Timing Diagrams (Continued)

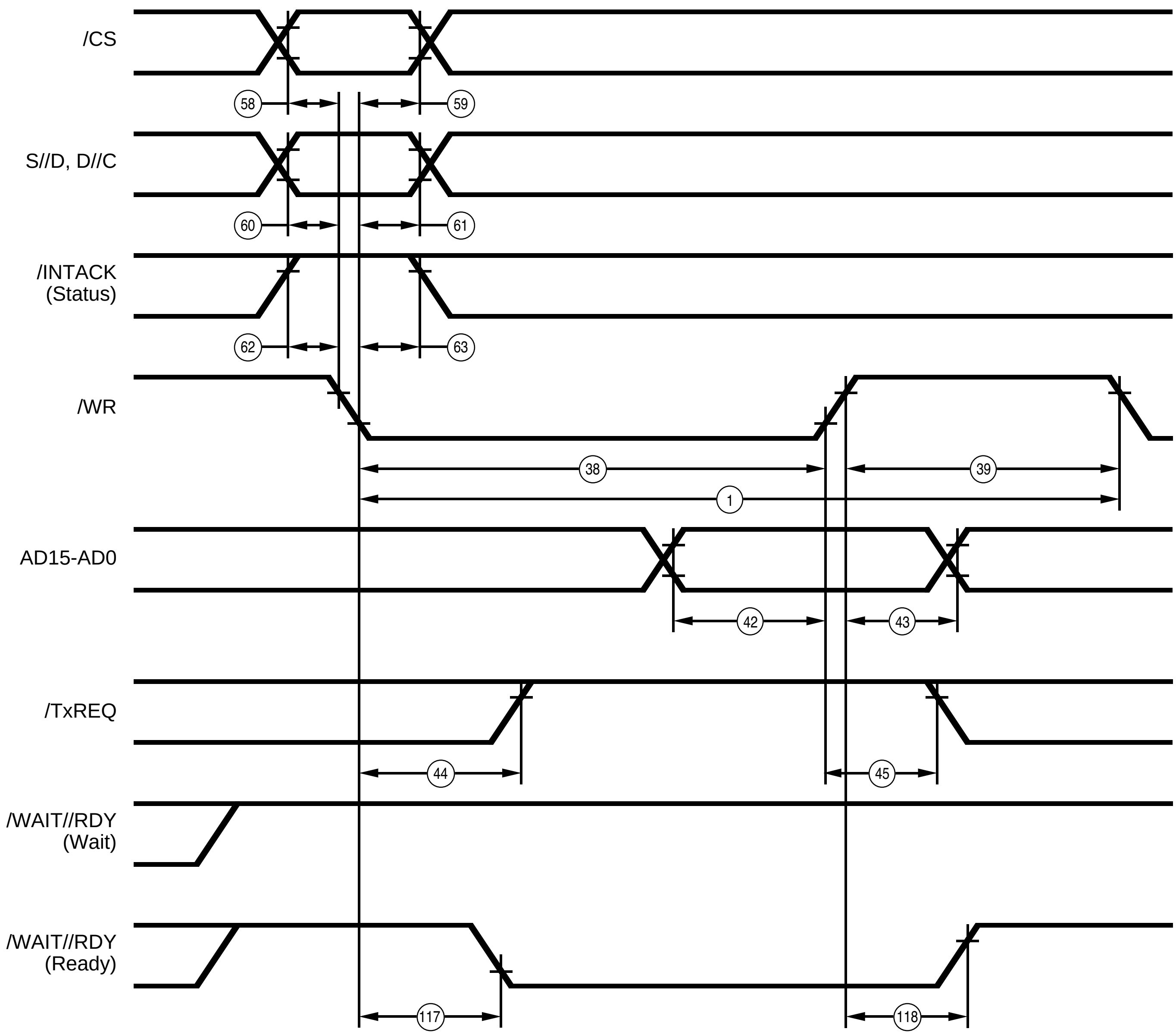



**Figure 91. Non-Multiplexed /WR Write Cycle**

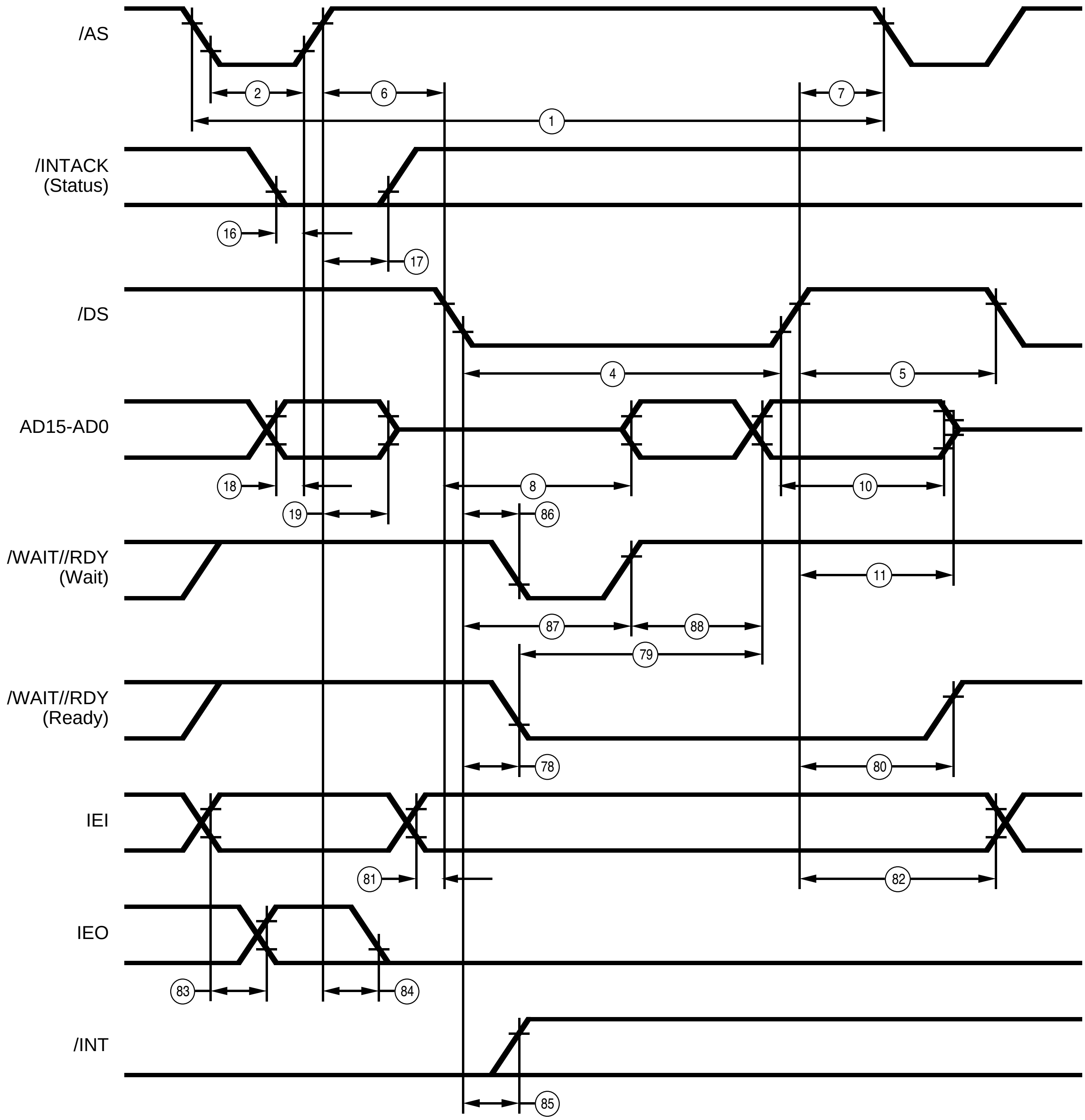



**Figure 92. Multiplexed /DS Interrupt Acknowledge Cycle**

## AC CHARACTERISTICS
Timing Diagrams (Continued)

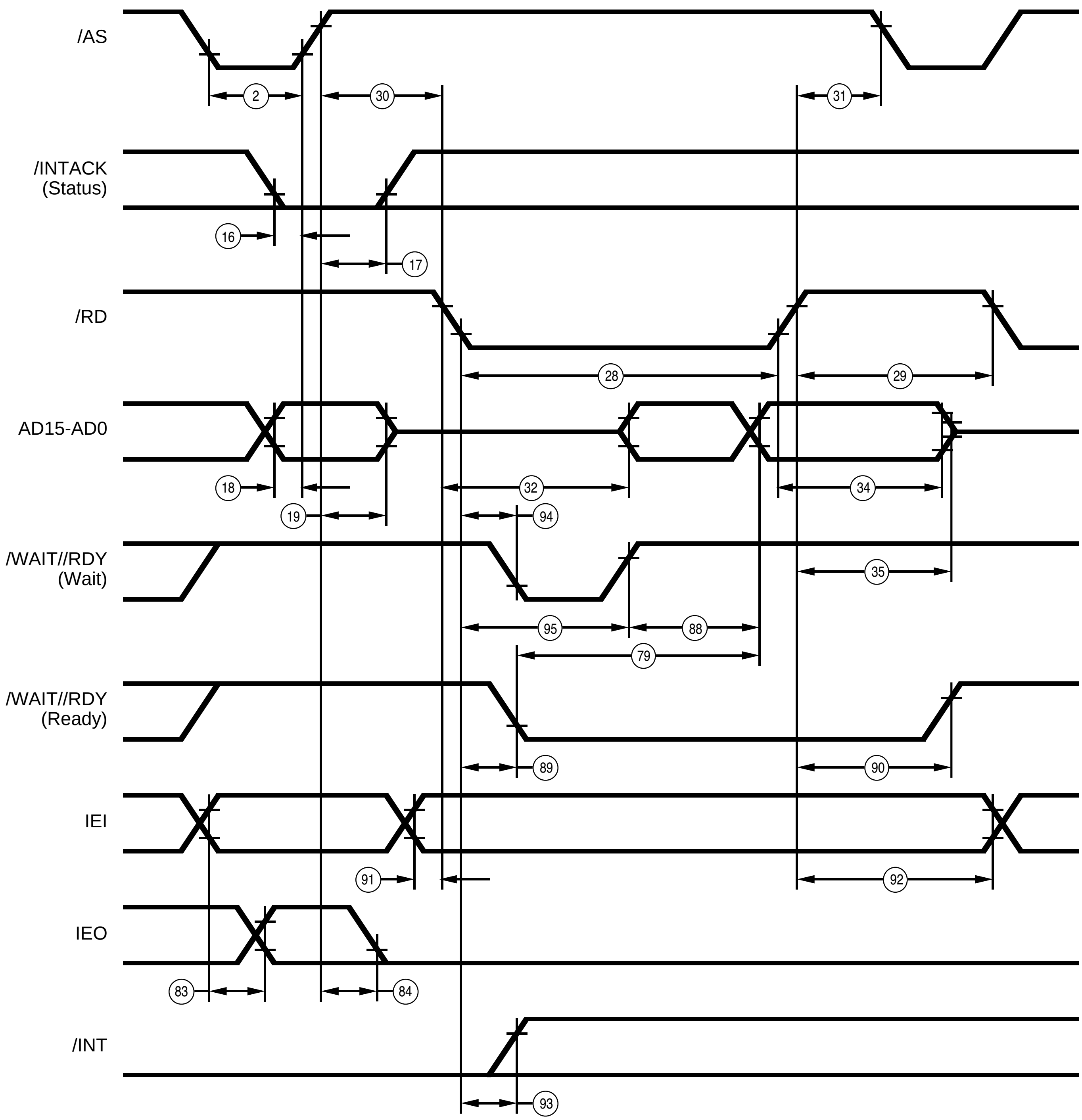



**Figure 93. Multiplexed /RD Interrupt Acknowledge Cycle**

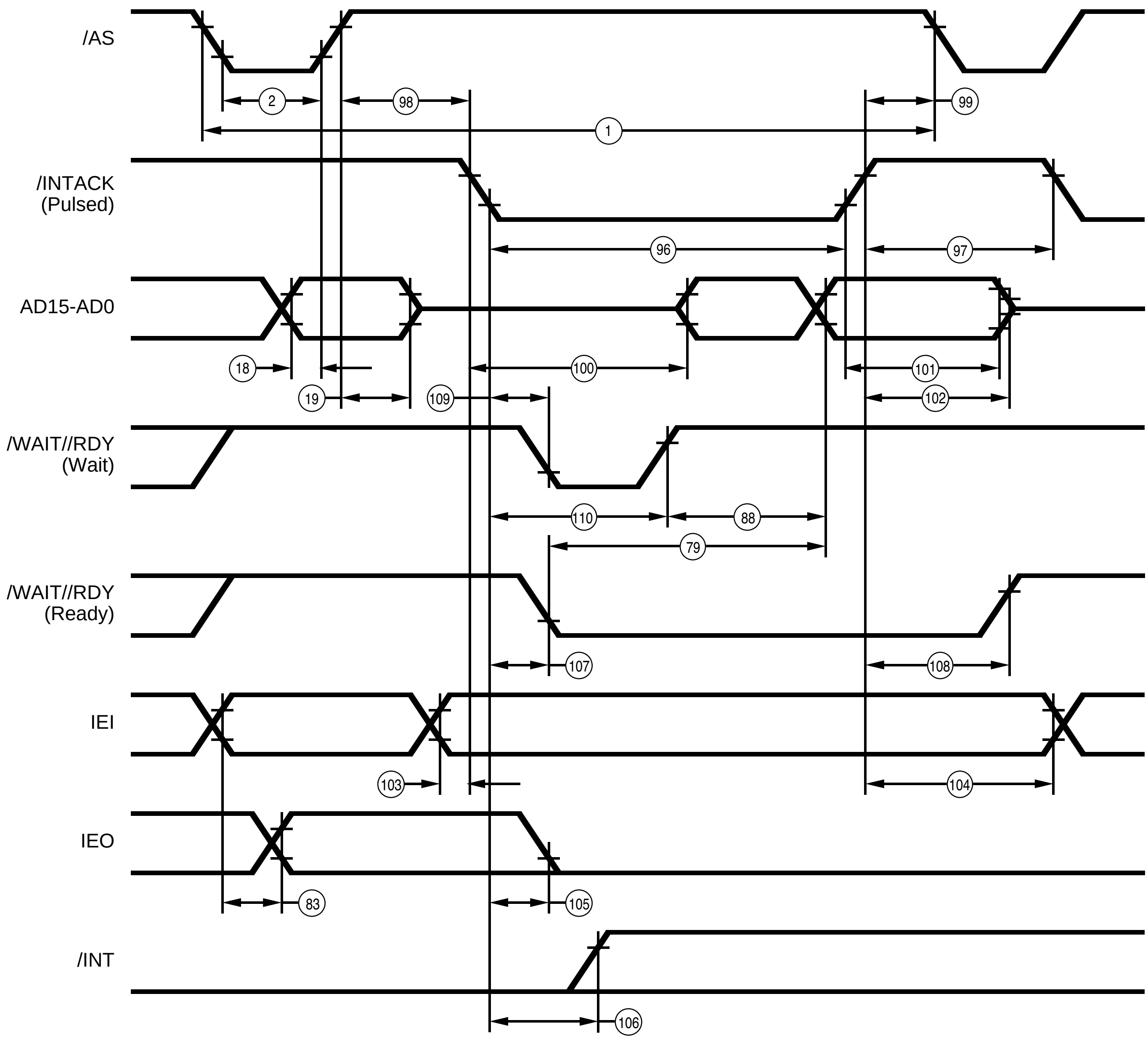



**Figure 94. Multiplexed Pulsed Interrupt Acknowledge Cycle**

## AC CHARACTERISTICS
Timing Diagrams (Continued)

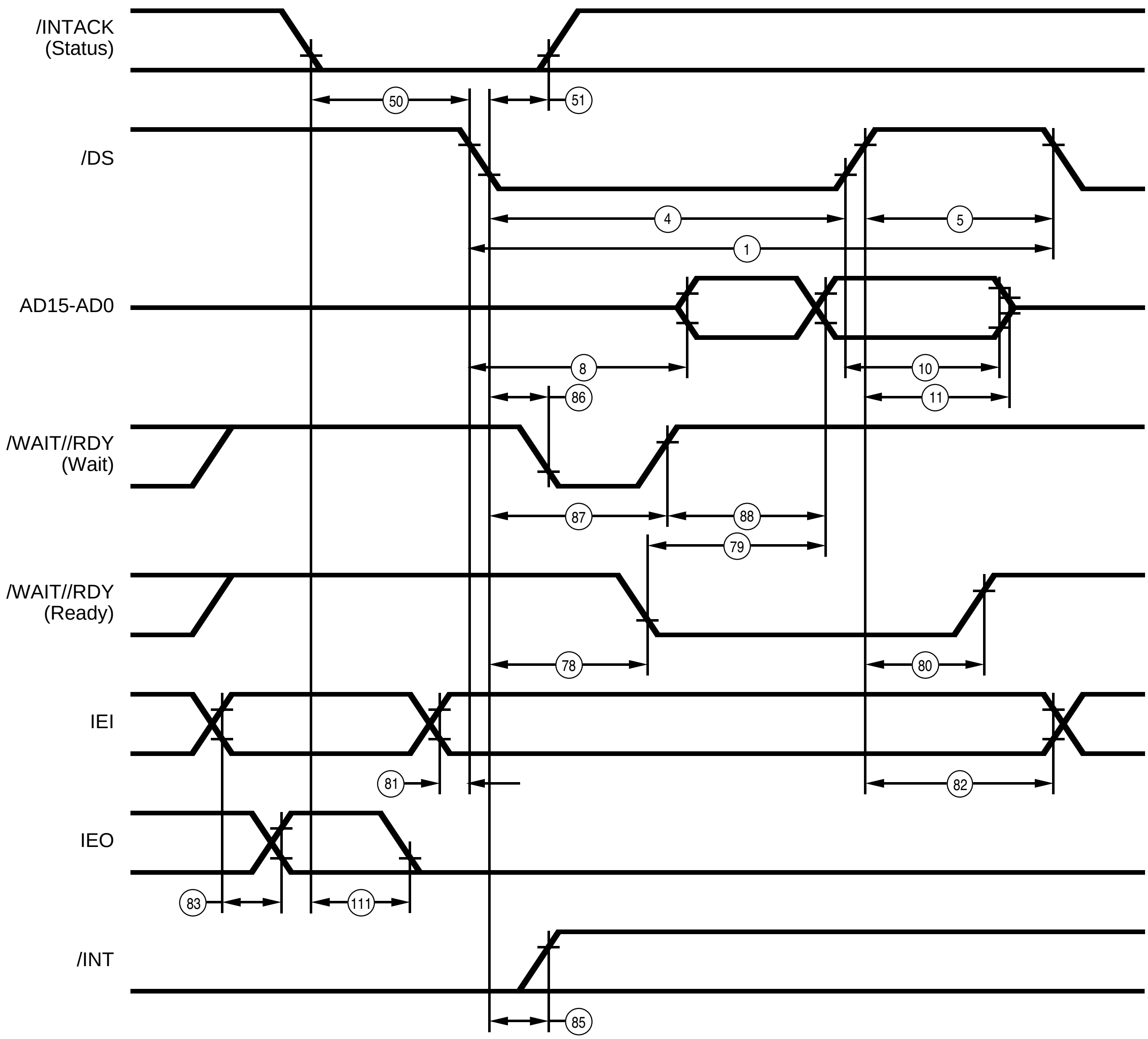



**Figure 95. Non-Multiplexed /DS Interrupt Acknowledge Cycle**

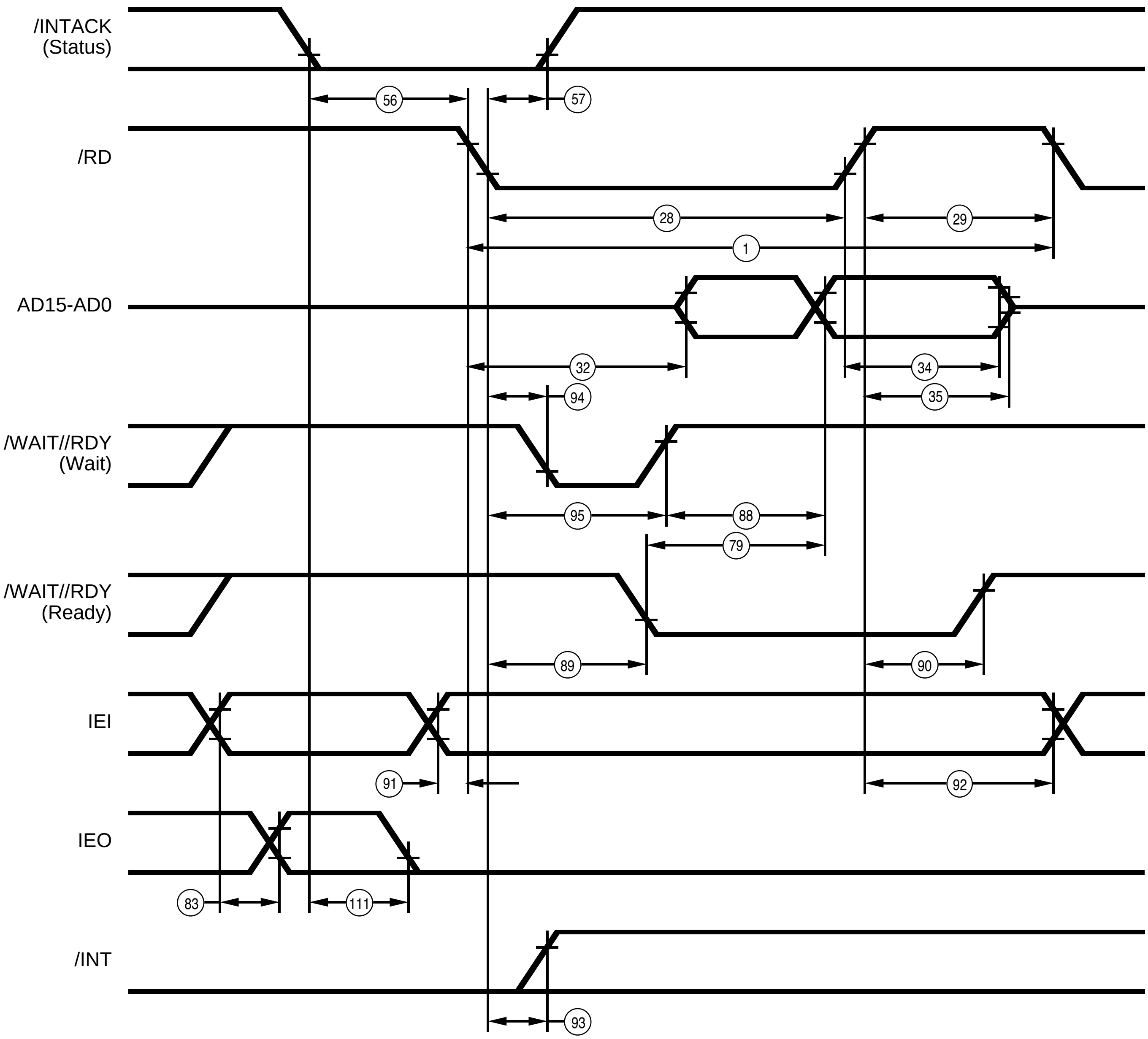



**Figure 96. Non-Multiplexed /RD Pulsed Interrupt Acknowledge Cycle**

# AC CHARACTERISTICS
## Timing Diagrams (Continued)

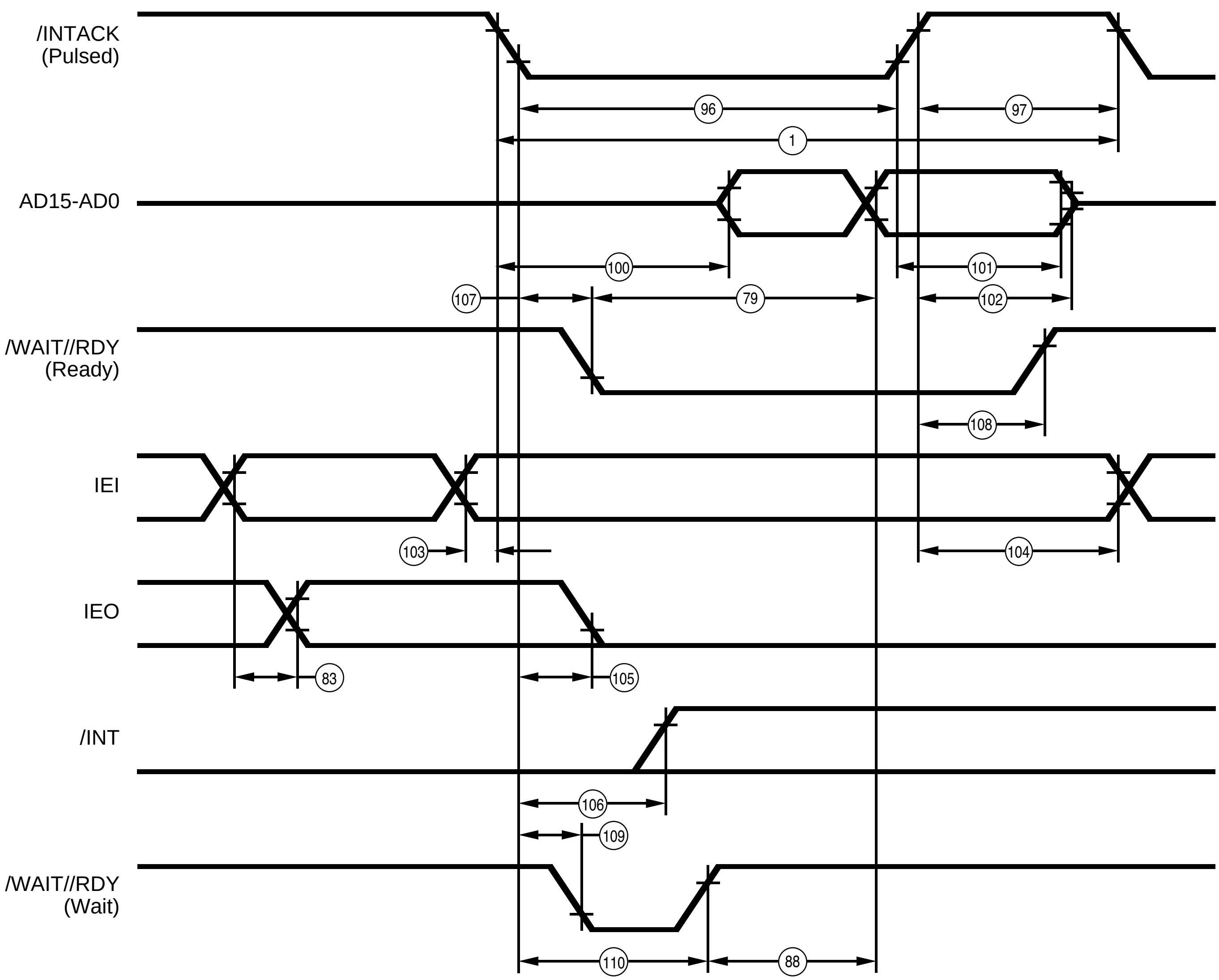



**Figure 97. Non-Multiplexed Pulsed Interrupt Acknowledge Cycle**

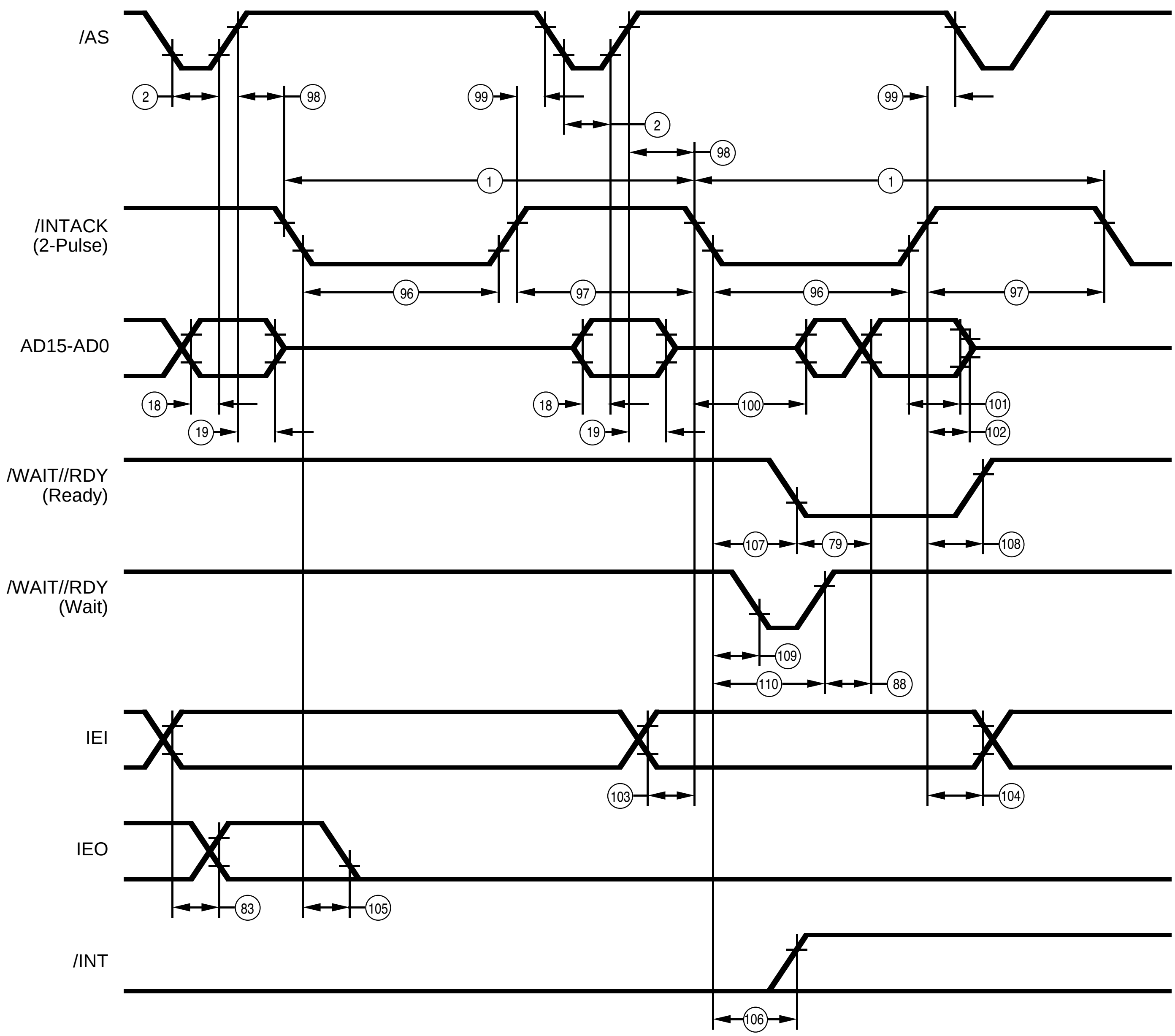



**Figure 98. Multiplexed Double-Pulse Intack Cycle**

# AC CHARACTERISTICS
## Timing Diagrams (Continued)

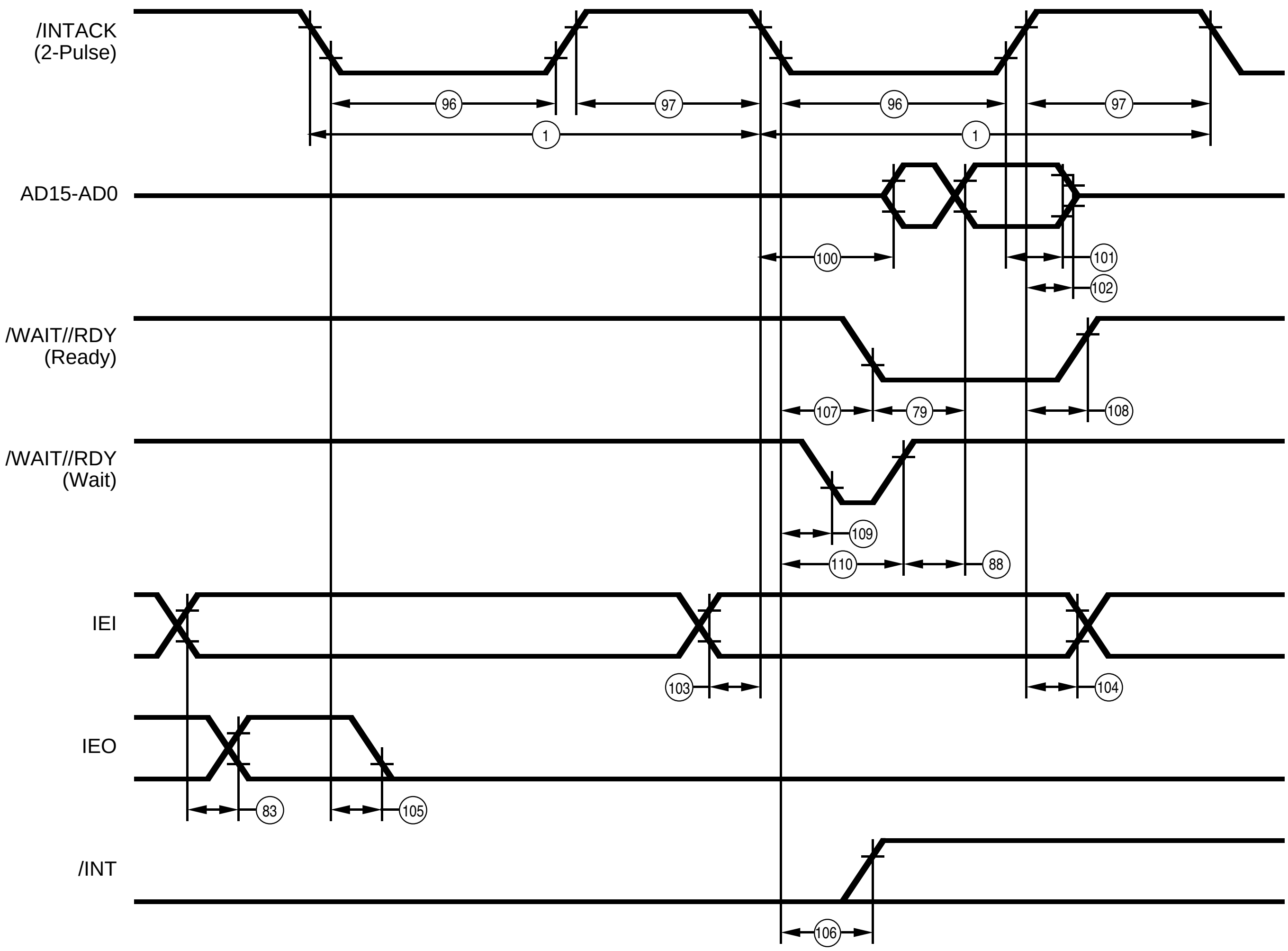



**Figure 99. Non-Multiplexed Double-Pulse Intack Cycle**

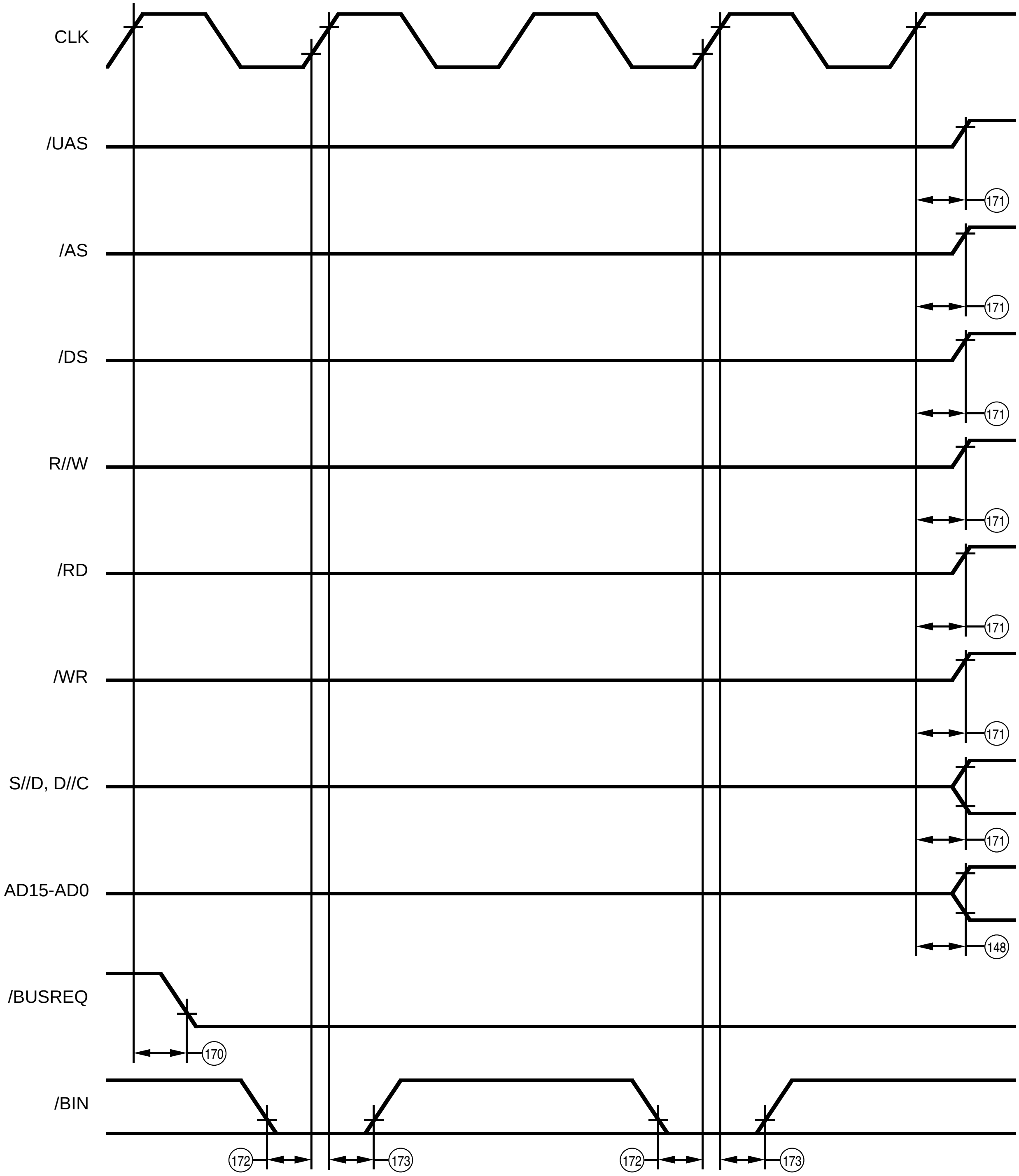



**Figure 100. DMA Start-Up**

## AC CHARACTERISTICS
Timing Diagrams (Continued)

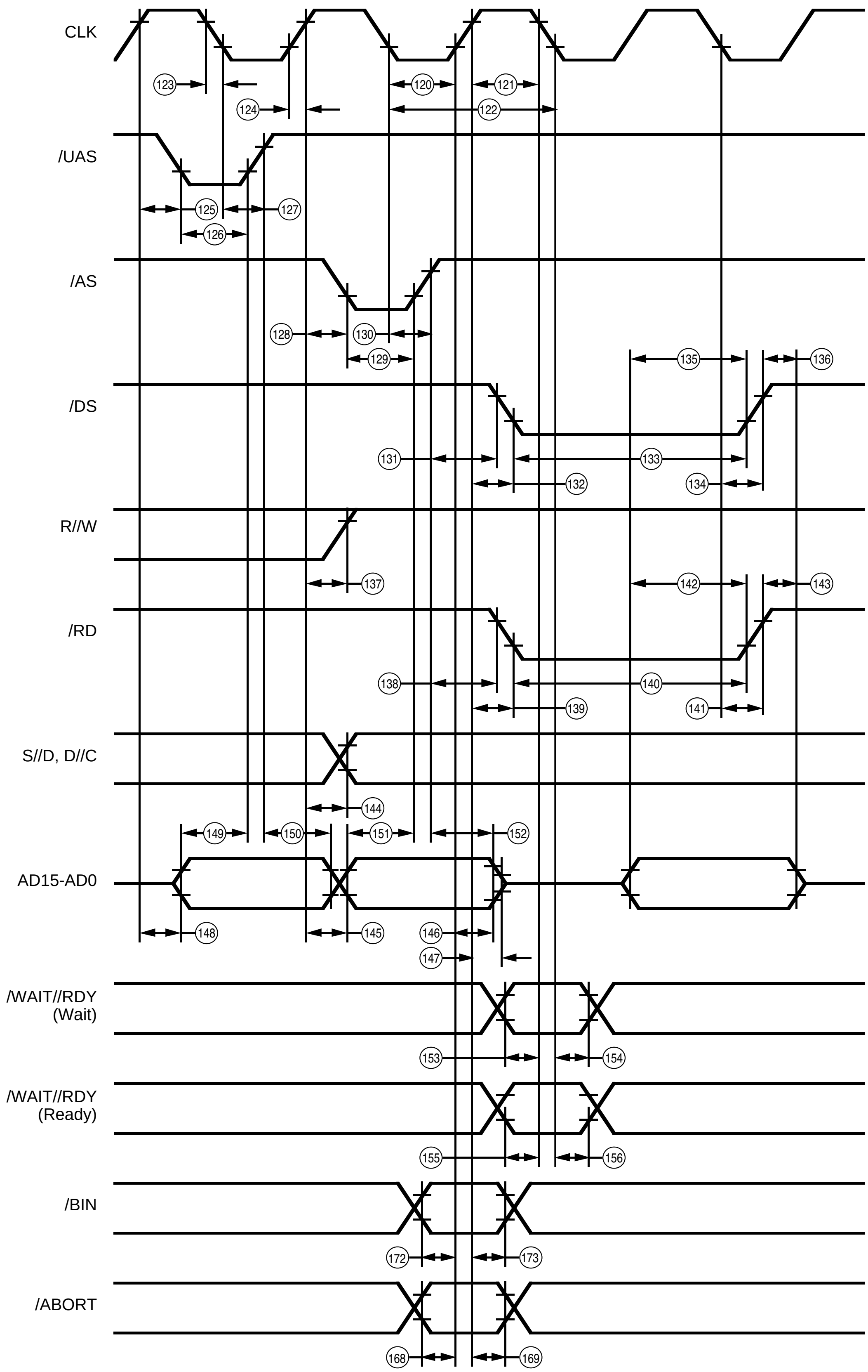



**Figure 101. Memory Read**

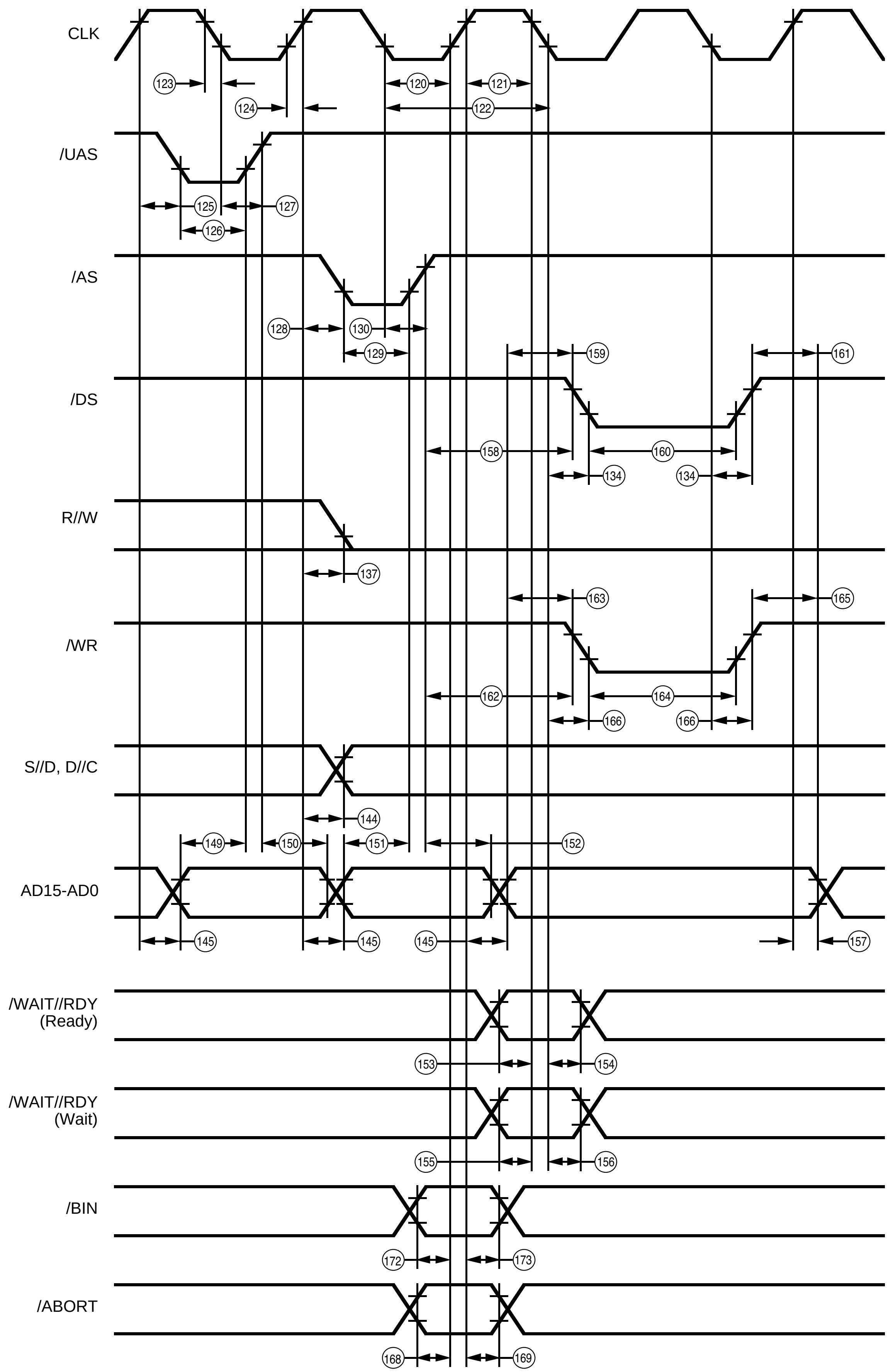



Figure 102. Memory Write

## AC CHARACTERISTICS
Timing Diagrams (Continued)

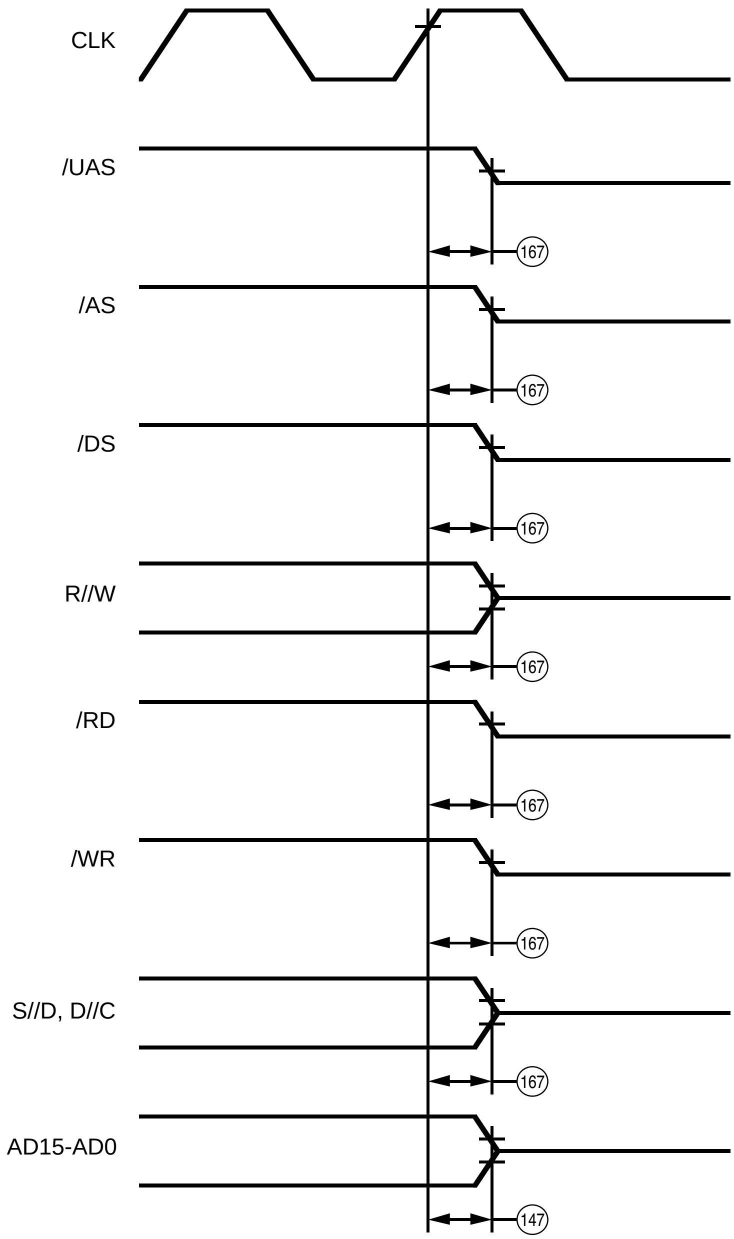



**Figure 103. Bus Release**

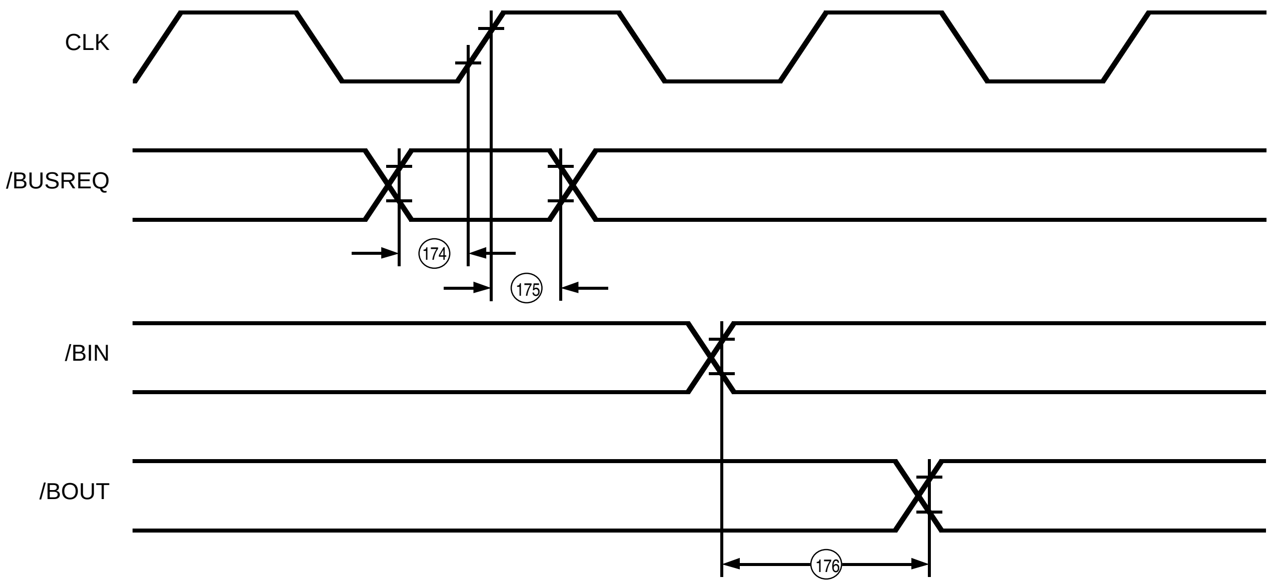



**Figure 104. Request Timing**

## AC CHARACTERISTICS
Timing Table

| No | Symbol | Parameter | Min | Max | Units | Note |
|---|---|---|---|---|---|---|
| 1 | Tcyc | Bus Cycle Time | 160 | | ns | |
| 2 | TwASl | /AS Low Width | 40 | | ns | |
| 3 | TwASh | /AS High Width | 90 | | ns | |
| 4 | TwDSl | /DS Low Width | 70 | | ns | |
| 5 | TwDSh | /DS High Width | 60 | | ns | |
| 6 | TdAS(DS) | /AS Rise to /DS Fall Delay Time | 5 | | ns | |
| 7 | TdDS(AS) | /DS Rise to /AS Fall Delay Time | 5 | | ns | |
| 8 | TdDS(DRa) | /DS Fall to Data Active Delay | 0 | | ns | |
| 9 | TdDS(DRv) | /DS Fall to Data Valid Delay | | 85 | ns | |
| 10 | TdDS(DRn) | /DS Rise to Data Not Valid Delay | 0 | | ns | |
| 11 | TdDS(DRz) | /DS Rise to Data Float Delay | | 20 | ns | |
| 12 | TsCS(AS) | /CS to /AS Rise Setup Time | 15 | | ns | |
| 13 | ThCS(AS) | /CS to /AS Rise Hold Time | 5 | | ns | |
| 14 | TsADD(AS) | Direct Address to /AS Rise Setup Time | 15 | | ns | [1] |
| 15 | ThADD(AS) | Direct Address to /AS Rise Hold Time | 5 | | ns | [1] |
| 16 | TsSIA(AS) | Status /INTACK to /AS Rise Setup Time | 15 | | ns | |
| 17 | ThSIA(AS) | Status /INTACK to /AS Rise Hold Time | 5 | | ns | |
| 18 | TsAD(AS) | Address to /AS Rise Setup Time | 15 | | ns | |
| 19 | ThAD(AS) | Address to /AS Rise Hold Time | 5 | | ns | |
| 20 | TsRW(DS) | R//W to /DS Fall Setup Time | 0 | | ns | |
| 21 | ThRW(DS) | R//W to /DS Fall Hold Time | 25 | | ns | |
| 22 | TsDSf(RRQ) | /DS Fall to /RxREQ Inactive Delay | | 60 | ns | [4] |
| 23 | TdDSr(RRQ) | /DS Rise to /RxREQ Active Delay | 0 | | ns | |
| 24 | TsDW(DS) | Write Data to /DS Rise Setup Time | 30 | | ns | |
| 25 | ThDW(DS) | Write Data to DS Rise Hold Time | 0 | | ns | |
| 26 | TdDSf(TRQ) | /DS Fall to /TxREQ Inactive Delay | | 60 | ns | [5] |
| 27 | TdDSr(TRQ) | /DS Rise to /TxREQ Active Delay | 0 | | ns | |
| 28 | TwRDl | /RD Low Width | 70 | | ns | |
| 29 | TwRDh | /RD High Width | 60 | | ns | |
| 30 | TdAS(RD) | /AS Rise to /RD Fall Delay Time | 5 | | ns | |
| 31 | TdRD(AS) | /RD Rise to /AS Fall Delay Time | 5 | | ns | |
| 32 | TdRD(DRa) | /RD Fall to Data Active Delay | 0 | | ns | |
| 33 | TdRD(DRv) | /RD Fall to Data Valid Delay | | 85 | ns | |
| 34 | TdRD(DRn) | /RD Rise to Data Not Valid Delay | 0 | | ns | |
| 35 | TdRD(DRz) | /RD Rise to Data Float Delay | | 20 | ns | |
| 36 | TdRDf(RRQ) | /RD Fall to /RxREQ Inactive Delay | | 60 | ns | [4] |
| 37 | TdRDr(RRQ) | /RD Rise to /RxREQ Active Delay | 0 | | ns | |
| 38 | TwWRl | /WR Low Width | 70 | | ns | |
| 39 | TwWRh | /WR High Width | 60 | | ns | |
| 40 | TdAS(WR) | /AS Rise to /WR Fall Delay Time | 5 | | ns | |
| 41 | TdWR(AS) | /WR Rise to /AS Fall Delay Time | 5 | | ns | |
| 42 | TsDW(WR) | Write Data to /WR Rise Setup Time | 30 | | ns | |
| 43 | ThDW(WR) | Write Data to /WR Rise Hold Time | 0 | | ns | |
| 44 | TdWRf(TRQ) | /WR Fall to /TxREQ Inactive Delay | | 60 | ns | [5] |

| No | Symbol | Parameter | Min | Max | Units | Note |
|---|---|---|---|---|---|---|
| 45 | TdWRr(TRQ) | /WR Rise to /TxREQ Active Delay | 0 | | ns | |
| 46 | TsCS(DS) | /CS to /DS Fall Setup Time | 0 | | ns | [2] |
| 47 | ThCS(DS) | /CS to /DS Fall Hold Time | 25 | | ns | [2] |
| 48 | TsADD(DS) | Direct Address to /DS Fall Setup Time | 5 | | ns | [1,2] |
| 49 | ThADD(DS) | Direct Address to /DS Fall Hold Time | 25 | | ns | [1,2] |
| 50 | TsSIA(DS) | Status /INTACK to /DS Fall Setup time | 5 | | ns | [2] |
| 51 | ThSIA(DS) | Status /INTACK to /DS Fall Hold Time | 25 | | ns | [2] |
| 52 | TsCS(RD) | /CS to /RD Fall Setup Time | 0 | | ns | [2] |
| 53 | ThCS(RD) | /CS to /RD Fall Hold Time | 25 | | ns | [2] |
| 54 | TsADD(RD) | Direct Address to /RD Fall Setup Time | 5 | | ns | [1,2] |
| 55 | ThADD(RD) | Direct Address to /RD Fall Hold Time | 25 | | ns | [1,2] |
| 56 | TsSIA(RD) | Status /INTACK to /RD Fall Setup Time | 5 | | ns | [2] |
| 57 | ThSIA(RD) | Status /INTACK to /RD Fall Hold Time | 25 | | ns | [2] |
| 58 | TsCS(WR) | /CS to /WR Fall Setup Time | 0 | | ns | [2] |
| 59 | ThCS(WR) | /CS to /WR Fall Hold Time | 25 | | ns | [2] |
| 60 | TsADD(WR) | Direct Address to /WR Fall Setup Time | 5 | | ns | [1,2] |
| 61 | ThADD(WR) | Direct Address to /WR Fall Hold Time | 25 | | ns | [1,2] |
| 62 | TsSIA(WR) | Status /INTACK to /WR Fall Setup Time | 5 | | ns | [2] |
| 63 | ThSIA(WR) | Status /INTACK to /WR Fall Hold Time | 25 | | ns | [2] |
| 78 | TdDSf(RDY) | /DS Fall (Intack) to /RDY Fall Delay | | 200 | ns | |
| 79 | TdRDY(DRv) | /RDY Fall to Data Valid Delay | | 40 | ns | |
| 80 | TdDSr(RDY) | /DS Rise to /RDY Rise Delay | | 40 | ns | |
| 81 | TsIEI(DSI) | IEI to /DS Fall (Intack) Setup Time | 60 | | ns | |
| 82 | ThIEI(DSI) | IEI to /DS Rise (Intack) Hold Time | 0 | | ns | |
| 83 | TdIEI(IEO) | IEI to IEO Delay | | 60 | ns | |
| 84 | TdAS(IEO) | /AS Rise (Intack) to IEO Delay | | 60 | ns | |
| 85 | TdDSI(INT) | /DS Fall to /INT Inactive Delay | | 200 | ns | |
| 86 | TdDSI(Wf) | /DS Fall (Intack) to /WAIT Fall Delay | | 40 | ns | |
| 87 | TdDSI(Wr) | /DS Fall (Intack) to /WAIT Rise Delay | | 200 | ns | |
| 88 | TdW(DRv) | /WAIT Rise to Data Valid Delay | | 40 | ns | |
| 89 | TdRDf(RDY) | /RD Fall (Intack) to /RDY Fall Delay | | 200 | ns | |
| 90 | TdRDr(RDY) | /RD Rise to /RDY Rise Delay | | 40 | ns | |
| 91 | TsIEI(RDI) | IEI to /RD Fall (Intack) Setup Time | 60 | | ns | |
| 92 | ThIEI(RDI) | IEI to /RD Rise (Intack) Hold Time | 0 | | ns | |
| 93 | TdRDI(INT) | /RD Fall (Intack) to /INT Inactive Delay | | 200 | ns | |
| 94 | TdRDI(Wf) | /RD Fall (Intack) to /WAIT Fall Delay | | 40 | ns | |
| 95 | TdRDI(Wr) | /RD Fall (Intack) to /WAIT Rise Delay | | 200 | ns | |
| 96 | TwPIAl | Pulsed /INTACK Low Width | 70 | | ns | |
| 97 | TwPIAh | Pulsed /INTACK High Width | 60 | | ns | |
| 98 | TdAS(PIA) | /AS Rise to Pulsed /INTACK Fall Delay Time | 5 | | ns | |
| 99 | TdPIA(AS) | Pulsed /INTACK Rise to /AS Fall Delay Time | 5 | | ns | |
| 100 | TdPIA(DRa) | Pulsed /INTACK Fall to Data Active Delay | 0 | | ns | |
| 101 | TdPIA(DRn) | Pulsed /INTACK Rise to Data Not Valid Delay | 0 | | ns | |
| 102 | TdPIA(DRz) | Pulsed /INTACK Rise to Data Float Delay | | 20 | ns | |

## AC CHARACTERISTICS (Continued)
Timing Table

| No | Symbol | Parameter | Min | Max | Units | Note |
|---|---|---|---|---|---|---|
| 103 | TsIEI(PIA) | IEI to Pulsed /INTACK Fall Setup Time | 10 | | ns | |
| 104 | ThIEI(PIA) | IEI to Pulsed /INTACK Rise Hold Time | 0 | | ns | |
| 105 | TdPIA(IEO) | Pulsed /INTACK Fall to IEO Delay | | 60 | ns | |
| 106 | TdPIA(INT) | Pulsed /INTACK Fall to /INT Inactive Delay | | 200 | ns | |
| 107 | TdPIAf(RDY) | Pulsed /INTACK Fall to /RDY Fall Delay | | 200 | ns | |
| 108 | TdPIAr(RDY) | Pulsed /INTACK Rise to /RDY Rise Delay | | 40 | ns | |
| 109 | TdPIA(Wf) | Pulsed /INTACK Fall to /WAIT Fall Delay | | 40 | ns | |
| 110 | TdPIA(Wr) | Pulsed /INTACK Fall to /WAIT Rise Delay | | 200 | ns | |
| 111 | TdSIA(INT) | Status /INTACK Fall to IEO Inactive Delay | | 200 | ns | [2] |
| 112 | TwSTBh | /Strobe High Width | 50 | | ns | [3] |
| 113 | TwRESl | /RESET Low Width | 170 | | ns | |
| 114 | TwRESh | /RESET High Width | 60 | | ns | |
| 115 | TdRES(STB) | /RESET Rise to /STB Fall | 60 | | ns | [3] |
| 116 | TdDSf(RDY) | /DS Fall to /RDY Fall Delay | | 50 | ns | |
| 117 | TdWRf(RDY) | /WR Fall to /RDY Fall Delay | | 50 | ns | |
| 118 | TdWRr(RDY) | /WR Rise to /RDY Rise Delay | | 40 | ns | |
| 119 | TdRDf(RDY) | /RD Fall to /RDY Fall Delay | | 50 | ns | |
| 120 | TwCLKl | CLK Low Width | 25 | | ns | |
| 121a | TwCLKh | CLK High Width | 25 | | ns | |
| 121b | TwCLKh | CLK High Width (Linked List Mode) | 35 | | ns | [12] |
| 122a | TcCLK | CLK Cycle Time | 50 | | ns | |
| 122b | TcCLK | CLK Cycle Time (Linked List Mode) | 60 | | ns | [12] |
| 123 | TfCLK | CLK Fall Time | | 5 | ns | |
| 124 | TrCLK | CLK Rise Time | | 5 | ns | |
| 125 | TdCLKr (UAS) | CLK Rise to /UAS Fall Delay | | 30 | ns | [6] |
| 126 | TwUASl | /UAS Low Width | 25 | | ns | [6,7,13] |
| 127 | TdCLKf(UAS) | CLK Fall to /UAS Rise Delay | | 30 | ns | [6] |
| 128 | TdCLKr(AS) | CLK Rise to /AS Fall Delay | | 30 | ns | [6] |
| 129 | TwASl | /AS Low Width | 25 | | ns | [6,7,13] |
| 130 | TdCLKf(AS) | CLK Fall to /AS Rise Delay | | 30 | ns | [6] |
| 131 | TdAS(DSr) | /AS Rise to /DS Fall (Read) Delay | 25 | | ns | [6,8] |
| 132 | TdCLKr(DS) | CLK Rise to /DS Delay | | 30 | ns | [6] |
| 133 | TwDSlr | /DS (Read) Low Width | 75 | | ns | [6,9,13] |
| 134 | TdCLKf(DS) | CLK Fall to /DS Delay | | 30 | ns | [6] |
| 135 | TsDR(DS) | Read Data to /DS Rise Setup Time | 30 | | ns | [6] |
| 136 | ThDR(DS) | Read Data to /DS Rise Hold Time | 0 | | ns | [6] |
| 137 | TdCLK(RW) | CLK Rise to R//W Delay | | 30 | ns | [6] |
| 138 | TdAS(RD) | /AS Rise to /RD Fall Delay | 25 | | ns | [6,8] |
| 139 | TdCLKr(RD) | CLK Rise to /RD Delay | | 30 | ns | [6] |
| 140 | TwRDl | /RD Low Width | 75 | | ns | [6,9] |
| 141 | TdCLKf(RD) | CLK Fall to /RD Delay | | 30 | ns | [6] |
| 142 | TsDR(RD) | Read Data to /RD Rise Setup Time | 30 | | ns | [6] |
| 143 | ThDR(RD) | Read Data to /RD Rise Hold Time | 0 | | ns | [6] |
| 144 | TdCLK(ADD) | CLK Rise to Direct Address Delay | | 30 | ns | [1,6] |
| 145 | TdCLK(AD) | CLK Rise to Address Delay | TdCLKf(DS) | 35 | ns | [6] |
| 146 | ThAD(PC) | Address to CLK Rise Hold Time | 0 | | ns | [6] |

| No | Symbol | Parameter | Min | Max | Units | Note |
|---|---|---|---|---|---|---|
| 147 | TdCLK(ADz) | CLK Rise to Address Float Delay | | 35 | ns | [6] |
| 148 | TdCLK(ADa) | CLK Rise to Address Active Delay | | 35 | ns | [6] |
| 149 | TsAD(UAS) | Address to /UAS Rise Setup Time | 10 | | ns | [6] |
| 150 | ThAD(UAS) | Address to /UAS Rise Hold Time | 10 | | ns | [6] |
| 151 | TsAD(AS) | Address to /AS Rise Setup Time | 10 | | ns | [6] |
| 152 | ThAD(AS) | Address to /AS Rise Hold Time | 10 | | ns | [6] |
| 153 | TsW(CLK) | /WAIT to CLK Fall Setup Time | 10 | | ns | [6] |
| 154 | ThW(CLK) | /WAIT to CLK Fall Hold Time | 15 | | ns | [6] |
| 155 | TsRDY(CLK) | /READY to CLK Fall Setup Time | 10 | | ns | [6] |
| 156 | ThRDY(CLK) | /READY to CLK Fall Hold Time | 15 | | ns | [6] |
| 157 | ThDW(CLK) | Write Data to CLK Rise Hold Time | 0 | | ns | [6] |
| 158 | TdAS(DSw) | /AS Rise to /DS Fall (Write) Delay | 40 | | ns | [6,10,13] |
| 159 | TsDW(DS) | Write Data to /DS Fall Setup Time | 25 | | ns | [6,7,13] |
| 160 | TwDSlw | /DS (Write) Low Width | 45 | | ns | [6,11,13] |
| 161 | ThDW(DS) | Write Data to /DS Rise Hold Time | 25 | | ns | [6,8] |
| 162 | TdAS(WR) | /AS Rise to /WR Fall Delay | 40 | | ns | [6,10,13] |
| 163 | TsDW(WR) | Write Data to /WR Fall Setup Time | 25 | | ns | [6,7,13] |
| 164 | TwWRl | /WR Low Width | 45 | | ns | [6,11,13] |
| 165 | ThDW(WR) | Write Data to /WR Rise Hold Time | 25 | | ns | [6,8] |
| 166 | TdCLK(WR) | CLK Fall to /WR Delay | | 30 | ns | [6] |
| 167 | TdCLK(BUSz) | CLK Rise to Bus Float Delay | | 30 | ns | [6] |
| 168 | TsABT(CLK) | /ABORT to CLK Rise Setup Time | 20 | | ns | [6] |
| 169 | ThABT(CLK) | /ABORT to CLK Rise Hold Time | 15 | | ns | [6] |
| 170 | TdCLK(BRQ) | CLK Rise to /BUSREQ Delay | | 30 | ns | [6] |
| 171 | TdCLK(BUSa) | CLK Rise to Bus Active Delay | | 30 | ns | [6] |
| 172 | TsBIN(CLK) | /BIN to CLK Rise Setup Time | 20 | | ns | [6] |
| 173 | ThBIN(CLK) | /BIN to CLK Rise Hold Time | 15 | | ns | [6] |
| 174 | TsBRQ(CLK) | /BUSREQ to CLK Rise Setup Time | 25 | | ns | [6] |
| 175 | ThBRQ(CLK) | /BUSREQ to CLK Rise Hold Time | 0 | | ns | [6] |
| 176 | TdBIN(BOT) | /BIN to /BOUT Delay | | 60 | ns | |

**Notes:**
AC Test Conditions:
$V_{CC}$ = 5V ±5% unless otherwise specified, over specified temperature range.
$V_{IH}$ = 2.0V $V_{OH}$ = 2.0V
$V_{IL}$ = 0.8V $V_{OL}$ = 0.8V
Float = +0.5V

[1] Direct Address is any of S//D, D//C or AD15-AD8 used as an address bus.
[2] The parameter applies only when /AS is not present.
[3] Strobe is any of /DS, /RD, /WR or Pulsed /INTACK.
[4] Parameter applies only if read empties the receive FIFO.
[5] Parameter applies only if write fills the transmit FIFO.
[6] Parameter applies only while the IUSC is bus master.
[7] Parameter is clock-cycle dependent, TwCLKh + TfCLK – 5.
[8] Parameter is clock-cycle dependent, TwCLKl + TrCLK –5
[9] Parameter is clock-cycle dependent, TcCLK + TwCLKh + TfCLK –5.
[10] Parameter is clock-cycle dependent, TcCLK –10.
[11] Parameter is clock-cycle dependent, TcCLK –5.
[12] Clock cycle parameters TwCLKh and TcCLK have unique values for Linked List Mode. In Linked List Mode, the system clock cycle is extended to 60 ns, and the system clock High pulse width is extended to 35 ns. This is due to the internal timing paths unique to the Linked List Mode. The transmit and receive bit rates are not affected.
[13] For Linked List Mode, the minimum for these values should be calculated using TwCLKh = 35 ns and TcCLK = 60 ns.

# AC CHARACTERISTICS

## General Timing Diagram

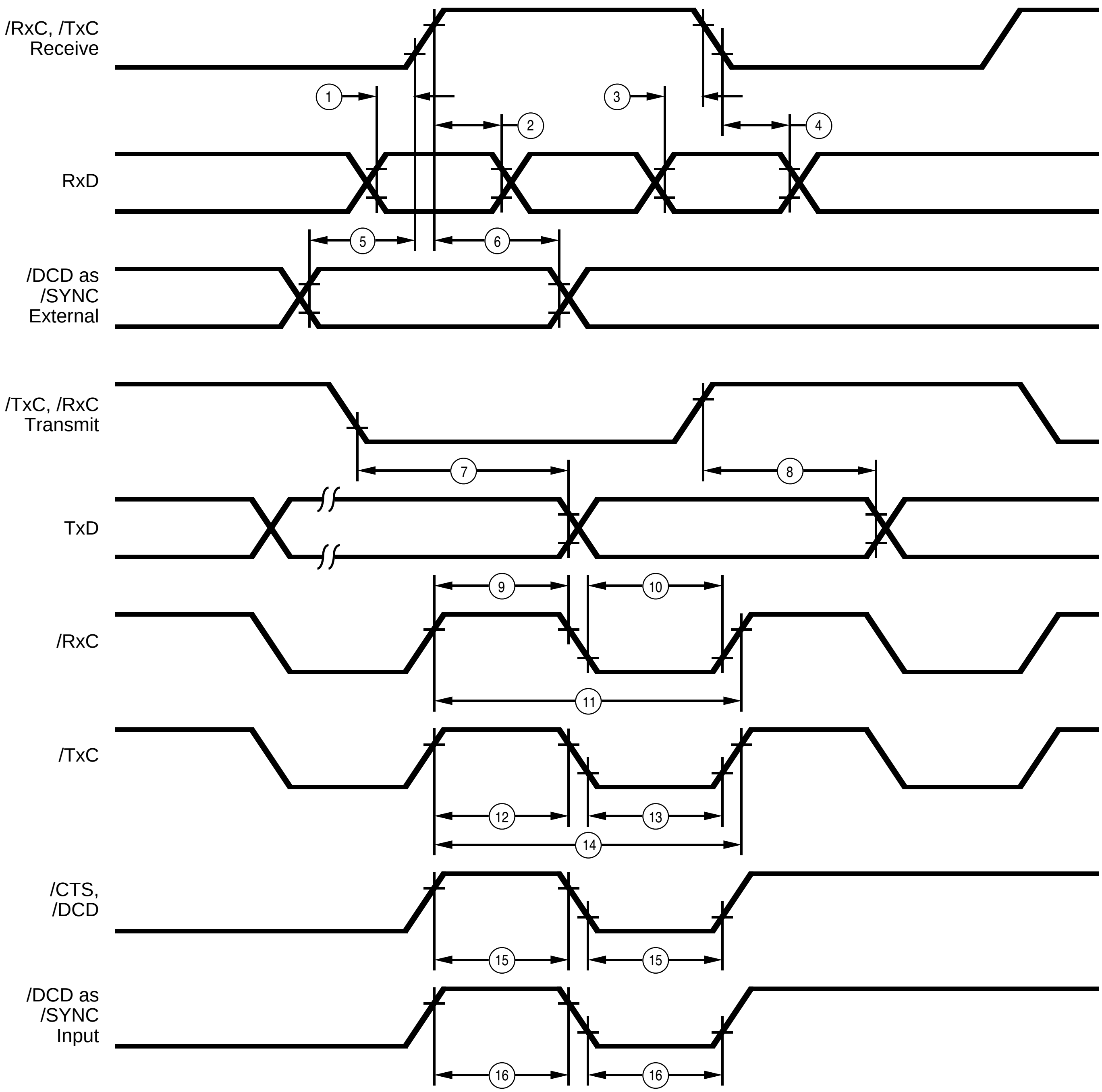



**Figure 105. General Timing**

## AC CHARACTERISTICS
### General Timing Table

| No | Symbol | Parameter | Min | Max | Units | Note |
|---|---|---|---|---|---|---|
| 1 | TsRxD(RxCr) | RxD to /RxC Rise Setup Time (x1 Mode) | 0 | | ns | [1] |
| 2 | ThRxD(RxCr) | RxD to /RxC Rise Hold Time (x1 Mode) | 20 | | ns | [1] |
| 3 | TsRxd(RxCf) | RxD to /RxC Fall Setup Time (x1 Mode) | 0 | | ns | [1,3] |
| 4 | ThRxD(RxCf) | RxD to /RxC Fall Hold Time (x1 Mode) | 20 | | ns | [1,3] |
| 5 | TsSy(RxC) | /DCD as /SYNC to /RxC Rise Setup Time | 0 | | ns | [1] |
| 6 | ThSy(RxC) | /DCD as /SYNC to /RxC Rise Hold Time (x1 Mode) | 20 | | ns | [1] |
| 7 | TdTxCf(TxD) | /TxC Fall to TxD Delay | | 25 | ns | [2] |
| 8 | TdTxCr(TxD) | /TxC Rise to TxD Delay | | 25 | ns | [2,3] |
| 9 | TwRxCh | /RxC High Width | 20 | | ns | |
| 10 | TwRxCl | /RxC Low Width | 20 | | ns | |
| 11 | TcRxC | /RxC Cycle Time | 50 | | ns | |
| 12 | TwTxCh | /TxC High Width | 20 | | ns | |
| 13 | TwTxCl | /TxC Low Width | 20 | | ns | |
| 14 | TcTxC | /TxC Cycle Time | 50 | | ns | |
| 15 | TwExT | /DCD or /CTS Pulse Width | 35 | | ns | |
| 16 | TWSY | /DCD as /SYNC Input Pulse Width | 35 | | ns | |

## AC CHARACTERISTICS

System Timing Diagram

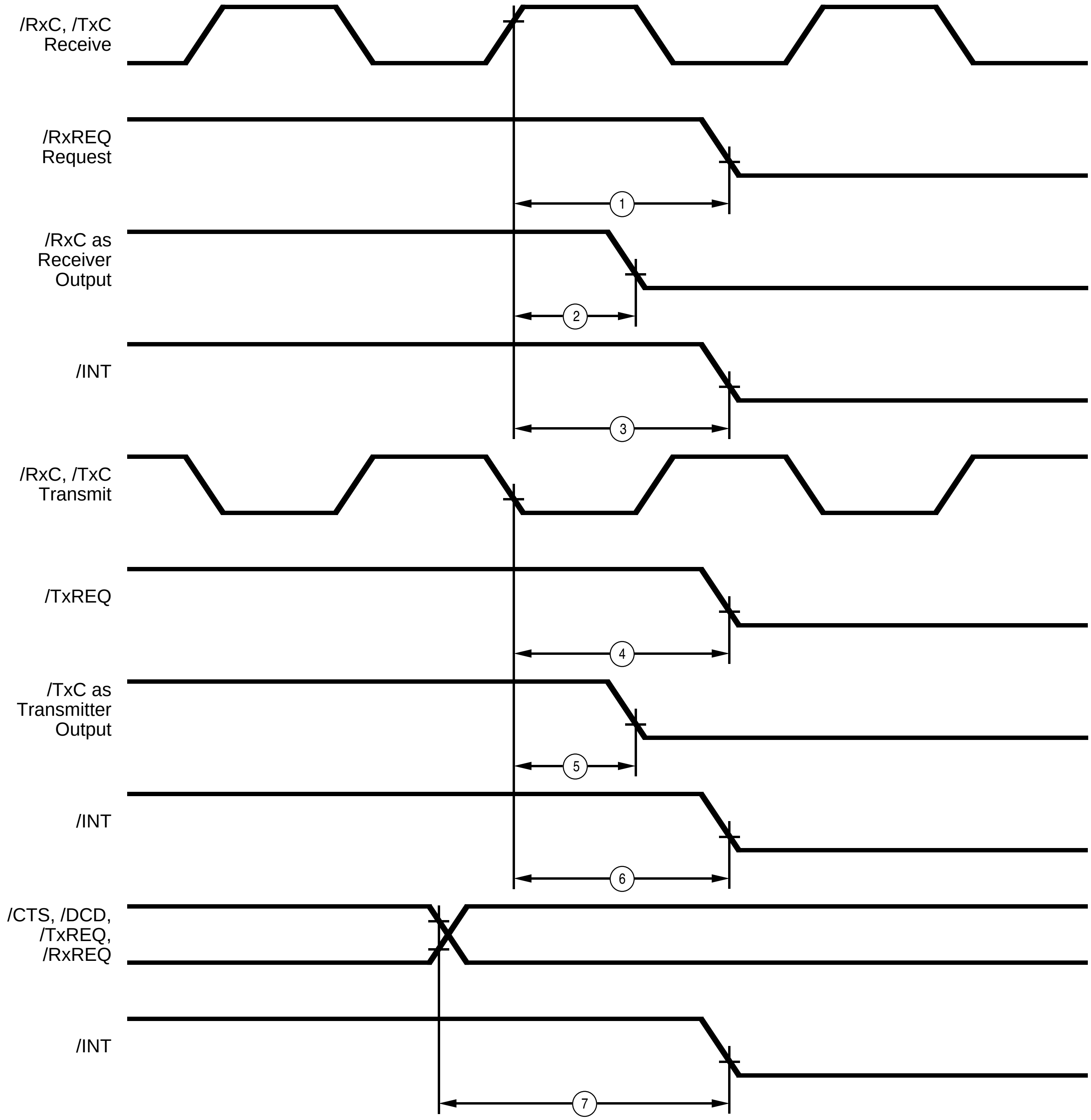



**Figure 106. Z16C32 System Timing**

# AC CHARACTERISTICS

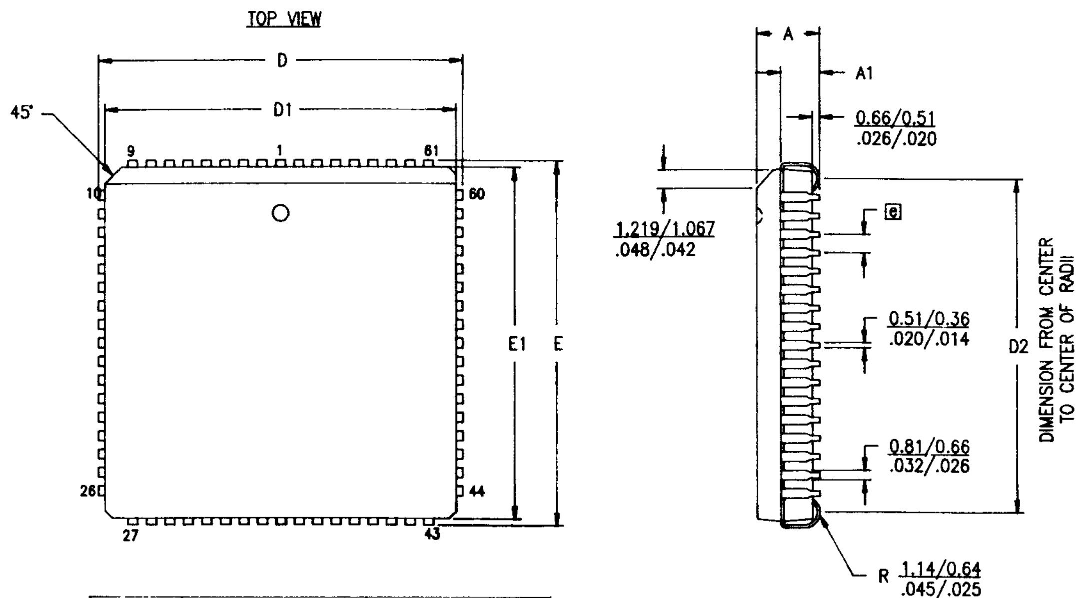
## System Timing Table

| No | Symbol | Parameter | Min | Max | Units | Note |
|---|---|---|---|---|---|---|
| 1 | TdRxC(REQ) | /RxC Rise to /RxREQ Valid Delay | | 50 | ns | [2] |
| 2 | TdRxC(RxC) | /TxC Rise to /RxC as Receiver Output Valid Delay | | 50 | ns | [2] |
| 3 | TdRxC(INT) | /RxC Rise to /INT Valid Delay | | 50 | ns | [2] |
| 4 | TdTxC(REQ) | /TxC Fall to /TxREQ Valid Delay | | 50 | ns | [2] |
| 5 | TdTxC(TxC) | /RxC Fall to /TxC as transmitter Output Valid Delay | | 50 | ns | |
| 6 | TdTxC(INT) | /TxC Fall to /INT Valid Delay | | 50 | ns | [2] |
| 7 | TdEXT(INT) | /CTS, /DCD, /TxREQ, /RxREQ transition to /INT Valid Delay | | 50 | ns | |

**Notes:**

[1] /RxC is /RxC or /TxC, whichever is supplying the receive clock.
[2] /TxC is /TxC or /RxC, whichever is supplying the transmit clock.
[3] Parameter applies only to FM encoding/decoding.

## PACKAGE INFORMATION




| 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 | |

NOTES:
1. CONTROLLING DIMENSIONS : INCH
2. LEADS ARE COPLANAR WITHIN .004 IN RANGE.
3. DIMENSION : MM / INCH

**68-Pin PLCC Package Diagram**

# ORDERING INFORMATION

## Z16C32 IUSC

### 20 MHz
**68-Pin PLCC**
Z16C3220VSC

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

### Package
V = Plastic Chip Carrier

### Temperature
S = 0°C to 70°C

### Speed
20 = 20 MHz

### Environmental
C = Plastic Standard

### Example:
**Z 16C32 20 V S C** is a Z16C32, 20 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow

Environmental Flow
Temperature
Package
Speed
Product Number
Zilog Prefix



Zilog, Inc. 210 East Hacienda Ave.
Campbell, CA 95008-6600
Telephone (408) 370-8000
Telex 910-338-7621
FAX 408 370-8056
Internet: http://www.zilog.com