SMC®

# Ethernet System Controller

- 83C795

Data Book

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## 1.0 GENERAL DESCRIPTION

The SMC 83C795 Ethernet System Controller implements the IEEE 802.3 protocol for networks such as Ethernet, Cheapernet, and 10BaseT. It is a highly integrated device that shrinks the essence of a LAN adapter card onto a single piece of silicon. It includes the 802.3 Media Access Control (MAC) functions, the Physical Layer Interface (PLI) for 10BASE-T media, and a host interface designed for simple connection to the Industry Standard Architecture (ISA) PC/AT bus.

To create a LAN adapter only the 83C795, a single buffer RAM, an EEROM chip, and an optional ROM for BIOS or IPL code storage are required. Transformers and supporting analog components complete an adapter design. All necessary control logic is provided by the 83C795.

The resulting LAN adapter appears to the host as a block of I/O registers with a block of shared memory, unless the I/O pipe is used. The base address for I/O registers is programmable as is the base address and size of the buffer memory.

This device is similar to the SMC 83C790 LAN controller with three major differences:

- A small memory cache has been added for host accesses to shared memory.
- An I/O pipe mode has been added to access the buffer memory.
- Auto-configurability logic has been added in order to comply with the new ISA Plug and Play specification.

As with the 83C790 chip, there are two basic modes of operation: normal and ALTEGO. In the normal mode, the LAN controller operates much like the 83C690 LAN Controller with received frames being buffered in a ring of contiguous, fixed-size buffers. When the ALTEGO feature is enabled, the device switches to a very different mode of operation. The differences are summarized here and explained in detail throughout the specification:

1. Linked-list style of buffering instead of ring buffers.
2. Different register map for LAN controller, exposing new registers for the linked-list buffering.

Figure 1-1 depicts the 83C795's functionality.

## 2.0 FEATURES

The basic features of the 83C795 chip are summarized here:

- Memory caching with time-shared access to buffer RAM.
- Compliant with the ISA Plug And Play specification
- Software compatible with 83C790 drivers
- Direct interface with ISA bus without TTL buffers
- I/O-mapped pipe access to buffer RAM
- Extended length option for the twisted-pair port
- Underrun detection in early receive mode
- Staggered address transfers supported
- Ring-empty bit supplied to host
- Automatic ring-wrapping
- PC-98 bus support through addition of a jumper
- Buffered 20 MHz clock output available through addition of a jumper
- Support for diskless workstations via Initial Program Load ROM
- Programmable base address and window size for buffer memory and IPL ROM
- Support for paging of buffer memory and IPL ROM
- Programmable I/O base address
- Programmable bus width of either 8 or 16 bits
- Zero wait state operation
- Automatic loading of host interface configuration and LAN address from external serial EEPROM
- Separate address and data busses to memory with no external address latches
- 7 programmable interrupt levels
- Clock oscillator
- Full 802.3 MAC layer protocol implementation with extended features
- Support for transmission and reception of frames up to 32K bytes long
- Transmit frame start at any location - no word alignment required
- Two modes of frame buffering: 83C690 mode and descriptor table mode
- Loopback modes - internal and external
- Full-duplex DMA capability in loopback mode
- Built-in AUI serial interface including drivers and receivers

- Built-in 10BASE-T serial interface for Ethernet on Twisted-Pair including drivers and receivers
- Manchester Encoder/Decoder with clock recovery circuitry
- Multicast addressing using 64-bit hashing algorithm
- I/O pin mapping enables rapid board test development
- 160-pin PQFP package for surface mounting.

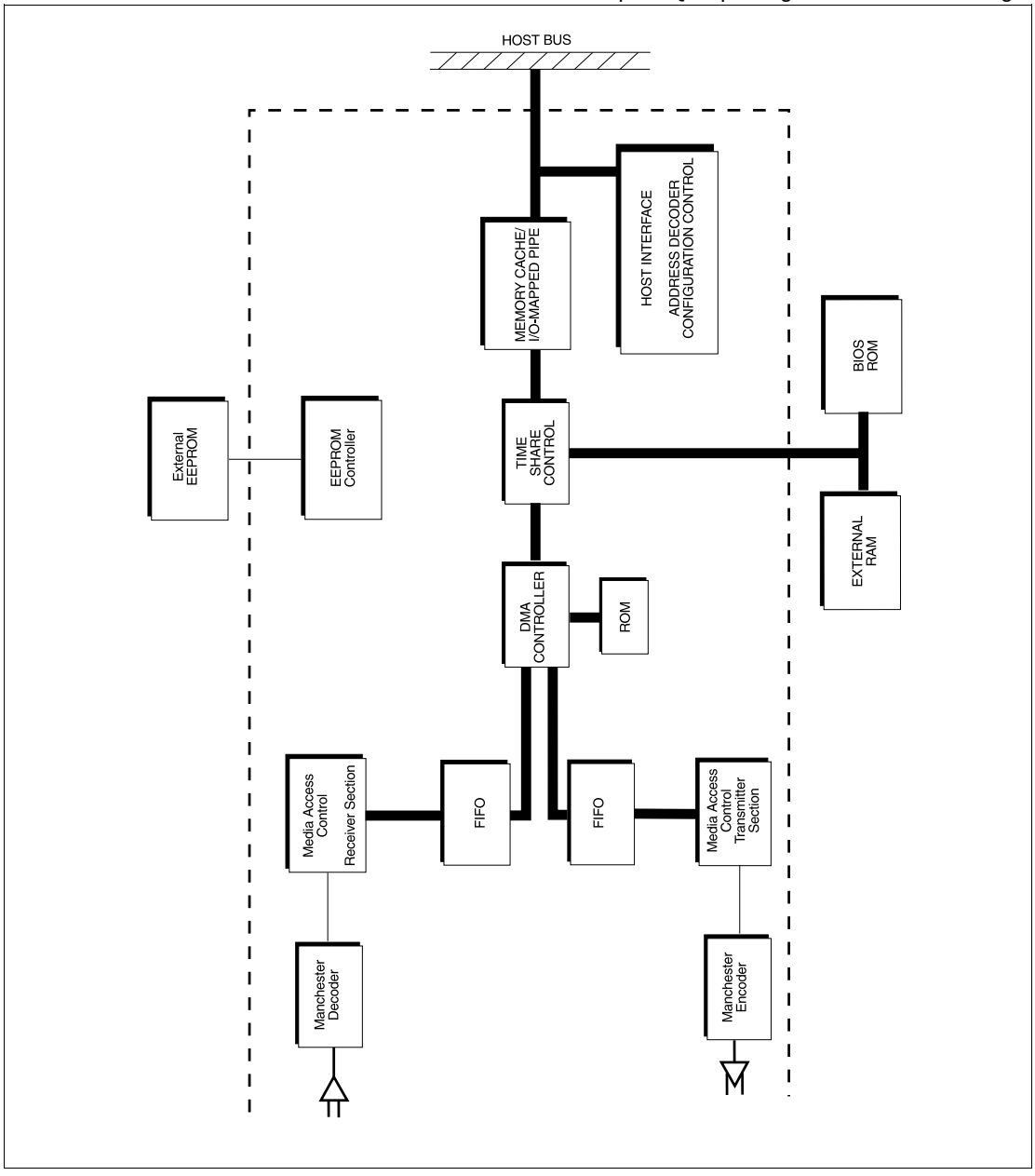


**FIGURE 1-1. 83C795 BLOCK DIAGRAM**

## 3.0 FUNCTIONAL DESCRIPTION

The principle sections of the device are the:

- Host interface which mediates host access to internal registers and buffer memory.
- LAN Controller which performs 802.3 MAC layer protocol and does supporting DMA transfers to and from local buffer memory.
- Serial line interface which supplies drivers and receivers for AUI and TP interfaces and port control logic for the 10BASE-T interface. In addition, it provides Manchester encoding and decoding.

### 3.1 DESCRIPTION OF DATA PATH

Figure 3-1 illustrates the data path for a 83C795-based board. All internal byte swapping is handled by the memory cache. The data bus drivers are designed to drive the ISA bus directly.

### 3.2 CONVENTIONS

A number of conventions are used in this databook.

1. A bit may be described as "low" or "logical 0" when its value is set to 0. A bit is described as "high" or "logical 1" when its value is set to 1.
2. The location of a bit is frequently described in this manner:

   <register>.<bit>

   For example, since the bit MENB is located in the CR Register, it might be described in this book as CR.MENB. Or, alternatively, since MENB is the sixth bit in the CR Register, it might also be described in this manner: CR.6

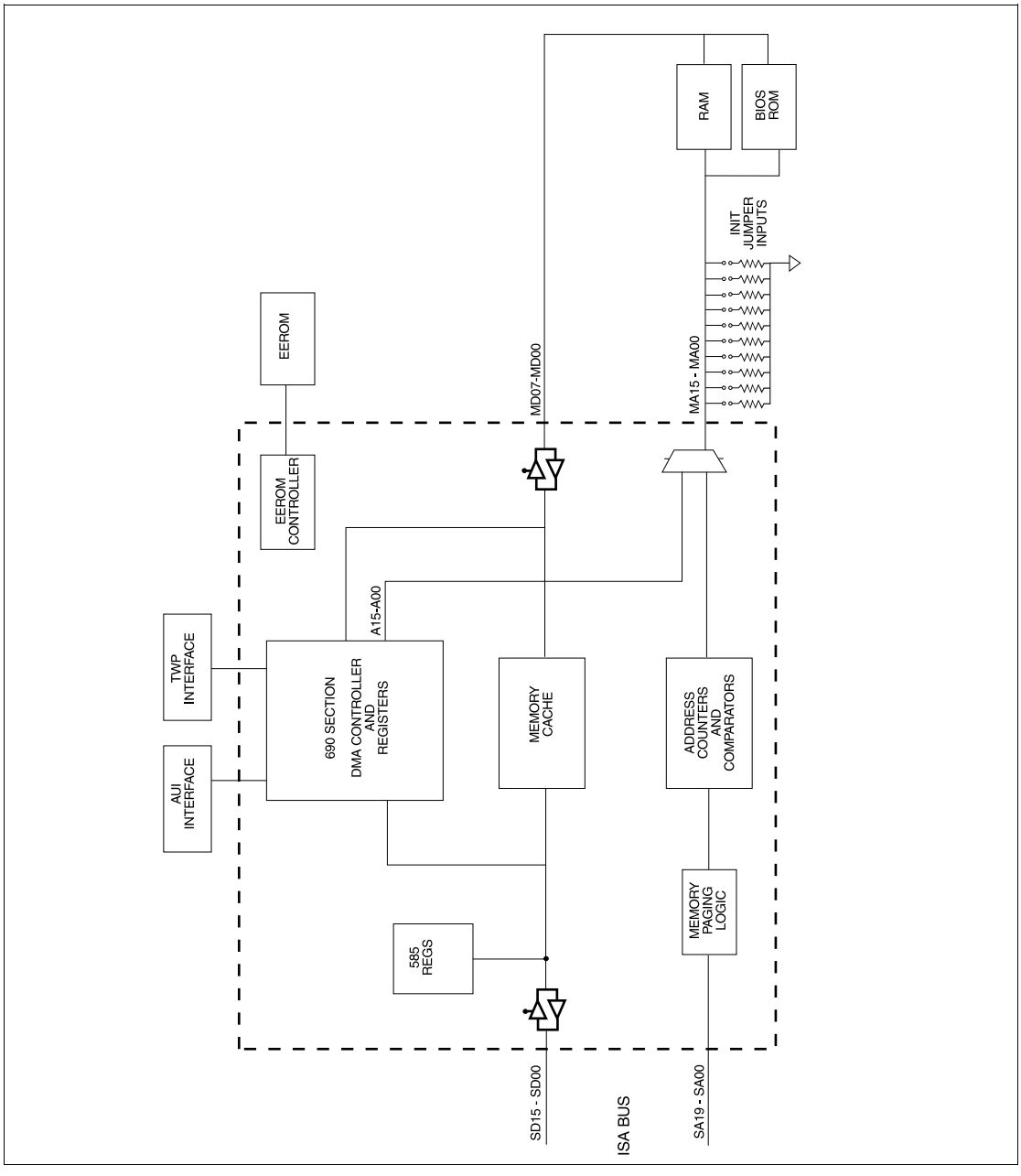


**FIGURE 3-1. 83C790 DATA PATH FLOW**

## 1.0 GENERAL DESCRIPTION

The SMC 83C795 Ethernet System Controller implements the IEEE 802.3 protocol for networks such as Ethernet, Cheapernet, and 10BaseT. It is a highly integrated device that shrinks the essence of a LAN adapter card onto a single piece of silicon. It includes the 802.3 Media Access Control (MAC) functions, the Physical Layer Interface (PLI) for 10BASE-T media, and a host interface designed for simple connection to the Industry Standard Architecture (ISA) PC/AT bus.

To create a LAN adapter only the 83C795, a single buffer RAM, an EEROM chip, and an optional ROM for BIOS or IPL code storage are required. Transformers and supporting analog components complete an adapter design. All necessary control logic is provided by the 83C795.

The resulting LAN adapter appears to the host as a block of I/O registers with a block of shared memory, unless the I/O pipe is used. The base address for I/O registers is programmable as is the base address and size of the buffer memory.

This device is similar to the SMC 83C790 LAN controller with three major differences:

- A small memory cache has been added for host accesses to shared memory.
- An I/O pipe mode has been added to access the buffer memory.
- Auto-configurability logic has been added in order to comply with the new ISA Plug and Play specification.

As with the 83C790 chip, there are two basic modes of operation: normal and ALTEGO. In the normal mode, the LAN controller operates much like the 83C690 LAN Controller with received frames being buffered in a ring of contiguous, fixed-size buffers. When the ALTEGO feature is enabled, the device switches to a very different mode of operation. The differences are summarized here and explained in detail throughout the specification:

1. Linked-list style of buffering instead of ring buffers.
2. Different register map for LAN controller, exposing new registers for the linked-list buffering.

Figure 1-1 depicts the 83C795's functionality.

## 2.0 FEATURES

The basic features of the 83C795 chip are summarized here:

- Memory caching with time-shared access to buffer RAM.
- Compliant with the ISA Plug And Play specification
- Software compatible with 83C790 drivers
- Direct interface with ISA bus without TTL buffers
- I/O-mapped pipe access to buffer RAM
- Extended length option for the twisted-pair port
- Underrun detection in early receive mode
- Staggered address transfers supported
- Ring-empty bit supplied to host
- Automatic ring-wrapping
- PC-98 bus support through addition of a jumper
- Buffered 20 MHz clock output available through addition of a jumper
- Support for diskless workstations via Initial Program Load ROM
- Programmable base address and window size for buffer memory and IPL ROM
- Support for paging of buffer memory and IPL ROM
- Programmable I/O base address
- Programmable bus width of either 8 or 16 bits
- Zero wait state operation
- Automatic loading of host interface configuration and LAN address from external serial EEPROM
- Separate address and data busses to memory with no external address latches
- 7 programmable interrupt levels
- Clock oscillator
- Full 802.3 MAC layer protocol implementation with extended features
- Support for transmission and reception of frames up to 32K bytes long
- Transmit frame start at any location - no word alignment required
- Two modes of frame buffering: 83C690 mode and descriptor table mode
- Loopback modes - internal and external
- Full-duplex DMA capability in loopback mode
- Built-in AUI serial interface including drivers and receivers

- Built-in 10BASE-T serial interface for Ethernet on Twisted-Pair including drivers and receivers
- Manchester Encoder/Decoder with clock recovery circuitry
- Multicast addressing using 64-bit hashing algorithm
- I/O pin mapping enables rapid board test development
- 160-pin PQFP package for surface mounting.

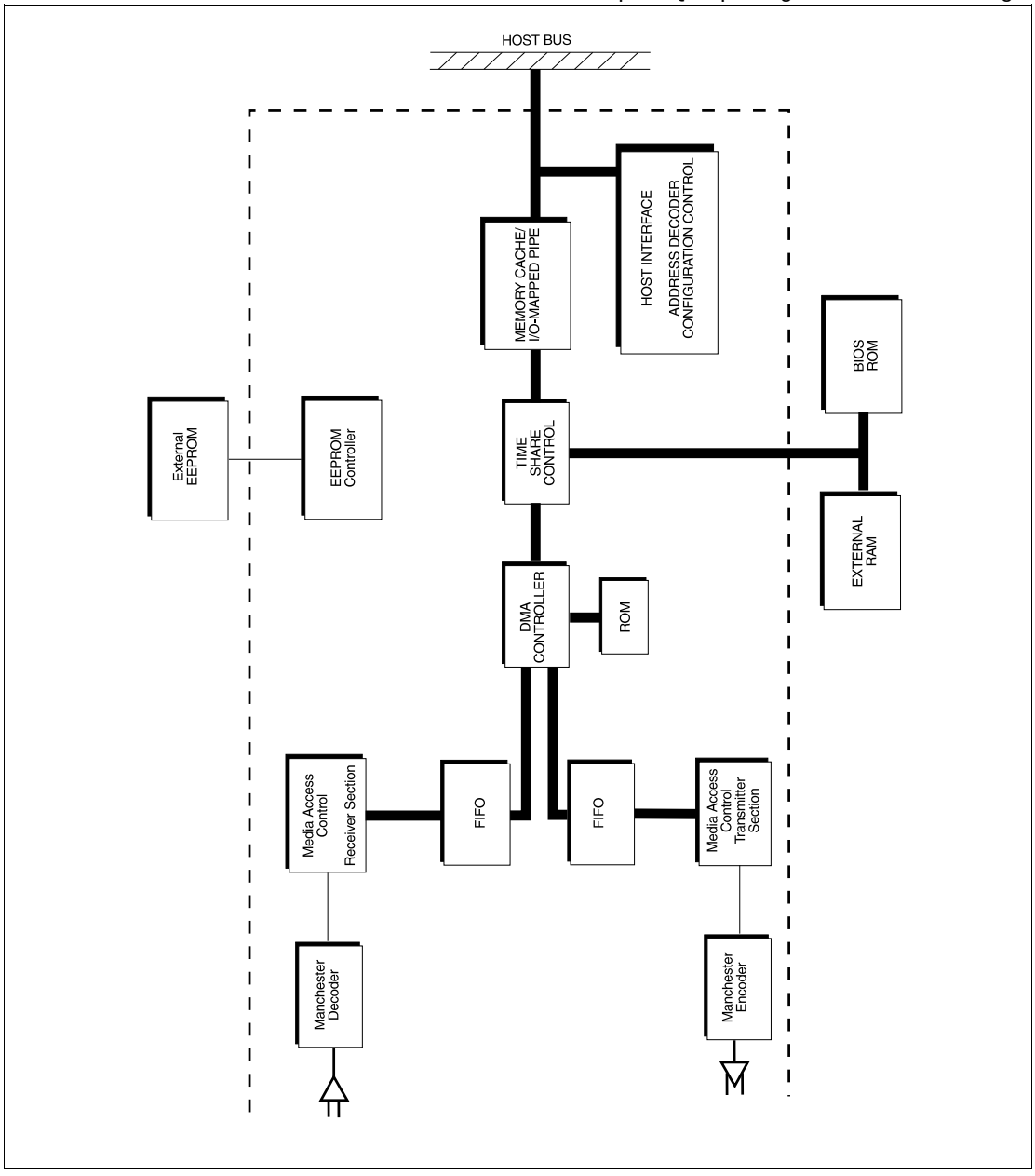


**FIGURE 1-1. 83C795 BLOCK DIAGRAM**

## 3.0 FUNCTIONAL DESCRIPTION

The principle sections of the device are the:

- Host interface which mediates host access to internal registers and buffer memory.
- LAN Controller which performs 802.3 MAC layer protocol and does supporting DMA transfers to and from local buffer memory.
- Serial line interface which supplies drivers and receivers for AUI and TP interfaces and port control logic for the 10BASE-T interface. In addition, it provides Manchester encoding and decoding.

### 3.1 DESCRIPTION OF DATA PATH

Figure 3-1 illustrates the data path for a 83C795-based board. All internal byte swapping is handled by the memory cache. The data bus drivers are designed to drive the ISA bus directly.

### 3.2 CONVENTIONS

A number of conventions are used in this databook.

1. A bit may be described as "low" or "logical 0" when its value is set to 0. A bit is described as "high" or "logical 1" when its value is set to 1.
2. The location of a bit is frequently described in this manner:

   <register>.<bit>

   For example, since the bit MENB is located in the CR Register, it might be described in this book as CR.MENB. Or, alternatively, since MENB is the sixth bit in the CR Register, it might also be described in this manner: CR.6

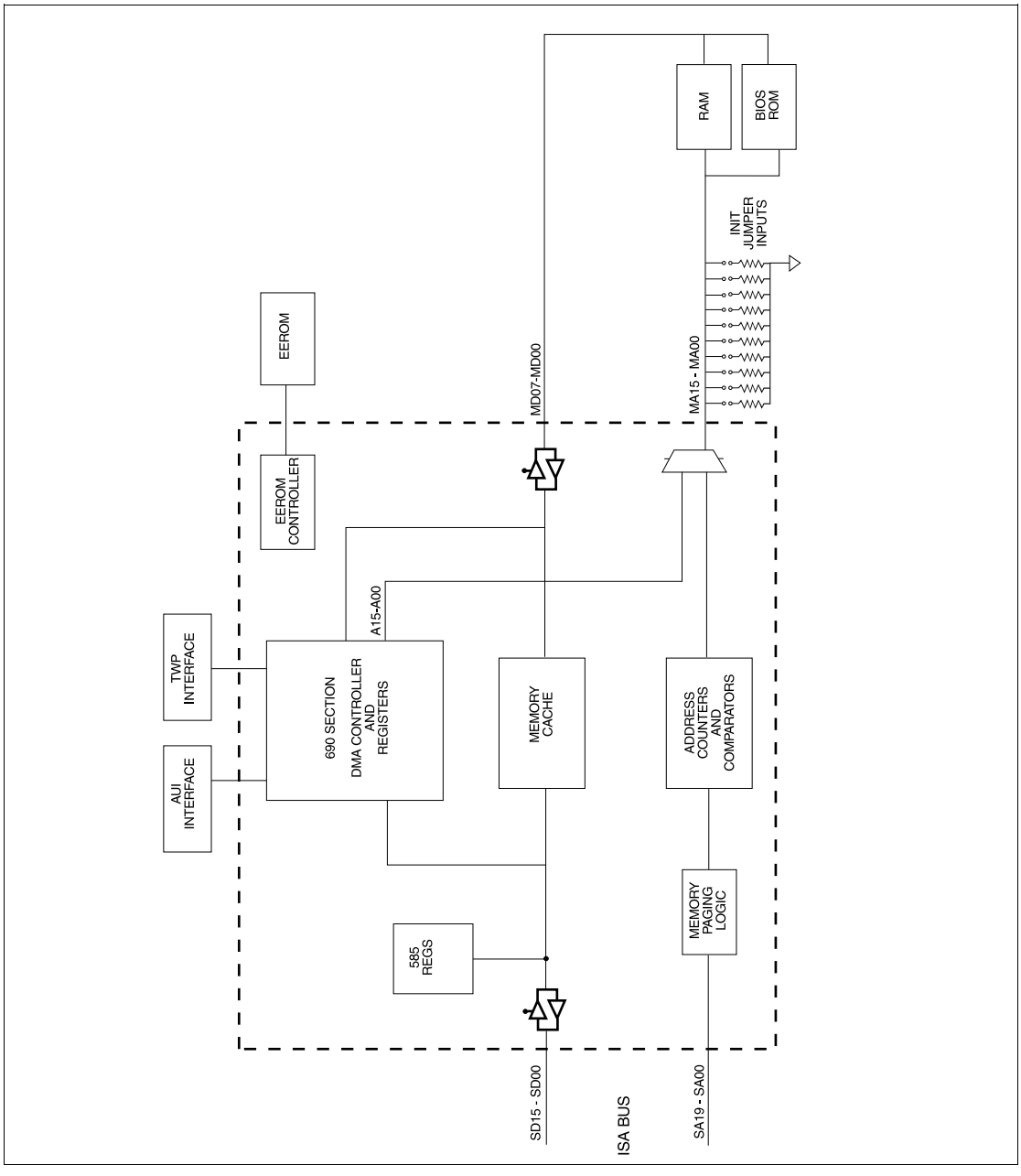


**FIGURE 3-1. 83C790 DATA PATH FLOW**

## 4.0 PIN LIST

This section provides the pin list and pin/signal descriptions for the 83C795. Section 4.1 describes the input-to-output pin mapping feature.

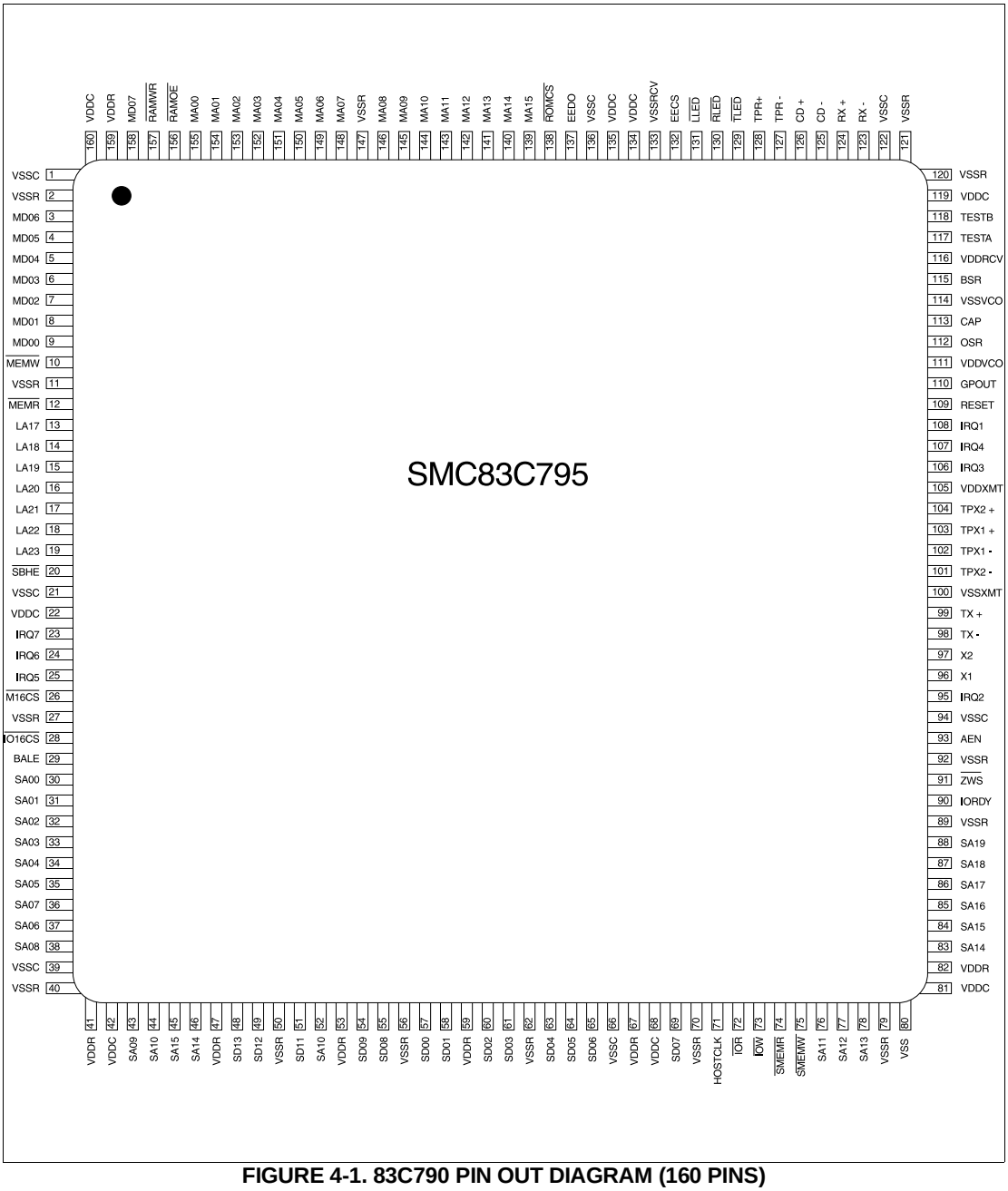


**FIGURE 4-1. 83C790 PIN OUT DIAGRAM (160 PINS)**

| Mnemonic | Pin Number | I/O | Description |
|---|---|---|---|
| $\overline{AEN}$ | 93 | I | PC ADDRESS ENABLE. Active low. When Address Enable is active the 83C795 responds to any host strobe (IOR, IOW, MEMR, MEMW, SMEMR, SMEMW). |
| BALE | 29 | I | PC ADDRESS LATCH ENABLE. Used to latch valid addresses from the LA bus. Passes LA signals through internal latches while high and latches them on falling edge. |
| BSR | 115 | I | BIAS RESISTOR. A resistor from BSR to VDD sets the internal bias levels. Nominal value is 10KΩ. |
| CAP | 113 | I | PLL FILTER CAP. A capacitor (nominal value .01µF) from CAP to ground is used as part of the filter for the internal phase lock loop. |
| CD+<br>CD- | 126<br>125 | I | AUI COLLISION. CD+ / CD- are used by the external transceiver to signal a collision by sending a 10MHz signal. |
| EECS | 132 | O | EEROM Chip Select. An external 9356 serial EEROM is used to store up to 2048 bits of configuration data. These signals (along with LLED and RLED) interface with that chip. |
| EEDO | 137 | I/O | EEROM DATA OUTPUT. |
| GPOUT | 110 | O | GENERAL PURPOSE OUTPUT. In some systems, this bit is wired to a shutdown control input of the DC/DC isolated power supply used in 10Base2 applications. In other systems, this supplies a control signal for switching power supplies. (The DC signal's polarity on the 83C795 is the opposite of the 83C790.) |
| HOSTCLK | 71 | I/O | Personal Computer BUS CLOCK. |
| $\overline{IO16CS}$ | 28 | O | 16 BIT I/O SELECTED. Active low. Indicates to the PC/AT bus that the I/O response will be 16 bits wide. Only used for the I/O pipe. |
| $\overline{IOR}$ | 72 | I/O | PC I/O READ. Active low. Reads an I/O register onto the PC data bus. |
| IORDY | 90 | O | I/O IORDY. Response to host access which can be used directly as 'I/O Channel Ready' when responding to a personal computer bus. It is pulled low (not ready) to lengthen I/O or memory cycles. When the 83C795 is ready to respond, the signal is driven high until the host access is completed, then becomes tristated. This signal is driven by a tri-state buffer capable of sinking 24 mA. |
| $\overline{IOW}$ | 73 | I/O | PC I/O WRITE. Active low. Writes an I/O register from the PC data bus. |
| IRQ1 | 108 | I/O | PC INTERRUPT REQUEST LINES. Active high. Tristated when not active. IRQ2-9 on the PC/AT bus. IRQ1 is the same as the XTXD pin in some test modes. |
| IRQ2 | 95 | I/O | IRQ3 on PC/AT bus. Same as XLOOP in some test modes. |
| IRQ3 | 106 | I/O | IRQ5 on PC/AT bus. Same as XCRS in some test modes. |
| IRQ4 | 107 | I/O | IRQ7 on PC/AT bus. Same as XRXC in some test modes. |
| IRQ5 | 25 | I/O | IRQ10 on PC/AT bus. Same as XRXD in some test modes. |
| IRQ6 | 24 | I/O | IRQ11 on PC/AT bus. Same as XCOL in some test modes. |
| IRQ7 | 23 | I/O | IRQ15 on PC/AT bus. Same as XTXC in some test modes. |

**TABLE 4-1. 83C795 PIN ASSIGNMENTS**

| Mnemonic | Pin Number | I/O | Description |
|---|---|---|---|
| LA17-LA23 | 13-19 | I/O | PCLA ADDRESS BUS. Advanced timing version of system address lines A23-A17 from PC/AT bus. These are not assumed to be stable during the entire host cycle and are latched internally by the BALE signal (falling edge). These signals are active high. |
| $\overline{\text{LLED}}$ | 131 | I/O | TWPR LINK STATUS. If valid data or Link Test pulses are received on TPR+ /TPR-, LLED is low (link status OK). When no data or Link Test pulses are received, LLED is high. The LLED pin can sink 4mA to drive an external LED. This pin also functions as the EEROM clock pin (formerly EESK) and functions as the shift control pin (formerly SHIFTIN) in scan mode. |
| $\overline{\text{M16CS}}$ | 26 | O | MEMORY 16 SELECTED. Active low. Indicates to the PC/AT bus that RAM access response is 16 bits wide. |
| MA15-MA10 | 139-144 | O | MEMORY ADDRESS LINES. These pins bring out the DMA address to memory or feed-through the host address as modified for paging. When dumping ROM contents, these pins present ROM data bits:<br>MA15 ROM30 then ROM31<br>MA14 ROM28 then ROM29<br>. . .<br>MA00 ROM00 then ROM01 |
| MA09/JMP9<br>MA08/JMP8<br>MA07/JMP7<br>MA06/JMP6<br>MA05/JMP5<br>MA04/JMP4<br>MA03/JMP3<br>MA02/JMP2<br>MA01/JMP1<br>MA00/JMP0 | 145<br>146<br>148<br>149<br>150<br>151<br>152<br>153<br>154<br>155 | I/O | Same as other MA lines except that during RESET, the drivers are disabled and the INIT jumpers are read through these pins and latched at the trailing edge of RESET. They are active high and are pulled up weakly by internal resistors (35kΩ–150kΩ).<br><br>To set a zero value on these pins, use an external pull-down resistor of 3.6KΩ. |
| MD7<br>MD6-MD0 | 158<br>3-9 | I/O | MEMORY DATA LINES. These pins connect to the data pins of the buffer RAM and the IPL or boot ROM. |
| $\overline{\text{MEMR}}$ | 12 | I/O | PC MEMORY READ for addresses exceeding 1 M. Active low. |
| $\overline{\text{MEMW}}$ | 10 | I/O | PC MEMORY WRITE for addresses exceeding 1 M. Active low. |
| OSR | 112 | I | VCO BIAS RESISTOR. A resistor from OSR to VCC biases the internal VCO current. Nominal value is 24.9KΩ. |
| $\overline{\text{RAMOE}}$ | 156 | O | RAM OUTPUT ENABLE. Active low. |
| $\overline{\text{RAMWR}}$ | 157 | O | RAM WRITE ENABLE. Active low. |
| RESET | 95 | I/O | SYSTEM RESET. Active high. |

**TABLE 4-1. 83C795 PIN ASSIGNMENTS (CONT.)**

| Mnemonic | Pin Number | I/O | Description |
|---|---|---|---|
| RLED | 130 | I/O | RECV LED DRIVER. When on, RLED drives low to turn on an external LED. If no data is received, RLED is off. If data is received, RLED goes active for approximately 50ms longer than the received packet length. This pin also serves as the EEROM data in pin (EEDI) and is used as the scan data input (formerly SCANIN) in scan mode. |
| ROMCS | 138 | O | BIOS ROM OUTPUT ENABLE. Active low. Chip Select for ROM BIOS. Used for XTXE in some test modes. |
| RX+<br>RX- | 124<br>123 | I | AUI RECEIVE. The Manchester encoded data from the external transceiver is received on RX+ / RX-. |
| SA19 - 14<br>SA13 - 11<br>SA10 - 09<br>SA08 - 06<br>SA05 - 00 | 88 - 83<br>78 - 76<br>44 - 43<br>38 - 37<br>35 - 30 | I/O | PC ADDRESS BUS. Normal timing version of system address lines. These signals are active high. |
| SBHE | 20 | I | PC BUS HIGH ENABLE. Active low. Indicates a transfer of data on the upper byte of the data bus, SD8 through SD15. |
| SD15 - 14<br>SD13 - 12<br>SD11 - 10<br>SD09 - 08<br>SD07<br>SD06 - 04<br>SD03 - 02<br>SD01 - 00 | 45 - 46<br>48 - 49<br>51 - 52<br>54 - 55<br>69<br>65 - 63<br>61 - 60<br>58 - 57 | I/O | SYSTEM DATA BUS. Data input and output to host and shared memory. These signals are active high. |
| SMEMR | 74 | I/O | LOW MEMORY READ STROBE. Reads buffer memory onto the PC data bus. It is active only when the memory address is within the first 1Mb of memory space. Active low. |
| SMEMW | 75 | I/O | LOW MEMORY WRITE STROBE. Writes buffer memory from the PC data bus. SMEMW is active only when the memory address is within the first 1Mb of memory space. Active low. |
| TESTA<br>TESTB | 117<br>118 | I | I/O-MAPPING CONTROL PINS. All signal pins on the 83C795 (except for the analog pins) have been divided into two groups, GROUPA and GROUPB. The functionality of these pins is controlled by the two test pins, as follows:<br><br>TESTA TESTB GROUPA GROUPB<br>1 1 NORMAL NORMAL<br>1 0 1 0<br>0 1 0 1<br>0 0 TRISTATE TRISTATE |

**TABLE 4-1. 83C795 PIN ASSIGNMENTS (CONT.)**

| Mnemonic | Pin Number | I/O | Description |
|---|---|---|---|
| $\overline{TLED}$ | 129 | O | TRANSMIT LED DRIVER. When on, $\overline{TLED}$ drives low to turn on an external LED. When there is no transmission (TXE inactive), TLED is off. When data is transmitted, TLED goes active for approximately 50ms longer than the transmitted packet length. TLED does not go active for Link Test pulses. This pin also serves as the scan data output (formerly SCANOUT) in scan mode, or as a 20 MHz buffered clock output if JUMPER8 is installed. |
| TPR+<br>TPR- | 128<br>127 | I | TWPR RECEIVE. In 10BaseT operation, Manchester encoded-data are received via TPR+ /TPR-. They are connected to the twisted pair medium through a transformer and filter. |
| TPX1+<br>TPX1- | 103<br>102 | O | TWPR TRANSMIT. TPX1+ and TPX1- are used for 10BaseT only. They are the high current positive and negative output pins. |
| TPX2+<br>TPX2- | 104<br>101 | O | TWPR TRANSMIT. TPX2+ and TPX2- are used for 10BaseT only. They are the low current positive and negative output pins. |
| TX+<br>TX- | 99<br>98 | O | AUI TRANSMIT. TX+ and TX- transmit differential, Manchester encoded data to the transceiver. These are current-driving outputs that furnish ECL level signals when connected to required external pullup resistors of 150Ω. |
| VDD<br>(14 pins) | 22, 41, 42, 47, 53, 59, 67, 68, 81, 82, 105, 111, 116, 119, 120, 134, 135, 159, 160 | | +5 VOLT SOURCES. Some are for logic, some power the pin drivers, and others provide power to the analog portions of the circuit. |
| VSS<br>(24 pins) | 1, 2, 11, 21, 27, 39, 40, 50, 56, 62, 66, 70, 79, 80, 89, 92, 94, 100, 114, 121, 122, 133, 136, 147 | | GROUND. Some are for logic, some power the pin drivers, and others provide power to the analog portions of the circuit. |
| X1<br>X2 | 96<br>97 | I<br>O | CRYSTAL OSCILLATOR. The crystal is attached across these two pins. Must be 20.000 MHz ±50 ppm. This clock operates the chip's logic and is divided by 2 internally to become the transmit clock. |
| $\overline{ZWS}$ | 91 | O | PC ZERO WAIT STATE. Active low. Zero Wait State signal tells the microprocessor that it can complete the present bus cycle without inserting any additional wait cycles. ZWS is driven by a tri-state driver capable of sinking 24 mA. |

**TABLE 4-1. 83C795 PIN ASSIGNMENTS (CONT.)**

### 4.1 SPECIAL INPUT-TO-OUTPUT PIN MAPPING

The 83C795 provides a special pin mapping feature which can be used for testing. The pins are divided into two groups, GROUPA and GROUPB, for the new I/O pin mapping scheme.

| GROUPA PINS | | GROUPB PINS | |
|---|---|---|---|
| **PIN** | **NAME** | **PIN** | **NAME** |
| 3 | MD06 | 4 | MD05 |
| 5 | MD04 | 6 | MD03 |
| 7 | MD02 | 8 | MD01 |
| 9 | MD00 | 10 | $\overline{\text{MEMW}}$ |
| 12 | $\overline{\text{MEMR}}$ | 13 | LA17 |
| 14 | LA18 | 15 | LA19 |
| 16 | LA20 | 17 | LA21 |
| 18 | LA22 | 19 | LA23 |
| 20 | $\overline{\text{SBHE}}$ | 23 | IRQ7 |
| 24 | IRQ6 | 25 | IRQ15 |
| 26 | $\overline{\text{M16CS}}$ | 28 | $\overline{\text{IO16CS}}$ |
| 29 | BALE | 30 | SA00 |
| 31 | SA01 | 32 | SA02 |
| 33 | SA03 | 34 | SA04 |
| 35 | SA05 | 36 | SA07 |
| 37 | SA06 | 38 | SA08 |
| 43 | SA09 | 44 | SA10 |
| 45 | SD15 | 46 | SD14 |
| 48 | SD13 | 49 | SD12 |
| 51 | SD11 | 52 | SD10 |
| 54 | SD09 | 55 | SD08 |
| 57 | SD00 | 58 | SD01 |
| 60 | SD02 | 61 | SD03 |
| 63 | SD04 | 64 | SD05 |
| 65 | SD06 | 69 | SD07 |
| 71 | BSCK | 72 | $\overline{\text{IOR}}$ |
| 73 | $\overline{\text{IOW}}$ | 74 | $\overline{\text{SMEMR}}$ |
| 75 | $\overline{\text{SMEMW}}$ | 76 | SA11 |
| 77 | SA12 | 78 | SA13 |
| 83 | SA14 | 84 | SA15 |
| 85 | SA16 | 86 | SA17 |
| 87 | SA18 | 88 | SA19 |

**TABLE 4-2. I/O PIN MAPPING SCHEME**

| GROUPA PINS | | GROUPB PINS | |
|---|---|---|---|
| **PIN** | **NAME** | **PIN** | **NAM,E** |
| 90 | IORDY | 91 | $\overline{ZWS}$ |
| 93 | AEN | 95 | IRQ2 |
| 106 | IRQ3 | 107 | IRQ4 |
| 108 | IRQ1 | 109 | RESET |
| 110 | GPOUT | 129 | RXM |
| 130 | $\overline{RLED}$ | 131 | $\overline{LLED}$ |
| 132 | EECS | 137 | EEDO |
| 138 | $\overline{ROMCS}$ | 139 | MA15 |
| 140 | MA14 | 141 | MA13 |
| 142 | MA12 | 143 | MA11 |
| 144 | MA10 | 145 | MA09 |
| 146 | MA08 | 148 | MA07 |
| 149 | MA06 | 150 | MA05 |
| 151 | MA04 | 152 | MA03 |
| 153 | MA02 | 154 | MA01 |
| 155 | MA00 | 156 | $\overline{RAMOE}$ |
| 157 | $\overline{RAMWR}$ | 158 | MD07 |

**TABLE 4-2. I/O PIN MAPPING SCHEME (CONT.)**

The I/O pin mapping test for this chip requires only two static vectors. The inputs you must set along with the desired output values are listed in Table 4-3.

| VECTOR | PIN DRIVEN HIGH | PINS DRIVEN LOW | PINS OUTPUTTING 1's | PINS OUTPUTTING 0's |
|---|---|---|---|---|
| 1 | TESTA, RXP, TPRP, CDP | TESTB, RXM, TPRM, CDM | GROUPA pins | GROUPB pins |
| 2 | TESTB, RXM, TPRM, CDM | TESTA, RXP, TPRP, CDP | GROUPB pins | GROUPA pins |

**TABLE 4-3. I/O PIN OUTPUT VALUES**

**Note**

This arrangement does not test the 83C795's analog outputs.

## 5.0 ETHERNET SYSTEM CONTROLLER REGISTERS

The register structure of the 83C795 is divided into two register groups: the Host Interface registers and the LAN Controller registers. The following sections describe the contents of each of these registers, detail the way they are accessed, and how they are used. Tables 5-1 through 5-3 provide a brief overview of these registers.

Refer to the Register Summary tables at the end of this section for a quick reference guide to all relevant register bit values.

| OFFSET | SWH = 0 | SWH = 1 |
|---|---|---|
| 00 | CR | CR |
| 01 | EER | EER |
| 02 | IOPL | IOPL |
| 03 | IOPH | IOPH |
| 04 | HWR | HWR |
| 05 | BPR | BPR |
| 06 | ICR | ICR |
| 07 | REV/IOPA | REV/IOPA |
| 08 | LAN0 | GCR2 |
| 09 | LAN1 | — |
| 0A | LAN2 | IAR |
| 0B | LAN3 | RAR |
| 0C | LAN4 | BIO |
| 0D | LAN5 | GCR |
| 0E | BDID | ERFAL |
| 0F | CKSM | ERFAH |

**TABLE 5-1. HOST INTERFACE REGISTERS SUMMARY**

| OFFSET | PAGE 0 READ | PAGE 0 WRITE | PAGE 1 READ | PAGE 1 WRITE | PAGE 2 READ | PAGE 2 WRITE | PAGE 3 READ | PAGE 3 WRITE |
|---|---|---|---|---|---|---|---|---|
| 10 | CMD | CMD | CMD | CMD | CMD | CMD | CMD | CMD |
| 11 | INCRL | RSTART | STA0 | STA0 | RSTART | INCRL | TEST | TEST |
| 12 | INCRH | RSTOP | STA1 | STA1 | RSTOP | INCRH | RTEST | RTEST |
| 13 | BOUND | BOUND | STA2 | STA2 | TCNTL | void | TTEST | TTEST |
| 14 | TSTAT | TSTARTH | STA3 | STA3 | TSTARTH | void | TEST2 | TEST2 |
| 15 | COLCNT | TCNTL | STA4 | STA4 | NEXT | NEXT | TSTARTL | TSTARTL |
| 16 | {0} | TCNTH | STA5 | STA5 | TCNTH | void | {0} | void |
| 17 | INTSTAT | INTSTAT | CURR | CURR | ENH | ENH | — | — |
| 18 | ERWCNT | ERWCNT | GROUP0 | GROUP0 | RADDL | RADDL | — | — |
| 19 | RENH | RENH | GROUP1 | GROUP1 | RADDH | RADDH | — | — |
| 1A | RCNTL | void | GROUP2 | GROUP2 | TADDL | TADDL | — | — |
| 1B | RCNTH | void | GROUP3 | GROUP3 | TADDH | TADDH | — | — |
| 1C | RSTAT | RCON | GROUP4 | GROUP4 | RCON | void | — | — |
| 1D | ALICNT | TCON | GROUP5 | GROUP5 | TCON | void | — | void |
| 1E | CRCCNT | DCON | GROUP6 | GROUP6 | DCON | void | — | void |
| 1F | MPCNT | INTMASK | GROUP7 | GROUP7 | INTMASK | void | MANCH | MANCH |

**TABLE 5-2. LAN CONTROLLER REGISTERS - NORMAL MODE**

| OFFSET | PAGE 0 READ | PAGE 0 WRITE | PAGE 1 READ | PAGE 1 WRITE | PAGE 2 READ | PAGE 2 WRITE | PAGE 3 READ | PAGE 3 WRITE |
|---|---|---|---|---|---|---|---|---|
| 10 | CMD | CMD | CMD | CMD | CMD | CMD | CMD | CMD |
| 11 | — | RBEGIN | STA0 | STA0 | RBEGIN | — | — | — |
| 12 | — | REND | STA1 | STA1 | REND | — | — | — |
| 13 | CURRL | CURRL | STA2 | STA2 | TBEGIN | void | — | — |
| 14 | TSTAT | TEND | STA3 | STA3 | TEND | void | — | — |
| 15 | COLCNT | TBEGIN | STA4 | STA4 | — | void | TSTARTL | TSTARTL |
| 16 | ERWCNT | ERWCNT | STA5 | STA5 | — | void | {0} | void |
| 17 | INTSTAT | INTSTAT | CURRH | CURRH | ENH | ENH | — | — |
| 18 | RTABL | RTABL | GROUP0 | GROUP0 | RDOWNL | RDOWNL | — | — |
| 19 | RTABH | RTABH | GROUP1 | GROUP1 | RDOWNH | RDOWNH | — | — |
| 1A | TTABL | TTABL | GROUP2 | GROUP2 | TDOWNL | TDOWNL | — | — |
| 1B | TTABH | TTABH | GROUP3 | GROUP3 | TDOWNH | TDOWNH | — | — |
| 1C | RSTAT | RCON | GROUP4 | GROUP4 | RCON | void | — | — |
| 1D | ALICNT | TCON | GROUP5 | GROUP5 | TCON | void | — | void |
| 1E | CRCCNT | — | GROUP6 | GROUP6 | DCON | void | — | void |
| 1F | MPCNT | INTMASK | GROUP7 | GROUP7 | INTMASK | void | MANCH | MANCH |

**TABLE 5-3. LAN CONTROLLER REGISTERS - LINKED-LIST MODE**

## 5.1 HOST INTERFACE INTERNAL REGISTERS

The following section describes the contents of the Host Interface Internal registers. This register set consists of 24 registers arranged in three groups of eight. These three groups are the LAN Address registers, the Hardware Configuration registers (which write to and read from the EEROM), and the Hardware Control registers. The Switch Register bit (HWR.SWH) determines whether the LAN Address Registers (SWH = 0) or the Hardware Configuration Registers (SWH = 1) are visible at any one time. (See HWR - Hardware Support Register on page 16 for more information on this bit.) The Hardware Control registers (including the CR, EER, HWR, BPR, ICR, and REV registers) are always visible.

Bits within registers may also have different functions depending on whether they are read from or written to.

Throughout this section, certain terms are used to describe each register and are defined below:

| Term | Definition |
|---|---|
| RESET | Value During RESET Time |
| INIT | If the INIT3, 2, 1, 0 jumpers = 1001, this value is loaded into the register immediately after RESET TIME. Some of these values are forced by hardware and others are recalled from EEROM. Recall time is on the order of 2 msec. While the initial recall is ongoing, register may have either the RESET or INIT values |
| RECALL | This value is loaded when a recall is performed other than after a RESET. |
| REGISTER VALUES | Possible register values are:<br>1 = logical 1<br>0 = logical 0<br>PIN = value unknown or wired to external pin.<br>EE = value loaded from EEROM.<br>— = not used. |

**TABLE 5-4. REGISTER TERM DEFINITIONS**

### 5.1.1 CR - Control Register

Read/Write Port = 00

This register has control over buffer memory enabling and soft reset of the LAN controller.

| BIT | CR | RESET |
|---|---|---|
| 7 | RNIC | 1 |
| 6 | MENB | 0 |
| 5 | — | 0 |
| 4 | CR4 | 0 |
| 3 | CR3 | 0 |
| 2 | RP15 | 0 |
| 1 | RP14 | 0 |
| 0 | RP13 | 0 |

**Bit 7: RNIC**, Reset Network Interface Controller

Set RNIC to 1 then back to 0 to force a hardware reset to the LAN Controller.

**Bit 6: MENB**, Memory Enable

Set MENB to 1 to enable host access to shared memory.

**Bit 4-3: CR4-CR3**, Reserved For Increase In RP Field

**Bit 2-0: RP15-RP13**, RAM Offset

A buffer address is created by adding the contents of this field to the difference between the buffer base address and the address supplied by the SA19-SA00 lines. This sum is treated as a movable page offset which is then inserted into the buffer window. This offset value should only be used when the memory provided is larger than the window selected.

### 5.1.2 EER - EEROM Register

Read/Write Port = 01

This register controls the stores to and recalls from EEROM. In addition, four input pins are visible through this register. Some bits have different functions when read than when written.

| BIT | EER READ | RESET |
|---|---|---|
| 7 | STO | 0 |
| 6 | RC | 0 |
| 5 | EA4 | 0 |
| 4 | UNLOCK | 0 |
| 3 | JMP3 | PIN |
| 2 | JMP2 | PIN |
| 1 | JMP1 | PIN |
| 0 | JMP0 | PIN |

| BIT | EER WRITE | RESET |
|---|---|---|
| 7 | STO | 0 |
| 6 | RC | 0 |
| 5 | EA4 | 0 |
| 4 | UNLOCK | 0 |
| 3 | EA3 | 0 |
| 2 | EA2 | 1 |
| 1 | EA1 | 1 |
| 0 | EA0 | 0 |

**Bit 7: STO**, Store Into Non-Volatile EEROM
Set this bit to 1 to store the 8 LAN address registers into the EEROM. The bit automatically resets when the store is complete. This function will not be performed unless the storage circuit has been armed by a series of accesses to the EER Register. (See Section 6.5.2.3 for details.)

**Bit 6: RC**, Recall EEROM
Set this bit to 1 to recall the 8 LAN address registers from EEROM. The bit will be automatically reset when the recall is complete.

**Bit 4: UNLOCK**, Unlock Store
This bit is used to unlock the EEPROM for storage operations. (See Section 6.5.2.3 for details.)

**Bits 3-0: JMP3-JMP0**, Initialization Jumpers
These bits are wired to the JMP input pins of the chip. The JMP field value is used by the recall logic to determine which bank of EEROM to load the host interface configuration registers from.

**Bits 5, 3-0: EA4-EA0**, EEROM Address Field
This field determines which bank of EEROM the LAN address registers are stored into or recalled from. This field does not change when a recall operation takes place. EA4 = 1 indicates an access to the Plug and Play resource string.

### 5.1.3 IOPL - I/O Pipe Data Location Low

Read/Write Port = 02

This register and IOPH determine the locations data is read from or written to.

| BIT | IOPL | RESET |
|---|---|---|
| 7 | IOP7 | 0 |
| 6 | IOP6 | 0 |
| 5 | IOP5 | 0 |
| 4 | IOP4 | 0 |
| 3 | IOP3 | 0 |
| 2 | IOP2 | 0 |
| 1 | IOP1 | 0 |
| 0 | IOP0 | 0 |

**Bits 7-0: IOP7-0**, I/O Pipe Data Location Low
When IOPEN (ICR.4) is set, these are the locations that data is read from or written to in the I/O pipe operation. All 8-bit operations must take place through IOPL only.

### 5.1.4 IOPH - I/O Pipe Data Location High

Read/Write Port = 03

This register and IOPL determine the locations data is read from or written to.

| BIT | IOPH | RESET |
|---|---|---|
| 7 | IOP15 | 0 |
| 6 | IOP14 | 0 |
| 5 | IOP13 | 0 |
| 4 | IOP12 | 0 |
| 3 | IOP11 | 0 |
| 2 | IOP10 | 0 |
| 1 | IOP9 | 0 |
| 0 | IOP8 | 0 |

**Bits 15-8: IOP15-8**, I/O Pipe Data Location High
When IOPEN (ICR.4) is set, these are the locations that data is read from or written to in the I/O pipe operation. All 8-bit operations must take place through the IOPL register.

### 5.1.5 HWR - Hardware Support Register

Read/Write Port = 04

This register is used to control general purpose outputs and to switch between configuration and LAN Address registers.

| BIT | HWR READ | RESET |
|---|---|---|
| 7 | SWH | 0 |
| 6 | — | 0 |
| 5 | ETHR | 1 |
| 4 | HOST16 | 0 |
| 3 | — | 0 |
| 2 | ISTAT | 0 |
| 1 | PNPJMP | JMP6 |
| 0 | GPOE | 0 |

| BIT | HWR WRITE | RESET |
|---|---|---|
| 7 | SWH | 0 |
| 6 | — | 0 |
| 5 | — | 1 |
| 4 | — | 0 |
| 3 | NUKE | 0 |
| 2 | — | 0 |
| 1 | — | 0 |
| 0 | GPOE | 0 |

**Bit 7: SWH**, Switch Register Set
This bit selects between the LAN Address registers and the Board Configuration registers in the register map where:

SWH = 0 - LAN Address Registers are visible
SWH = 1 - Configuration Registers are visible

**Bit 5: ETHER**, MAC Protocol Type
When ETHER = 1, the 802.3 protocol is provided by this device. The 83C795 supports only 802.3.

**Bit 4: HOST16**
This bit reports whether the 83C795 believes it is connected to an 8-bit or 16-bit host. The chip determines this by looking at the MEMR pin for activity. Where:

HOST16 = 0 - 8-bit host
HOST16 = 1 - 16-bit host

**Bit 3: NUKE**, Restart
This bit is 'OR'ed together with the RESET pin signal to form the internal RESET for the chip. Setting this bit has the same effect on the 83C795 as cycling power on the host machine. The chip will reset to its initial condition and reload from the EEROM. This bit is cleared when the reset becomes effective. The NUKE reset executes for 256 chip-clock cycles to allow the MA bus to float and the initialization jumpers to achieve their true values.

**Note**

Do not try to access this chip during reset.

**Bit 2: ISTAT**, Interrupt Status
ISTAT returns to 1 when Network Interface Controller has an interrupt active.

**Bit 1: PNPJMP, Plug and Play Jumper Installed**
A read-only bit that returns a 1 if jumper 6 is installed. If PNPJMP = 1 and the PNPEN bit (ERFAL.0) is set, then Plug and Play hardware is enabled.

**Bit 0: GPOE**, GPX Pin Output Enable
The output is enabled when GPOE = 1.

### 5.1.6 BPR - BIOS Page Register

Read/Write Port = 05

This register controls mapping of ROM window to ROM address and other miscellaneous controls.

| BIT | BPR | RESET |
|---|---|---|
| 7 | M16EN | 0 |
| 6 | BP15 | 0 |
| 5 | BP14 | 0 |
| 4 | BP13 | 0 |
| 3 | — | 0 |
| 2 | — | 0 |
| 1 | SOFT1 | 0 |
| 0 | SOFT0 | 0 |

**Bit 7: M16EN**, Memory 16-bit Enable
Set M16EN to 1 to enable 16-bit memory access by the host. This should be only set when all interrupts have been disabled. See page 17 for details.

**Bits 6-4: BP15-13**, ROM Offset
ROM address is created by adding the contents of this field to the difference between ROM base address and the address supplied on the SA19-SA00 lines. This sum serves as a movable page offset into the ROM window. It is intended that the offset be used only when the ROM provided is larger than the window selected.

**Bits 1-0: SOFT1-SOFT0**
These bits are written and read by software. They may be used as 'claim' bits for allocation of drivers to multiple LAN connections within a common backplane.

### 5.1.7 ICR - Interrupt Control Register

Read/Write Port = 06

This register enables and masks interrupts. It is not used to select IRQ lines. That function is performed through the GCR Register.

| BIT | ICR | RESET |
|---|---|---|
| 7 | MCTEST | 0 |
| 6 | STAG | 0 |
| 5 | IOPAV | 0 |
| 4 | IOPEN | 0 |
| 3 | SINT | 0 |
| 2 | MASK2 | 0 |
| 1 | MASK1 | 0 |
| 0 | EIL | 0 |

**Bit 7: MCTEST**, Memory Cache Test Bit
The memory cache counters are accelerated when MCTEST=1. Use this bit only for test purposes.

**Bit 6: STAG**, Staggered Address Enable
When STAG = 1, the lowest bit in the buffer counter is forced to 1 on memory cache misses.

**Bit 5: IOPAV**, I/O Pipe Address Visible
When IOPAV = 1, it allows the I/O pipe's temporary address register to be read out of the REV Register.

**Bit 4: IOPEN**, I/O Pipe Enable
When IOPEN = 1, the I/O pipe is enabled. Regular memory accesses should be disabled when this bit is set.

**Bit 3: SINT,** Software Interrupt
Set SINT = 1 to create an interrupt under software control. Set to zero to remove interrupt. For more details, refer to page 52.

**Bit 2: MASK2**, Mask Interrupt Sources
Set MASK2 to 1 to mask out interrupt from the NIC.

**Bit 1: MASK1**, Not used

**Bit 0: EIL**, Enable Interrupts
Set to 1 to enable interrupts from this device. This enable controls SINT and interrupts from the LAN controller. For more details, refer to page 52.

### 5.1.8 REV/IOPA - Revision/I/O Pipe Address Register

Read/Write Port = 07

This register serves two functions:

1. It provides the host with revision information about the chip (CHIP3-0 and REV3-0).
2. It provides a port for loading the I/O pipe address into the buffer counter. For more on this, see Section 6.2.

The revision information is read-only and will be returned on reads to this location when the IOPAV bit (ICR.5) is zero. This information is detailed as follows:

| BIT | REV READ | RESET |
|---|---|---|
| 7 | CHIP3 | 0 |
| 6 | CHIP2 | 1 |
| 5 | CHIP1 | 0 |
| 4 | CHIP0 | 0 |
| 3 | REV3 | 0 |
| 2 | REV2 | 0 |
| 1 | REV1 | 0 |
| 0 | REV0 | 0 |

**Bits 7-4: CHIP3-CHIP0**, Chip Type
Depending on the condition of Jumper 9, this field yields either 0100 or 0010. A value of 0100 indicates to the host that this is an 83C795 device; a value of 0010 indicates an 83C790 device.

**Bits 3-0: REV3-REV0**, Revision Number
These bits initialize to the revision number of this chip.

When the I/O pipe is enabled – that is, when IOPEN is set (ICR.4) – the I/O pipe address is loaded into the buffer counter through this register. Since the buffer counter is 16-bits wide, two consecutive writes are required to accomplish this using this method.

- The first write, which contains the lower byte of the address, is stored in a temporary register.
- The second write, which contains the upper byte of the address, is then transferred, along with the contents of the temporary register, into the buffer counter.

Any host access to the chip between the first and second writes will automatically reset the process.

When IOPAV is set, the contents of the temporary register can be read from this location.

| BIT | IOPA | RESET |
|---|---|---|
| 7 | IOPA7 | 0 |
| 6 | IOPA6 | 0 |
| 5 | IOPA5 | 1 |
| 4 | IOPA4 | 0 |
| 3 | IOPA3 | 0 |
| 2 | IOPA2 | 0 |
| 1 | IOPA1 | 1 |
| 0 | IOPA0 | 0 |

**Bits 7-0: IOPA7-IOPA0**, I/O Pipe Address
This register provides the location of the I/O Pipe address.

### 5.1.9 LAN0 - LAN5 - LAN Address Registers

Read/Write Ports = 08 - 0D SWH = 0

These six LAN address registers (along with the BDID and CHKSUM registers) recall or store general-purpose data from the EEROM and, during normal use, recall the permanently-assigned LAN address for the adapter.

| REG | LN | RESET | INIT | RECALL |
|---|---|---|---|---|
| LAN0 | LN07-LN00 | 0 | EE | EE |
| LAN1 | LN15-LN08 | 0 | EE | EE |
| LAN2 | LN23-LN16 | 0 | EE | EE |
| LAN3 | LN31-LN24 | 0 | EE | EE |
| LAN4 | LN39-LN32 | 0 | EE | EE |
| LAN5 | LNMSB, LN46-LN40 | 0 | EE | EE |

**Bits 0-7: LN07-LN00**
In normal use, these are the least significant bits of the globally-assigned LAN address block.

**Bits 8-15: LN08-LN15**
In normal use, LN8-LN15 are part of the globally assigned LAN address block.

**Bits 16-23: LN16-LN23**
In normal use, LN16-LN23 are part of the globally assigned LAN address block.

**Bits 24-31: LN24-LN31**
In normal use, LN24-LN31 is part of the unique LAN address for each adapter (LN47-LN24) and may be assigned at the time of manufacture for the end product.

**Bits 32-39: LN32-LN39**
In normal use, LN32-LN39 is part of the unique LAN address for each adapter (LN47-LN24) and may be assigned at the time of manufacture for the end product.

**Bits 40-46: LN40-LN46**
LN40-LN46 forms part of the unique LAN address for each adapter (LN47-LN24) and may be assigned at the time of manufacture for the end product.

**Bit 47: LNMSB**
LNMSB is the most significant digit of the unique LAN address block which comprises LN24 through LN47.

### 5.1.10 BDID - Board ID Register

Read Port = OE SWH=0

This register is similar to the LAN registers except that it contains an 8-bit code identifying the board type for software purposes. The administration of this ID byte is beyond the scope of this specification. This register is stored and recalled along with the LAN registers.

| BIT | BDID | RESET | INIT | RECALL |
|---|---|---|---|---|
| 7 | BDID7 | 0 | EE | EE |
| 6 | BDID6 | 0 | EE | EE |
| 5 | BDID5 | 0 | EE | EE |
| 4 | BDID4 | 0 | EE | EE |
| 3 | BDID3 | 0 | EE | EE |
| 2 | BDID2 | 0 | EE | EE |
| 1 | BDID1 | 0 | EE | EE |
| 0 | BDID0 | 0 | EE | EE |

### 5.1.11 CKSM - Checksum Register

Read/Write Port = OF SWH=0

Before storing a LAN address, CKSM should be programmed with an 8-bit checksum which causes the 2's complement sum of all eight second-group register contents to be FFH. The sum must include this register. This register is stored and recalled along with the LAN registers. It is recommended that on recall of the LAN address, the register's integrity should be confirmed by computing (in software) the checksum of the second group of registers.

| BIT | CKSM | RESET | INIT | RECALL |
|---|---|---|---|---|
| 7 | CHK7 | 1 | EE | EE |
| 6 | CHK6 | 1 | EE | EE |
| 5 | CHK5 | 1 | EE | EE |
| 4 | CHK4 | 1 | EE | EE |
| 3 | CHK3 | 1 | EE | EE |
| 2 | CHK2 | 1 | EE | EE |
| 1 | CHK1 | 1 | EE | EE |
| 0 | CHK0 | 1 | EE | EE |

**Bits 7-0: CHK7-CHK0**, Checksum Register
The 83C795 stores the checksum amount in this register for reference and comparison with the LAN registers' amounts.

### 5.1.12 GCR2 - General Control Register 2

Read/Write Ports = 08 SWH=1

This register is used to hold general control information.

| BIT | GCR2 | RESET | INIT |
|---|---|---|---|
| 7 | — | 0 | 0 |
| 6 | — | 0 | 0 |
| 5 | — | 0 | 0 |
| 4 | — | 0 | 0 |
| 3 | — | 0 | 0 |
| 2 | — | 0 | 0 |
| 1 | — | 0 | 0 |
| 0 | PNPIOP | 0 | 0 |

**Bit 0: PNPIOP,** PNP and I/O Mapped Pipe
This bit is used to communicate to the Plug and Play logic that the adapter uses the I/O-mapped mode. When PNPIOP = 1, the Plug and Play RAM Control registers are disabled and the ROM Control registers are moved from 48h–4Ch to 40h–44h.

### 5.1.13 IAR - I/O Address Register

Read/Write Port = 0A SWH=1

This register programs the base I/O address for the chip.

| BIT | IAR | RESET | INIT |
|---|---|---|---|
| 7 | IA15 | 0 | 0 |
| 6 | IA14 | 0 | 0 |
| 5 | IA13 | 0 | 0 |
| 4 | IA8 | 0 | 0 |
| 3 | IA7 | 1 | 1 |
| 2 | IA6 | 0 | 0 |
| 1 | IA5 | 0 | 0 |
| 0 | PNPBOOT | 0 | 0 |

**Bits 7-5: IA15-IA13**, I/O Address Lines
These bits are compared against the A15-A13 lines from the host when IOR or IOW are active and AEN is not. To access the chip, the lines must match.

**Bits 4-1: IA8-IA5**, I/O Address Lines
These bits are compared against the A8-5 lines from the host when IOR or IOW are active and AEN is not. To access the chip, the lines must match.

**Bit 0: PNPBOOT,** Plug and Play Boot Bit
PNPBOOT = 1 to indicate to the Plug and Play hardware that the adapter is a boot card. This allows the 83C795 address decoders to be active without waiting for the Plug and Play hardware activate command.

### 5.1.14 RAR - RAM Address Register

Read/Write Port = 0B SWH=1

This register controls the base address and window size for the buffer RAM.

| BIT | RAR | RESET | INIT |
|---|---|---|---|
| 7 | HRAM | 0 | 0 |
| 6 | RA17 | 0 | 0 |
| 5 | RAMSZ1 | 0 | 0 |
| 4 | RAMSZ0 | 0 | 0 |
| 3 | RA16 | 0 | 0 |
| 2 | RA15 | 1 | 1 |
| 1 | RA14 | 0 | 0 |
| 0 | RA13 | 0 | 0 |

**Bit 7: HRAM**, High RAM Address
This bit provides a means of locating the buffer memory above the 1MB DOS limit. When HRAM = 0, the buffer address decoder matches LA23-LA20 against zero. When HRAM = 1, the LA23-LA20 lines are matched against the value F. This field is not supported by the Plug and Play hardware.

**Bits 5-4: RAMSZ1-RAMSZ0**, Buffer Window Size Field
This encoded field determines the apparent size of the buffer RAM. It is decoded in the following manner:

| SZ1 | SZ0 | Window Size |
|---|---|---|
| 0 | 0 | 8K Bytes |
| 0 | 1 | 16K Bytes |
| 1 | 0 | 32K Bytes |
| 1 | 1 | 64K Bytes |

**TABLE 5-5. BUFFER WINDOW SIZE FIELD**

**Bits 6, 3-0: RA17, RA16-RA13**, RAM Base Address Field
These bits form part of the base address for the buffer RAM decoder along with the fixed value of '11' for RA19-RA18. When SA19-SA13 has a value between this base address and the base plus the window size, the request for memory is recognized. Once the SA19-SA13 value is no longer in this range, the host ends the access.

**Note**

The 64K window size is not supported by Plug and Play.

### 5.1.15 BIO - ROM Control Register

Read/Write Port = 0C SWH = 1

This register programs the base address and window size for the external ROM.

| BIT | BIO | RESET | INIT | RECALL |
|---|---|---|---|---|
| 7 | FINE16 | 0 | 0 | EE |
| 6 | BA17 | 0 | 0 | EE |
| 5 | BIOSZ1 | 1 | 1 | EE |
| 4 | BIOSZ0 | 1 | 1 | EE |
| 3 | BA16 | 0 | 0 | EE |
| 2 | BA15 | 0 | 0 | EE |
| 1 | BA14 | 0 | 0 | EE |
| 0 | BA13 | 0 | 0 | EE |

**Bit 7: FINE16**, Fine Decode
When FINE16 = 1, MEM16CS's response is generated only when the actual RAM window is being addressed. It includes SA16-SA13 in the address decoding process. See page 51 for more details.

**Bits 5-4: BIOSZ1-BIOSZ0**, ROM Window Size Field
These two bits determine the ROM window size and are decoded in this manner:

| SZ1 | SZ0 | ROM Window Size |
|---|---|---|
| 0 | 0 | 8K |
| 0 | 1 | 16K |
| 1 | 0 | 32K |
| 1 | 1 | Disabled |

**TABLE 5-6. ROM WINDOW SIZE FIELD**

**Bits 6, 3-0: BA17, BA16-BA13**, Base Address Field
These bits form part of the base address for the ROM decoder along with the fixed value of '11' for BA19-BA18. When SA19-SA13 has a value between this base address and the base plus the window size and SMEMR are active, a request for the ROM is recognized. A chip select will be generated to the ROM if not disabled. Memory access at the same address is blocked if ROM is enabled.

### 5.1.16 GCR - General Control Register

Read/Write Port = OD SWH = 1

This register controls interrupt level selection, zero wait state response, and several other functions.

| BIT | GCR | RESET | INIT | RECALL |
|---|---|---|---|---|
| 7 | XLENGTH | 0 | 0 | EE |
| 6 | IR2 | 0 | 0 | EE |
| 5 | ZWSEN | 0 | 0 | EE |
| 4 | RIPL | 0 | 0 | EE |
| 3 | IR1 | 0 | 0 | EE |
| 2 | IR0 | 0 | 0 | EE |
| 1 | GPOUT | 0 | 0 | EE |
| 0 | LIT | 0 | 0 | EE |

**Bit 7: XLENGTH**, Extended Length Bit Enable
When XLENGTH = 1, the extended length option is enabled as specified by the 802.3 specification (refer to page 69).

**Bit 5: ZWSEN**, Zero Wait State Enable
This bit is set to 1 to enable the chip to generate a ZWS response when the RAM is accessed and available to the host.

**Bit 4: RIPL**, Software Flag

**Bits 6, 3-2: IR2-IR0**, Interrupt Request Field
These bits form an encoded field to select through which IRQ pin the interrupt output is channeled. Because of Plug and Play logic, it is necessary to connect the interrupt pins to specific lines on the ISA bus, as shown below. The interrupt request pins and their corresponding ISA lines are decoded in this fashion:

| IRQ2 | IRQ1 | IRQ0 | IRQ Pin Selected | ISA Bus Line |
|---|---|---|---|---|
| 0 | 0 | 0 | None | None |
| 0 | 0 | 1 | IRQ1 | IRQ2/9 |
| 0 | 1 | 0 | IRQ2 | IRQ3 |
| 0 | 1 | 1 | IRQ3 | IRQ5 |
| 1 | 0 | 0 | IRQ4 | IRQ7 |
| 1 | 0 | 1 | IRQ5 | IRQ10 |
| 1 | 1 | 0 | IRQ6 | IRQ11 |
| 1 | 1 | 1 | IRQ7 | IRQ15 |

**TABLE 5-7. INTERRUPT REQUEST FIELD**

**Bit 1: GPOUT**, General Purpose Output
This bit controls the GPOUT pin of the chip. When GPOUT = 1, it causes the GPOUT pin to drive low. In some systems, this bit is wired to a shutdown control input for DC/DC isolated power supply used in 10Base2 applications. For more information on this feature, refer to Section 6.7.

**Bit 0: LIT**, Link Integrity Test
This bit controls the Link Integrity Test. In STARLAN-10 networks, the Link Integrity Test should be disabled.

LIT = 0 - Link Test is Disabled
LIT = 1 - Link Test is Enabled.

Disabling the Link Integrity Test (LIT) forces the 83C795 to select the twisted-pair interface. When LIT is enabled, the twisted-pair interface will be automatically selected when link activity is found and the AUI interface will be selected when the twisted-pair link enters the 10BASE-T link fail state.

### 5.1.17 ERFAL - Early Receive Fail Address Low Register

Read/Write Port = OE SWH = 1

This register contains the lower eight bits for the address at which the early-receive logic detected an underrun. This register also contains a control bit for the Plug and Play logic.

| BIT | ERFAL | RESET | RECALL |
|---|---|---|---|
| 7 | ERFA7 | — | — |
| 6 | ERFA6 | — | — |
| 5 | ERFA5 | — | — |
| 4 | ERFA4 | — | — |
| 3 | ERFA3 | — | — |
| 2 | ERFA2 | — | — |
| 1 | — | 0 | 0 |
| 0 | PNPEN | 1 | EE |

**Bits 7-2: ERFA7-2**, Early Receive Failure Address
This register contains the lower eight bits of the address where the early receive logic detected an underrun. The comparison has a granularity of 4 bytes so the least significant two bits are zero. This value is read-only.

**Bit 0: PNPEN**, Plug and Play Enable
When PNPEN = 1 along with the installation of JUMPER6, Plug and Play logic is enabled. This bit is readable but can only be set by the initial EEROM load.

### 5.1.18 ERFAH - Early Receive Fail Address High Register

Read/Write Port = OF SWH = 1

This register contains the higher eight bits for the address at which the early receive logic detected an underrun.

| BIT | ERFAH | RESET | RECALL |
|---|---|---|---|
| 7 | ERFA15 | — | — |
| 6 | ERFA14 | — | — |
| 5 | ERFA13 | — | — |
| 4 | ERFA12 | — | — |
| 3 | ERFA11 | — | — |
| 2 | ERFA10 | — | — |
| 1 | ERFA09 | — | — |
| 0 | ERFA08 | — | — |

**Bits 7-0: ERFA15-8**, Early Receive Failure Address
This register contains the higher eight bits of the address where the early receive logic detected an underrun.

## 5.2 LAN CONTROLLER REGISTER DESCRIPTIONS

To simplify the programming model for the LAN controller and retain compatibility with the SMC 83C690 LAN Controller, the internal registers are divided into two address maps. The default address map is used for Ring-style buffering (like the 83C690). Those registers needed for linked-list buffering are grouped together in the alternate address map and are enabled through the Enhancement (ENH) register described starting on page 26.

Each map provides access to all registers necessary for operating that particular buffering mode. Many registers are visible in both maps, although not always at the same address in each. To facilitate manufacturing test of the device, many internal registers can be accessed in one or both of these maps. Within each map, the registers are organized into 4 pages of 16 registers each. Only one page is visible at a time. Page selection is made through the Command (CMD) register described starting on page 24.

The addresses listed in this specification are in an abbreviated form. The first hex digit is really a two-bit 'page' value which is written into the LAN COMMAND (CMD) register to access the 16 registers visible for that page. The digits after the colon are the offset within the 83C795's LAN Controller I/O segment in this manner:

*page*:*offset*

To determine the correct address, you must first know the 83C795's base address then select the correct page and finally select the correct offset. So, for example, "3:1C" indicates that the address for this particular register is found on page 3 at the offset value 1C.

In the following descriptions, the most significant bit position is numbered '7'. The line labelled RESET shows the initial values loaded into the register by assertion of the RESET pin. The symbol '0' denotes void bits which always return zero when read.

### 5.2.1 ALICNT - Alignment Error Counter Register

Normal Map Read Port = 0:1D Link-List Map Read Port = 0:1D

This register is the alignment error counter. It is incremented by the receive unit when a packet is received with a frame alignment error. Only packets whose addresses are recognized will be included in this tally. The counter will increment to 255 and stop if additional alignment errors are detected. The counter is cleared when read.

| BIT | ALICNT | RESET |
|---|---|---|
| 7 | CT7 | 0 |
| 6 | CT6 | 0 |
| 5 | CT5 | 0 |
| 4 | CT4 | 0 |
| 3 | CT3 | 0 |
| 2 | CT2 | 0 |
| 1 | CT1 | 0 |
| 0 | CT0 | 0 |

### 5.2.2 BOUND - Receive Boundary Page Register

Normal Map Read/Write Port = 0:13

The Receive Boundary Page Register points to the oldest used receive buffer in the ring. It is used to prevent overflow in the buffer ring. The DMA compares the contents of this register to the next buffer address when linking buffers together for storage of a received frame. If the contents match the next buffer address, the DMA operation is aborted. Only A08-A15 are specified since all buffers are aligned on 256-byte boundaries. For more information, refer to page 85.

| BIT | BOUND | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.3 CMD - Command Register

Normal Map Read/Write Port = X:10
Linked-List Map Read/Write Port = X:10

| BIT | CMD | RESET |
|---|---|---|
| 7 | PS1 | 0 |
| 6 | PS0 | 0 |
| 5 | RFU | 0 |
| 4 | ENETCH | 0 |
| 3 | DISETCH | 0 |
| 2 | TXP | 0 |
| 1 | STA | 0 |
| 0 | STP | 1 |

The Command register is used to initialize the 83C795 chip, start transmissions, and switch pages.

**Bits 7-6: PS1-PS0**, Page Select
This 2-bit field designates which of 4 pages is showing. They decode as follows:

| PS1 | PS0 | Page Select |
|---|---|---|
| 0 | 0 | Page 1 |
| 1 | 0 | Page 2 |
| 0 | 1 | Page 3 |
| 1 | 1 | Page 4 |

**TABLE 5-8. PAGE SELECT FIELD**

**Bit 5: RFU,** Reserved for Future Use
This bit is not used by 83C795 and always returns zero when read.

**Bit 4: ENETCH**, Enable Early Transmit Checking
By setting this bit to 1, it enables comparison of transmit DMA address against the host memory write address. Once set, this bit can be cleared and the 83C795 continues to check transmission addresses until DISETCH is set. See page 78 for more details.

**Bit 3: DISETCH**, Disable Early Transmit Checking
By setting this bit to 1, it disables the early transmit address checking. Once set, the bit can be cleared and transmit address checking is suppressed until ENETCH is set. See page 78 for more details.

**Bit 2: TXP**, Transmit packet
Set this bit after loading the Transmit Buffer and Control registers to initiate transmission of a packet. The 83C795 clears this bit upon completion or abortion of the transmission.

**Bit 1: STA**, Start Bit
Set the STA bit to activate the 83C795 after power up or when the 83C795 has been reset by a software command. No frames can be sent or received until this bit has been set. The user's software should set up the other registers prior to bringing the device on line, but setting this bit is the actual command which brings the Transmit and Receive portions of the device online. Once set, this bit may be cleared and the 83C795 will continue to remain online.

**Bit 0: STP**, Stop Bit
Set the STP bit to take the chip offline and disengage from the LAN. Frames partially transmitted or received are completed before reset occurs. INTSTAT.RST is set high when the Transmit and Receive section have completed all outstanding operations (see page 28). No frames will be received or transmitted until the start bit has been set.

### 5.2.4 COLCNT - Collision Count Register

Normal Map Read Port = 0:15
Linked-List Map Read Port = 0:15

This register contains the number of collisions detected while attempting to transmit the current (or most recent) packet. It is cleared to zero at the start of transmission. If no collisions are detected, the counter will read zero. For each collision encountered, the count is incremented. If more than 15 collisions occur, the abort bit of TSTAT is set and the count is reset to zero (see page 38).

| BIT | COLCNT | RESET |
|---|---|---|
| 7 | T10 | 0 |
| 6 | T9 | 0 |
| 5 | T8 | 0 |
| 4 | T7 | 0 |
| 3 | CT3 | 0 |
| 2 | CT2 | 0 |
| 1 | CT1 | 0 |
| 0 | CT0 | 0 |

**Bits 7-4: T10-T7**, Backoff Counter
These 4 consecutive bits always return zero.

**Bits 3-0: CT3-CT0**, Collision Counter
These bits indicate the value of the collision counter. They are always readable.

### 5.2.5 CRCCNT - CRC Error Counter

Normal Map Read Port = 0:1E
Linked-List Map Read Port = 0:1E

This register is incremented by the receive unit when a packet is received with a CRC error. Only packets whose address is recognized will be included in this tally. When a 'runt' frame is received with a CRC error, CRCCNT is incremented if RCON.RUNTS is enabled (see page 32). The counter will increment to 255 and stick if additional CRC errors are detected. The counter is cleared when read.

| BIT | CRCCNT | RESET |
|---|---|---|
| 7 | CT7 | 0 |
| 6 | CT6 | 0 |
| 5 | CT5 | 0 |
| 4 | CT4 | 0 |
| 3 | CT3 | 0 |
| 2 | CT2 | 0 |
| 1 | CT1 | 0 |
| 0 | CT0 | 0 |

### 5.2.6 CURR - Current Frame Buffer Pointer Register

Normal Map Read/Write Port = 1:17

This register points to the first buffer used for storage of the present frame. It is used by DMA as a backup address for recovering buffers in case of a flawed packet and facilitates storage of buffer header information. The CURR register should be initialized to the same value as RSTART (see page 33) and not altered thereafter by the user unless the controller is reset. Only A08-A15 are specified since all buffers are aligned on 256-byte boundaries.

| BIT | CURR | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.7 CURRH - Current Frame Buffer Descriptor Pointer Register High

Linked-List Map Read/Write Port = 1:17

This register is one of a pair of registers (CURRH and CURRL) that point to the first buffer descriptor used for storage of the present frame. They are used by DMA as a backup address for recovering buffers in the case of a flawed packet and to facilitate storage of buffer header information. Neither the CURRH nor CURRL registers should be altered by the user. They are accessible for test purposes only.

| BIT | CURRH | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.8 CURRL - Current Frame Buffer Descriptor Pointer Register Low

Linked-List Map Read/Write Port = 0:13

This register is one of a pair of registers (CURRH and CURRL) that point to the first buffer descriptor used for storage of the present frame. They are used by DMA as a backup address for recovering buffers in the case of a flawed packet and to facilitate storage of buffer header information. Neither the CURRH nor CURRL registers should be altered by the user. They are accessible for test purposes only.

| BIT | CURRL | RESET |
|---|---|---|
| 7 | A07 | X |
| 6 | A06 | X |
| 5 | A05 | X |
| 4 | A04 | X |
| 3 | A03 | X |
| 2 | A02 | X |
| 1 | A01 | X |
| 0 | A00 | X |

### 5.2.9 DCON - Data Configuration Register

Linked-List Map Read Port = 2:1E Linked-List Map Write Port = 0:1E

This register always returns 41h. In the 83C790 this register controlled DMA burst lengths; however, the 83C795 is hardwired for 8-byte bursts. Refer to page 65 for more information.

### 5.2.10 ENH - Enhancement Register

Normal Map Read/Write Port = 2:17
Linked-List Map Read/Write Port = 2:17

This register enables enhancement features.

| BIT | ENH | RESET |
|---|---|---|
| 7 | — | 0 |
| 6 | — | 1 |
| 5 | ALTEGO | 0 |
| 4 | SLOT1 | 0 |
| 3 | SLOT0 | 0 |
| 2 | EOTINT | 0 |
| 1 | — | 0 |
| 0 | SBACK | 0 |

**Bits 7-6: Unused**

**Bit 5: ALTEGO**, Buffering Format Selection

ALTEGO = 0 -
Designates ring buffering and single frame transmission format. This is essentially 8390/83C690 compatibility mode.

ALTEGO = 1 -
Designates linked-list receive buffering and multiple frame transmission format. The register address map is selected with this bit, exposing the registers associated with the selected buffering mode.

**Bits 4-3: SLOT1-0**, Slot Time Selection
This two-bit field selects the slot time according to Table 5-10.

| SLOT1 | SLOT0 | Slot Time |
|---|---|---|
| 0 | X | 512 bit times (Ethernet) |
| 1 | 0 | 256 bit times |
| 1 | 1 | 1024 bit times |

**TABLE 5-9. SLOT TIME SELECTION FIELD**

**Bit 2: EOTINT**, Interrupt on End-of-Transmit

EOTINT = 1 -
Interrupt on End-of-Transmit chain instead of each transmitted frame. This bit is ignored if not operating in multiple frame transmission mode.

EOTINT = 0 -
Interrupt on each transmitted frame.

**Bit 0: SBACK**, Enable Stop Backup Modifications

SBACK = 1 -
Enable the Stop Backoff modifications to the backoff timer.

SBACK = 0 -
Normal backoff.

### 5.2.11 ERWCNT - Early Receive Warning Threshold Register

Normal Map Read/Write Port = 0:18
Linked-List Map Read/Write Port = 0:18

This register contains the Received Byte Count threshold at which the Early Receive Warning interrupt is generated. The ERW interrupt is generated when RBC ≥ ERW. Bits 3-0 of RBC are ignored. For more information on this register, refer to page 71.

| BIT | ERWCNT | RESET |
|---|---|---|
| 7 | ERW11 | 0 |
| 6 | ERW10 | 0 |
| 5 | ERW9 | 0 |
| 4 | ERW8 | 0 |
| 3 | ERW7 | 0 |
| 2 | ERW6 | 0 |
| 1 | ERW5 | 0 |
| 0 | ERW4 | 0 |

### 5.2.12 GROUP0-GROUP7 - Multicast Filter Table Registers

| GROUP Register | Normal Map Port Address | | Linked-List Map Port Address | |
|---|---|---|---|---|
| | Read | Write | Read | Write |
| GROUP0 | 1:18 | 1:18 | 1:18 | 1:18 |
| GROUP1 | 1:19 | 1:19 | 1:19 | 1:19 |
| GROUP2 | 1:1A | 1:1A | 1:1A | 1:1A |
| GROUP3 | 1:1B | 1:1B | 1:1B | 1:1B |
| GROUP4 | 1:1C | 1:1C | 1:1C | 1:1C |
| GROUP5 | 1:1D | 1:1D | 1:1D | 1:1D |
| GROUP6 | 1:1E | 1:1E | 1:1E | 1:1E |
| GROUP7 | 1:1F | 1:1F | 1:1F | 1:1F |

These 8 registers hold the node's Multicast filter table. See Table 5-10 for the registers' bit assignments.

### 5.2.13 INTMASK - Interrupt Mask Register

Normal Map Read Port = 2:1F Normal Map Write Port = 0:1F
Linked-List Map Read Port = 2:1F Linked-List Write Port = 0:1F

The Interrupt Mask Register is used to mask out certain interrupt sources selectively. Mask bits set to '1' allow the corresponding interrupts to cause an IRQ. Mask bits set to '0' block their respective interrupt sources.

| BIT | INTMASK | RESET |
|---|---|---|
| 7 | — | 0 |
| 6 | ERWE | 0 |
| 5 | CNTE | 0 |
| 4 | OVWE | 0 |
| 3 | TXEE | 0 |
| 2 | RXEE | 0 |
| 1 | PTXE | 0 |
| 0 | PRXE | 0 |

**Bit 6: ERWE**, Early Receive Warning Enable
When ERWE = 1, this bit enables Early Receive Warning as defined by the ERW bit in the Interrupt Status Register. (See the next register, INTSTAT.)

**Bit 5: CNTE**, Counter Overflow Enable
When CNTE = 1, this bit enables Counter Overflow as defined by the CNT bit in the Interrupt Status Register. (See the next register, INTSTAT.)

**Bit 4: OVWE**, Overwrite Warning Enable
When OVWE = 1, this bit enables Overwrite Warning as defined by the OVW bit in the Interrupt Status Register. (See the next register, INTSTAT.)

| GROUP Registers | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|
| GROUP0 | GA07 | GA06 | GA05 | GA04 | GA03 | GA02 | GA01 | GA00 |
| GROUP1 | GA15 | GA14 | GA13 | GA12 | GA11 | GA10 | GA09 | GA08 |
| GROUP2 | GA23 | GA22 | GA21 | GA20 | GA19 | GA18 | GA17 | GA16 |
| GROUP3 | GA31 | GA30 | GA29 | GA28 | GA27 | GA26 | GA25 | GA24 |
| GROUP4 | GA39 | GA38 | GA37 | GA36 | GA35 | GA34 | GA33 | GA32 |
| GROUP5 | GA47 | GA46 | GA45 | GA44 | GA43 | GA42 | GA41 | GA40 |
| GROUP6 | GA55 | GA54 | GA53 | GA52 | GA51 | GA50 | GA49 | GA48 |
| GROUP7 | GA63 | GA62 | GA61 | GA60 | GA59 | GA58 | GA57 | GA56 |

**TABLE 5-10. GROUP REGISTER BITS**

**Bit 3: TXEE**, Transmit Error Enable
When TXEE = 1, this bit enables Transmit Error as defined by the TXE bit in the Interrupt Status Register. (See the next register, INTSTAT.)

**Bit 2: RXEE**, Receive Error Enable
When RXEE = 1, this bit enables Receive Error as defined by the RXE bit in the Interrupt Status Register. (See the next register, INTSTAT.)

**Bit 1: PTXE**, Packet Transmitted Enable
When TXEE = 1, this bit enables Packet Transmitted as defined by the PTX bit in the Interrupt Status Register. (See the next register, INTSTAT.)

**Bit 0: PRXE**, Packet Received Enable
When PRXE = 1, this bit enables Packet Received as defined by the PRXE bit in the Interrupt Status Register. (See the next register, INTSTAT.)

### 5.2.14 INTSTAT - Interrupt Status Register

Normal Map Read/Write Port = 0:17
Linked-List Map Read/Write Port = 0:17

The Interrupt Status Register enables the host to determine the cause of an interrupt and to evaluate pending or masked interrupts. Masked-out interrupts are visible in this register although they will not generate an IRQ to the host. Pending interrupts can be cleared by writing '1' to the associated bit of this register. The IRQ signal is active as long as any unmasked interrupt bit remains set. For more details, see page 80.

| BIT | INTSTAT | RESET |
|---|---|---|
| 7 | RST | 1 |
| 6 | ERW | 0 |
| 5 | CNT | 0 |
| 4 | OVW | 0 |
| 3 | TXE | 0 |
| 2 | RXE | 0 |
| 1 | PTX | 0 |
| 0 | PRX | 0 |

**Bit 7: RST**, Reset Status
This bit is set by 83C795 when its Transmit and Receive sections are stopped in response to the assertion of the RESET pin or the setting of the CMD.STP bit. The RST bit does not generate an interrupt.

**Bit 6: ERW**, Early Receive Warning
When this bit is set it indicates that the number of bytes received in the current frame has exceeded the programmable limit of the ERWCNT register.

**Bit 5: CNT**, Counter Overflow
When this bit is set it indicates that the MSB of one or more network error counters has been set.

**Bit 4: OVW**, Overwrite Warning
This bit is set when the receive DMA must abort frame reception due to a lack of receive buffers.

**Bit 3: TXE**, Transmit Error
This bit is set when excessive collisions, out-of-window collisions, FIFO underrun, or early transmit address violations prevent transmission of a packet.

**Bit 2: RXE**, Receive Error
This bit is set when a packet is received with one or more of the following errors:

- CRC error (happens when SEP is enabled)
- Frame alignment error (happens when SEP is enabled)
- FIFO overrun
- Missed packet (monitor mode)

This interrupt will not be posted if a DMA Abort occurs, a condition indicated by the assertion of an OVW interrupt. If RXE is previously set, it will not be changed due to OVW.

**Bit 1: PTX**, Packet Transmitted
This bit is set when a packet is transmitted successfully. When the bit ENH.EOTINT is set in Multiple Packet Transmit mode (see page 5-26), setting of this interrupt is deferred until the entire transmit chain has been processed. PTX is then set if any packet in the chain was transmitted successfully, or if a zero length transmit chain was processed.

**Bit 0: PRX**, Packet Received
When PRX = 1, it indicates that a packet was received with no errors.

### 5.2.15 MANCH - Manchester Management Register

Normal Map Read/Write Port = 3:1F
Linked-List Map Read/Write Port = 3:1F

This register allows the reading back of the 10BaseT status LED drivers to support network and station management functions. It also enables and controls the 83C795's internal Manchester encoder/decoder.

| BIT | MANCH | RESET |
|---|---|---|
| 7 | MANDIS | 0 |
| 6 | SEL | 1 |
| 5 | 0 | 0 |
| 4 | ENAPOL | 1 |
| 3 | PLED | 0 |
| 2 | LLED | 0 |
| 1 | RLED | 0 |
| 0 | TLED | 0 |

**Bit 7: MANDIS**, Manchester Disable
MANDIS = 1 - disables the internal Manchester Encoder/Decoder. When disabled, the LAN Controller uses the decoder serial interface consisting of these lines: XTXD, XTXE, XTXC, XRXD, XRXC, XCRS, XCOL, XLOOP.

MANDIS = 0 - Enables the Manchester Encoder/Decoder.

**Bit 6: SEL**, Select AUI Mode For Idle State
SEL = 0 - TX+ is positive in relation to TX-.

SEL = 1 - TX+ = TX-.

**Bit 4: ENAPOL**, Automatic Polarity Correct
ENAPOL = 1 - Enable Auto Polarity Correct

ENAPOL = 0 - Disable Auto Polarity Correct

**Bit 3: PLED**, TPRX Polarity LED Readback
When PLED = 1, this LED is on.

**Note**

There is no $\overline{\text{PLED}}$ pin onto which this signal might be driven.

**Bit 2: LLED**, Link Status LED Readback
When LLED = 1, this LED is on (output sinking current).

**Bit 1: RLED**, Receive LED Readback
When RLED = 1, this LED is on (output sinking current).

**Bit 0: TLED**, Transmit LED Readback
When TLED = 1, this LED is on (output sinking current).

### 5.2.16 MPCNT - Missed Packet Error Counter Register

Normal Map Read Port = 0:1F Linked-List Map Read Port = 0:1F

This register is incremented by the receive unit whenever it cannot receive a packet due to a lack of receive buffers, receive FIFO overflow, or because the receiver is in monitor mode. Only packets whose address is recognized will be included in this tally. The counter will increment to 255 and stick if additional packets are missed. The counter is cleared when read.

| BIT | MPCNT | RESET |
|---|---|---|
| 7 | CT7 | 0 |
| 6 | CT6 | 0 |
| 5 | CT5 | 0 |
| 4 | CT4 | 0 |
| 3 | CT3 | 0 |
| 2 | CT2 | 0 |
| 1 | CT1 | 0 |
| 0 | CT0 | 0 |

### 5.2.17 NEXT - DMA Controller Next Buffer Register

Normal Map Read/Write Port = 2:15

This is a working register of the DMA controller. It holds a pointer to the next buffer to be opened.

| BIT | NEXT | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.18 RADDH - Receive Burst Starting Address High Register

Normal Map Read/Write Port = 2:19

This is the higher 8 bits of a register pair used internally by the DMA controller as a scratch pad for the burst address of the receive process. Writing to the RADDH and RADDL registers while communication is taking place may cause errors in the DMA process.

| BIT | RADDH | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.19 RADDL - Receive Burst Starting Address Low Register

Normal Map Read/Write Port = 2:18

This is the lower 8 bits of a register pair used internally by the DMA controller as a scratch pad for the burst address of the receive process. Writing to the RADDH and RADDL registers while communication is taking place may cause errors in the DMA process.

| BIT | RADDL | RESET |
|---|---|---|
| 7 | A07 | X |
| 6 | A06 | X |
| 5 | A05 | X |
| 4 | A04 | X |
| 3 | A03 | X |
| 2 | A02 | X |
| 1 | A01 | X |
| 0 | A00 | X |

### 5.2.20 RBEGIN - Receive Buffer Starting Address Register

Linked-List Map Read Port = 2:11 Linked-List Map Write Port = 0:11

This register holds the upper 8 bits of the starting address of the receive buffer descriptor table. The lower 8 bits are assumed to be zero.

| BIT | RBEGIN | RESET |
|---|---|---|
| 7 | RB15 | X |
| 6 | RB14 | X |
| 5 | RB13 | X |
| 4 | RB12 | X |
| 3 | RB11 | X |
| 2 | RB10 | X |
| 1 | RB09 | X |
| 0 | RB08 | X |

### 5.2.21 RCNTH - Receive Byte Count High Register

Normal Map Read Port = 0:1B

This register contains the upper 8 bits of the receive unit's count of bytes received in the most recent frame. It is cleared by the receive unit at the start of reception.

| BIT | RCNTH | RESET |
|---|---|---|
| 7 | CT15 | 0 |
| 6 | CT14 | 0 |
| 5 | CT13 | 0 |
| 4 | CT12 | 0 |
| 3 | CT11 | 0 |
| 2 | CT10 | 0 |
| 1 | CT09 | 0 |
| 0 | CT08 | 0 |

### 5.2.22 RCNTL - Receive Byte Count Low Register

Normal Map Read Port = 0:1A

This register contains the lower 8 bits of the receive unit's count of bytes received in the most recent frame. It is cleared by the receive unit at the start of reception.

| BIT | RCNTL | RESET |
|---|---|---|
| 7 | CT07 | 0 |
| 6 | CT06 | 0 |
| 5 | CT05 | 0 |
| 4 | CT04 | 0 |
| 3 | CT03 | 0 |
| 2 | CT02 | 0 |
| 1 | CT01 | 0 |
| 0 | CT00 | 0 |

### 5.2.23 RCON - Receive Configuration Register

Normal Map Read Port = 2:1C Normal Map Write Port = 0:1C
Linked-List Map Read Port = 2:1C Linked-List Map Write Port = 0:1C

The Receive Configuration Register defines optional behavior of the receive unit. It controls address recognition and the acceptance of abnormal packets. These bits can be set independently, although the monitor mode takes precedence over the other bits.

| BIT | RCON | RESET |
|---|---|---|
| 7 | 0 | 0 |
| 6 | RCA | 0 |
| 5 | MON | 0 |
| 4 | PROM | 0 |
| 3 | GROUP | 0 |
| 2 | BROAD | 0 |
| 1 | RUNTS | 0 |
| 0 | SEP | 0 |

**Bit 7: Unused**
This bit is unused in 83C795. When read, it always returns zero.

**Bit 6: RCA**, Receive Abort Frame on Collision
Setting this bit allows the receiver unit to abort reception of any frame in which the COL pin is active after the start of frame delimiter. Reception of any frame whose prefix contains consecutive '0' bits is also aborted. Neither cause results in RXE being set. It is not overridden by the SEP bit. This bit was unused in the 83C690.

**Bit 5: MON**, Check Addresses/CRC Without Buffering
MON = 1 - This bit enables the receive unit to check addresses and CRC on incoming packets without buffering them to memory. The Missed Packet Counter (MPCNT) will be incremented for each recognized packet.

MON = 0 - This is normal operation.

**Bit 4: PROM**, Promiscuous Reception
When PROM = 1, this bit enables promiscuous reception of all frames having individual addresses.

**Bit 3: GROUP**, Receive Multicast Frames
When GROUP = 1, this bit enables reception of all frames that:

- have multicast addresses
- pass the multicast address hashing filter

**Bit 2: BROAD**, Receive Broadcast Frames
When BROAD = 1, this bit enables reception of all frames having a Broadcast (all '1's) destination address.

**Bit 1: RUNTS**, Receive Runts Frames
When RUNTS = 1, this bit allows reception of frames having less than 64 bytes, provided that they otherwise meet the requirements of the 802.3 protocol.

**Bit 0: SEP**, Save Errored Packets
When SEP = 1, it directs the receive unit to save packets having CRC or frame alignment errors in the buffers.

### 5.2.24 RDOWNH - Buffer Room Remaining High Register

Linked-List Map Read/Write Port = 2:19

This register contains the upper 8 bits of a register pair used by the DMA controller as a scratch pad for the buffer room remaining count during the reception process.

**Note**

Writing to these registers while communication is taking place may cause errors in the DMA process.

| BIT | RDOWNH | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.25 RDOWNL - Buffer Room Remaining Low Register

Linked-List Map Read/Write Port = 2:18

This register contains the lower 8 bits of a register pair used by the DMA controller as a scratch pad for the buffer room remaining count during the reception process.

**Note**

Writing to these registers while communication is taking place may cause errors in the DMA process.

| BIT | RDOWNL | RESET |
|---|---|---|
| 7 | A07 | X |
| 6 | A06 | X |
| 5 | A05 | X |
| 4 | A04 | X |
| 3 | A03 | X |
| 2 | A02 | X |
| 1 | A01 | X |
| 0 | A00 | X |

### 5.2.26 REND - Receive Buffer End Register

Linked-List Map Read Port = 2:12 Linked-List Map Write Port = 0:12

This register holds the upper 8 bits of the first address beyond the end of the receive buffer descriptor table. The lower 8 bits are assumed to be zero. The table lies between the numbers (RBEGIN * 256) and (REND * 256 - 1). Refer to page 88 for more details.

| BIT | REND | RESET |
|---|---|---|
| 7 | RE15 | X |
| 6 | RE14 | X |
| 5 | RE13 | X |
| 4 | RE12 | X |
| 3 | RE11 | X |
| 2 | RE10 | X |
| 1 | RE9 | X |
| 0 | RE8 | X |

### 5.2.27 RSTART - Receive Start Page Register

Normal Map Read Port = 2:11 Normal Map Write Port = 0:11

Receive Start Page register points to the start of the receive buffer ring. Only A08-A15 are specified since all buffers are aligned on 256-byte boundaries. Refer to page 88 for more information.

| BIT | RSTART | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.28 RSTAT - Receive Packet Status Register

Normal Map Read Port = 0:1C Linked-List Map Read Port = 0:1C

This register reports the status of the most-recently received packet. It categorizes any errors that were detected and reports on the type of address recognized. All bits are cleared at the start of reception except for DIS.

| BIT | RSTAT | RESET |
|---|---|---|
| 7 | DRF | 0 |
| 6 | DIS | 0 |
| 5 | GROUP | 0 |
| 4 | MPA | 0 |
| 3 | OVER | 0 |
| 2 | FAE | 0 |
| 1 | CRC | 0 |
| 0 | PRX | 0 |

**Bit 7: DFR**, Deferring IGSM
This bit is set when the Interframe Gap State Machine (IGSM) is deferring. If the transceiver has asserted the CD line as a result of jabber, this bit will stay set indicating the jabber condition.

**Bit 6: DIS**, Receiver Disabled
This bit is set when the receiver is in Monitor Mode. It is cleared when the receiver leaves Monitor Mode.

**Bit 5: GROUP**, Group Address Recognized
This bit is set when the recognized address was either a group address (multicast) or broadcast. It is cleared to indicate an individual (physical) address match.

**Bit 4: MPA**, Missed Packet
This bit is set when a packet intended for this station cannot be accepted by the device due to a lack of receive buffers or because the device is in monitor mode. The Missed Packet Counter (MPCNT) is also incremented when this occurs.

**Bit 3: OVER**, FIFO Overrun
This bit is set when the receiver attempts to write into a FIFO that is already full. This occurs when the DMA fails to keep up with the received data.

**Bit 2: FAE**, Frame Alignment Error
When FAE = 1, it indicates that the incoming packet did not end on a byte boundary and the CRC did not match at the last byte boundary. The Alignment Error Counter is incremented when this condition occurs.

**Bit 1: CRC**, CRC Error
When this bit is set, it indicates that the frame's computed CRC failed to correspond with the CRC appended to the end of the frame. This error also causes the CRC Counter to be incremented.

**Bit 0: PRX**, Packet Received Intact
When set to '1', this bit indicates that a packet was received without error. This means that CRC = FAE = OVER = MPA = 0.

### 5.2.29 RSTOP - Receive Stop Page Register

Normal Map Read Port = 2:12
Normal Map Write Port = 0:12

The Receive Stop Page Register points to the first address beyond the last receive buffer in the ring before wrapping around to the RSTART buffer. Only A08-A15 are specified since all buffers are aligned on 256-byte boundaries.

| BIT | RSTOP | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.30 RTABH - Receive Buffer Table Pointer High Register

Linked-List Map Read/Write Port = 0:19

This register contains the upper 8 bits for the register pair used as a pointer to the receive buffer descriptors table. These registers should be initialized to the same value as the RBEGIN register when the descriptor table is created and thereafter left unaltered unless the receiver buffer pool is rebuilt. For more information, refer to page 89.

| BIT | RTABH | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.31 RTABL - Receive Buffer Table Pointer Low Register

Linked-List Map Read/Write Port = 0:18

This register contains the lower 8 bits for the register pair used as a pointer to the receive buffer descriptors table. These registers should be initialized to the same value as the RBEGIN register when the descriptor table is created and thereafter left unaltered unless the receiver buffer pool is rebuilt. For more information, refer to page 89.

| BIT | RTABL | RESET |
|---|---|---|
| 7 | A07 | X |
| 6 | A06 | X |
| 5 | A05 | X |
| 4 | A04 | X |
| 3 | A03 | X |
| 2 | A02 | X |
| 1 | A01 | X |
| 0 | A00 | X |

### 5.2.32 STA0-STA5 - Station Address Registers

| STA Register | Normal Map Port Address | | Linked-List Map Port Address | |
|---|---|---|---|---|
| | Read | Write | Read | Write |
| STA0 | 1:11 | 1:11 | 1:11 | 1:11 |
| STA1 | 1:12 | 1:12 | 1:12 | 1:12 |
| STA2 | 1:13 | 1:13 | 1:13 | 1:13 |
| STA3 | 1:14 | 1:14 | 1:14 | 1:14 |
| STA4 | 1:15 | 1:15 | 1:15 | 1:15 |
| STA5 | 1:16 | 1:16 | 1:16 | 1:16 |

These 6 registers hold the node's individual station address. Table 5-11 shows the bits defined for these registers.

| STA Registers | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|
| STA0 | DA07 | DA06 | DA05 | DA04 | DA03 | DA02 | DA01 | DA00 |
| STA1 | DA15 | DA14 | DA13 | DA12 | DA11 | DA10 | DA09 | DA08 |
| STA2 | DA23 | DA22 | DA21 | DA20 | DA19 | DA18 | DA17 | DA16 |
| STA3 | DA31 | DA30 | DA29 | DA28 | DA27 | DA26 | DA25 | DA24 |
| STA4 | DA39 | DA38 | DA37 | DA36 | DA35 | DA34 | DA33 | DA32 |
| STA5 | DA47 | DA46 | DA45 | DA44 | DA43 | DA42 | DA41 | DA40 |
| RESET | X | X | X | X | X | X | X | X |

**TABLE 5-11. STATION ADDRESS REGISTER BITS**

### 5.2.33 TADDH - Transmit Burst Starting Address High Register

Normal Map Read/Write Port = 2:1B

This is the higher 8 bits of a register pair used internally by the DMA controller as a scratch pad for the burst address of the transmit process. Writing to the TADDH and TADDL Registers while communication is taking place may cause errors in the DMA process.

| BIT | TADDH | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.34 TADDL - Transmit Burst Starting Address Low Register

Normal Map Read/Write Port = 2:1A

This register contains the lower 8 bits for a register pair used internally by the DMA controller as a scratch pad for the burst address of the transmit process. Writing to the TADDH and TADDL registers while communication is taking place may cause errors in the DMA process.

| BIT | TADDL | RESET |
|---|---|---|
| 7 | A07 | X |
| 6 | A06 | X |
| 5 | A05 | X |
| 4 | A04 | X |
| 3 | A03 | X |
| 2 | A02 | X |
| 1 | A01 | X |
| 0 | A00 | X |

### 5.2.35 TBEGIN - Transmit Buffer Starting Address Register

Linked-List Map Read Port = 2:13
Linked-List Map Write Port = 0:15

This register holds the upper 8 bits of the starting address of the transmit buffer descriptor table. The lower 8 bits are assumed to be zero. Refer to page 80 for more information.

| BIT | TBEGIN | RESET |
|---|---|---|
| 7 | TB15 | X |
| 6 | TB14 | X |
| 5 | TB13 | X |
| 4 | TB12 | X |
| 3 | TB11 | X |
| 2 | TB10 | X |
| 1 | TB09 | X |
| 0 | TB08 | X |

### 5.2.36 TCNTH - Transmit Frame Length High Register

Normal Map Read Port = 2:16 Normal Map Write Port = 0:16

This register contains the upper 8 bits of a two-register set that holds the byte count for the frame to be transmitted. This byte count must include the DA, SA, and data fields. If CRC generation is inhibited, this count must also include the CRC field in the buffer.

| BIT | TCNTH | RESET |
|---|---|---|
| 7 | L15 | X |
| 6 | L14 | X |
| 5 | L13 | X |
| 4 | L12 | X |
| 3 | L11 | X |
| 2 | L10 | X |
| 1 | L09 | X |
| 0 | L08 | X |

### 5.2.37 TCNTL - Transmit Frame Length Low Register

Normal Map Read Port = 2:13 Normal Map Write Port = 0:15

This register contains the lower 8 bits of a two-register set that holds the byte count for the frame to be transmitted. This byte count must include the DA, SA, and data fields. If CRC generation is inhibited, this count must also include the CRC field in the buffer.

| BIT | TCNTL | RESET |
|---|---|---|
| 7 | L07 | X |
| 6 | L06 | X |
| 5 | L05 | X |
| 4 | L04 | X |
| 3 | L03 | X |
| 2 | L02 | X |
| 1 | L01 | X |
| 0 | L00 | X |

### 5.2.38 TCON - Transmit Configuration Register

Normal Map Read Port = 2:1D Normal Map Write Port = 0:1D
Linked-List Map Read Port = 2:1D Linked-List Map Write Port = 0:1D

This register controls loopback options and transmitter mode operations.

| BIT | TCON | RESET |
|---|---|---|
| 7 | 0 | 0 |
| 6 | 0 | 0 |
| 5 | 0 | 0 |
| 4 | 0 | 0 |
| 3 | 0 | 0 |
| 2 | LB1 | 0 |
| 1 | LB0 | 0 |
| 0 | CRCN | 0 |

**Bits 2-1: LB1, LB0**, Loopback Test Selection
These two bits are decoded as shown in Table 5-12.

| LB1 | LB0 | Operation |
|---|---|---|
| 0 | 0 | Normal (no loopback) |
| 0 | 1 | Internal loopback (before MAN CODEC) |
| 1 | 0 | Internal loopback, LOOP pin is high (after MAN CODEC) |
| 1 | 1 | External loopback with LOOP pin low |

**TABLE 5-12. LOOPBACK TEST SELECTION**

**Bit 0: CRCN**, CRC Generation Inhibition
Setting this bit inhibits generation of CRC during transmission of frame. The user is responsible for calculating the frame's CRC and placing it in the buffer in such a way that when the last 4 bytes of the buffer are shifted out, they form the correct CRC for the frame. Note that the serializer shifts bytes out LSB first whereas the CRC must be shifted MSB first. The operation of the receiver is not affected by this bit.

### 5.2.39 TDOWNH - Transfer Count High Register

Linked-List Map Read/Write Port = 2:1B

This register contains the upper 8 bits for the register pair used by the DMA controller as a scratch pad for the bytes remaining to transfer count during the transmission process. They can be accessed for manufacturing test purposes.

**Note**

Writing to these registers while communication is taking place may cause errors in the DMA process.

| BIT | TDOWNH | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.40 TDOWNL - Transfer Count Low Register

Linked-List Map Read/Write Port = 2:1A

This register contains the lower 8 bits for the register pair used by the DMA controller as a scratch pad for the bytes remaining to transfer count during the transmission process. They can be accessed for manufacturing test purposes.

**Note**

Writing to these registers while communication is taking place may cause errors in the DMA process.

| BIT | TDOWNL | RESET |
|---|---|---|
| 7 | A07 | X |
| 6 | A06 | X |
| 5 | A05 | X |
| 4 | A04 | X |
| 3 | A03 | X |
| 2 | A02 | X |
| 1 | A01 | X |
| 0 | A00 | X |

### 5.2.41 TEND - Transfer Buffer End Register

Linked-List Map Read Port = 2:14
Linked-List Map Write Port = 0:14

This register holds the upper 8 bits of the first address beyond the end of the transmit buffer descriptor table. The lower 8 bits are assumed to be zero. The table lies between (TBEGIN * 256) and (TEND * 256 - 1). Refer to page 80 for more information.

| BIT | TEND | RESET |
|---|---|---|
| 7 | TE15 | X |
| 6 | TE14 | X |
| 5 | TE13 | X |
| 4 | TE12 | X |
| 3 | TE11 | X |
| 2 | TE10 | X |
| 1 | TE9 | X |
| 0 | TE8 | X |

### 5.2.42 TLEVEL - Transmit FIFO Track Register

Normal Map Read Port = 3:1E
Linked-List Map Read Port = 3:1E

This counter tracks the number of empty bytes in the transmit FIFO. An empty FIFO has 10h in this counter. A full FIFO has 00h.

| BIT | TLEVEL | RESET |
|---|---|---|
| 7 | — | 0 |
| 6 | — | 0 |
| 5 | — | 0 |
| 4 | CT04 | 0 |
| 3 | CT03 | 0 |
| 2 | CT02 | 0 |
| 1 | CT01 | 0 |
| 0 | CT00 | 0 |

### 5.2.43 TSTARTH - Transmit Start Page High Register

Normal Map Read Port = 2:14
Normal Map Write Port = 0:14

This register is the higher 8 bits of a register pair that points to the assembled packet to be transmitted. To retain compatibility with 83C690 drivers, the user should start all frames on 256-byte boundaries and not write to TSTARTL.

| BIT | TSTARTH | RESET |
|---|---|---|
| 7 | A15 | 0 |
| 6 | A14 | 0 |
| 5 | A13 | 0 |
| 4 | A12 | 0 |
| 3 | A11 | 0 |
| 2 | A10 | 0 |
| 1 | A09 | 0 |
| 0 | A08 | 0 |

### 5.2.44 TSTARTL - Transmit Start Page Low Register

Normal Map Read/Write Port = 3:15
Linked-List Map Read/Write Port = 3:15

This register is the lower 8 bits of a register pair that points to the assembled packet to be transmitted. To retain compatibility with 83C690 drivers, the user should start all frames on 256-byte boundaries and only write to the TSTARTH register.

| BIT | TSTARTL | RESET |
|---|---|---|
| 7 | A07 | 0 |
| 6 | A06 | 0 |
| 5 | A05 | 0 |
| 4 | A04 | 0 |
| 3 | A03 | 0 |
| 2 | A02 | 0 |
| 1 | A01 | 0 |
| 0 | A00 | 0 |

### 5.2.45 TSTAT - Transmit Status Register

Normal Map Read Port = 0:14
Linked-List Map Read Port = 0:14

The Transmit Status Register reports events that occur on the media at the end of packet transmission. All bits are cleared prior to transmission of a packet and are set as needed. When ALTEGO = 1, the register is cleared only at the beginning of a transmit chain and is set after each packet has completed.

| BIT | TSTAT | RESET |
|---|---|---|
| 7 | OWC | 0 |
| 6 | CDH | 0 |
| 5 | UNDER | 0 |
| 4 | CRL | 0 |
| 3 | ABORT | 0 |
| 2 | TWC | 0 |
| 1 | NDT | 0 |
| 0 | PTX | 0 |

**Bit 7: OWC**, Out of Window Collision
This bit is set if a collision is detected more than one slot time after the start of transmission. Transmission is aborted under these conditions.

**Bit 6: CDH**, Collision Detect Heartbeat
This bit is set to a '1' during transmission of each packet. It is set to '0' if a collision is detected within 3.6 µsec of the end of each packet transmission. If no collision is detected within this window, it remains '1'.

**Bit 5: UNDERFIFO**, FIFO or Buffer Underrun
When this bit is set, it means either:

- a FIFO underrun condition has occurred. This condition results when the transmit unit attempts to read from an empty FIFO prior to receiving the transmit done flag from DMA. This means that the FIFO failed to supply enough data for the serializer to maintain frame generation.
- a Buffer underrun has occurred. This condition happens when the transmit DMA accesses an address that is greater than or equal to the most recent host-written location in memory, provided that the Early Transmit Check feature is enabled.

**Bit 4: CRL**, Carrier Sense Lost
This bit is set if the carrier is lost during packet transmission. Carrier sense is monitored from its rising edge at the start of the outgoing frame's echo. Transmission is not aborted upon loss of carrier. It is reported for statistical purposes.

**Bit 3: ABORT**, Abort Transmission
This bit is set if the transmission is aborted due to excessive collisions.

**Bit 2: TWC**, Transmitted With Collisions
This bit is set if the frame collided at least once with another frame on the network. It is not set for either out-of-window collisions or excessive collision aborts.

**Bit 1: NDT**, Non-deferred Transmission
This bit is set if the frame was transmitted successfully without deferring. A deferred transmission can only occur the first time an attempt is made to send a packet. Collisions are not deferred transmissions.

**Bit 0: PTX**, Packet Transmitted
This bit is set to indicate transmission of a packet without excessive collisions or a FIFO underrun.

### 5.2.46 TTABH - Transmit Buffer Pointer High Register

Linked-List Map Read/Write Port = 0:1B

This register contains the higher 8 bits of the register pair used as a pointer to the transmit buffer descriptors table. These registers should be initialized to the same value as TBEGIN when the descriptor table is created, and not altered thereafter by the user unless the transmit buffer pool is rebuilt. For more information, refer to page 80.

| BIT | TTABH | RESET |
|---|---|---|
| 7 | A15 | X |
| 6 | A14 | X |
| 5 | A13 | X |
| 4 | A12 | X |
| 3 | A11 | X |
| 2 | A10 | X |
| 1 | A09 | X |
| 0 | A08 | X |

### 5.2.47 TTABL - Transmit Buffer Pointer Low Register

Linked-List Map Read/Write Port = 0:1A

This register contains the lower 8 bits of the register pair used as a pointer to the transmit buffer descriptor table. These registers should be initialized to the same value as TBEGIN when the descriptor table is created, and not altered thereafter by the user unless the transmit buffer pool is rebuilt.

| BIT | TTABL | RESET |
|---|---|---|
| 7 | A07 | X |
| 6 | A06 | X |
| 5 | A05 | X |
| 4 | A04 | X |
| 3 | A03 | X |
| 2 | A02 | X |
| 1 | A01 | X |
| 0 | A00 | X |

### 5.2.48 RENH - Receive Enhancement Register

Normal Map Read/Write Port = 0:19

The Receive Enhancement Register contains several bits required for the new receive features of the 83C795 chip.

| BIT | RENH | RESET |
|---|---|---|
| 7 | — | 0 |
| 6 | — | 0 |
| 5 | — | 0 |
| 4 | — | 0 |
| 3 | — | 0 |
| 2 | REMPTY | 0 |
| 1 | ERFBIT | 0 |
| 0 | WRAPEN | 0 |

**Bit 2: REMPTY**, Ring Bit Empty
When REMPTY = 1, this read-only bit indicates that the receive buffer ring has no completely received frames.

**Bit 1: ERFBIT**, Early Receive Fail Bit
When ERFBIT = 1 it indicates that an underrun has occurred during the reception of a frame. The host clears this bit after reading the address where the failure occurred from the ERFA Registers.

**Bit 0: WRAPEN**, Automatic Ring-Wrap Enable
When WRAPEN = 1 it enables the auto-wrapping feature. For more information on Automatic Ring-Wrap, refer to page 87.

| ADDRESS | FUNCTION | NAME | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 00 | CONTROL | CR | RNIC | MENB | — | CR4 | CR3 | RP15 | RP14 | RP13 |
| 01 | EEROM | RD ER | STORE | RC | EA4 | UNLOCK | JMP3 | JMP2 | JMP1 | JMP0 |
| 01 | EEROM | WR ER | STORE | RC | EA4 | UNLOCK | EA3 | EA2 | EA1 | EA0 |
| 02 | IO PIPE LOW | IOPL | IOP7 | IOP6 | IOP5 | IOP4 | IOP3 | IOP2 | IOP1 | IOP0 |
| 03 | IO PIPE HIGH | IOPH | IOP15 | IOP14 | IOP13 | IOP12 | IOP11 | IOP10 | IOP9 | IOP8 |
| 04 | HW SUPP | RD HWR | SWH | — | ETHER | HOST16 | — | ISTAT | PNPJMP | GPOE |
|  | HW SUPP | WR HWR | SWH | — | — | — | NUKE | — | — | GPOE |
| 05 | BIOS PAGE | BPR | M16EN | BP15 | BP14 | BP13 | — | — | SOFT1 | SOFT0 |
| 06 | INT CONTROL | ICR | MCTEST | STAG | IOPAV | IOPEN | SINT | MASK2 | MASK1 | EIL |
| 07 | REVISION | REV | CHIP3 | CHIP2 | CHIP1 | CHIP0 | REV3 | REV2 | REV1 | REV0 |
|  | IO PIPE ADDR | IOPA | IOPA7 | IOPA6 | IOPA5 | IOPA4 | IOPA3 | IOPA2 | IOPA1 | IOPA0 |
| 08 SWH=0 | LAN ADDR0 | LAN0 | LN07 | LN06 | LN05 | LN04 | LN03 | LN02 | LN01 | LN00 |
| 09 SWH=0 | LAN ADDR1 | LAN1 | LN15 | LN14 | LN13 | LN12 | LN11 | LN10 | LN09 | LN08 |
| 0A SWH=0 | LAN ADDR2 | LAN2 | LN23 | LN22 | LN21 | LN20 | LN19 | LN18 | LN17 | LN16 |
| 0B SWH=0 | LAN ADDR3 | LAN3 | LN31 | LN30 | LN29 | LN28 | LN27 | LN26 | LN25 | LN24 |
| 0C SWH=0 | LAN ADDR4 | LAN4 | LN39 | LN38 | LN37 | LN36 | LN35 | LN34 | LN33 | LN32 |
| 0D SWH=0 | LAN ADDR5 | LAN5 | LNMSB | LN46 | LN45 | LN44 | LN43 | LN42 | LN41 | LN40 |
| 0E SWH=0 | BOARD ID | BDID | BDID7 | BDID6 | BDID5 | BDID4 | BDID3 | BDID2 | BDID1 | BDID0 |
| 0F SWH=0 | CHECKSUM | CKSM | CHK7 | CHK6 | CHK5 | CHK4 | CHK3 | CHK2 | CHK1 | CHK0 |
| 08 SWH=1 | GENERAL CNTL2 | GCR2 | — | — | — | — | — | — | — | PNPIOP |
| 0A SWH=1 | I/O ADDRESS | IAR | IA15 | IA14 | IA13 | IA8 | IA7 | IA6 | IA5 | PNPBOOT |
| 0B SWH=1 | RAM BASE | RAR | HRAM | RA17 | RAMSZ1 | RAMSZ0 | RA16 | RA15 | RA14 | RA13 |
| 0C SWH=1 | BIOS BASE | BIO | FINE | BA17 | BIOSZ1 | BIOSZ0 | BA16 | BA15 | BA14 | BA13 |
| 0D SWH=1 | GEN CONTROL | GCR | XLENGTH | IR2 | OWS | RIPL | IR1 | IR0 | GPOUT | LIT |
| 0E SWH=1 | ERF ADDR LOW | ERFAL | ERFA7 | ERFA6 | ERFA5 | ERFA4 | ERFA3 | ERFA2 | — | PNPEN |
| 0F SWH=1 | ERF ADDR HIGH | ERFAH | ERFA15 | ERFA14 | ERFA13 | ERFA12 | ERFA11 | ERFA10 | ERFA9 | ERFA8 |

**TABLE 5-13. HOST INTERFACE REGISTER SUMMARY**

| REGISTER | RING | | LINKED | | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ALICNT | 0:1D | — | 0:1D | — | CT7 | CT6 | CT5 | CT4 | CT3 | CT2 | CT1 | CT0 |
| BOUND | 0:13 | 0:13 | — | — | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| CMD | X:10 | X:10 | X:10 | X:10 | PS1 | PS0 | 0 | ENETCH | DISETCH | TXP | STA | STP |
| COLCNT | 0:15 | — | 0:15 | — | T10 | T9 | T8 | T7 | CT3 | CT2 | CT1 | CT0 |
| CRCCNT | 0:1E | — | 0:1E | — | CT7 | CT6 | CT5 | CT4 | CT3 | CT2 | CT1 | CT0 |
| CURR | 1:17 | 1:17 | — | — | CURR15 | CURR14 | CURR13 | CURR12 | CURR11 | CURR10 | CURR09 | CURR08 |
| CURRH | — | — | 1:17 | 1:17 | CURR15 | CURR14 | CURR13 | CURR12 | CURR11 | CURR10 | CURR09 | CURR08 |
| CURRL | — | — | 0:13 | 0:13 | CURR07 | CURR06 | CURR05 | CURR04 | CURR03 | CURR02 | CURR01 | CURR00 |
| DCON | 2:1E | 0:1E | 2:1E | 0:1E | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| ENH | 2:17 | 2:17 | 2:17 | 2:17 | 0 | 1 | ALTEGO | SLOT1 | SLOT0 | EOTINT | — | SBACK |
| ERWCNT | 0:18 | 0:18 | 0:18 | 0:18 | ERW11 | ERW10 | ERW9 | ERW8 | ERW7 | ERW6 | ERW5 | ERW4 |
| GROUP0 | 1:18 | 1:18 | 1:18 | 1:18 | GA07 | GA06 | GA05 | GA04 | GA03 | GA02 | GA01 | GA00 |
| GROUP1 | 1:19 | 1:19 | 1:19 | 1:19 | GA15 | GA14 | GA13 | GA12 | GA11 | GA10 | GA09 | GA08 |
| GROUP2 | 1:1A | 1:1A | 1:1A | 1:1A | GA23 | GA22 | GA21 | GA20 | GA19 | GA18 | GA17 | GA16 |
| GROUP3 | 1:1B | 1:1B | 1:1B | 1:1B | GA31 | GA30 | GA29 | GA28 | GA27 | GA26 | GA25 | GA24 |
| GROUP4 | 1:1C | 1:1C | 1:1C | 1:1C | GA39 | GA38 | GA37 | GA36 | GA35 | GA34 | GA33 | GA32 |
| GROUP5 | 1:1D | 1:1D | 1:1D | 1:1D | GA47 | GA46 | GA45 | GA44 | GA43 | GA42 | GA41 | GA40 |
| GROUP6 | 1:1E | 1:1E | 1:1E | 1:1E | GA55 | GA54 | GA53 | GA52 | GA51 | GA50 | GA49 | GA48 |
| GROUP7 | 1:1F | 1:1F | 1:1F | 1:1F | GA63 | GA62 | GA61 | GA60 | GA59 | GA58 | GA57 | GA56 |
| INTMASK | 2:1F | 0:1F | 2:1F | 0:1F | 0 | ERWE | CNTE | OVWE | TXEE | RXEE | PTXE | PRXE |
| INTSTAT | 0:17 | 0:17 | 0:17 | 0:17 | RST | ERW | CNT | OVW | TXE | RXE | PTX | PRX |
| MANCH | 3:1F | 3:1F | 3:1F | 3:1F | MANDIS | SEL | 0 | ENAPOL | TPOL | LNK | RLED | XLED |
| MPCNT | 0:1F | — | 0:1F | — | CT7 | CT6 | CT5 | CT4 | CT3 | CT2 | CT1 | CT0 |
| NEXT | 2:15 | 2:15 | — | — | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| RADDH | 2:19 | 2:19 | — | — | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| RADDL | 2:18 | 2:18 | — | — | A07 | A06 | A05 | A04 | A03 | A02 | A01 | A00 |
| RBEGIN | — | — | 2:11 | 0:11 | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| RCNTH | 0:1B | — | — | — | CT15 | CT14 | CT13 | CT12 | CT11 | CT10 | CT09 | CT08 |
| RCNTL | 0:1A | — | — | — | CT07 | CT06 | CT05 | CT04 | CT03 | CT02 | CT01 | CT00 |
| RCON | 2:1C | 0:1C | 2:1C | 0:1C | 0 | RCA | MON | PROM | GROUP | BROAD | RUNTS | SEP |
| RDOWNH | — | — | 2:19 | 2:19 | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| RDOWNL | — | — | 2:18 | 2:18 | A07 | A06 | A05 | A04 | A03 | A02 | A01 | A00 |
| REND | — | — | 2:12 | 0:12 | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| RENH | 0:19 | 0:19 | | | — | — | — | — | — | REMPTY | ERFBIT | WRAPEN |
| RSTART | 2:11 | 0:11 | — | — | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| RSTAT | 0:1C | — | 0:1C | — | DFR | DIS | GROUP | MPA | OVER | FAE | CRC | PRX |
| RSTOP | 2:12 | 0:12 | — | — | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| RTABH | — | — | 0:19 | 0:19 | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |

**TABLE 5-14. LAN CONTROLLER REGISTER SUMMARY**

| REGISTER | RING | | LINKED | | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RTABL | — | — | 0:18 | 0:18 | A07 | A06 | A05 | A04 | A03 | A02 | A01 | A00 |
| STA0 | 1:11 | 1:11 | 1:11 | 1:11 | DA07 | DA06 | DA05 | DA04 | DA03 | DA02 | DA01 | DA00 |
| STA1 | 1:12 | 1:12 | 1:12 | 1:12 | DA15 | DA14 | DA13 | DA12 | DA11 | DA10 | DA09 | DA08 |
| STA2 | 1:13 | 1:13 | 1:13 | 1:13 | DA23 | DA22 | DA21 | DA20 | DA19 | DA18 | DA17 | DA16 |
| STA3 | 1:14 | 1:14 | 1:14 | 1:14 | DA31 | DA30 | DA29 | DA28 | DA27 | DA26 | DA25 | DA24 |
| STA4 | 1:15 | 1:15 | 1:15 | 1:15 | DA39 | DA38 | DA37 | DA36 | DA35 | DA34 | DA33 | DA32 |
| STA5 | 1:16 | 1:16 | 1:16 | 1:16 | DA47 | DA46 | DA45 | DA44 | DA43 | DA42 | DA41 | DA40 |
| TADDH | 2:1B | 2:1B | — | — | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| TADDL | 2:1A | 2:1A | — | — | A07 | A06 | A05 | A04 | A03 | A02 | A01 | A00 |
| TBEGIN | — | — | 2:13 | 0:15 | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| TCNTH | 2:16 | 0:16 | — | — | L15 | L14 | L13 | L12 | L11 | L10 | L09 | L08 |
| TCNTL | 2:15 | 2:15 | — | — | L07 | L06 | L05 | L04 | L03 | L02 | L01 | L00 |
| TCON | 2:1D | 0:1D | 2:1D | 0:1D | — | — | — | — | — | LB1 | LB0 | CRCN |
| TDOWNH | — | — | 2:1B | 2:1B | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |
| TDOWNL | — | — | 2:1A | 2:1A | A07 | A06 | A05 | A04 | A03 | A02 | A01 | A00 |
| TEND | — | — | 2:14 | 0:14 | TE15 | TE14 | TE13 | TE12 | TE11 | TE10 | TE09 | TE08 |
| TLEVEL | 3:1E | | 3:1E | | — | — | — | CT04 | CT03 | CT02 | CT01 | CT00 |
| TSTARTH | 2:14 | 0:14 | — | — | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| TSTARTL | 3:15 | 3:15 | 3:15 | 3:15 | A07 | A06 | A05 | A04 | A03 | A02 | A01 | A00 |
| TSTAT | 0:14 | — | 0:14 | — | OWC | CDH | UNDER | CRL | ABORT | TWC | NDT | PTX |
| TTABH | — | — | 0:1B | 0:1B | A15 | A14 | A13 | A12 | A11 | A10 | A09 | A08 |
| TTABL | — | — | 0:1A | 0:1A | A07 | A06 | A05 | A04 | A03 | A02 | A01 | A00 |

**TABLE 5-14. LAN CONTROLLER REGISTER SUMMARY (cont.)**

## 6.0 HOST INTERFACE SECTION

The Host Interface is a configurable interface between an Industry Standard Architecture bus (like IBM PC/XT/AT) and the LAN controller with its buffer memory. The interface is a slave peripheral with shared RAM and support for an Initial Program Load ROM.

The basic functions of the host interface section are the:

- Address decode
- Memory address generation
- Retrieval and storage of configuration parameters and LAN address
- Interrupt mapping and control
- Control over certain HW functions for support circuitry

### 6.1 MEMORY CACHE

The memory cache in the 83C795 consists of a 4-byte-deep FIFO which serves as an intermediate buffer between the ISA bus and the local buffer RAM. For read operations, the cache acts as a small prefetch buffer which fills itself with data from locations in the buffer RAM that depend on the address of the last data location read by the host. For write operations, the cache acts as several temporary registers that can be asynchronously written by the host, then synchronously flushed to buffer RAM as time permits. This method provides several advantages over previous methods:

- Host accesses to shared memory can be treated more like register accesses, thus simplifying zero-wait state timing.
- A single 8-bit wide buffer RAM is used but the chip can accommodate 8- or 16-bit accesses by the host.
- Asynchronous arbitration between the host and the DMA controller for access to the buffer RAM is not necessary.

In addition to the data FIFO, the cache requires two address counters for its operation. The first, called the Host Counter, is compared with the incoming host address. The second, called the Buffer Counter, is used to generate the address to the local buffer RAM. The same data FIFO is used for both reads and writes resulting in two different modes of operation: Read Mode and Write Mode.

READ MODE

If the host address does not equal the value in the Host Counter, then both counters are loaded with the incoming address. Then the cache is filled with data from the buffer RAM a byte at a time by incrementing the Buffer Counter. The host access is stalled during this time by driving the IORDY signal low. Once the cache has a valid word of data from the buffer RAM, the IORDY line is driven high. This signals the host that the data is valid, and the host, in turn, ends the access. Once the host has finished, the Host Counter is incremented (the increment step will be either 1 or 2 depending on whether the host access was for a byte or for a word), and the FIFO pointers are updated. The cache continues to fill with data as long as there is room in the FIFO. If the host address matches the value in the Host Counter (and there is valid data in the cache), the read can be serviced immediately.

WRITE MODE

The Write Mode is handled like the Read Mode except that the data moves in the opposite direction through the FIFO. Also, if an address miss occurs in write mode, the cache must first flush all valid data in the FIFO out to the buffer RAM before loading new values into the address counters.

The IORDY signal is used when the cache needs to stall a host access. This signal is outputted by a high current, tri-statable driver which is normally turned off between accesses to the board. It drives low to indicate that the board is not IORDY and drives high when making the transition from 'not ready' to 'ready'.

When the access completes, the IORDY line is tri-stated by the ending of the host's strobes. Figure 6-1 depicts the memory cache arrangement.

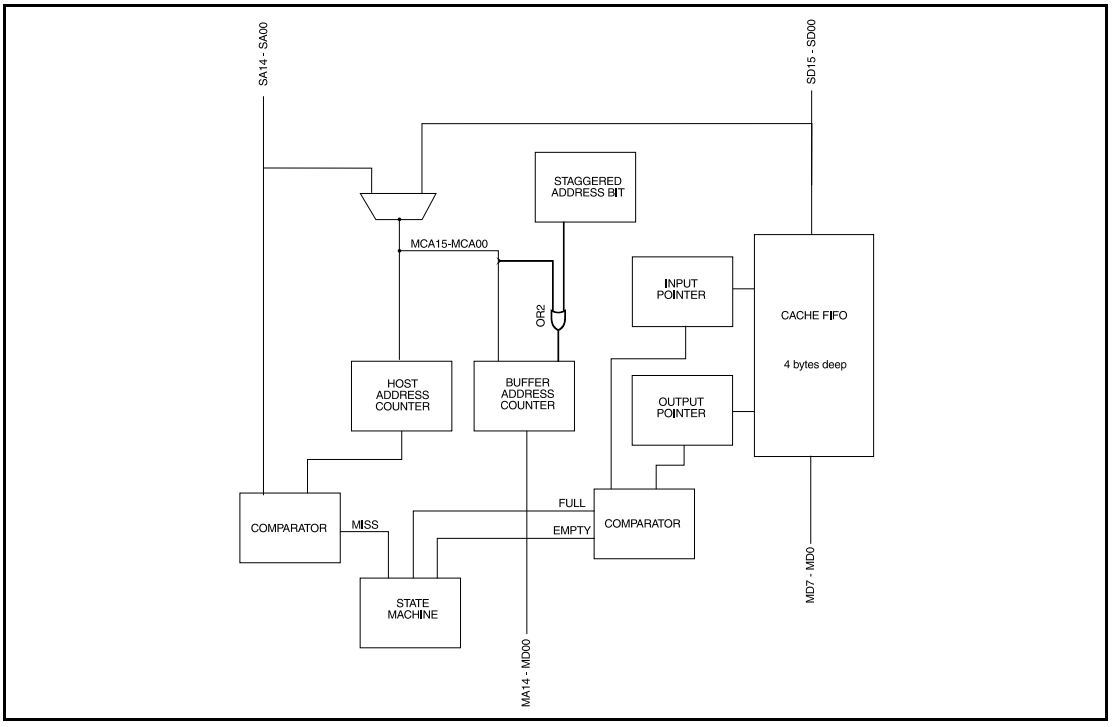


**FIGURE 6-1. MEMORY CACHE ARRANGEMENT**

### 6.1.1 Zero Wait State Response to Host

The Zero Wait State signal tells the microprocessor that it can complete the present bus cycle without inserting any additional wait cycles. For 16-bit memory access, this means zero wait states are inserted by the host bus logic and the access cycle completes in 2 bus clocks. When asserted for an 8 bit memory access, an ISA bus automatically inserts the minimum of 2 wait states.

The response algorithm for the $\overline{ZWS}$ line depends upon the memory width, the host access type and whether the board has been enabled to act as a 16-bit device. The appropriate $\overline{ZWS}$ response logic is selected on the basis of the BPR.M16EN control bit and whether the board is in an 8- or 16-bit slot.

The memory cache can accommodate zero wait state timing if the following conditions are met:

1. The type of host access matches the current mode of the cache,
2. The host address matches the value in the host counter, and
3. The cache either contains at least one valid data word for reads or has room for at least one more word for writes.

For writes, zero wait states are also always possible if the cache is in read mode, or if it is currently empty.

There is a Zero Wait Enable bit in one of the host interface registers (CR.ZWSEN) which can be used to prevent the 83C795 from asserting the Zero Wait State signal.

### 6.1.2 Staggered Address Transfers

Staggered address transfers occur when the host attempts 16-bit data transfers from system memory to the local buffer RAM and finds that the address of the system data differs from the local address in the least significant bit (one is even, one is odd). In consequence, the ISA bus forbids 16-bit accesses to odd locations and breaks the transfer into two 8-bit cycles which run considerably slower.

To overcome this on the 795:

1. Make sure the system address is even. If the address comes out odd, transfer one byte.
2. Set the STAG bit (ICR.6). This forces a '1' into bit 0 of the Buffer Counter when the address is loaded

**NOTE**

This only happens on a cache miss.

This makes it possible for the host to perform an even-to-even transfer, but the internal address to the local RAM is transformed to an odd address.

### 6.1.3 Operation on Micro-Channel Adapters

Do not use this chip for micro-channel applications. A future variant may be created with the necessary interface logic.

## 6.2 I/O-MAPPED PIPE

The I/O-mapped pipe provides another method for accessing the local buffer RAM. When enabled, all memory accesses take place through two I/O registers in the host interface I/O space (IOPL and IOPH). The data in these two I/O locations corresponds to the the location in the buffer memory indicated by the Buffer Counter. When run in this mode, the memory-space address decoders are disabled, so the adapter will not use any host memory space for the buffer RAM.

The mechanism used by the I/O-pipe is similar to that used by the memory cache except for address handling. In this method, the address is loaded into the Buffer Counter by performing two consecutive writes to the IOPA register. The first write stores the lower half of the address into a temporary register. The second write stores data directly into the upper half of the buffer counter and moves the temporary register into the lower half. Any access to the chip between the two writes will cause the state machine to not load the address. The Host Counter is not used during this process.

To use the I/O-pipe, the IOPEN bit (ICR.4) must be set to 1, and the MENB bit (CR.6) must be set to 0. All 8-bit transfers must take place through IOPL only. Also, it is imperative that when switching from Read Mode to Write Mode, the address must be reloaded even if the counter holds the correct value.

## 6.3 ADDRESS DECODERS

Three address decoders are used to detect host accesses to the buffer memory, I/O registers, and IPL ROM. These decoders observe the SA19-SA05 lines to decode access within a range of addresses (a window). The buffer and IPL ROM decoders allow placement of their respective windows on any 8K boundary between C0000H and EFFFFH regardless of window size. This allows windows to start on even or odd 8K boundaries. The RAM and IPL ROM are scrollable (and therefore can be paged) through their programmable window size as shown in Table 6-1 below.

| DECODER | MIN BASE | MAX BASE | INCREMENT | WINDOW SIZES |
|---|---|---|---|---|
| BUFFER | C0000H | EE000H | 2000H | 8K<br>16K<br>32K<br>64K*<br>DISABLED |
| IPL ROM | C0000H | EE000H | 2000H | 8K<br>16K<br>32K<br>DISABLED |
| I/O BASE | 0200H | E3E0H | 20H, 2000H | 32 Bytes |

**TABLE 6-1. HOST INTERFACE ADDRESS DECODERS**

* Plug and Play cannot utilize this window size.

Because of the 16-bit memory mechanism employed in ISA buses, do not allow the memory window to overlap the DFFFFH to E0000H boundary when using 16-bit memory. If the boundary crosses, host memory accesses to portions above E0000H are made into 8-bit cycles.

**Note**

Use caution when overlapping the ROM and RAM windows if the RAM is 16-bit wide and both windows are enabled. With 16-bit access enabled, the M16CS is asserted for all accesses within the same 128K address block as the RAM base window address. Should the access actually be intended to the ROM, the host falsely expects the ROM to return 16-bit data. THIS CAN CRASH THE OPERATING SYSTEM. To avoid such problems on 16-bit boards, copy ROM code to system RAM and disable the ROM window. Alternatively, map the ROM into other 128K address block.

Program control enables buffer memory decoding through a register in the host interface section. When connected to a 16-bit bus, this comparison is qualified by MEMR, MEMW, and the inverse of AEN. When connected to an 8-bit bus, the qualification is by SMEMR, SMEMW, and the inverse of AEN. The buffer memory window size is program-selectable as 8K, 16K, 32K, 64K bytes or disabled. The buffer base address can be set to any 8K boundary from C0000H through EE000H.

By setting a bit in the RAM Address Register (RAR.HRAM), the decoded buffer range can be changed to the range FC0000h - FEE000h. (This range is not possible when using Plug and Play.) When connected to a PC/XT bus having no LA lines, it is required that the BPR.M16EN be kept a zero (inactive).

The IPL ROM decoding is enabled by program control through the BIO register in the host interface section. Decoder qualification is by SMEMR and an inverted AEN. ROM window size is program selectable from 8K, 16K, 32K bytes or disabled. The ROM window placement in host memory space

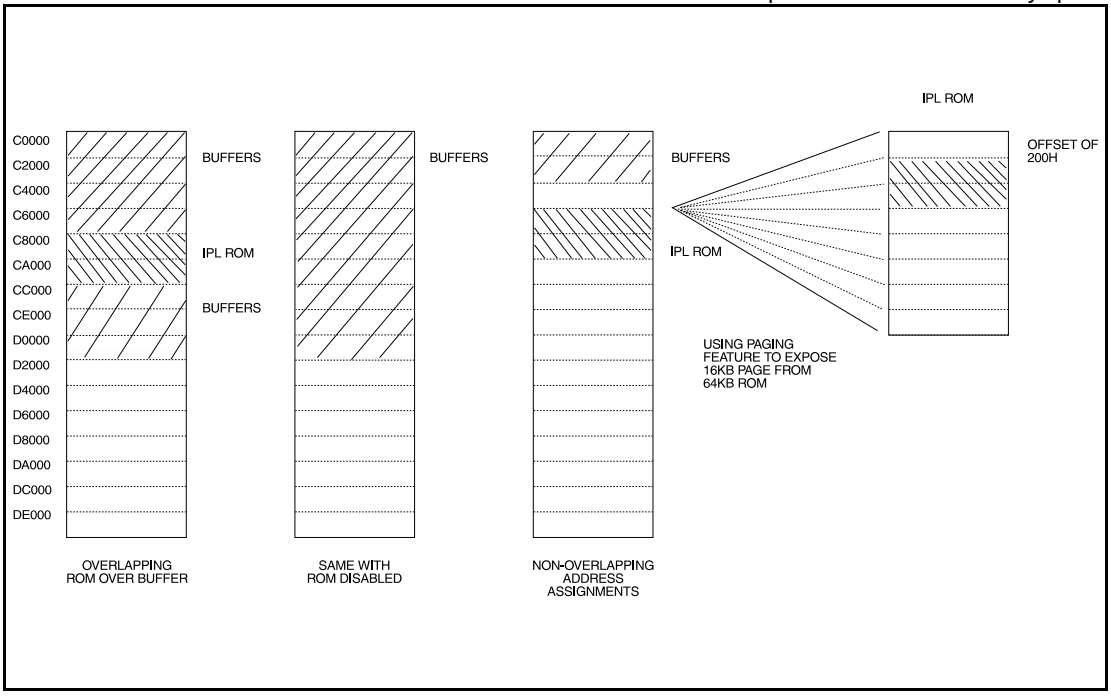


**FIGURE 6-2. OVERLAPPING ADDRESS STRUCTURE**

is programmable on any 8K boundary from C0000H to EE000H. Where ROM and RAM decoding overlaps, ROM takes precedence. Figure 6-2 shows how the overlapping address structure works.

You can relocate RAM and ROM base addresses in tandem above 1M (100000H) with the application of an external cascaded address decoder. This is illustrated in Figure 6-3.

**Note**

Relocation of the memory windows in this way is not supported when Plug and Play is enabled.

### 6.3.1 Memory Address Generation

ROMs and buffer memories larger than their programmed windows can be scrolled (paged) by using programmable address modifiers (adders) that lie between the host address and the ROM or buffer memory. These address modifiers can add independently to either address any of the following values and expose different parts of the targeted memory within the window:

0000H
2000H
4000H
6000H
8000H
A000H
C000H
E000H

The lower address lines (SA12-0) from the host are multiplexed with the DMA address lines to generate part of the memory address. Two potential memory addresses are generated by subtracting the respective RAM and ROM base address bits (17-13) from SA17-SA13, then adding the RP15-RP13 field of the CR Register for the RAM

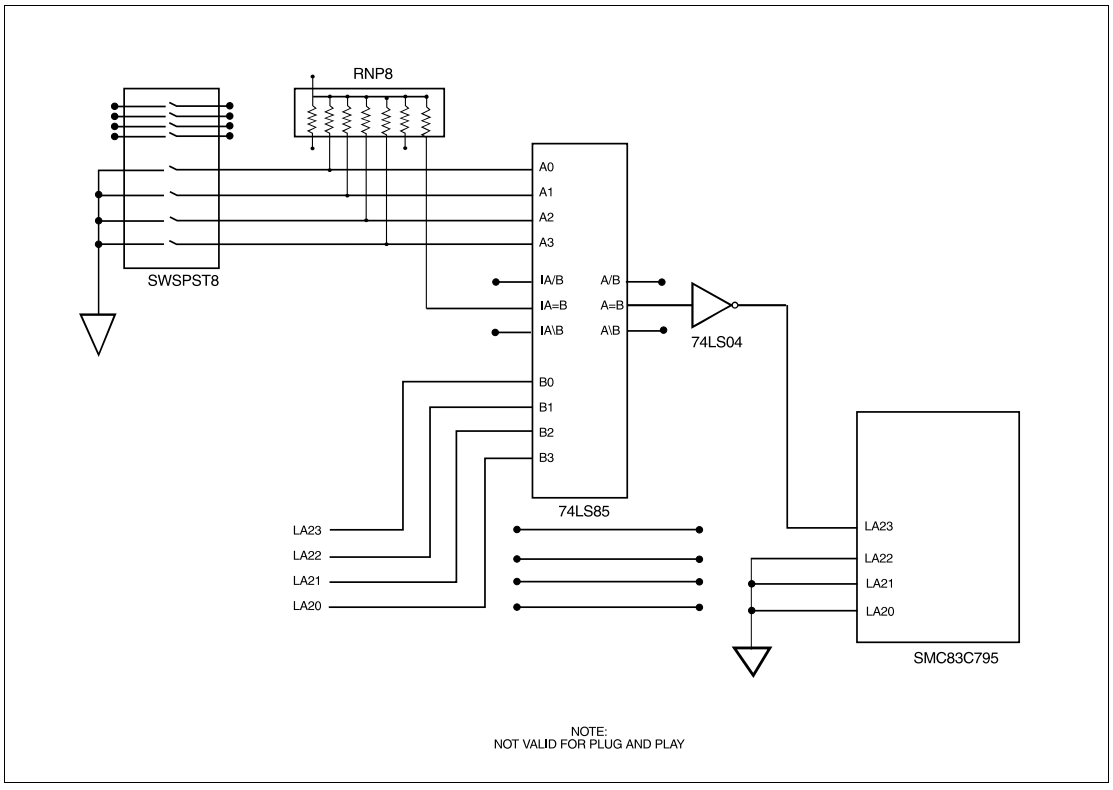


**FIGURE 6-3. EXTERNAL CASCADED ADDRESS DECODER**

address or the BP15-BP13 field of the BPR Register for the ROM address. Depending on whether the buffer memory or the ROM window is being accessed, one of those two possible sums becomes MA15-MA13 to the memory cache counters.

Refer to Figure 6-4 for an illustration of the address generation path.

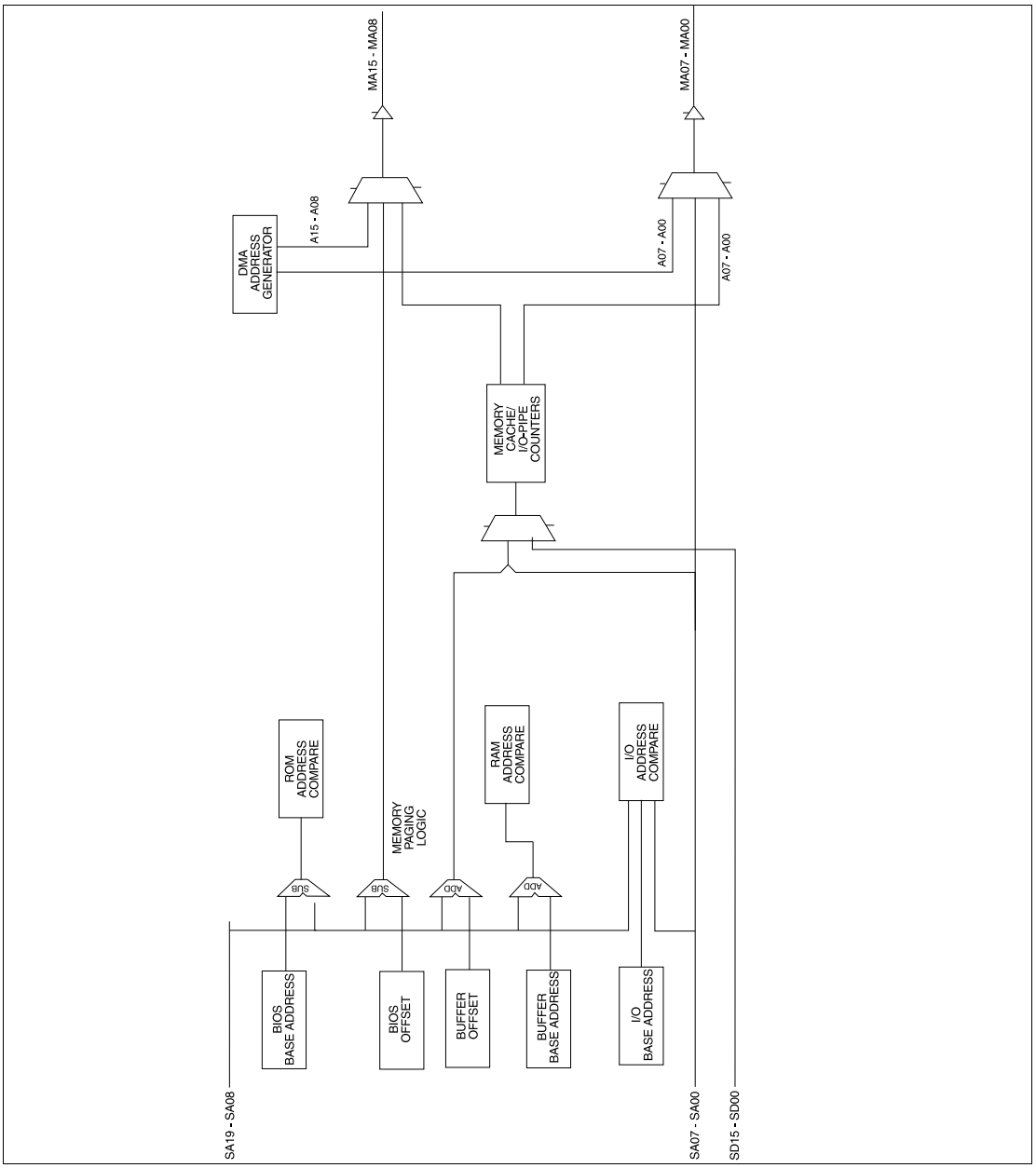


**FIGURE 6-4. ADDRESS GENERATION PATH**

#### 6.3.2 I/O Address Decode

The I/O address decoder compares the system address lines SA15-SA13 and SA8-SA5 against a programmable value. SA9 is compared to '1'. The lower group of lines gives a window size of 32 bytes located on 32-byte boundaries over the range of 200H to 3E0H. The comparison with upper address bits allows the window to be located outside the base I/O area in the event there are multiple LAN cards on the same backplane. This comparison is qualified by the IOR, IOW, and the inverse of the AEN lines. I/O base location possibilities are:

0200, 0220, 0240, ..., 03E0
2200, 2220, 2240, ..., 23E0
4200, 4220, 4240, ..., 43E0
...
E200, E220, E240, ..., E3E0

**Note**

Only the first base location option is supported by Plug and Play.

The I/O address is further decoded to resolve between the LAN controller and registers associated with the host interface based on the A4 address line.

##### 6.3.2.1 PC-98 Bus Support

This feature allows the I/O address decoding to be changed to support the NEC PC-98 bus. This is done by installing JUMPER7, which connects an external resistor between MA7 and ground. When enabled, the SA9-SA1 lines replace the SA8-SA0 lines, the SA12-SA10 lines must be all 1's, and the SA0 line must be zero for an I/O access to occur. This remapping only affects I/O accesses and leaves memory decoding unchanged.

### 6.4 BUS CONTROL SIGNALS

Two signals control much of the bus activity. They are I/O Channel Ready (IORDY) and Zero Wait State (ZWS). Each is explained below.

#### 6.4.1 IORDY

The IORDY output is a high current, tri-state driver which is normally turned off between accesses to the board. It will actively drive low to indicate that the board is not IORDY and drives high when making the transition from 'not ready' to 'ready'.

Access to the internal registers of the LAN controller is arbitrated by the LAN controller. This arbitration is transparent to the host.

When host access is completed, IORDY is tri-stated by the ending of the host's strobes.

#### 6.4.2 Zero Wait State Response To Host

The Zero Wait State (ZWS) signal tells the microprocessor that it can complete the present bus cycle without inserting any additional wait cycles. The response algorithm for the ZWS line depends on the memory width, the host access type, and whether the board has been enabled to act as a 16-bit device. The appropriate type of ZWS response logic is selected on the basis of memory width and the M16EN control bit state.

There is a Zero Wait Enable bit in one of the host interface registers (CR.ZWSEN) which can be used to prevent the 83C795 from asserting the Zero Wait State signal.

## 6.5 MEMORY BUS STRUCTURE AND CONFIGURATION

### 6.5.1 Memory Bus Width Control

Because of the 83C795's memory cache, the width of the memory path is fixed at 8 bit. By means of the memory cache, the I/O pipe can service either an 8-bit or 16-bit host access. The Host and host interface logic are programmed for a specific memory width by setting bit 7 in the BIOS Page Register, M16EN. (See page 17 for more on this bit.)

The 83C795 calculates the width of the host bus by observing the MEMR line for transitions. An internal flag (EEROM.HOST16) is set to indicate a 16-bit host bus after 2 rising edges are seen on this pin. When connected to an 8-bit host, this pin is left unconnected or is tied to VDD and should not have any transitions.

### 6.5.2 16-Bit Response To Host Access

The BPR.M16EN bit in the BIOS Page Register tells the host interface logic whether to make the LAN adapter respond as a 16-bit or an 8-bit peripheral to the host. The 83C795 responds to either host bus width.

When the host accesses memory, an address comparator within the 83C795 looks at the LA23-LA17 lines to determine if the 83C795's memory territory is being accessed. If it is and if MPR.M16EN is set, the M16CS line is activated to tell the host to run a 16-bit transfer cycle. When this decision is based only on the LA address lines, it is possible that the M16CS will be sent out when the host is accessing a device other than 83C795 within the same address range.

To allow finer resolution for the M16CS decode, there is an optional means of including the decoding of SA16-SA13 lines in the generation of the M16CS response. This can be enabled by the FINE16 bit of the Memory Page Register. Because the SA lines are not guaranteed stable as early as LA lines, this form of decoding can lead to erroneous results.

Be careful when you use this method. To avoid bus width conflicts between buffer memory and the ROM as well as conflicts with other cards in the system, the 16-bit response should be turned on by software only when that software can guarantee that no access to the ROM is taking place and that the only accesses within the 128K memory range are to 16-bit devices. This may mean ensuring that no access to any other card can take place. In existing drivers, this is done by performing all 16-bit transfers within interrupt service routines that keep all other interrupts disabled during the transfer.

Take special care when writing IPL ROM code. If the code actually gets executed out of ROM, the ROM can potentially be configured within the same 128K block. The best advice is to copy code from ROM to system memory outside the block or to write code that does not enable 16-bit transfers.

The Host is provided with the ZWS signal in accordance with whether the memory cache can accommodate the transfer. The timing of this signal is dependent upon the width of the transfer being performed with the host.

To meet the memory bandwidth required by the ISA bus, it is necessary to implement the buffer memory

with fast (35 nsec) RAMs. For more details, see the AC timing specs in Section 10.

### 6.6 INTERRUPT REQUEST CONTROL LOGIC

There are two sources of interrupt requests to the host: the LAN Controller and a programmable bit (SINT) in the ICR register. The LAN controller section provides for the masking, polling, and clearing of its individual interrupt conditions. The sum of the masked LAN interrupt conditions is 'OR'ed with the programmable interrupt from the host interface section (SINT) and gated by the EIL bit from the ICR register prior to turning on one of the seven program-selectable tri-state drivers. The driver selection is made via bits in the GCR register.

Interrupt disabling should be accomplished via the ICR.EIL bit, not by changing the interrupt level to '0', because during the transition from an active level to tri-state, false interrupts may be generated.

The Interrupt Request Control logic is depicted in Figure 6-5.

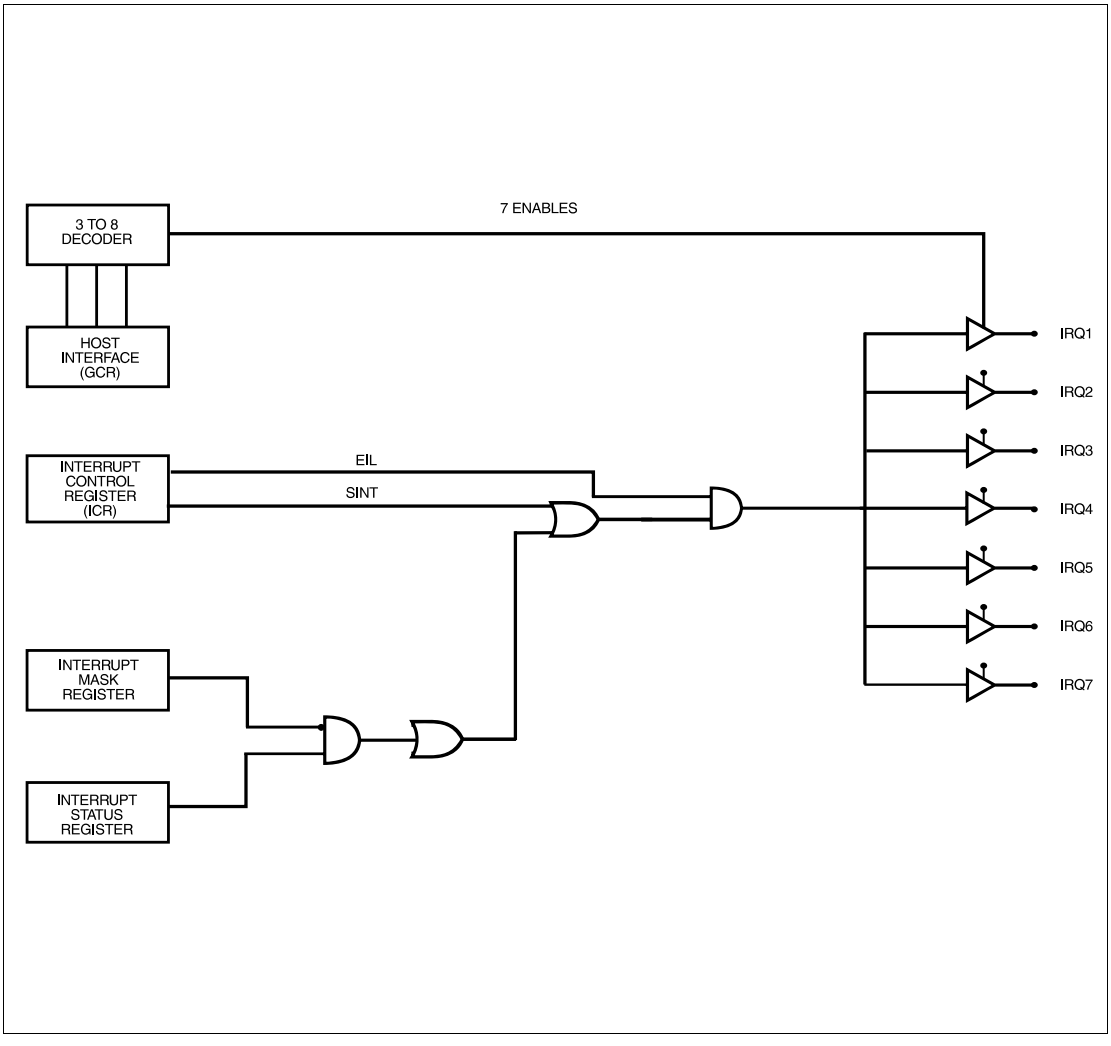


**FIGURE 6-5. INTERRUPT CONTROL LOGIC**

### 6.7 EEROM CONTROLLER AND ITS UTILIZATION

The 83C795 is designed to operate in conjunction with a serial EEROM memory that stores the configuration of the host interface and the permanently-assigned LAN station address. It can reduce the number of jumpers needed on a board and allows for reconfiguration without removing the board from the system. The EEROM is used to initialize some of the host interface configuration registers at reset time.

#### 6.7.1 Initialization Of 83C795

Activation of the reset pin forces the internal state of the 83C795 to a known value. There is a group of option bits that can be configured by attaching resistive pull-downs to four of the memory address output lines (MA09-MA00). These pins are sometimes referred to as the INIT pins. They still perform their function as memory address lines but also act as input pins during the RESET process. Each pin has a high impedance internal pull-up resistor that causes the pin to read as '1' unless a lower-impedance external pull-down resistor brings the natural state of the line to a logic '0' level. It is expected that an application of this chip would use a set of jumpers to selectively connect these pins to the external pull-downs, as shown in Table 6-2 below.

| Jumpers | Pins | Effect |
|---|---|---|
| *These jumpers are installed to mark 0 bits* | | |
| JMP0-3 | MA0-3 | EEROM Config. field, bits 0-3. |
| *These jumpers are installed to activate features* | | |
| JMP4 | MA4 | Switching PS HIDUTY option. |
| JMP5 | MA5 | Switching PS output from GPOUT. |
| JMP6 | MA6 | Plug and Play logic enable. |
| JMP7 | MA7 | NEC PC-98 Bus support. |
| JMP8 | MA8 | Drive 20 MHz clock out $\overline{\text{TLED}}$ pin. |
| JMP9 | MA9 | Use 83C790 Chip ID field instead of the 83C795 ID field. |

**TABLE 6-2. JUMPER EXAMPLE**

At the end of reset, these INIT pins are sampled and latched. One of these combinations determines whether the EEROM is read into the host interface registers. Ordinarily, the 83C795 loads its configuration registers from EEROM, but selection of one special combination of jumpers (when all INITs are pulled down) provides a means of bypassing the EEROM load to allow rapid simulation and testing of the device. Utilize the bypass mode to produce a less expensive adapter design which does not retain configuration or address permanently.

There is also a means of accelerating the EEROM clock and associated EEROM interface pins (EEDO, EECS, LLED, $\overline{\text{RLED}}$) for test purposes. LLED is connected to the EEROM clock pin (EESK) and is normally the primary clock (20 MHz) divided down by 128 during EEROM accesses. The IOR and IOW pins are latched on the rising edge of RESET. If both are active (low) at the same time, the clock accelerates to 10 MHz. To restore the normal 156 kHz clock rate, the chip is reset without activating IOR or IOW. $\overline{\text{RLED}}$ is connected to the EEROM's data-in pin, EEDI.

Unless bypassed, the EEROM data is read automatically into the host interface registers just after the 83C795 is reset. This takes approximately 2 milliseconds. During this time, memory, ROM, and register access is disabled. I/O accesses to any host interface register return garbage except for bit 6, which will return '1' until the registers are loaded at which time bit 6 returns '0'. To determine when the initial load of registers is completed, poll the EER Register. If the RECALL bit (Bit 6) equals 0, the initial load is complete.

The registers start out with their reset values and are changed one register at a time as the EEROM is read out. The buffer memory is always disabled upon reset and must be enabled by software.

The first time an EEROM is powered up, it has random data. The 83C795 can be accessed at known initial addresses if a special setting of the INIT pins is chosen when RESET pin is active. This is explained in more detail in the following section.

### 6.7.2 Retrieval And Storage Of Host Configuration Registers

#### 6.7.2.1 EEROM Interface Overview

An external 9356 serial EEROM is used to store up to 256 bytes of data. It takes about 2 ms to read all 16 registers after the end of the reset pulse. It takes about 200 ms to store the EEROM. The LAN Controller should not be online (transmitting or able to receive) while EEROM recall or store operations are ongoing, nor should any of the registers in the LAN controller or host interface section be accessed during that time. Unpredictable results may occur because the internal data busses will be supporting the data movement to or from the EEROM during that time interval.

An exception to this rule is made in the case of the EER register, which may be polled to determine when the recall or store operation completes. When the EER register is read, the EER.STO and EER.RC bits are visible to the host. Other bits from that register are meaningful only when there is no ongoing EEROM operation.

All 256 bytes of EEROM can be written to and read from. They are read into the LAN Address registers 8 bytes at a time. Once there, they can be changed and stored.

The EEROM controller can be operated under program control to do partial retrievals from and save configuration data into the EEROM.

EEROMs have a limited number of storage cycles. The store operation should only take place at initial board configuration or at initial installation in a customer's computer.

#### 6.7.2.2 EEROM Recall Operation Details

All recalls from EEROM into host interface registers are made in groups of either 8 or 16 registers. The choices are programmed into the EER register in the host interface section according to Table 6-3.

| Reset | Recall | Action |
|---|---|---|
| 0 | 0 | No Recall |
| 0 | 1 | Recall from bank 'EA' into LAN ADDR registers |
| 1 | X | Recall from bank '6' into LAN ADDR registers and from bank 'INIT' into configuration registers. |

**TABLE 6-3. EEROM RECALL OPERATIONS**

The recall of host interface configuration registers (addresses 08-0Fh, SWH=1) are done from the bank selected by 4 INIT jumpers. Table 6-4 defines which bank of configuration registers corresponds to each arrangement of the INIT pins.

| INIT: MA3-0 | Bank | Notes |
|---|---|---|
| 0000 | NONE | All pins jumpered to ground. This is the bypass condition. |
| 0001 | 14 | |
| 0010 | 13 | |
| 0011 | 12 | |
| 0100 | 11 | |
| 0101 | 10 | |
| 0110 | 9 | |
| 0111 | 8 | |
| 1000 | 7 | |
| 1001 | 6 | Initial recall defaults:<br>I/O = 280h<br>ROM= disabled at C0000h<br>RAM = 8K at C8000h<br>INT = 0 (no interrupt mapped)<br>LIT = disabled, GPOUT = 0<br><br>Configuration Registers (IAR, RAR, BIO, GCR, GCR2) are not recalled after this RESET. |
| 1010 | 5 | |
| 1011 | 4 | |
| 1100 | 3 | |
| 1101 | 2 | |
| 1110 | 1 | |
| 1111 | 0 | No jumpers are attached to pins. |

**TABLE 6-4. CONFIG REGISTER/INIT PINS**

The recall of LAN address registers are from the bank selected by the 4-bit EA field and are written to registers at I/O locations 08H-0FH, SWH=0. Upon RESET, the EA field points to Bank 6. Unless the EEROM load is bypassed by jumpers (INIT = 0000), the 83C795 recalls either 8 or 16 registers. When Config #6 is jumpered, only 8 registers are recalled and the hardware defaults are used for the configuration registers. All other initial configurations result in 16 register loads. These are done automatically. Since EA always initializes to Bank 6, the initial load from the EEROM always pulls in the LAN address from same location.

Refer to Figure 6-6 for a depiction of the EEROM register logic.

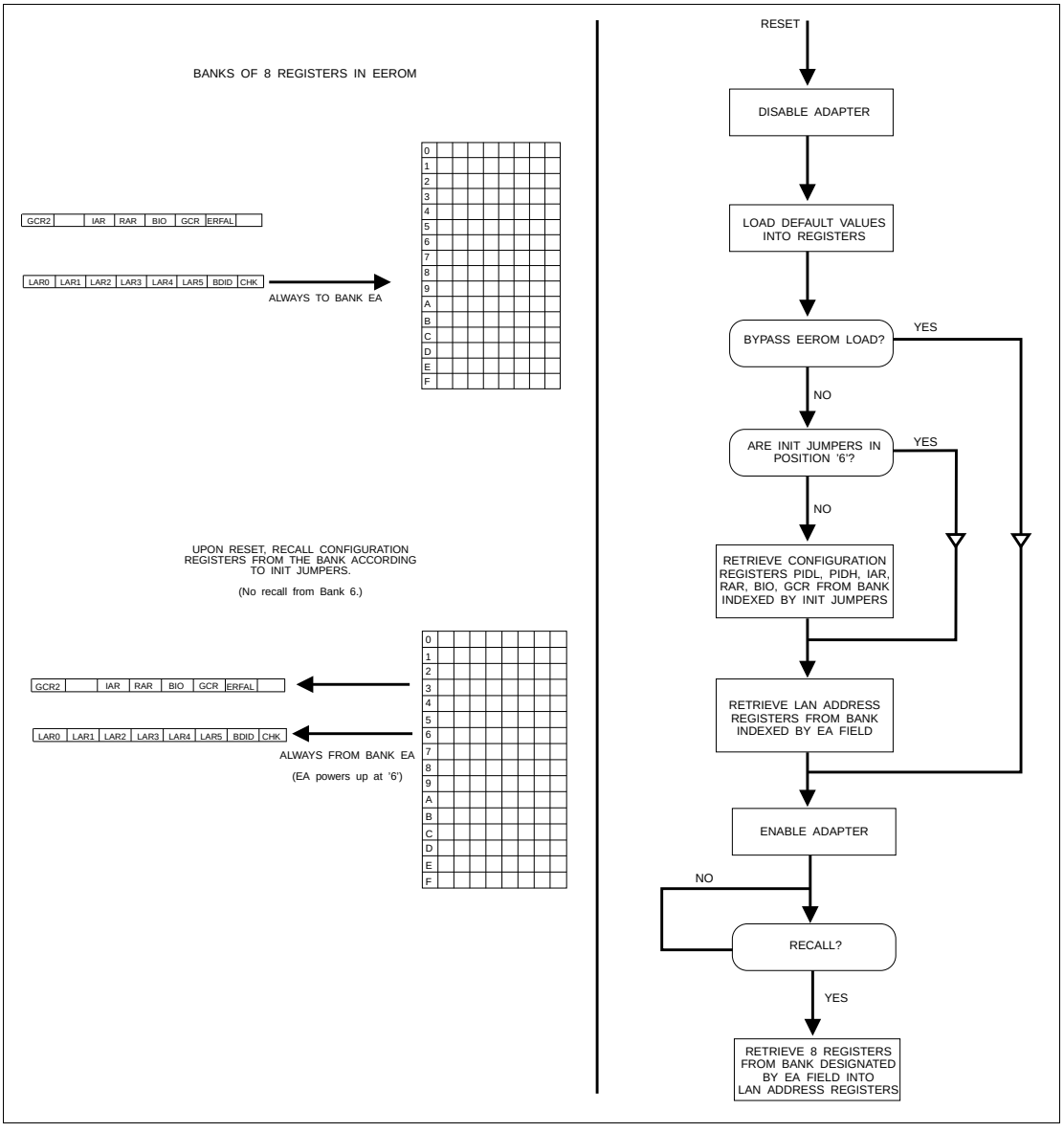


**FIGURE 6-6. EEROM REGISTER LOGIC**

#### 6.7.2.3 Storage Operations

The Store operation only copies the 8 LAN address registers to the EEROM. This does not depend upon INIT jumper settings. The store operation always moves the LAN Address registers into the bank selected by the 4-bit EA field.

Table 6-5 defines how each bank of 8 EEROM locations has been allocated:

| BANK | ALLOCATION |
|---|---|
| 00 | USER PROGRAMMABLE ('SOFT') CONFIGURATION |
| 01 | CONFIGURATION 1 |
| 02 | CONFIGURATION 2 |
| 03 | CONFIGURATION 3 |
| 04 | CONFIGURATION 4 |
| 05 | CONFIGURATION 5 |
| 06 | PERMANENT LAN ADDRESS |
| 07 | CONFIGURATION 7 |
| 08 | CONFIGURATION 8 |
| 09 | CONFIGURATION 9 |
| 0A | RESERVED FOR DRIVER CONFIGURATION STORAGE |
| 0B | CONFIGURATION 11 |
| 0C | CONFIGURATION 12 |
| 0D | CONFIGURATION 13 |
| 0E | CONFIGURATION 14 |
| 0F | RESERVED FOR DRIVER CONFIGURATION STORAGE |
| 10-1F | PNP DATA |

**TABLE 6-5. EEROM LOCATION ALLOCATION**

Unlocking the EEROM For Write Operations

This convoluted method is used to protect the EEROM from getting accidentally erased by someone else's software.

The EEROM register (EER) looks like this:

| BIT | EER WRITE | RESET |
|---|---|---|
| 7 | STO | 0 |
| 6 | RC | 0 |
| 5 | EA4 | 0 |
| 4 | UNLOCK | 0 |
| 3 | EA3 | 0 |
| 2 | EA2 | 1 |
| 1 | EA1 | 1 |
| 0 | EA0 | 0 |

The scheme requires a sequence of three writes into the EEROM register to start a store operation. The first two writes are required to set the UNLOCK bit and the third write executes the store.

The 1st write must be:

STORE=0, RECALL=0, EA4=X, UNLOCK=0, EA[3–0] = Ch.

The 2nd write must be:

STORE=0, RECALL=0, EA4=X, UNLOCK=1, EA[3–0] = Ah.

The 3rd write is:

STORE=1, RECALL=0, UNLOCK=0, EA[4–0] = EEROM bank.

The value for the EA4-EA0 field is the EEROM bank you intend to store in it.

The writing of any other value to the EER other than the required ones in the sequence will set the locking mechanism back to its initial condition and all three writes will once again be required in sequence to unlock the write protection. Intervening I/O operations to other registers and reads of the EEROM register do not affect the unlock sequence.

Storage of User-Defined Initial Configurations

To define any of the 15 recallable configurations, follow these steps.

1. Build an image of the desired configuration registers in the LAN register bank (SWH=0, addr=8:F).
2. Write the unlock/store sequence with the final EA field of the EEROM register selecting the desired configuration bank. Do not use Bank 6 because that is reserved for the permanent device LAN address. Do not use Bank 10 either because that is reserved for driver-related information storage and cannot be recalled as a board configuration.
3. Wait about 200 msec or poll the STORE bit to determine whether that operation has completed.

Storage of User-Defined LAN Address

To store a user-defined LAN address, follow these steps.

1. Program the desired LAN address, board ID register, and checksum register with desired values.
2. Write the unlock store sequence with the final EA field of EEROM register selecting the Bank 6.
3. Wait about 200 msec or poll the STORE bit to determine that operation has completed.

Storage of User-Defined Data

In some applications, there may be other data you need to save in the EEROM, like board type and revision numbers, multicast filter acceptance mask, software driver parameters, and the host machine type. Since some EEROM locations may not be needed for configuration or LAN address storage, they can be used for this purpose.

1. Overwrite LAN address, board ID, and checksum registers with the data you need to save.
2. Store the data as if it were a board configuration. The driver setup program uses this method to store driver related parameters into EEROM Bank 10.

To recall this user defined data, program the EA field for the bank and do a Recall operation.

## 6.8 PLUG AND PLAY

The 83C795 supports the Plug and Play ISA Specification. This specification provides full and interactive configuration of all PnP-compliant boards installed on the ISA bus. The essential steps in this process are the abilities to:

- Power up and reset Plug and Play devices
- Send out an initiation key to bring all PnP devices into a known state
- Isolate each ISA adapter in turn
- Read the adapter's resource requirements
- Arbitrate the available resources to all of the PnP cards
- Identify each board and configure its resources
- Activate the cards on the ISA bus
- Locate a suitable driver for the adapter, if necessary

To be effective in this new environment, the 83C795 contains the logic necessary to prepare any board on which it is placed for Plug and Play standards. This logic will be active only when both the PNPEN bit (ERFAL.0 as read from EEROM) is set and a 3.6kΩ resistor is connected to the MA[6] pin (JMP6) between MA6 and GND.

This section contains a brief overview of how Plug and Play works, along with information specific to the 83C795's implementation of PnP. For more detail on the Plug and Play process and protocol, please refer to the latest version of the *Plug and Play ISA Specification*.

### 6.8.1 Auto-Configuration Ports

The Plug and Play protocol requires that all logic not related to PnP on a PnP board not respond to any ISA bus access until the PnP logic activates the card, except for devices required for boot (see Section 6.8.5). Until this happens, the only access to the card is through the PnP auto-configurations ports. These ports consist of three 8-bit I/O registers, as shown in Table 6-6.

| Port Name | Location | Type |
|---|---|---|
| ADDRESS | Fixed at 0279h (LPT2 Status Port) | Write-only |
| WRITE_DATA | Fixed at 0A79h (LPT2 Status Port + 0800h) | Write-only |
| READ_DATA | Relocatable in range 0200h – 03FFh | Read-only |

**TABLE 6-6. AUTO-CONFIGURATION PORTS**

All writes to the ADDRESS port are stored in an address register. The value in this register is used as an index into the internal 256-byte range that holds the PnP configuration registers. Any accesses to the WRITE_DATA or READ_DATA port will be to the PnP configuration register that is currently indexed by the address register. See section 6.8.3 for details on the configuration registers.

### 6.8.2 Plug And Play States

The main Plug and Play state machine contains four states and is illustrated in Figure 6-7.

All cards will enter the WaitForKey state in response to either a power-up reset or the Reset and Wait for Key commands. No commands are active in this state until the Initiation Key is detected on the ISA bus. The WaitForKey state is the default state for Plug and Play cards during normal system operation. The Initiation Key places the Plug and Play state machine into the Sleep state. The Initiation Key consists of a predefined series of writes to the ADDRESS port. The write sequence is decoded by on-card logic. If the proper sequence of I/O writes is detected, the Plug and Play auto-configuration ports are enabled. The sequence of writes that are expected is:

```
0x6A,0xB5,0xDA,0xED,0xF6,0xFB,0x7D,0xBE,
0xDF,0x6F,0x37,0x1B,0x0D,0x86,0xC3,0x61,
0xB0,0x58,0x2C,0x16,0x8B,0x45,0xA2,0xD1,
0xE8,0x74,0x3A,0x9D,0xCE,0xE7,0x73,0x39
```

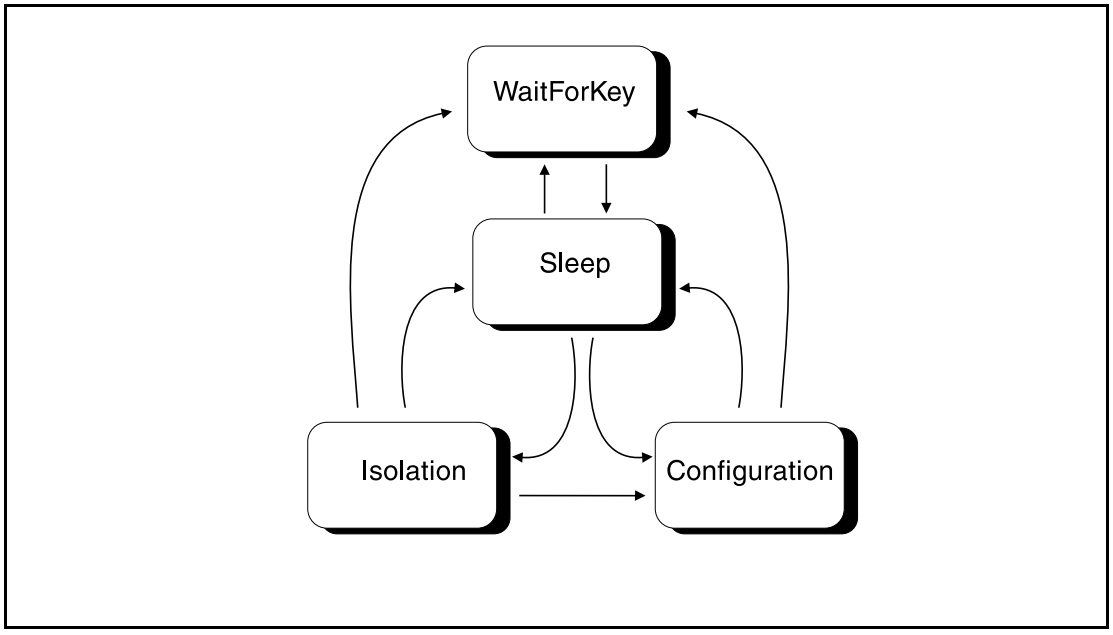


**FIGURE 6-7. PLUG AND PLAY STATE MACHINE**

Any writes to the ADDRESS port that do not match the Initiation Key sequence will cause the logic to reset back to the start of the Key. While the PnP state machine is in the WaitForKey state, all access to the READ_DATA or WRITE_DATA ports is disabled.

Once the state machine is in the Sleep state, the board responds to a WAKE[CSN] command. Each PnP board has a register to store a Card Select Number, and this register contains a zero at power-up. The card responds to a WAKE[CSN] command only if the CSN in the command matches the value in the Card Select Number register. If the card's CSN does not match the CSN in the WAKE[CSN], it goes into the sleep state. The card stays in the sleep state until awakened by the correct WAKE[CSN]. If the CSN is zero, then the card enters the Isolation state, otherwise it moves into the Configuration state.

#### 6.8.2.1 Isolation

A simple algorithm is used to isolate each Plug and Play card. The isolation protocol requires that each card contain a unique number, referred to as the *serial identifier*. This is a 72-bit number composed of two 32-bit fields and an 8-bit checksum. The first field is a vendor identifier. The second can be any value - for example, a serial number or part of a LAN address. The number is accessed serially, bit by bit, by the isolation logic and is used to differentiate the adapters.

The PnP software begins the isolation by sending out a Wake[0] command. This will cause all PnP cards that have not been isolated to transition to the Isolation state. If this is the first pass through the protocol, then the software will pick an initial location for the READ_DATA port at this time. The software then issues two reads to the ISOLATION register for each bit in the serial identifier. If the current bit is a '1', then the PnP board is expected to return 0x55 on the first read, and 0xAA on the second. If the current bit is a '0', then the PnP logic will drive nothing on the bus, but will instead observe the bus to see if another card is driving the 0x55, 0xAA pattern. If it does see that pattern for the two reads, then that card must put itself back into the Sleep state. This ensures that only one PnP card will be in the Isolation state at the end of the protocol.

The PnP software will recognize the 0x55, 0xAA pattern as a '1' for that bit position, and any other pattern as a '0'. Once all 72 bits have been read, the PnP software will then verify that the checksum matches the data. If it does, then the software will assign a unique, non-zero CSN to the one card that made it to the end of the protocol. This will cause that one card to transistion to the Configuration state. If the checksum does not match, then the software will move the location of the READ_DATA port, and start the protocol over.

#### 6.8.2.2 Configuration And Activation

The Isolation protocol ensures that only one card can be in the Configuration state at a time. This makes it possible to read out the resource string byte-serially when in this state. This is done through the Resource_Data Register (location 0x04), after polling the status bit (location 0x05, bit 0) to make sure the data in the register is valid. The PnP software will use this method to read the entire resource string from the PnP card. This string lists the resource requirements of the card, along with what the card is capable of using (see section 6.8.5 for more on the resource string). The software repeats this process with all of the PnP cards in the system, and thus obtains an image of all of the resource requirements in the system. The software then arbitrates the available resources to meet the needs of each card. If a configuration can be found, then the assigned configuration for each card will be written to the cards. The software then activates the card by setting the Activate bit (location 0x30, bit 0). On the 83C795, this causes the software to transfer the appropriate settings in the PnP configuration registers to the 83C795's configuration registers by way of a shift chain. Once this transfer is complete, the rest of the logic in the 83C795 becomes active.

### 6.8.3 Configuration Registers

Figure 6-8 contains a map showing all of the configuration registers implemented by the 83C795. This figure also illustrates the relationship between the auto-configuration ports, the PnP secondary address space, and the resource string.

Most of the configuration registers can only be accessed when the PnP state machine is in certain states. Any unused registers or bits in the PnP register space must return zeros when read. The

register map shown in Table 6-7 changes when the adapter is run in I/O-mapped mode (see section 6.8.6). Also, when the ROM space is disabled (when BIO.4 and BIO.5 are both set), the ROM configuration registers become read-only and always read zeros.

There are 22-bits of information in the PnP configuration registers that are used for configuring the 83C795. These bits are shifted from the PnP section to the 83C795 section whenever the chip is activated. The same bits are shifted from the 83C795 section back to the PnP section at the end of the initial EEROM load (to get the default values into the PnP registers). Some of these bits may have to be remapped before they are shifted so as to match the formats of the two different register

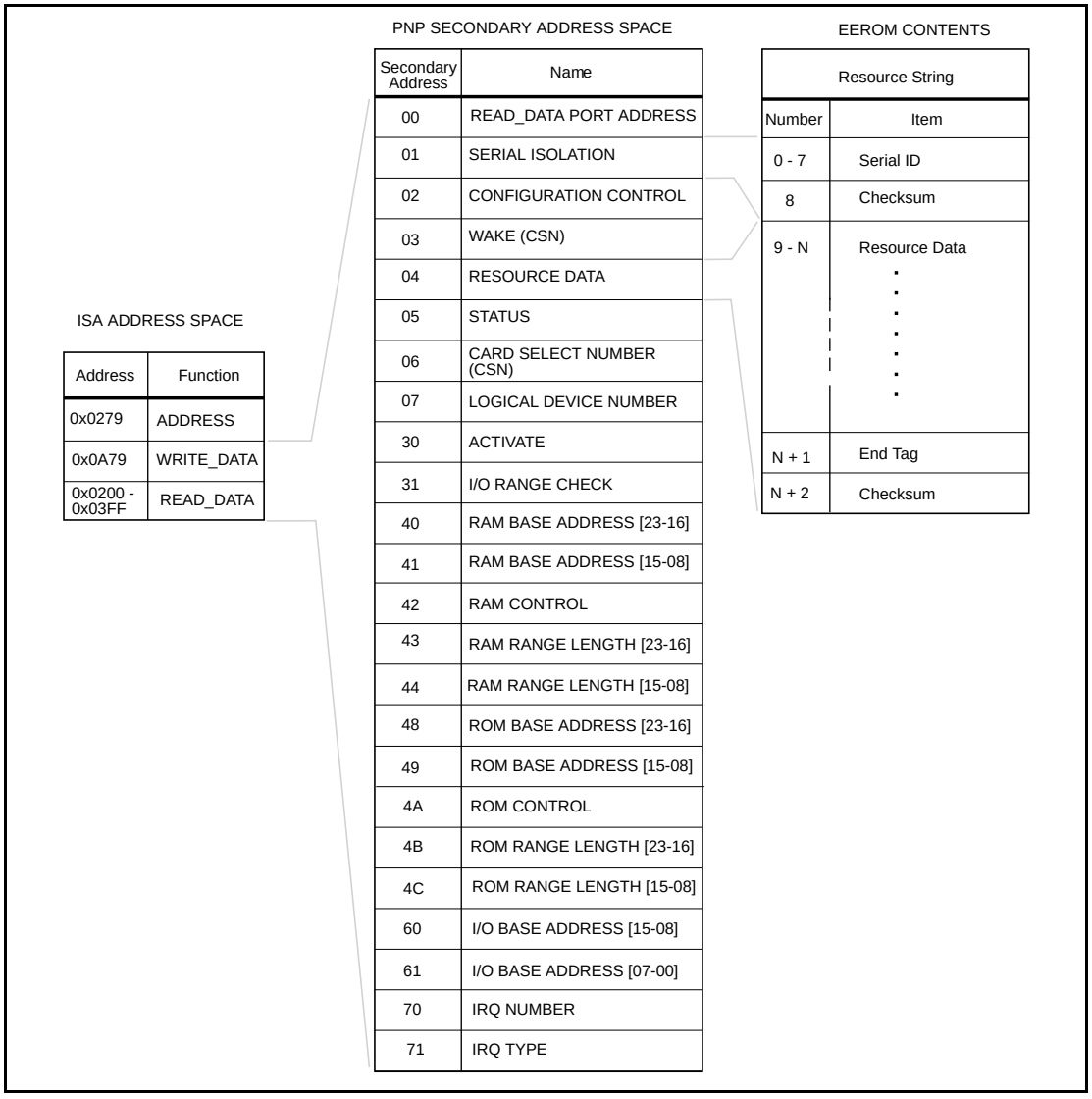


**FIGURE 6-8. PLUG AND PLAY CONFIGURATION REGISTERS**

| Field Name | No. | PnP Bit Location | 795 Bit Location | Mapping Function |
|---|---|---|---|---|
| | | | | PNP Bits / 795 Bits |
| RAM Base | 1 | 0x40.1 | RAR.6 | direct mapping |
| | 2 | 0x40.0 | RAR.3 | |
| | 3 | 0x41.7 | RAR.2 | |
| | 4 | 0x41.6 | RAR.1 | |
| | 5 | 0x41.5 | RAR.0 | |
| RAM Size | 6 | 0x44.6 | RAR.5 | |
| | 7 | 0x44.5 | RAR.4 | |

| 6 | 7 | RAR.5 | RAR.4 |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| ROM Base | 8 | 0x48.1 | BIO.6 | direct mapping |
| | 9 | 0x48.0 | BIO.3 | |
| | 10 | 0x49.7 | BIO.2 | |
| | 11 | 0x49.6 | BIO.1 | |
| | 12 | 0x49.5 | BIO.0 | |
| ROM Size | 13 | 0x4C.6 | BIO.5 | |
| | 14 | 0x4C.5 | BIO.4 | |

| 13 | 14 | BIO.5 | BIO.4 |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| I/O Base | 15 | 0x60.0 | IAR.4 | direct mapping |
| | 16 | 0x61.7 | IAR.3 | |
| | 17 | 0x61.6 | IAR.2 | |
| | 18 | 0x61.5 | IAR.1 | |

| IRQ Level Used | 19 | 0x70.3 | GCR.6 | 19 | 20 | 21 | 22 | GCR.6 | GCR.3 | GCR.2 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 20 | 0x70.2 | GCR.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 21 | 0x70.1 | GCR.2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| | 22 | 0x70.0 | | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| | | | | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| | | | | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| | | | | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| | | | | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | | | | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**TABLE 6-7. PLUG AND PLAY BIT REMAPPING**

sets. These bits are listed in Table 6-7, along with any necessary mapping information.

### 6.8.4 Resource String

The Plug and Play resource string is stored in EEROM and can be accessed either bitwise through the Serial Isolation register or byte-serially through the Resource Data register. This string should completely describe the resource needs and options for a 83C795 based card.

A sample version of this structure is listed in Table 6-8. Once the 83C795 has been activated by the PnP logic, the resource string is accessed in the

| Byte | Value | Swapped Value | Name | Description |
|---|---|---|---|---|
| 00 | 4D | B2 | Vendor_ID.0 | Serial Identifier |
| 01 | A3 | C5 | Vendor_ID.1 | |
| 02 | 84 | 21 | Vendor_ID.2 | |
| 03 | 16 | 68 | Vendor_ID.3 | |
| 04 | C0 | 03 | LAN_Addr.0 | |
| 05 | 01 | 80 | LAN_Addr.1 | |
| 06 | 23 | C4 | LAN_Addr.2 | |
| 07 | 45 | A2 | LAN_Addr.3 | |
| 08 | 4C | 32 | Checksum | |
| 09 | 0A | 50 | PnP_Version_Type | Plug and Play Version Descriptor |
| 0A | 10 | 08 | PnP_Version | |
| 0B | 10 | 08 | Vendor_Version | |
| 0C | 82 | 41 | ANSI_Identifier_Type | ANSI Descriptor for Card |
| 0D | 0D | 30 | Descriptor_Length [7-0] | |
| 0E | 00 | 00 | Descriptor_Length [15-8] | |
| 0F | 53 | CA | ANSI Character | – S |
| 10 | 4D | B2 | ANSI Character | – M |
| 11 | 43 | C2 | ANSI Character | – C |
| 12 | FF | FF | ANSI Character | – |
| 13 | 38 | 1C | ANSI Character | – 8 |
| 14 | 34 | 2C | ANSI Character | – 4 |
| 15 | 31 | 8C | ANSI Character | – 1 |
| 16 | 36 | 6C | ANSI Character | – 6 |
| 17 | FF | FF | ANSI Character | – |
| 18 | 43 | C2 | ANSI Character | – C |
| 19 | 61 | 86 | ANSI Character | – a |
| 1A | 72 | 4E | ANSI Character | – r |
| 1B | 64 | 26 | ANSI Character | – d |

**TABLE 6-8. PLUG AND PLAY RESOURCE STRING STRUCTURE**

| Byte | Value | Swapped Value | Name | Description |
|---|---|---|---|---|
| 1C | 15 | A8 | Logical_Device_Type | Logical Device Descriptor |
| 1D | 4D | B2 | Logical_Device_ID.0 | |
| 1E | A3 | C5 | Logical_Device_ID.1 | |
| 1F | 84 | 21 | Logical_Device_ID.2 | |
| 20 | 16 | 68 | Logical_Device_ID.3 | |
| 21 | 02 | 40 | Logical_Device_Flags | (Non-boot, does I/O range checking) |
| 22 | 23 | C4 | Interrupt_Descriptor_Type | Interrupt Format Descriptor |
| 23 | A8 | 15 | IRQ_Mask [7-0] | – Interrupts supported: 3, 5, 7 |
| 24 | 8E | 71 | IRQ_Mask [15-8] | – Interrupts supported: 9, 10, 11, 15 |
| 25 | 16 | 68 | IRQ_Information | |
| 26 | 47 | E2 | I/O_Port_Descriptor_Type | I/O Port Descriptor |
| 27 | 00 | 00 | I/O_Port_Info | |
| 28 | 00 | 00 | I/O_Min_Base_Addr [7-0] | |
| 29 | 02 | 40 | I/O_Min_Base_Addr [15-8] | – Minimum Base Address = 0x0200 |
| 2A | E0 | 07 | I/O_Max_Base_Addr [7-0] | |
| 2B | 03 | C0 | I/O_Max_Base_Addr [15-8] | – Maximum Base Address = 0x03E0 |
| 2C | 20 | 04 | I/O_Base_Alignment | – Alignment = 32-byte blocks |
| 2D | 20 | 04 | I/O_Range_Length | – Length = 32 bytes |
| 2E | 81 | 18 | Memory_Range_Descriptor_Type | Memory Range Descriptor #1 |
| 2F | 09 | 90 | Descriptor_Length [7-0] | (two would be required for cards using |
| 30 | 00 | 00 | Descriptor_Length [15-8] | both ROM and shared RAM) |
| 31 | 11 | 88 | Memory_Range_Information | – 8-/16-bit, writable |
| 32 | 00 | 00 | Mem1_Min_Base_Addr [15-8] | |
| 33 | 0C | 30 | Mem1_Min_Base_Addr [23-16] | – Minimum Base Address = 0x0C0000 |
| 34 | E0 | 07 | Mem1_Max_Base_Addr [15-8] | |
| 35 | 0F | F0 | Mem1_Max_Base_Addr [23-16] | – Maximum Base Address = 0x0FE000 |
| 36 | 00 | 00 | Mem1_Base_Alignment [7-0] | |
| 37 | 40 | 02 | Mem1_Base_Alignment [15-8] | – Alignment = 16-kbyte blocks |
| 38 | 40 | 02 | Mem1_Range_Length [7-0] | |
| 39 | 00 | 00 | Mem1_Range_Length [15-8] | – Length = 16 kbytes |
| 3A | 79 | 9E | End_Tag_Type | End Tag |
| 3B | 0E | 70 | Checksum | – Covers all data bytes |

**TABLE 6-8. PLUG AND PLAY RESOURCE STRING STRUCTURE (cont.)**

same manner (for both reads and writes) as the rest of the EEROM contents, except that EA[4] (EER.5) must be set.

### 6.8.5 Configuring As A Boot Card

Many LAN adapters are configured as boot cards. Since these cards must be visible to the BIOS at boot time, they may have to be activated before the PnP software has had a chance to run. The PNPBOOT bit (IAR.0) was implemented to do this. When this bit is read out of EEROM as set, the chip becomes active on the ISA bus. This bit also forces bit 0 of the IAR register to '1' which identifies the card as a boot card to the PnP software.

### 6.8.6 Configuring With An I/O-Mapped Pipe

When the 83C795 is configured with an I/O-mapped pipe instead of shared memory, the board will not use any memory address space for the buffer RAM. Therefore, the PnP RAM control registers (usually locations 40h–44h) are not necessary. If this card requires a ROM, then the descriptor for the ROM memory window will be the first one in the resource string. This requires that the PnP ROM control registers be re-mapped from 48h-4Ch to 40h-44h. This is accomplished by setting the PNPIOP bit (GCR2.0). This bit must be read out of EEROM because it must have the correct value before the chip is activated by the PnP logic. Setting this bit also causes locations 48h-4Ch to read as zeroes.

### 6.8.7 Buffer Memory Limitations

Normally, buffer memory can be located above the 1MB DOS limit by setting the buffer address line (LA23-LA20) decoder to match at 'F' rather than zero (done by setting the HRAM bit, RAR.7). This is not supported by the PnP hardware as implemented in the 83C795.

## 6.9 EXTERNAL POWER SUPPLY CONTROL

The GPOUT pin can be used to control an external power supply supporting a 10Base2 MAU circuit. This pin can be used to source either a simple DC control signal or a pulse train. The DC signal is used to enable or disable a controllable power supply.

**Note**

The DC signal's polarity on the 83C795 is the opposite of the 83C790's.

The pulse train is designed to be the switching control signal for a specific design of switching power supply. The pulse train includes a start-up sequence as well as a choice of two normal operating pulse trains.

Before the GPOUT pin can emit a pulse train, install the INIT5 jumper to pull down the MA5 pin. The INIT4 jumper determines which of the two pulse trains is emitted. The pulse train is turned on or off using the GCR.GPOUT bit (see page 21 for details).

The pulse train is a 312 KHz signal with 1/8 duty cycle for the first 1024 clocks after GPOUT is enabled. This is followed by 1/4 duty cycle for 1K clocks, 3/8 duty cycle for 1K clocks, and 1/2 (50%) duty cycle thereafter.

Installation of the INIT4 jumper causes the final duty cycle to be 17/32 (53%) instead of 1/2.

## 7.0 LAN CONTROLLER OVERVIEW

The LAN Controller consists of 3 basic blocks: DMA controller, transmitter, and receiver. Each of these blocks consists of sub-sections. The DMA controller includes a memory interface unit, control registers, and a micro-coded sequencer that handles data buffering for the transmitter and receiver sections.

The transmitter block has a MAC (Media Access Control) section that performs the IEEE 802.3 transmission protocol and a Physical Layer Interface (PLI) section that does Manchester encoding and drives the cables.

The receiver block has a MAC section that performs the 802.3 reception protocol and a PLI section that converts line level differential signals to internal logic signals while doing clock recovery, and manchester decoding.

### 7.1 DMA CONTROLLER

The DMA controller handles data movement between the FIFOs and buffer memory for transmission and reception of frames. All DMA data traffic is 8-bit wide. One DMA controller is shared between the transmit and receive functions. The controller groups memory transfers into bursts of 8 bytes for both transmit and receive functions. The DMA controller always accesses memory by doing two single-byte transfers in a row. The burst size and its trigger levels are shown in Table 7-1.

| BURST | TRIGGER LEVEL RX | TX |
|---|---|---|
| 8 bytes | R ≥ 8 | T ≤ 8 |

**TABLE 7-1. DMA BURST LENGTH FIELD**

Though internally 8 bits wide, the DMA controller generates 16-bit addresses. It accesses memory in 2 cycles of the chip's master clock (per byte).

When conducting a loop-back test, this controller can handle full-duplex buffering of full length frames at serial data rates up to 10 Mbps. It does not handle the general case of independent (concurrent) transmit and reception processes.

#### 7.1.1 Assembly and Disassembly Latches

These latches are used to match up the internal 8-bit data path with the external data bus. Assembly latches build a 16-bit word out of two 8-bit words or supplies the consecutive bytes when interfacing to an 8-bit bus. Disassembly latches perform the inverse function. These are used during DMA operations and are bypassed when the chip's registers are written or read.

#### 7.1.2 Memory Interface Unit

The memory interface unit (MIU) transfers data from buffer memory to the internal disassembly latches and from the internal assembly latches to buffer memory. It is a part of the DMA controller. This block generates the memory strobes (RAMOE, RAMWR) when the DMA is accessing the buffer RAM.

MIU operation is initiated by the DMA controller after it sets up the address for the transfer and puts outgoing data (receiver functions) into the assembly latches. The MIU then performs the memory transfer in the next time slot assigned to the DMA.

The basic DMA cycles are in Figure 7-1. Real details can be found in the AC timing section.

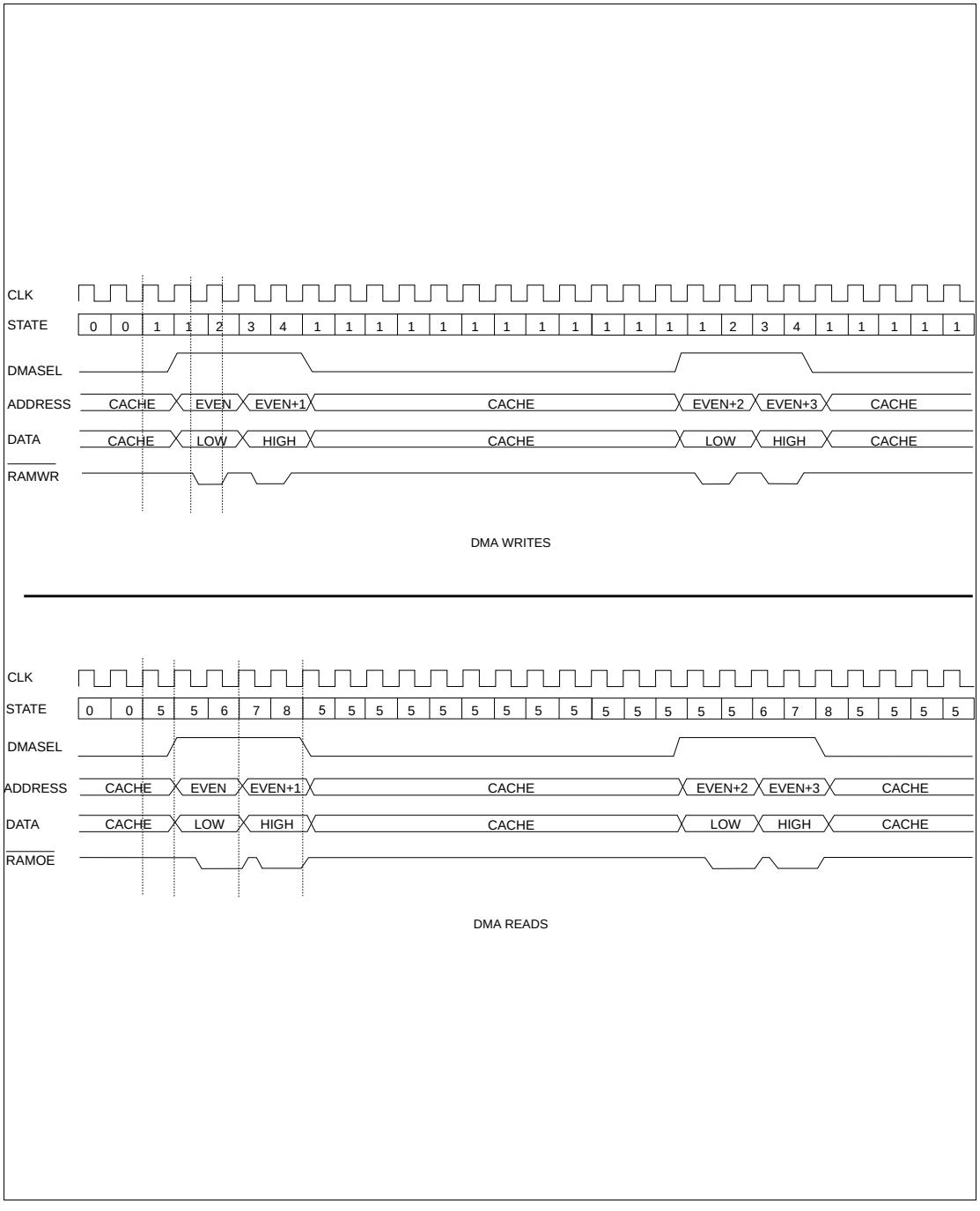


**FIGURE 7-1. BASIC DMA CYCLES**

### 7.1.3 LAN Controller Internal Bus Arbitration

This portion of the 83C795 is used to resolve conflicts that can occur on data buses used by both the DMA controller and host accesses to the internal registers of the LAN Controller section.

The LAN bus arbitration section observes a LAN select (CS) signal derived from the memory bus arbiter and provides a 'ready' handshake signal in return. It also controls internal data flow within the LAN controller and holds off the DMA microcontroller during I/O accesses.

### 7.1.4 DMA Microcontroller

The core of the DMA controller is a ROM-based microcontroller which includes an address counter for the memory position, comparators for internal address comparisons, some decrementers for loop control, registers for storage of operating variables, and I/O control signals that attach to many circuits within the LAN controller section of the chip.

In addition to the microcode associated with normal transmit, receive and loopback processes, there is additional code to facilitate testing of the LAN controller.

### 7.1.5 How to Access Registers

A request for LAN register access is made when the host presents an I/O address that decodes to a register within the upper 16 bytes of the 83C795's I/O block and a valid IOR or IOW is presented.

The chip will respond with an I/O Channel Not Ready signal (IORDY) while internal arbitration proceeds. It remains NOT IORDY until the desired transfer is ready to be completed.

Access to the registers of the LAN controller section is allowed after any ongoing DMA burst is completed. At that time, the DMA may wish to become active again in response to new needs, but the arbitration logic will allow host access to the chip until the I/O strobe becomes false. The arbiter generates the IORDY signal as an indication to the host that the internal bus has been made available and that the requested I/O access has been made. Between accesses to the chip, IORDY is undriven.

To read from a register, an I/O address is placed on the SAxx pins and IOR is asserted by the host (must be asserted after a valid address) and recognized by the bus arbitration logic which enables data flow from the addressed register to the D00-D07 pins. Register reads are always done through the D00-D07 pins, except for 16-bit I/O pipe accesses. The D08-D15 pins will be tri-stated during read operations. IORDY will indicate when the host may sample data and terminate the read operation.

To write to a register, an I/O address is placed on the SAxx pins and IOW is asserted by the host and recognized by the bus arbiter. Address must become stable before IOW is asserted. When the bus is free for the transfer, IORDY is asserted. Data is latched into an intermediate transfer latch with the trailing edge of IOW and then transferred to the destination register two clocks later.

This delayed write operation requires an internal recovery period between host accesses to registers. This period is documented in the detailed timing diagrams.

### 7.1.6 Memory Interface

The internal DMA controller moves packets between buffer memory and the FIFOs.

### 7.2 FIFOS

There are two FIFO structures used to expedite the transfer process between the DMA processor and the Transmit/Receive units. Each FIFO is 16-bytes deep. Associated with each FIFO are up/down counters that keep track of how full each FIFO is. The normal loading source for each FIFO receives an overflow indication when it occurs and the normal unloading destination has an empty signal to prevent drawing of nonexistent data.

### 7.3 RECEIVER NETWORK INTERFACE (PHY-TO-MAC)

This section of the 83C795 has squelched, differential receivers for the AUI and TWP interfaces. The inputs from these interfaces are multiplexed together with a feedback path to a common phase-locked-loop circuit for clock recovery and data decoding. Figure 7-2 shows the AUI and Twisted-Pair interface relationship.

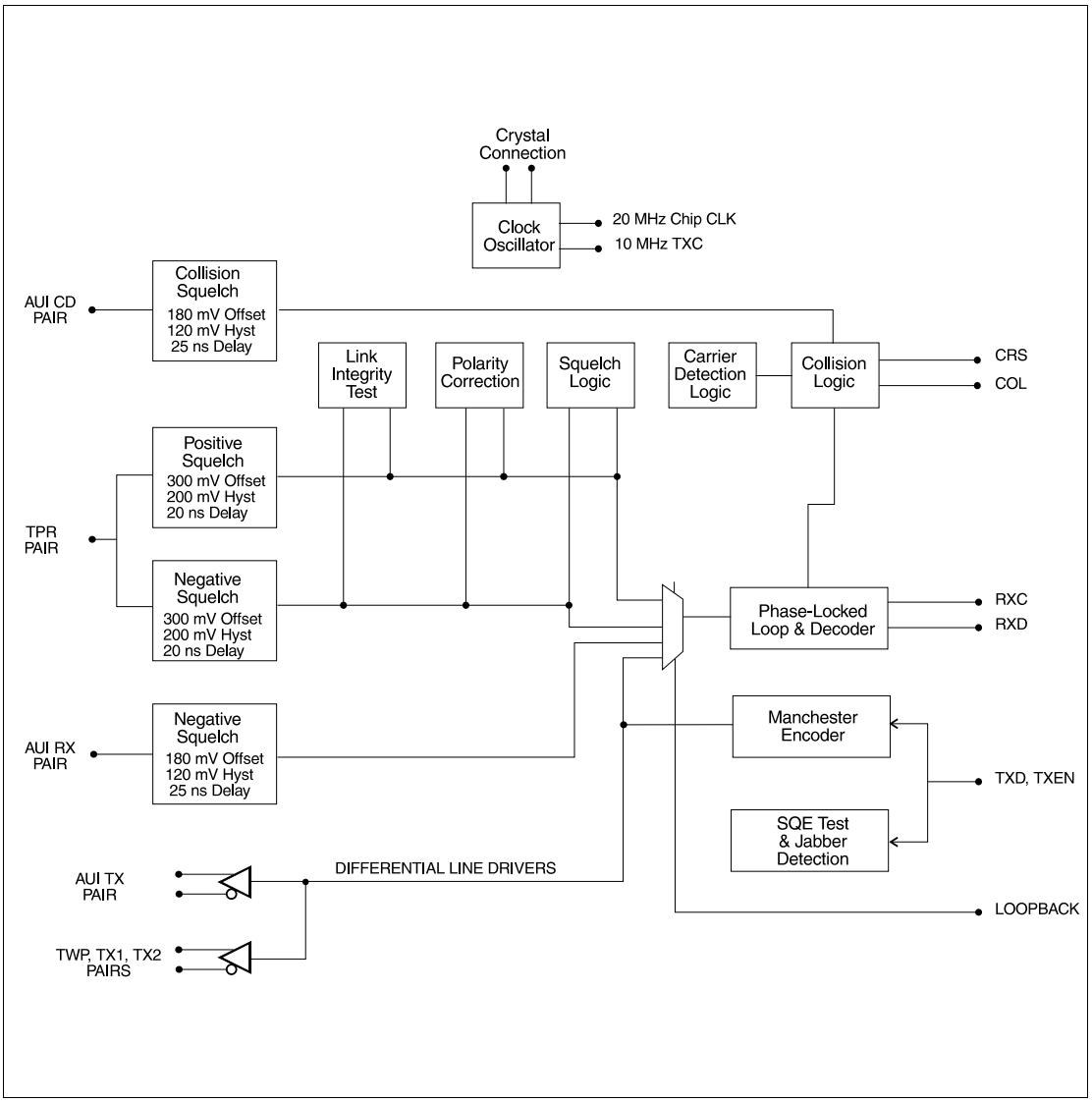


**FIGURE 7-2. AUI/TWISTED-PAIR INTERFACE**

### 7.3.1 AUI Differential Receiver

With the standard 78Ω transceiver AUI cable, the differential input must be externally terminated. This requirement may be satisfied by connecting two 39Ω resistors in series with one optional common mode bypass capacitor.

To prevent noise at the AUI RX+/- input from falsely triggering the decoder, a squelch circuit rejects signals with pulse widths less than 25 nsec (negative going), or with levels less than 175 mV.

When the input exceeds the squelch levels, the analog phase-locked loop locks into the incoming signal and Manchester decoding takes place. The internal carrier sense signal is activated and the receive data (RXD) and the receive clock (RXC) become available within seven bit times. At the end of a frame when normal mid-bit transition on the differential input ceases, carrier sense is deactivated. The receive clock remains active for an additional 5 bit times.

### 7.3.2 Twisted-Pair Differential Receiver

The received signal from the unshielded cable may be noisy, so minimum voltage and timing limits must be met before the receiver logic is enabled. A 'Smart Squelch' digital noise filter is used in addition to the analog squelch circuit in the receiver.

If the input polarity is reversed it will be automatically detected and corrected. When polarity is normal, the PLED (MANCH.3) bit will be set.

The phase-locked loop and Manchester decoder are the same circuits used by the AUI receiver.

#### 7.3.2.1 Extended Length For Twisted-Pair

Setting the XLENGTH bit (GCR.7) increases the squelch levels used by the TP receiver. This enables the adapter to be connected to a cable that is longer than specified by the 802.3 standard.

### 7.3.3 Manchester Decoder

Decoding is accomplished by an analog phase-locked loop that separates the Manchester-encoded data stream into clock signals and NRZ data. This loop can tolerate up to 20 nsec of jitter on each signal edge, exceeding the 802.3 requirements.

### 7.3.4 Carrier Sense

The AUI interface determines carrier presence by requiring the differential received signal to exceed the negative squelch level for a minimal period, nominally 25 nsec. The twisted-pair interface requires that the positive and negative squelch levels be exceeded several times for a period of at least 20 nsec.

When operating in loopback mode, internal transmit enable signal is returned to the MAC as the carrier indication.

### 7.3.5 Collision Detect

Collision detection for the AUI interface is indicated when the differential signal on the CD input pair exceeds the negative squelch threshold. Collision detection for the twisted-pair interface is indicated when there is carrier sensed while the transmitter is active.

### 7.3.6 Loopback Mode

When the LAN controller is programmed to operate in loopback mode 2, it provides a signal to the line receiver section which causes the Manchester decoder to derive its input from the encoded transmit data instead of the differential receivers of AUI and TP interfaces. The AUI and TP interfaces are ignored while the loopback indication is active.

## 7.4 MAC RECEIVER

### 7.4.1 Basic Functions

The 83C795's receiver section processes a serial stream of NRZ data. The start of the frame is identified, destination address is checked against the station address, recognized frames are transferred to memory and checked for valid formation. Error conditions are reported.

### 7.4.2 Interface to the MAC Receiver

The serial interface to the MAC receiver section is handled by four signals: Carrier Sense (XCRS), Collision Detect (XCOL), Receive Data (XRXD), and Receive Clock (XRXC). The group may come from one of three sources:

- internally from the Manchester Decoder
- direct from the pins
- internally from the transmit section.

All sources are treated equally. Selection of source is done by programming configuration registers. Refer to the TCON and MANDIS Registers for further information.

**Note**

These signals are multiplexed with the IRQ pins and are used for testing purposes only.

### 7.4.3 Loopback Paths

The 83C795 has 3 loopback modes.

Mode 1 provides the path between receiver and transmitter inside the LAN controller. In this mode, NRZ data from the transmitter connects to the receiver's RXD input, bypassing Manchester encoder/decoder. XRXC is generated internally for this mode by dividing 20 Mhz clock by two. The minimum frame length for mode 1 loopback is 25 bytes.

Mode 2 connects transmit and receive data through the Manchester encoder/decoder. The serial data is wrapped around just inside the device pins with neither AUI nor TP interfaces actually driving the outside world. The minimum frame length for mode 2 loopback is 25 bytes.

Mode 3 has transmitter and receiver pins active with loopback pins inactive. The DMA controller will run its special loopback code to allow reception of the outgoing transmission if it is echoed back. In this mode, the DMA can handle delay of the echo provided that the frame's length exceeds echo delay by at least 200 bit times (25 bytes). When run in this mode, the board being tested should be connected to an 802.3-compliant cable. That cable may or may not have other 802.3 nodes attached.

**Note**

Caution is advised when running this test on a live network or when other nodes on the test cable could send a frame to the node under test. Reception of a frame destined to the loopback node could confuse the results of the loopback test as the node will be able to receive the incoming frame.

### 7.4.4 Receive Deserialization

Receive deserializer is activated by carrier sense. Byte alignment is determined by a synch circuit which detects the Start-of-Frame Delimiter (SFD) when it sees the serial sequence '10101011' after the start of carrier sense. This pattern marks the first octet boundary and determines byte alignment for the entire frame.

Incoming RXD bits are clocked into an 8 bit wide serial-to-parallel shift register. The bits are received in order from least significant to most significant within each byte. When an octet is complete, parallel data is loaded into the receiver FIFO. When Carrier is lost, the frame is considered to have terminated; and all remaining serial data are dropped. Serial data is passed to the CRC checker which is initialized upon recognition of the SFD.

This process ordinarily discards all bits that precede the SFD pattern without prejudicing reception of the frame. Some exceptions can be made to this process to improve the robustness of the receiver. They are discussed below.

There is a selectable mode (using the RCON.RCA bit) during the receiver operation which adds robustness by monitoring the XCOL signal continuously throughout reception, starting after the Start-of-Frame delimiter. If collision is detected, reception of the frame is aborted.

Operating in ALTEGO mode engages a second improvement to the reception process. The receiver checks for corruption of the preamble and terminates reception of any frame which has consecutive '0' bits. All valid preambles have an alternating '10' pattern followed by the Start-of-Frame delimiter ('11'). The above is checked immediately on the assertion of the internal carrier sense without any grace periods.

Neither of these causes for abort forces logic to set RXE.

### 7.4.5 CRC Checker

The Receiver computes the CRC of an incoming frame serially. CRC computation includes address, data, and CRC frame fields. It excludes preamble and SFD. Computation stops after reception of last whole octet following loss of carrier. The final value of the CRC must be "C704DD7B" for the packet to be validated.

The CRC polynomial used is AUTODIN II:

$$X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X1 + 1.$$

If the received frame's CRC does not check out, a CRC error is indicated in the Status Register and the CRC Error Counter is incremented. Frame reception will terminate unless the Receive-with-Errors mode is enabled. In addition, if the number of bits received in the last octet (when the carrier is terminated) is greater than one and less than 8 (a full octet), and the CRC check for all complete octets fails, the frame is also labeled as an 'alignment error' and an error flag is set in the Status Register; if this occurs, the Alignment Error Counter will be incremented.

### 7.4.6 Address Recognition Logic

Destination addresses that are 'individual' are compared to a 6-byte station address stored in registers. If all bits match or if the PROMISCUOUS mode is enabled, the frame is received.

A snapshot is taken of the partially-computed CRC as the end of the destination field passes through the CRC checker. If the address has the 'group' or 'multicast' bit set, 6 bits of this checksum are used as a hashed index to a 64-bit Multicast Filter table. If reception of multicast frames has been enabled and if the 6-bit partial CRC points to a bit in the table that has been turned on, the multicast frame will be recognized. Broadcast frames are received when the Broadcast Enable bit (RCON.BROAD) is active or when the hashed bit in the Multicast Filter table has been set.

To cause promiscuous reception of multicast and broadcast frames, the entire Group Address table should be turned on and reception of multicast frames enabled.

If the address is rejected, the frame is also rejected and none of the frame is transported to the buffer memory. If the address is recognized, buffering of the frame begins.

### 7.4.7 Received Byte Counter and Early Receive Warning Interrupt

This circuit counts the number of bytes in each completed frame and filters out runt frames (less than 64 bytes) unless the runt filter is defeated by setting the Accept Runt bit in the Receive Configuration Register (RCON.RUNTS).

The Early Receive Warning (ERW) interrupt is generated when the received byte count equals or exceeds a value specified in the Early Receive Warning Count Register (ERWCNT). The value of ERWCNT is left-shifted four bits (multiplied by 16) before it is compared to the receive byte count. The value is 8 bits wide, allowing the ERW threshold to be set between 0 and 4K with a resolution of 16 bytes.

The ERW interrupt is only generated if an active reception is in progress. Once an early receive interrupt has been set, it may be cleared and will not be set again until another packet exceeds the ERW threshold or the ERWCNT Register is written to. Writing a value to ERWCNT that is less than the current receive byte count (while reception is in progress) will automatically set the ERW interrupt.

The ERW interrupt is mapped to bit 6 in the Interrupt Status Register (INSTAT). The ERW interrupt is enabled or disabled like all other interrupts by the corresponding bit in the Interrupt Mask Register (INTMASK).

#### 7.4.7.1 Early Receive Failure Detection

During the reception of a frame with early receive enabled, it is possible for the host to read frame data from the buffer RAM before the DMA writes it if the early receive threshold is set too low. The failure detection logic enables the host to detect if this has happened; if so, it goes back and recopies the correct data.

The logic required for this is similar to the logic used for early transmit underrun detection. The local memory address is latched every time the DMA writes to the buffer RAM. When the memory cache or I/O pipe reads data from the RAM, the local memory address is compared to the last latched address. (The least significant 2 bits are not compared so the detection mechanism has a granularity of 4 bytes). This comparison is turned off when the DMA finishes placing the frame in shared memory. If the two addresses are equal, ERFBIT (UBRCV.1) is set and the latched address is stored in the ERFA registers (ERFAL and ERFAH). When the host reads that ERFBIT is set, it should begin recopying data from a point at or before ERFA and reset ERFBIT.

**Note**

The value in ERFA contains the local memory address where the failure occurred, not the host address.

The ERFA registers remain set until the host clears the ERFBIT. For more on ERFBIT, see page 40. For more on the ERFA registers, see page 22.

### 7.4.8 Receive Protocol FSM

The Receive Protocol FSM controls reception of frames, checks for errors, and posts status to a register after completion of each reception. It operates counters for the number of bytes in the frame and for three types of error conditions. The receiver protocol FSM can be configured via a register, allowing some flexibility as to which frames are to be received.

The received byte counter is 16-bits wide. The three error counters are each 8-bits wide. These will count from zero up to 255, where they stick to avoid wrap-around. The error counters are self-clearing when read and they can generate a shared interrupt condition when any of them have counted up to 192.

The Receive Protocol FSM interfaces with the DMA section to coordinate buffering of received frames. It informs the DMA of abort conditions, should they occur as well as valid completions of received frames. After the frame has been buffered to memory by the DMA, the DMA section copies the Receiver Status Register (RSTAT) and the number of bytes received from the receiver into the header of the buffer.

The receiver FIFO is monitored for overflow conditions and if one occurs, frame reception is terminated and an error flag is posted to the Status Register.

The Receiver section is enabled by setting the Start and clearing the Stop bits in the Command Register - CMD.STA and CMD.STP. Until enabled, the receiver section ignores incoming frames. Once the Start bit has been set, it remains true internally until the Stop bit is set or the chip is reset. Clearing the Start bit in Command Register does not cause the receiver section to stop operating.

If the Stop bit is set while receiver is operational, it will complete the handling of any ongoing frame and then go to a soft reset condition, ignoring new incoming frames. The receiver will clear the RSTAT Register when the current frame is finished and posted. When both transmitter and receiver sections are stopped, the RST bit in the interrupt status register will be set. It should be noted that the DMA controller may remain active while Stop is set.

The protocol machine can be configured to operate in a "Monitor Mode" which checks validity of incoming frames and maintains error statistics for them but does not store them in memory. Each time an acceptable frame is completed while in this mode, the Missed Packet Counter (MPCNT) is incremented. This counter is not incremented by FIFO overflows.

### 7.4.9 Reception Process

IEEE 802.3 packets consist of these fields and are therefore processed in this order:

- Preamble field
- SFD field
- DA field
- SA field

- Data field
- CRC field

Each of these fields is explained in the following sections.

#### 7.4.9.1 Start of Frame

The preamble field is used to train the Manchester decoder and to detect carrier. If carrier is detected, preamble passes through the receive deserializer which discards it while searching for the Start-of-Frame Delimiter (SFD) symbol. On detecting a good SFD, a VALID_FRAME signal is asserted and the receive FIFO is cleared to accept the received frame. The receive FIFO is loaded by the deserializer with octets (bytes) starting with the first bit after SFD.

While the destination address (DA field) is being checked for recognition, the receive DMA is disabled. If the address is recognized, the DMA is enabled and transfer to memory begins when the FIFO fills to the programmed burst level. If the frame's address is not recognized, the receive unit clears out the FIFO, stops filling it, and waits for the start of the next frame.

The source address and data fields are passed to buffer memory. In some protocols, the first 2 bytes of the data field denote a frame length. These bytes are not interpreted by the SMC795. They are treated as ordinary data.

#### 7.4.9.2 End of Frame

If there is a loss of carrier sense, 3 dribble clocks (receive clocks that occur after the loss of carrier sense) are needed to ensure the synchronizations of all line signals to the receiver circuits. When using the internal Manchester decoder (either 10BASE-T or AUI interfaces), this decoder automatically supplies sufficient dribble clocks to the receiver to complete processing of the frame. When the Manchester decoder is bypassed, it is necessary to supply dribble clocks at the XRXC pin after XCRS terminates.

The CRC from the received frame is sent to memory with the frame via DMA and is included in the byte count posted in the buffer header.

If the receive unit detects errors in the frame (such as an incorrect CRC, an alignment problem, a foreshortened frame), it can abort reception depending on the configuration of the Save Errored Packets and Accept Runt Frames bits of the Receive Configuration Register - RCON.SEP and RCON.RUNTS respectively. Certain other types of errors (including FIFO overflow and Receiver Buffer Overwrite) *always* abort reception.

If reception is aborted, the DMA controller stops sending bytes to the buffer, the Receive Status Register (RSR) and the Interrupt Status Register (ISR) are updated, and the receive unit waits for the next frame to begin.

No buffer header will be posted for the frames that have not been accepted; the previous contents of the header location will remain unchanged.

The received packet length should be less than 32,764 bytes, including DA, SA, data, and CRC. The receiver does not reject longer frames but it may be hard to fit the contents into available buffer space. The buffer ring must always have enough space to contain the entire frame with a 4-byte header. Packets larger than the available buffer space will not be received, regardless of the SEP bit in the RCON Register. Such frames will be posted as ring overwrites and causes the OVW interrupt to be set.

Receiver interrupts (RXE for frames with errors and PRX for frames without errors) indicate the DMA has completely posted the frame to memory. If the DMA aborts, these interrupts are not set for the current frame. If set previously, they remain unchanged. Packets shorter than 64 bytes will be received only when the Accept Runts bit (RCON.RUNTS) is enabled.

### 7.4.10 Receiver Blinding

The Receiver Carrier Sense function is blinded for a period of 4.0 µsec starting at the end of (XCRS + XCOL) when the device has transmitted a frame. This allows the heartbeat to be detected without resetting the deference timer and ensures that an improperly-spaced frame will not interfere with proper posting of status for a new reception.

## 7.5 TRANSMITTER NETWORK INTERFACE (MAC-TO-PHY)

### 7.5.1 Oscillator

A 20 Mhz parallel resonant crystal can be connected between pins X1 and X2; or an external clock can be connected at X1. The oscillator's 20 Mhz output is divided in half internally to provide the clock signals for the encoding and decoding circuits. Operation at serial data rates other than 10 MHz requires an externally-supplied clock since the oscillator is only tuned for 20 MHz. The IEEE 802.3 standard requires 0.01% absolute accuracy on the transmitted signal frequency; however, stray capacitance can shift the crystal's frequency out of range causing it to exceed the 0.01 tolerance.

### 7.5.2 Manchester Encoder

Data encoding and transmission begin when the internal transmit enable signal from the LAN controller goes high and continues as long as it remains high. Transmission ends when the Transmit Enable signal goes low. The last transition occurs at the center of the bit cell if the last bit is '1' or at the boundary of the bit cell if the last bit is '0'.

### 7.5.3 AUI Differential Driver

The AUI differential line driver has the ability to drive up to 50 meters of twisted-pair AUI/Ethernet transceiver cable. These drivers provide emitter-coupled logic (ECL) level signals. The outputs consist of current drivers that must be loaded with external 150Ω pull-up resistors. The interface can be programmed to operate in either half-step or full-step mode in the idle state. This is done via the SEL bit in the MANCH Register. In full-step mode, TX+ is positive in relation to TX- when idle. In half-step mode, TX+ and TX- are equal, resulting in nearly zero differential output voltage.

By setting a bit in the MANCH Register, Manchester encoder/decoder logic can be bypassed completely. External circuitry should drive XTXC, XRXC, XCRS, XRXD, and XCOL pins.

### 7.5.4 Collision Translator

When the 83C795 is used as an AUI device, a separate Ethernet transceiver (MAU) detects collisions on the coaxial cable and generates a 10 Mhz signal which is monitored by the 83C795 through the collision detect pins (CD+, CD-). The presence of this signal activates the internal collision detect signal (CD) connected to the LAN controller. The collision detect signal is deactivated within 160 nsec after the absence of the 10 Mhz signal.

With the standard 78Ω transceiver AUI cable, the CD+/CD- differential input pair must be externally terminated. This requirement may be satisfied by connecting two 39Ω resistors in series with one optional common mode bypass capacitor.

When 83C795 is used in twisted-pair configuration, the collision is generated if the Manchester decoder detects carrier while the transmit enable is active.

### 7.5.5 Twisted-Pair Differential Driver

The TP Driver can transmit through up to 100 meters of unshielded twisted pair cable. The driver includes a circuit for transmit equalization which attenuates the transmit waveform's low-frequency components. This reduces the received signal's zero-crossing jitter and makes the receiver design simpler.

In the transmitter, phase compensation is used to reduce jitter. This is accomplished by using external resistors to determine the drive strength during the second half of a double-width pulse as compared to the drive strength of the first half. There are two pairs of twisted-pair transmit drivers: TPX1 and TPX2. TPX2 is a much weaker driver than TPX1. During the first half-bit-time of each pulse, both pairs of drivers drive out the encoded transmit data. If the pulse is a full-bit-time in length, TPX2 switches polarity during the second half of the pulse and acts to reduce the amplitude of the transmitted signal.

A simplified example of external transmit circuitry is shown in Figure 7-3.

### 7.5.6 Link Integrity Test Function

Each TP driver transmits a short positive pulse periodically when it is not sending data. These pulses are received at the other end of the twisted pair cable, signaling that the link is operating correctly. The time between link test pulses is compared to the expected range at the receiver to help ignore noise pulses. If the link test fails (no pulses or data received within a fixed time period) then the LLED pin is set high and AUI interface is selected. If the link is restored, the chip automatically selects TP interface.

The Link Integrity Test is also used to correct the receiver connection's polarity. The Link Integrity Test signal and Start-of-Idle both have positive polarity. The polarity correction state machine looks at both of these to determine whether to flip receiver polarity. Polarity correction can be disabled by a bit in the MANCH Register (MANCH.ENAPOL).

The Link Integrity Test can be disabled by a bit in the General Control Register (GCR.LIT). Disabling Link Test forces the 83C795 to select the twisted-pair interface.

### 7.5.7 Jabber Protection

If the internal transmit enable signal is active for more than 46 msec, the twisted-pair transmitter will be disabled and a collision indication is sent to the MAC transmitter circuit. When the internal transmit enable signal has been inactive for more than 368 msec, the internal collision indicator will become inactive and the twisted-pair transmitter will be re-enabled. Jabber protection for the AUI port is provided by an external MAU.

### 7.5.8 SQE Test (Heartbeat Test)

In twisted-pair operation, a brief internal collision indication will be sent to the MAC transmitter after each packet is transmitted. When an AUI port is in use, an external MAU will provide this signal.

### 7.5.9 Status Indications

To assist in installation and management of the network, indicator LEDs can be driven directly by four outputs from the 83C795. These show the result of Link Test, polarity check, transmit and receive activity. The LED outputs can be read back through the MANCH register to support network management functions.

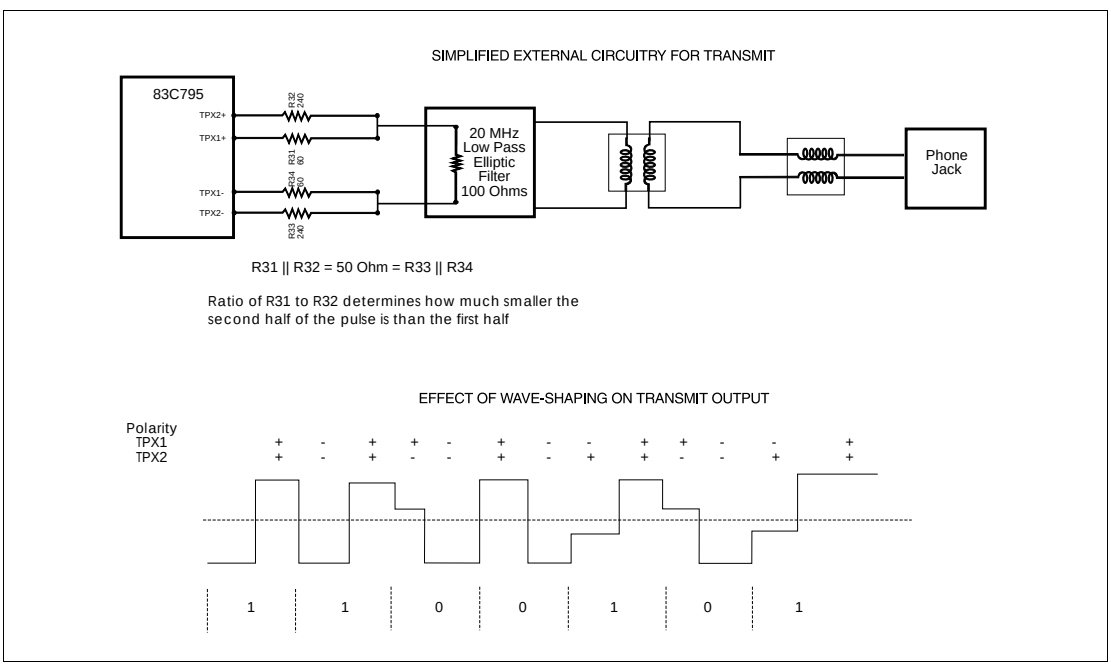


**FIGURE 7-3. SIMPLIFIED TRANSMIT CIRCUITRY**

## 7.6 TRANSMITTER SECTION

### 7.6.1 Basic Function

The transmitter section generates serial stream of NRZ data. It produces preamble and the SFD field at the beginning of a frame, then data is shifted from the FIFO serially followed by the CRC field. The transmitter checks for collisions and retransmits the frames if necessary, it counts interframe gap and implements random backoff algorithm. It maintains the transmit statistics and generates status information on each attempted transmission. Selection of optional operating modes of the Transmitter section is done primarily by programming the Transmit Configuration register.

### 7.6.2 Preamble Generator

At the beginning of each frame, the transmitter generates 56 bits of preamble (an alternating '1010' pattern). Immediately after this, it generates a Start Frame Delimiter sequence which is '10101011.'

### 7.6.3 Transmit Serializer

The Transmit Serializer converts 8 bits of parallel data from the FIFO into serial transmit NRZ data. Data is shifted out least-significant-bit (LSB) first. Serial data is clocked onto an internal signal (TXD) by the rising edge of the transmit clock. This signal passes to the Manchester Encoder which encodes it and drives the selected serial interface. When the encoder is being bypassed, the serial data drives the XTXD pin directly.

### 7.6.4 CRC Generator

The transmitter calculates the CRC serially and appends it to each frame. CRC is clocked out most-significant-bit (MSB) first. The transmitter can be configured to exclude attachment of the computed CRC by setting the CRCN option bit in the Transmit Configuration Register (TCON.CRCN). This is useful for some bridging applications in which the original checksum must remain attached to the packet until the final destination.

### 7.6.5 Transmit Protocol FSM

Transmit Protocol FSM controls transmission of frames, defers to active carriers and collisions, monitors collision conditions, and initiates both backoff and re-transmission when needed.

#### 7.6.5.1 Interframe Gap and Deference

Deference is initiated when both XCRS and XCOL have terminated at the end of a frame.

The transmitter deference logic initiates a 2-part timer at the end of network activity. While this timer is running, no frame transmission will be initiated. The first part of the timer (interFrameSpacingPart1) is used to observe the network for transmission activity by other stations. If this station is transmitting, carrier is sensed, or collision is detected during this part of the timer, the timer will be reset to zero and held there until the termination of line activity. When the first part of the timer elapses, line activity is no longer observed and the timer runs to completion.

If any frame is queued up for transmission at the moment of timer expiration, transmission will be initiated regardless of line activity.

The combination of interFrameSpacingPart1 and interFrameSpacingPart2 makes up the Inter-Frame Gap (IFG) as defined by the 802.3 specification. The interFrameSpacingPart1 is 6.0 µsec and interFrameSpacingPart2 is 3.6 µsec.

#### 7.6.5.2 Collision Handling Logic

When collision is detected by the transmitter section during the first slot time of an active transmission, the transmission does not terminate immediately. Instead, the preamble is allowed to finish and the jam sequence is transmitted. The jam sequence consists of 32 bits of logic '1's. If collision is detected after the slot time is passed, the 83C795 will abort the transmission with jam and without retry and 'Out of Window Collision' bit is set in the transmission status register.

### 7.6.6 Timers

#### 7.6.6.1 Slot Timer

During transmit, the slot timer starts counting once the receiver recognizes that a carrier is present at the start of a returning preamble. When backing off, the slot timer starts with the end of TXE for the collided frame and does not get reset by any other incoming frames.

Slot time is programmable through the Enhancement Register. The choices are 256-, 512-, and 1024-bit times. The default value is 512-bit times.

#### 7.6.6.2 Backoff Timer

After a transmission is terminated because of a collision, a retransmission is attempted. Timing of it is determined by the 'truncated binary exponential backoff' algorithm. This algorithm is:

draw random integer r: 0 <= r < 2**k

where *k* equals the number of retries already on this transmission. K starts at 0. Wait 'r' number of slot times and then start normal transmit deferral.

The backoff timer is a 12-bit counter that is initialized to a random number when an attempted transmission results in a collision. The counter decrements once per slot time until it reaches zero. The Transmit Protocol State machine utilizes this timer to insert a variable amount of delay ahead of its attempt to retransmit the frame.

There is a selectable enhancement to backoff timer operation which causes it to suspend counting while there is network activity and resume during idle times. In this mode of operation, the backoff timer continues to operate while a carrier sense remains from the initial collision but does not operate during any other carrier indication. This is referred to as the 'Stop Backoff' algorithm. It may put stations that use it at a disadvantage when operating on the same network with stations not utilizing it and caution in its use is advised. This algorithm can be enabled by setting the SBACK bit in the ENH Register.

#### 7.6.6.3 Collision Counter

All retransmission attempts are counted by the collision counter. After the maximum number of attempts is reached (16), the transmission of the frame is aborted, an interrupt is generated and event is reported as an error in the transmit status register.

#### 7.6.6.4 Heartbeat Detection

After each transmission, the transmit logic opens a window 3.6 µsec long during which it looks for a pulse on the XCOL signal. This pulse is normally generated by the MAU and is received through the AUI interface. If the pulse is received, the CDH status bit of the TSTAT Register is cleared. If no pulse is received during the window, the CDH bit is set.

### 7.6.7 Transmitter Operation

#### 7.6.7.1 Transmission Initialization

Packets to be transmitted are built in buffer memory by the host. These packets must include the DA, SA, and data fields. CRC is not read from buffer memory unless CRC generation is disabled.

The transmitter requests the frame from the DMA when the TXP bit of the CMD Register is set by the host. The TSTART and TCNT Registers must be properly programmed prior to setting TXP.

Once set by the host, TXP can be cleared only by the DMA after the transmitter has signalled completion of an attempted transmission.

#### 7.6.7.2 Transmission Process

The DMA starts to fill the transmit FIFO with bursts of data then notifies the transmitter that it is ready for transmission. The transmitter defers until the media is clear and an interframe gap has passed then generates preamble and SFD fields. It then pulls bytes out of the transmit FIFO, serializes them, and shifts bits to the output pins while computing the CRC on the packet. The DMA also monitors the amount of room remaining in the FIFO and initiates a burst of memory transfers when there is enough room for the entire burst to fit.

Once the DMA has filled the Transmit FIFO with the last byte of the packet, it sets a flag. When the FIFO becomes empty, it signifies the end of the frame. CRC computation stops and the CRC is appended serially to the frame, most significant bit first.

#### 7.6.7.3 Transmit Underrun

If the FIFO becomes empty before the internal flag is set, it is considered a transmit underflow and is posted as a transmit error in the Transmit Status (TSTAT) Register. In this case, transmission of the packet is aborted and an interrupt can be generated.

#### 7.6.7.4 Early Transmit Underrun Protection

This feature is used to facilitate initiation of transmission prior to completion of assembly of the outgoing frame in the transmit buffer.

Early transmit underrun protection is controlled by two bits in the Command Register - CMD.DISETCH and CMD.ENETCH. Setting DISETCH to '1' disables early transmit underrun checking and setting ENETCH to '1' enables checking. Writing both bits to zero leaves transmit checking in its previous state. Setting both bits to '1' is illegal. This operation works the same as the Command Register start and stop bits for bringing the chip on and offline.

While early transmit underrun checking is enabled, the memory address is latched each time the host does a write to the buffer memory (the actual memory address is used, not the host address). When the DMA reads packet data from the buffer memory, the memory address is compared to the most recently-latched memory write address (written from the host with ETCHON only). The DMA distinguishes between accesses to descriptor table entries and actual packet data.

If early transmit checking is on, and the DMA's memory read address is greater than the absolute value of the latched memory write address, a "buffer underrun" condition is set. This condition aborts the transmitter which in turn aborts the DMA. The condition is cleared when the DMA detects the abort and clears the transmit FIFO. The transmit abort is reported as though it were a FIFO underrun and both the TSTAT.UNDER and INTSTAT.RXE flag bits are set.

#### 7.6.7.5 Collisions

When a collision is reported on the CD pin, the transmitter sends a 32-bit sequence composed of all '1' bits as a jam signal, then terminates its transmission. If collision occurs during the preamble of a frame, the remainder of the preamble is sent before sending the jam signal.

If the collision occurred after the end of one slot time, transmission is aborted without retry after sending a jam pattern. This is considered an out-of-window collision and posts a status bit in the TSTAT register (TSTAT.OWC) and is a contributor to the TXE flag in the INTSTAT Register.

For collisions that occur within the first slot time of a frame, a counter of retries is incremented and checked against the retry limit (16). If the number of retries is less than the limit, a back-off delay (in units of slot-time) is chosen at random. The transmitter then requests the frame's retransmission from memory and delay is initiated.

The DMA controller clears out the transmit FIFO, loads its pointer to the start of frame in memory, and waits for the abort signal to subside. The FIFO is loaded in the same manner as it was initially. If the maximum number of collisions (16) is exceeded, transmission is aborted without further retries and no back-off delay is executed.

#### 7.6.7.6 Extensions Beyond 802.3 10Base5 Protocol

The 802.3 10Base5 protocol utilizes frame lengths between 64 and 1518 bytes inclusively. The transmitter section is capable of sending frames greater than 17 and less than 32,768 bytes in length. Transmission of longer or shorter frames than permitted within the 10Base5 definition may be useful in other variations of 802.3 protocols.

When considering the transmission of giant frames on a non-802.3 network, be aware that very long frames can activate jabber detectors in existing 10BaseX MAUs and repeaters. The use of such frames in non-standard networks requires considerable planning and some caution.

To support non-802.3 protocols, the 83C795's slot time is program-selectable. Choose from 256-bit, 512-bit, or 1024-bit times.

The Stop Backoff algorithm is selectable for backoff modification following collisions.

When operating in ALTE GO Mode, detection of any pair of consecutive '0' bits within the preamble causes the reception of that frame to be abandoned. No error is reported.

The RCON.RCA bit enables the receiver to abandon reception of any frame which causes a collision. No error is reported.

#### 7.6.7.7 Extended Length

When the XLENGTH bit (GCR.7) is set, the twisted-pair port can be connected to cables longer than the 100-meter limit specified by the 802.3 specification

# 8.0 BUFFER STRUCTURING AND DATA MOVEMENT PROCESSES

## 8.1 TRANSMIT PACKETS

### 8.1.1 Single Packet Transmission

A packet for transmission is placed by the host into buffer memory. This packet must include the DA, SA, and data fields. The preamble, SFD, and CRC (normally) are not included in the buffer. If CRC generation is suppressed, the CRC field for the packet is also supplied by the host. The packet is placed in a contiguous block of memory in the buffer, starting on a 256-byte boundary.

Valid 802.3 packets have at least 48 bytes of data. If less data is to be transmitted on an 802.3 network, it is the responsibility of the host to build a packet with pad data included. The 83C795 will transmit frames of any programmed length (greater than 17 bytes), even those which are too short to be valid frames in an 802.3 network. DMA will transfer the number of bytes programmed into the TCNTH and TCNTL Register pair starting from address (TSTART * 100H).

### 8.1.2 Multiple Packet Transmissions

To support multiple transmissions per command, a transmit queue can be enabled by setting the ALTEGO bit in the Enhancement Register (ENH.5). In this mode, a table of frame descriptors defines the starting location and length of all enqueued transmissions. This descriptor table is processed in a circular manner by the LAN controller.

The table is treated as a ring of entries whose starting and ending points are defined by a pair of registers (TBEGIN and TEND) in the LAN controller. These registers are initialized with the upper 8-bits of address for the first location of the table and the first location after the end of the table. TEND is not within the table. When table processing reaches the location defined by TEND, it is switched back to TBEGIN. The table must be aligned with 256 byte boundary in the buffer memory. Each entry is 8 bytes long. The format of this buffering is defined in Figure 8-1.

To send multiple transmissions, the driver builds the frames in buffer memory in the same contiguous form presently expected. The driver then adds an entry for each frame into a table of transmit descriptors. This entry contains the starting location and length, and transmit configuration for each frame in the transmit queue. Places are provided in the table for return of the Transmit Status (TSTAT) Register and collision count associated with each transmission. A simple semaphore protocol will be used to control ownership of transmit buffers.

The LAN controller keeps a pointer in the TTABH and TTABL Registers to the transmit descriptor table. This pointer is initialized by the driver when the table is first built and should not need re-initialization thereafter. When transmit command has been set and device is online, transmit begins from the entry pointed to by the TTABH and TTABL Registers. The LAN controller first checks the TSTAT field. If it encounters a field equal to FF, it will attempt to transmit the frame pointed to by the entry. The status field will be changed to zero after the remainder of the entry has been read. When it encounters a TSTAT field not equal to FF, no frame will be sent, the transmit complete interrupt will be sent and the field will not be altered.

If the frame is marked for transmission, the DMA controller loads its TSTARTH, TSTARTL, TCNTH, TCNTL, and TCON Registers from the descriptor. TSTAT gets marked as having been opened by the LAN controller and transmission proceeds as with single transmissions except that when the transmission has completed, the transmit status and collision count are moved by DMA into the table. The table pointer is updated and transmission of next entry begins.

If a transmit abort occurs (too many collisions) the transmitter will stop processing the chain and post the current transmit and interrupt status. If the CMD.STP bit is set, the transmission of any ongoing frame proceeds until completion or abort but no successive frames in the chain are processed. The TTAB indices will point to the first unprocessed frame in the table so that none are lost.

An alternative mode of controlling the transmit interrupt can be enabled by the EOTINT bit in the Enhancement (ENH) Register. When enabled, the transmit interrupt will be generated only upon

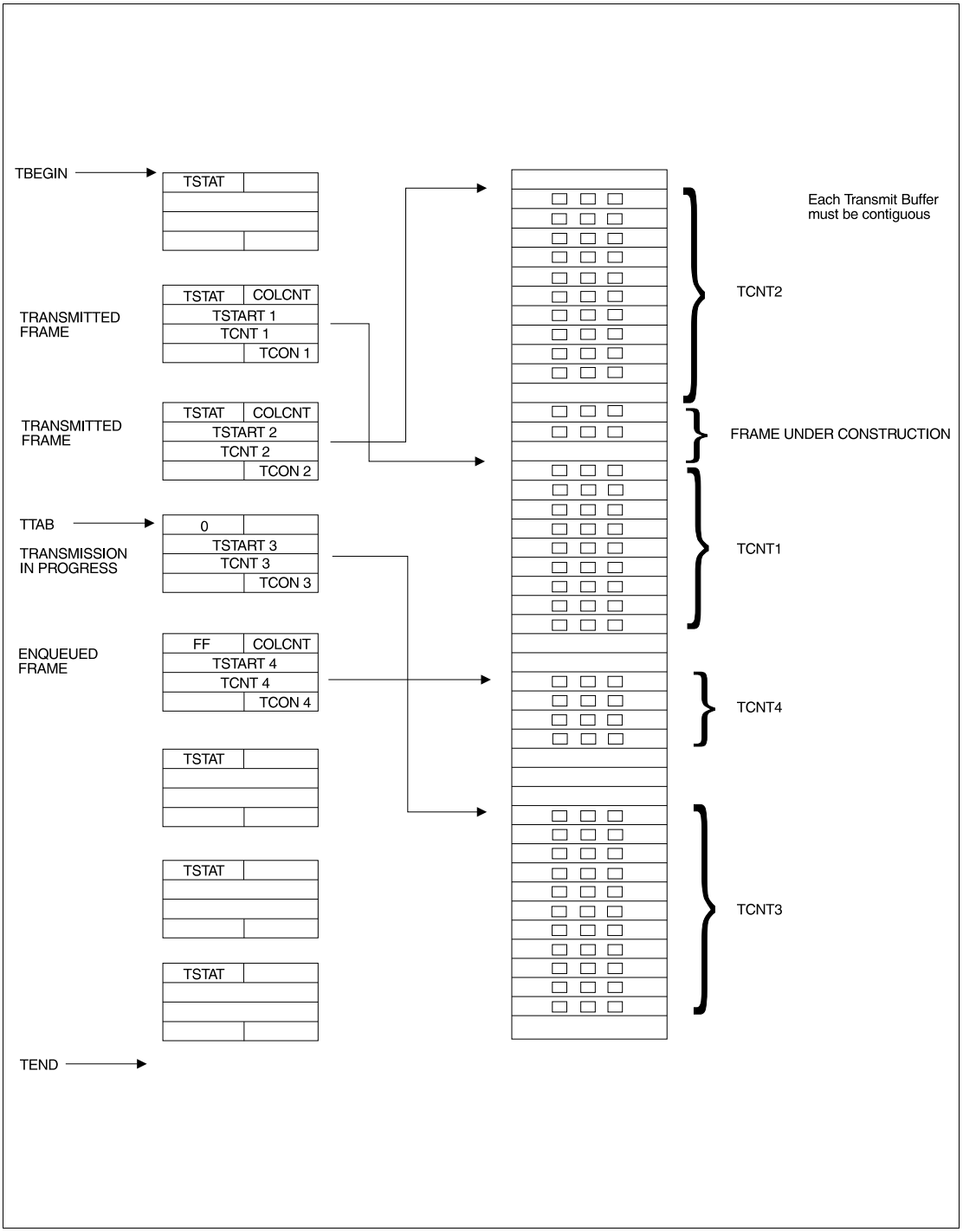


**FIGURE 8-1. MULTIPLE FRAME TRANSMIT BUFFER FORMAT**

completion of the transmit chain. Normal masking applies on top of this delay mechanism.

Without enabling this new mode of controlling transmit interrupts, an interrupt will be generated on a frame by frame basis but the interrupt status may not be current by the time it can be read by the driver software.

#### 8.1.2.1 Ownership of Buffers

The TSTAT field of the table entry is used to control ownership of the frame buffers. Handshake over control of transmit frame buffers is governed by the following conventions:

| TSTAT Field Value | Meaning |
|---|---|
| TSTAT field = 00 | 83C795 has begun transmission. |
| TSTAT field > 00 and < FFh | Frame completed. |
| TSTAT field = FF | Assembled frame, not yet transmitted. |

**TABLE 8-1. TSTAT FIELD VALUES**

The driver software fills the TSTAT table entry with FFh when it releases the frame to the LAN controller for transmission. When transmit command has been set and device is online, the DMA looks at TSTAT field. If it encounters a field = FF, it will attempt to transmit the frame pointed to by the entry. The status field will be changed to zero after descriptor entry is read and transmitter commits to sending. If DMA encounters a TSTAT field greater than or less than FFh, no frame will be transmitted, the transmit complete interrupt will be sent. The field is not altered.

When a transmission completes, the contents of TSTAT register will be moved into that location.

#### 8.1.2.2 Modifying the Transmit Queue

To add a frame to the transmit queue, build the frame in buffer memory then find the table entry following the last enqueued frame. Enter the descriptor for the new frame with a TSTAT that is neither 00 nor FF in value. When descriptor is complete, write TSTAT with FF. The descriptor entry following the last enqueued frame entry must have its TSTAT value marked with a non-FF value. That defines the end of the enqueued frames for the DMA.

Set the TXP bit of the COMMAND register to ensure that the new transmission goes out. The TXP bit can be written regardless of completion status and will ensure that the latest frame does get transmitted. If the LAN controller reaches the end of the transmit queue before the new frame has been added, a transmit complete interrupt is generated for the old portion of the queue and another transmit complete interrupt will be generated when the added portion completes.

The driver should not attempt to alter any buffered frame whose TSTAT is either 00 or FF while CMD.TXP is turned on. Wait until all transmissions are complete or set the CMD.STP bit and wait for the STOP status to be confirmed in the Interrupt Status register. Examine TSTAT table entries to determine which frames have been transmitted. Those whose entries are FF have not been opened by transmitter.

Refer to Tables 8-2 and 8-3 for a summary of the transmit descriptor table format.

| 8-BIT MEMORY | 16-BIT MEMORY | |
|---|---|---|
| | D15-D08 | D07-D00 |
| COLCNT | TSTAT | COLCNT |
| TSTAT | TSTARTH | TSTARTL |
| TSTARTL | TCNTH | TCNTL |
| TSTARTH | not used | TCON |
| TCNTL | | |
| TCNTH | | |
| TCON | | |
| not used | | |

**TABLE 8-2. FORMAT OF TRANSMIT DESCRIPTOR TABLE**

| REG | DESCRIPTION |
|---|---|
| COLCNT | Written with the number of collisions experiences by the LAN controller while attempting to transmit this frame. |
| TSTAT | This entry in the table is used to control ownership of the frame buffer:<br>TSTAT = 00 means the 83C795 has begun transmission<br>TSTAT>00 but <FFh means the frame has been completed and the contents of the transmit status register have been moved into that location.<br>TSTAT = FFh means assembled frame has not yet been transmitted. |
| TSTARTH,L | Pointer to start of transmit buffer. 16-bits wide. |
| TCNTH, L | Number of bytes to transmit, exclusive of computed FCS. |
| TCON | Transmit configuration desired for this frame. This entire byte is DMA'd into the internal TCON register and applies to this frame. |
| not used | Ignored by the LAN controller. |

**TABLE 8-3. MEANING OF DESCRIPTOR TABLE**

## 8.2 RECEIVE PACKET BUFFERING

Two alternative modes of buffering received packets are supported: ring of buffers and linked-list. Selection between these two modes is made by a bit in the ENH Register, ENH.ALTEGO.

### 8.2.1 Ring of Buffers

All received packets are stored in one circularly connected, contiguous set of 256-byte buffers. The number and location of buffers in the ring is determined by the values written into the RSTART and RSTOP Registers by the host when the 83C795 is initialized. RSTART points to the first buffer in the ring and RSTOP points to the buffer after the last one in the ring.

Each packet received will be stored into one or more of these buffers, with a 4-byte header inserted at the start of the first buffer.

Figure 8-2 details the format of a received packet in memory.


**FIGURE 8-2. RECEIVER BUFFER FORMAT**

Frames that extend to the buffer designated by RSTOP are continued in the buffer designated by RSTART and successive locations. RSTOP may be either greater than RSTART+1 or less than RSTART. Making RSTOP equal to RSTART or RSTART + 1 leads to unpredictable results. The relationship of these registers to ring placement in memory is illustrated in Figure 8-3.

Up to 254 buffers can be allocated to the ring. The receiver DMA will use as many as required to store a packet. This allows the chip to be configured to receive frames nearly as long as 64K bytes. This can be useful in customized CSMA networks; however, allocating so many buffers to the reception process leaves very little capability for buffering transmit frames although it is within the capability of the 83C795.

The receive DMA utilizes two additional registers to manage the buffer ring. These are the Current Page Register (CURR) and the Boundary Page Register (BOUND). The CURR Register points to the first buffer that is not part of a completely received packet. When RDMA is storing a frame, this register points to the start of the frame being stored. When RDMA is not storing a frame, it points to the first buffer that will be used for the next frame to be received.

The BOUND Register protects received frames from being overwritten by later frames. It points to the first buffer in the ring that is not to be overwritten.

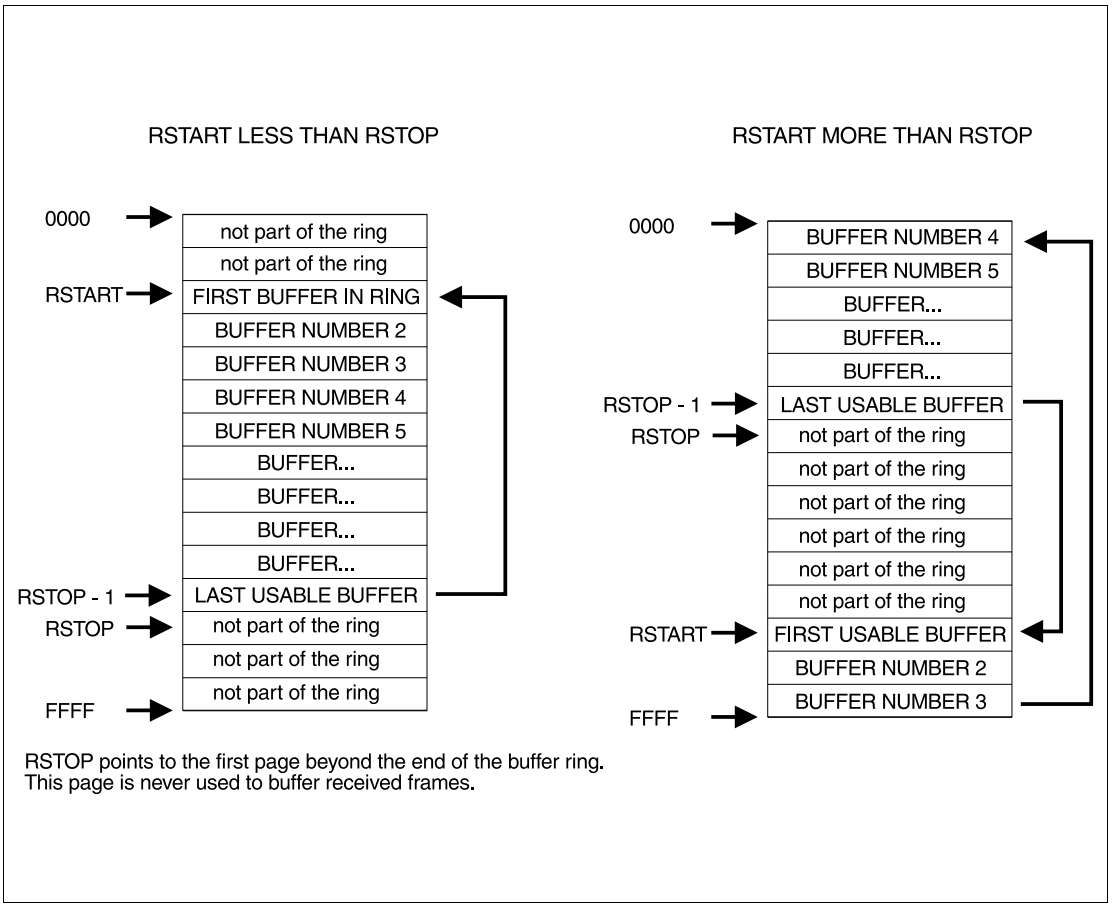


**FIGURE 8-3. RING BUFFER STRUCTURE**

When the receive DMA process attempts to open the buffer that BOUND points to for storage of a packet, it aborts the reception and sets the INTSTAT.OVW flag (in the Interrupt Status Register) and the RSTAT.MPA bit (in the Receiver Status Register). The protected buffer is not written to.

Normally, BOUND is set up to point to the oldest received packet in the ring. This pointer can be managed by the host. To discard an unwanted frame, the host may simply rewrite BOUND to point to the next packet. A good practice is to write zeros into the first byte of the discarded packet to prevent future interpretation as a received packet.

When BOUND and CURR have the same value, the ring may be either full or empty. The 83C795 can distinguish between full and empty rings. CURR is updated by the DMA controller at the end of a frame reception. The ring is considered full if these two registers are equal and the DMA controller updated CURR more recently than the host updated BOUND. Conversely, when data is being removed from the ring, the host updates BOUND after removal. When it has been advanced past the end of the last received frame, it should have the same value as CURR. The chip treats the ring as empty when these two register values are equal and BOUND has been updated after CURR.

When initializing the buffer ring, BOUND and CURR may be given the same or different values. These registers may be initialized to point to any buffer within the ring. They may point to RSTART but may *not* point to RSTOP. If CURR points outside the ring, the RDMA will store frames outside the ring in an unpredictable manner. If BOUND points outside the ring, the received frames will not be protected from overwrites by later frames.

Figure 8-4 illustrates the buffer ring in two common states. The top ring, INITRBUF, illustrates the relationship between these pointers in a typical ring initialization. The bottom ring, RBUF, shows a ring that has received a few frames - the normal condition for the ring.

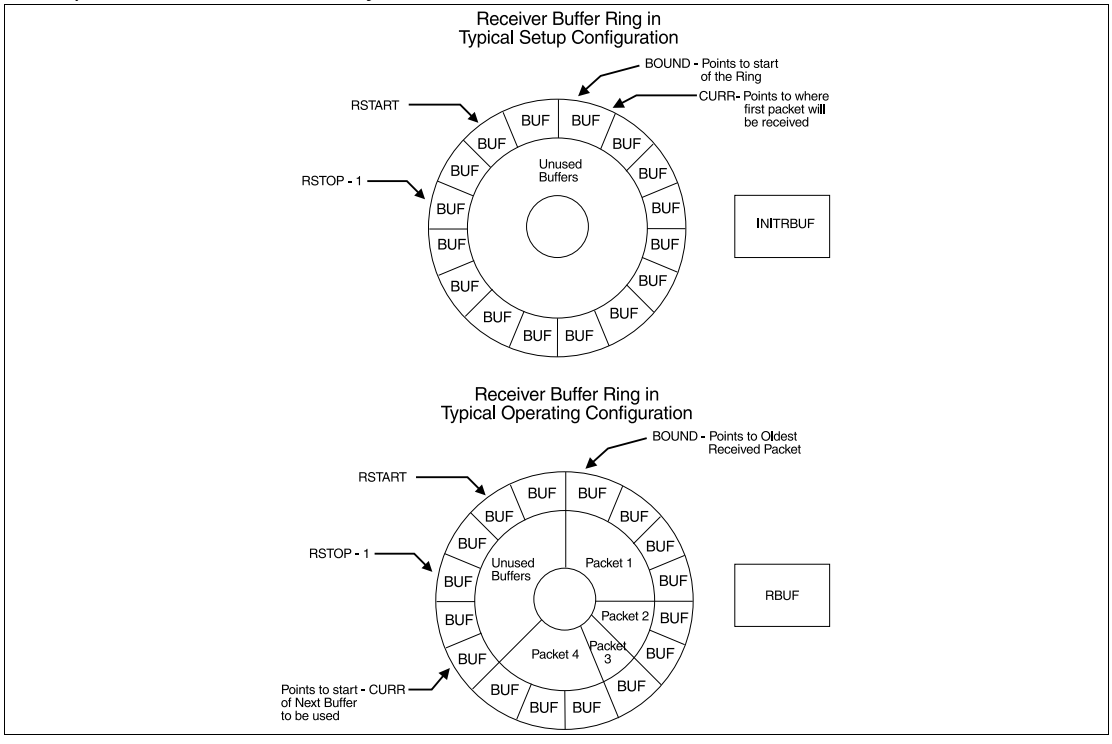


**FIGURE 8-4. RECEIVER BUFFER RING 1**

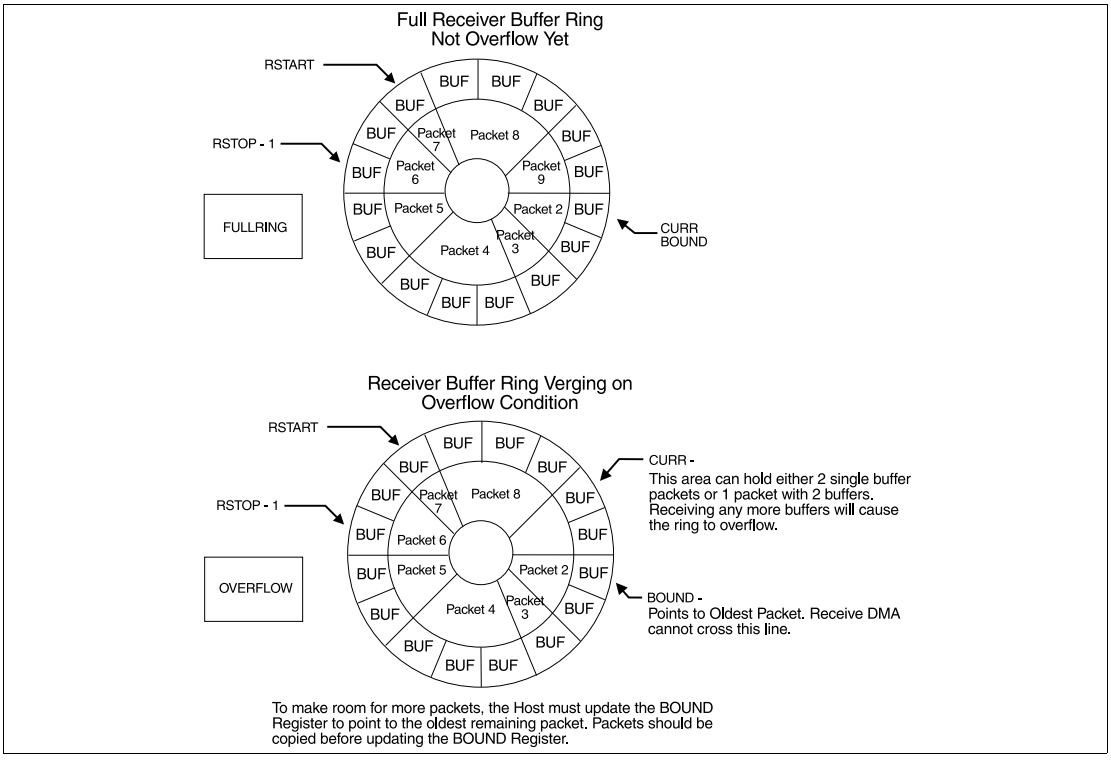


**FIGURE 8-5. RECEIVER BUFFER RING 2**

Figure 8-5 shows the same ring after proper removal of the oldest received packet. The top ring, FULLRING, shows a ring that is completely full. The bottom ring, OVERFLOW, shows a ring on the verge of overflow.

#### 8.2.1.1 Automatic Ring Wrapping

Automatic ring wrapping enables the host to read a contiguous block of data from the buffer RAM without having to check whether the block is so long that the access wraps around the end of the buffer ring. This is accomplished by checking whether the next value to be loaded into the memory cache's buffer counter is equal to the register RSTOP. If this is true, then the value of the RSTART register is loaded into the counter instead. The memory cache's Host Counter is left unchanged so the host can continue accessing memory past the end of the ring, but instead receives the correct data from the start of the ring. The comparison is only made when the counter is incrementing so it is not possible to start the block of data past the ring's end. Also, since the host is sending addresses that are greater than the end of the ring, it is required that the 83C795's memory space extend beyond the end of the ring. This can be accomplished by arranging the memory space such that the transmit buffers come after the receive buffer ring. This feature is enabled by setting the WRAPEN bit (UBRCV.0). For more on this, refer to page 40.

#### 8.2.1.2 Ring-Empty Bit

This is a read-only bit located at UBRCV.2 which indicates to the host that there are no completely received frames in the buffer ring yet. The host checks this bit after it finishes receiving a frame and quickly determines whether there are more frames to copy. This bit is set when BOUND equals CURR. BOUND is updated after CURR.

Since this bit is only cleared for completely received frames, another method must be used to determine if there is a partial frame in the buffer that exceeds

the early receive threshold (early receive mode only). This is done by checking the interrupt status register after checking the ring-empty bit.

### 8.2.2 Linked-List Receiver Buffering

Linked-List Receiver Buffering is enabled by a bit in the Enhancement register, ENH.ALTEGO, and is an alternative to receive ring form of buffering. In this mode, the receiver directs its input to a group of individually-sized buffers that are not necessarily contiguous. All buffers need not be the same size. Multiple buffers can be chained together as needed to receive an incoming frame.

These buffers are linked together via a table of buffer descriptors which define the starting location, size, and usage of all receiver buffers. This table controls both the received frames and the available buffer pool.

The descriptor table is treated as a ring of entries whose starting and ending points are defined by a pair of registers (RBEGIN and REND) in the LAN controller. These registers are initialized with the upper 8-bits of address for the first location of the table and the first location after the end of the table. REND is not within the table. When table processing reaches the location defined by REND, it is switched back to RBEGIN. The table must be aligned with 256-byte boundary in the buffer memory. Each entry is 8-bytes long. The format of this buffering arrangement is depicted in Figure 8-6.

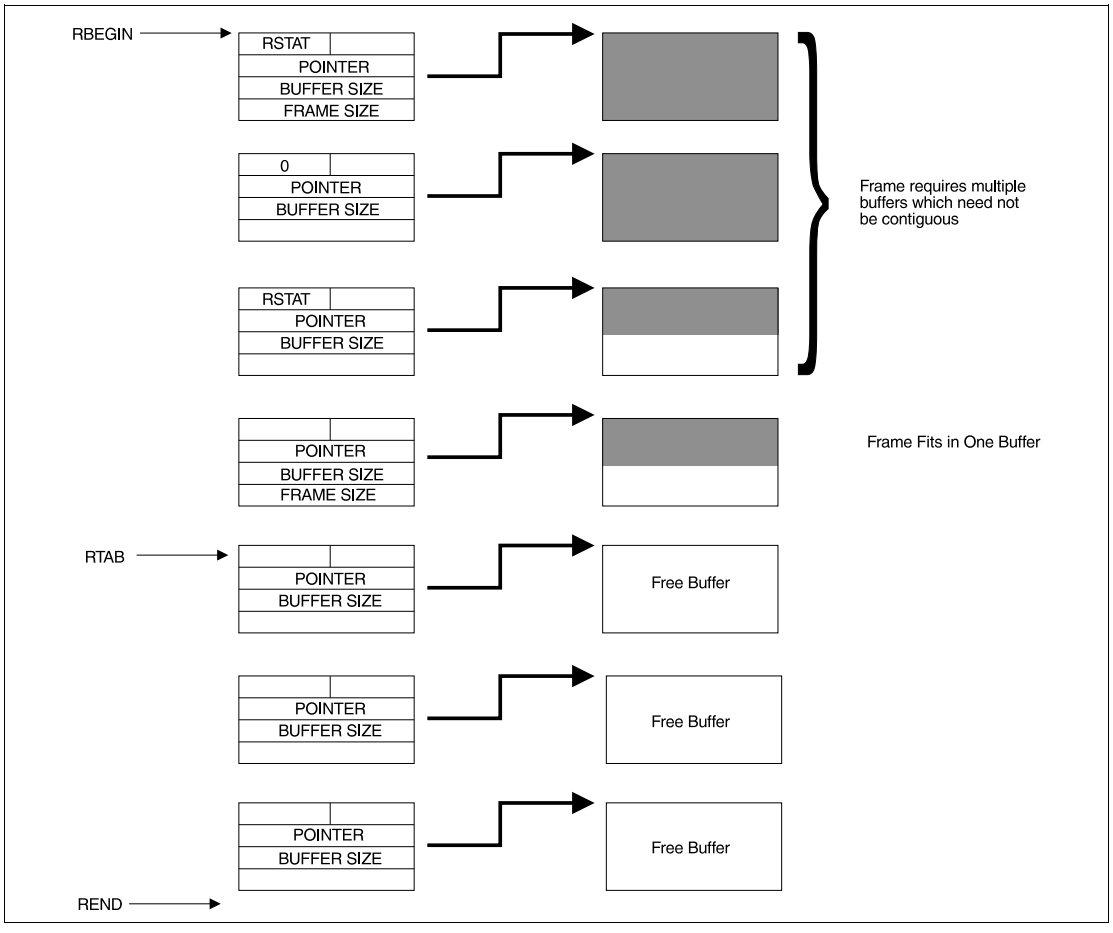


**FIGURE 8-6. LINKED-LIST BUFFER FORMAT**

The LAN controller keeps a pointer (Registers RTABH and RTABL) to the Receive Descriptor table. This pointer is initialized by the driver when the table is first built and should not need re-initialization thereafter.

When the LAN controller's DMA becomes active to receive a frame, it will use RTAB as a pointer to the first free receive buffer. RSTAT field of descriptor is checked to see if the buffer is free (RSTAT=00). If free, the size and starting location of the buffer are loaded from the descriptor and rounded down to even values. Storage of the frame into the buffer begins. The value of RTAB is saved at the start of reception for each frame in case the frame gets aborted and RTAB has already been updated to the next table entry.

While a frame is being buffered, the LAN controller will follow the entries in the descriptor table to obtain additional buffers as needed to receive the entire frame. Upon completion, the descriptor for the first buffer of that frame is written with the receiver status register and the total byte count for that frame. The byte count includes storage of the received FCS.

Should a partially received frame be rejected by the LAN controller, it will reclaim the buffers by resetting the RTAB pointer to its value previous to start of reception of this frame.

To add free buffers to the end of the descriptor table, fill in the first new table entry with RSTAT=non-zero, build all other table entries with RSTAT=0, then change the first RSTAT to 00 to indicate availability.

## 9.0 ELECTRICAL SPECIFICATIONS

### 9.1 ABSOLUTE MAXIMUM RATINGS

Supply Voltage ($V_{cc}$) . . . . . 6V
TTL Input Voltage. . . . . 0 – 5.5V
Differential Input Voltage . . . . . -0.5 – 5.5V
Differential Output Voltage. . . . . 0 – 16V
Differential Output Current. . . . . -40mA
Storage Temperature . . . . . -65°C (-85°F) to 150°C (302°F)

Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operating at these limits is not recommended; operation should be limited to conditions specified under "DC Operating Characteristics."

### 9.2 RECOMMENDED OPERATING CONDITIONS

Supply Voltage ($V_{cc}$) . . . . . 5V ± 5%
Ambient Temperature. . . . . 0°C (32°F) to 70°C (158°F)

### 9.3 DC OPERATING CHARACTERISTICS

$T_a$ = 0° C (32° F) to 70° C (158° F)
$V_{dd}$ = +5V ± 5%

**Note**

All currents into device pins are positive. All currents out of device pins are negative. All voltages are referenced to ground unless otherwise specified.

#### 9.3.1 Input Pins

| SYMBOL | PARAMETER | MIN | MAX | UNIT | CONDITIONS |
|---|---|---|---|---|---|
| $I_{dd}$ | Supply Current | | 110 | mA | |
| $V_{ih}$ | Input High Voltage (MA09-MA00) | 2.0 | — | V | Internal pull-down resistor value between 35KΩ and 150KΩ |
| | Input High Voltage (MA07-MA00, MEMR, MEMW, SBHE) | 1.65<br>1.94 | —<br>— | V<br>V | when $V_{cc}$=4.5V<br>when $V_{cc}$=5.5V for Schmitt-triggered inputs, TTL-compatible levels |
| | Input High Voltage (X1) | 3.5 | — | V | CMOS level input |
| | Input High Voltage (All other inputs) | 2.0 | — | V | TTL levels, high impedance |

| SYMBOL | PARAMETER | MIN | MAX | UNIT | CONDITIONS |
|---|---|---|---|---|---|
| $V_{il}$ | Input Low Voltage (MA09-MA00) | — | 0.8 | V | Internal pull-down resistor value between 35KΩ and 150KΩ |
| | Input Low Voltage (MA07-MA00, MEMR, MEMW, SBHE) | —<br>— | 1.04<br>1.20 | V | when $V_{dd}$=4.5V<br>when $V_{dd}$=5.5V<br>Internal pull-down resistor value between 35KΩ and 150KΩ |
| | Input Low Voltage (X1) | — | 1.5 | V | CMOS level input |
| | Input Low Voltage (All other inputs) | —<br>— | 1.04<br>1.20 | V<br>V | when $V_{dd}$=4.5V<br>when $V_{dd}$=5.5V |
| $V_{hys}$ | Input Voltage Hysteresis (Schmitt Inputs) | 0.40<br>0.39 | 0.51<br>0.59 | V<br>V | when $V_{dd}$=4.5V<br>when $V_{dd}$=5.5V |
| $V_{t+}$ | Input Voltage TTL + (Schmitt Inputs) | 1.45<br>1.65 | 1.65<br>1.94 | V<br>V | when $V_{dd}$=4.5V<br>when $V_{dd}$=5.5V |
| $V_{t-}$ | Input Voltage TTL- (Schmitt Inputs) | 1.04<br>1.20 | 1.18<br>1.35 | V<br>V | when $V_{dd}$=4.5V<br>when $V_{dd}$=5.5V |
| $V_{ds}$ | Differential Squelch Threshold (RX±, CD±) | -175 | -300 | mV | |
| | Differential Squelch Threshold (TPR±) | 300 | 500 | mV | |
| $I_{il}$ | Input Low Current (RX±, CD±, TPR±) | — | -500 | uA | |
| | Input Low Current (MA09-MA00, MD07-MD00, MEMR, MEMW, SBHE) | -25 | -200 | uA | |
| | Input Low Current (X1) | — | -50 | uA | |
| | Input Low Current (All other inputs) | — | -10 | uA | |
| $I_{ih}$ | Input High Current (RX±, CD±, TPR±) | — | 500 | uA | |
| | Input High Current (X1) | — | 50 | uA | |
| | Input High Current (All other inputs) | — | 10 | uA | |
| $I_{in}$ | Input Current (CAP) | -1.0 | -1.0 | mA | when $V_{in}$ = 2.5V |

**TABLE 9-1. INPUT PIN VALUES**

### 9.3.2 Output Pins

| SYMBOL | PARAMETER | MIN | MAX | UNIT | CONDITIONS |
|---|---|---|---|---|---|
| $V_{ol}$ | Output Low Voltage (X2) | | 0.7 | V | Iol = 8mA |
| | Output Low Voltage (TPX1±) | | 0.6 | V | $I_{ol}$ = 30mA |
| | Output Low Voltage (TPX2±) | | 0.75 | V | $I_{ol}$ = 14mA |
| | Output Low Voltage (ZWS) | | 0.4 | V | $I_{ol}$ = 24mA |
| | Output Low Voltage (GPOUT, IORDY, IO16CS, M16CS, SD00-SD16) | | 0.4 | V | $I_{ol}$ = 24mA |
| | Output Low Voltage (All other outputs) | | 0.4 | V | $I_{ol}$ = 4mA |
| $V_{oh}$ | Output High Voltage (X2) | 3.5 | | V | Ioh = -100mA |
| | Output High Voltage (TPX1±) | $V_{dd}$-0.6 | | V | $I_{oh}$ = 30mA |
| | Output High Voltage (TPX2±) | $V_{dd}$-0.75 | | V | $I_{oh}$ = 14mA |
| | Output High Voltage (ZWS) | 2.4 | | V | $I_{oh}$ = 24mA |
| | Output High Voltage (GPOUT, IORDY, IO16CS, M16CS, SD00-SD16) | 2.4 | | V | $I_{oh}$ = 24mA |
| | Output High Voltage (All other outputs) | 2.4 | | V | $I_{oh}$ = 4mA |
| $V_{od}$ | Differential Output Voltage (TX±) | -500<br>500 | -1200<br>1200 | mV<br>mV | 78Ω termination and 150Ω from each output to $V_{dd}$. |
| $V_{out}$ | Output Voltage (BSR) | 2.25 | 2.75 | V | 10.0 KΩ external resistor to $V_{dd}$. |
| | Output Voltage (OSR) | 2.25 | 2.75 | V | 24.9 KΩ external resistor to $V_{dd}$. |

**TABLE 9-2. OUTPUT PIN VALUES**

## 10.0 AC OPERATING CHARACTERISTICS AND TIMING

This sections provides timing diagrams and parameters for 83C795 signals. Table 10-1 indicates the timing diagrams included in this section. Table 10-2 indicates the timing parameters applying to each timing diagram. Except where otherwise noted, timing units are in nanoseconds.

| FIGURE | TITLE |
|---|---|
| 10-1 | System Clock Timing |
| 10-2 | Register Access Timing – Read |
| 10-3 | Register Access Timing – Write |
| 10-4 | 16-bit Register Access (I/O Pipe Only) |
| 10-5 | Host Memory Access (16-bit, ZWS) |
| 10-6 | Host Memory Access (8-bit, No ZWS) |
| 10-7 | Host Memory Access (8-bit, ZWS) |
| 10-8 | Host Memory Access (8-bit, No ZWS) |
| 10-9 | Rom Access (8-bit Only, Read Only) |
| 10-10 | DMA or Memory Cache Writes |
| 10-11 | DMA or Memory Cache Reads |
| 10-12 | EEPROM Interface |
| 10-13 | Transmit Timing – Start Of Transmission |
| 10-14 | Transmit Timing – End Of Transmission (last bit = 1) |
| 10-15 | Transmit Timing – End Of Transmission (last bit = 0) |
| 10-16 | Receive Timing – Start Of Packet |
| 10-17 | Receive Timing – End Of Packet |
| 10-18 | Collision Timing – AUI |
| 10-19 | Collision Timing – TP |
| 10-20 | Loopback Timing |
| 10-21 | SQE Test Timing |
| 10-22 | Link Test Pulse |
| 10-23 | ROM Dump (Test Mode) |

**TABLE 10-1. LIST OF TIMING DIAGRAMS**

| PARAMETER | DESCRIPTION | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| T1 | Register Read: data valid delay | 3 | – | 5 | Cycles |
| T2 | Register Read: data hold time | – | – | 15 | nsec |
| T6 | Address setup for register I/O | – | – | 18 | |
| T7 | Address hold for register I/O | – | – | 0 | |
| T8 | IORDY inactive delay from I/O strobe | – | – | 20 | |
| T9 | IORDY active delay from X1 | – | – | 28 | |
| T10 | IORDY tristate delay from I/O strobe | – | – | 14 | |
| T11 | Register write: data set up time | 15 | – | – | |
| T12 | Register write: data hold time | 20 | – | – | |
| T13 | IOW active time | 300 | – | – | |
| T14 | IOR active time | 300 | – | – | |
| T15 | $\overline{\text{MCS16}}$ active from LA address ($\overline{\text{FINE16}}$=0) | – | – | 15 | |
| T20 | $\overline{\text{M16CS}}$ inactive from LA address | – | – | 13 | |
| T21 | $\overline{\text{IO16CS}}$ active from SA address | – | – | 17 | |
| T22 | $\overline{\text{IO16CS}}$ inactive from SA address | – | – | 15 | |
| T23 | IORDY active delay from HOSTCLK | – | – | 25 | |
| T27 | $\overline{\text{ZWS}}$ tristate delay from SMEMx strobe | – | – | 11 | |
| T28 | $\overline{\text{ZWS}}$ active from MEMx strobe (16 bit) | – | 9 | 13 | |
| T29 | $\overline{\text{ZWS}}$ tristate from MEMx strobe | – | – | 11 | |
| T30 | $\overline{\text{SMEMR}}$ active to inactive | 425 | – | – | |
| T31 | $\overline{\text{SMEMR}}$ active to $\overline{\text{ROMCS}}$ active | – | – | 155 | |
| T32 | ROMCS active to SD valid | – | – | 270 | |
| T33 | ROM read: data hold time | 15 | – | – | |
| T35 | SA address setup for MEMx strobe | 18 | – | – | |
| T36 | SA adress hold for MEMx strobe | – | – | 0 | |
| T41 | IORDY inactive delay from MEM strobe | – | – | 23 | |
| T42 | IORDY active delay from X1 | – | – | 27 | |
| T43 | IORDY tristate delay from MEM strobe | – | – | 17 | |
| T46 | $\overline{\text{ZWS}}$ active delay from HOSTCLK | – | – | 17 | |
| T48 | EECS set up time | 100 | – | – | |
| T49 | EECS hold time | 100 | – | – | |
| T50 | $\overline{\text{RLED}}$ set up time | 100 | – | – | |
| T51 | $\overline{\text{RLED}}$ hold time | 100 | – | – | |
| T52 | EEDO delay | – | – | 200 | |

**TABLE 10-2. TIMING PARAMETERS**

| PARAMETER | DESCRIPTION | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| T53 | MA address active delay from X1 | – | – | 26 | nsec |
| T54 | MA address inactive delay from X1 | – | – | 10 | |
| T57 | $\overline{\text{RAMWR}}$ active delay from $\overline{\text{X1}}$ | – | – | 11 | |
| T58 | $\overline{\text{RAMWR}}$ inactive delay from X1 | – | – | 9 | |
| T59 | MA address set up to $\overline{\text{RAMWR}}$, $\overline{\text{RAMOE}}$ | 10 | – | – | |
| T60 | MA address hold from $\overline{\text{RAMWR}}$ | 15 | – | – | |
| T61 | Data valid delay time from X1 | – | – | 24 | |
| T62 | Data hold time from $\overline{\text{RAMWR}}$ | 15 | – | – | |
| T64 | $\overline{\text{RAMOE}}$ active delay from X1 | – | – | 12 | |
| T65 | $\overline{\text{RAMOE}}$ inactive delay from $\overline{\text{X1}}$ | – | – | 9 | |
| T66 | Data setup time to X1 | 12 | – | – | |
| T67 | Data hold time to X1 | 0 | – | – | |
| T68 | XTXE setup time to XTXC | 25 | – | – | |
| T69 | XTXE hold time from XTXC | 0 | – | – | |
| T70 | XTXD hold time from XTXC | 0 | – | – | |
| T71 | TX outputs delay to idle | – | – | 310 | |
| T72 | TX outputs stay high before idle | 240 | – | – | |
| T73 | XTXD setup time to XTXC | 20 | – | – | |
| T74 | TX outputs delay from XTXC - AUI | – | – | 100 | |
| | TX outputs delay from XTXC - TP | – | – | 100 | |
| T75 | XCOL active delay - AUI | – | – | 60 | |
| T76 | XCOL active delay - TP | – | – | 900 | |
| T77 | XCOL inactive delay - AUI | – | – | 350 | |
| T78 | XCOL inactive delay - TP | – | – | 160 | |
| T79 | XCRS active delay - AUI | – | – | 300 | |
| | XCRS active delay - TP | – | – | 60 | |
| T80 | XCRS inactive delay - AUI | – | – | 250 | |
| | XCRS inactive delay - TP | – | – | 160 | |
| T81 | Differential input reject pulse width - AUI | 8 | – | 35 | |
| | Differential input reject pulse width - TP | 8 | 20 | 30 | |
| T82 | Acquisition time from PLL - AUI | – | – | 700 | |
| | Acquisition time from PLL - TP | – | – | 950 | |
| T83 | XRXD stable from XRXC | ±40 | – | – | |
| T84 | SQE test start delay - TP | – | 900 | – | |
| T85 | SQE test duration - TP | – | 1000 | – | |
| T86 | Loopback setup time | – | – | – | |
| T87 | Loopback hold time | – | – | – | |

**TABLE 10-2. TIMING PARAMETERS (cont.)**

| PARAMETER | DESCRIPTION | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| T88 | X1 clock period | 45 | 50 | – | nsec |
| T89 | X1 clock width high | 22.5 | 25 | – | |
| T90 | X1 clock rise time | – | – | 3 | |
| T91 | X1 clock fall time | – | – | 3 | |
| T92 | X1 clock width low | 22.5 | 25 | – | |
| T93 | Link test pulse width | – | 100 | – | |
| T94 | I/O write recovery time | 2 | – | – | cycles |

**TABLE 10-2. TIMING PARAMETERS (cont.)**

**NOTES**

1. All numbers are in nanoseconds except where otherwise designated.
2. All outputs are measured under 50pF load.
3. The external Manchester Encoder/Decoder port is multiplexed out on other pins in certain test modes only. Use the table below to determine which Manchester signals correspond to which pin names.

| MANCHESTER SIGNALS | 83C795 I/O PINS |
|---|---|
| XTXD | IRQ1 |
| XLOOP | IRQ2 |
| XCRS | IRQ3 |
| XRXC | IRQ4 |
| XRXD | IRQ5 |
| XCOL | IRQ6 |
| XTXC | IRQ7 |
| XTXE | $\overline{\text{ROMCS}}$ |

**TABLE 10-3. TEST PIN I/O MATCHING**

Refer to Chapter 4 for more details.

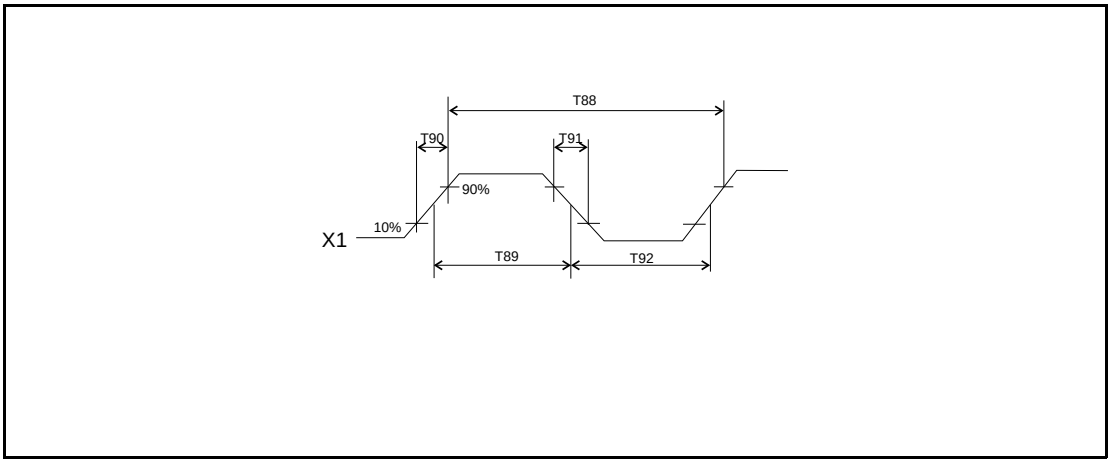


**FIGURE 10-1. SYSTEM CLOCK TIMING**

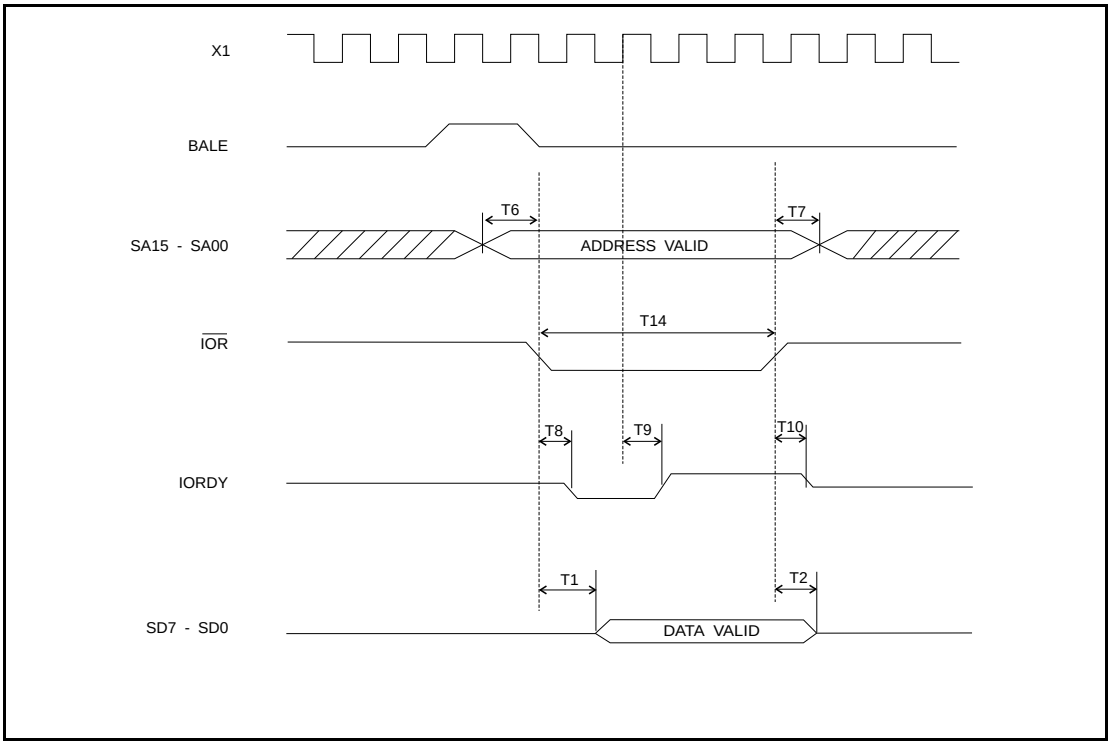


**FIGURE 10-2. REGISTER ACCESS TIMING - READ**

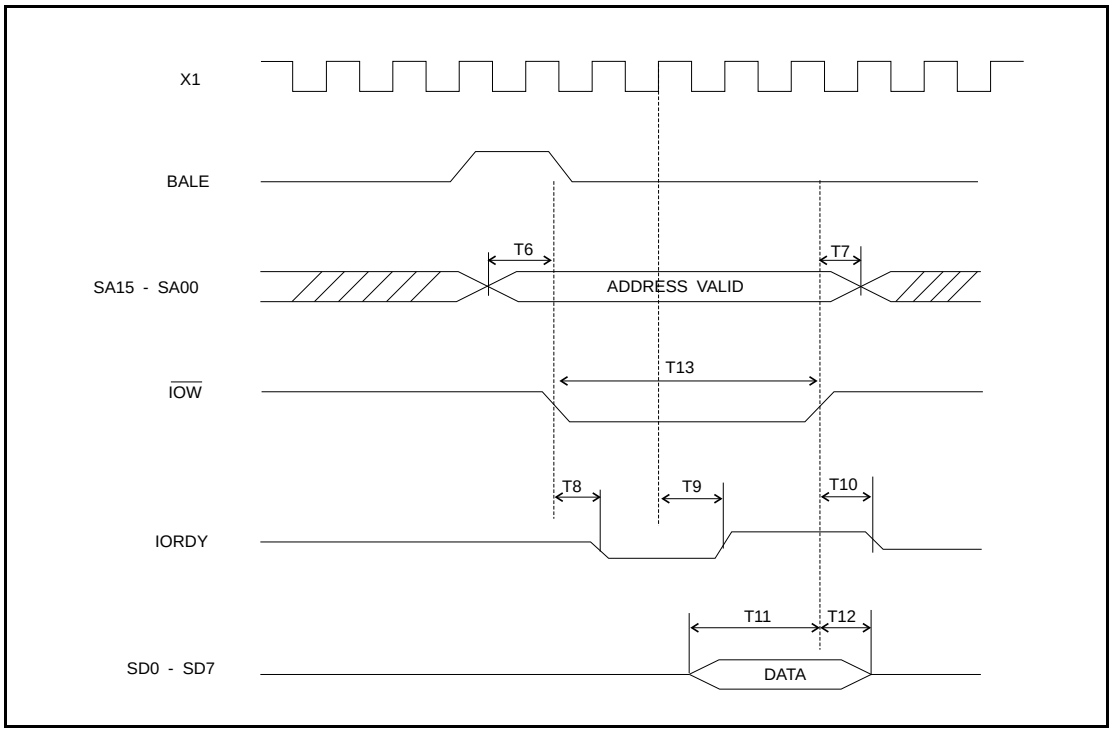


**FIGURE 10-3. REGISTER ACCESS TIMING - WRITE**

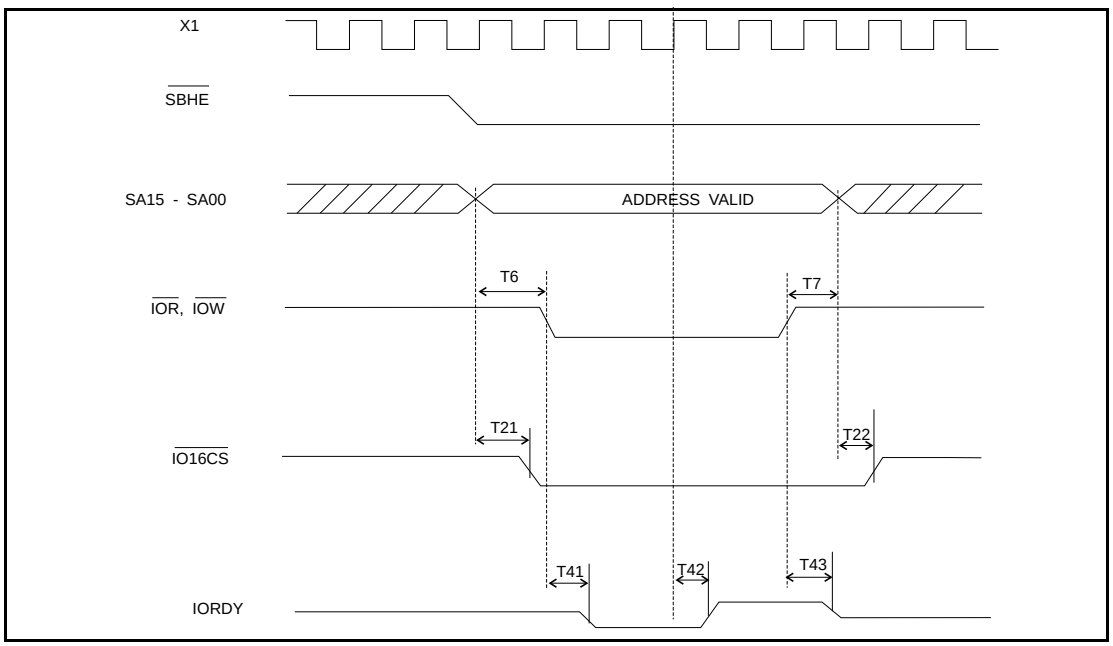


**FIGURE 10-4. 16-BIT REGISTER ACCESS (I/O PIPE ONLY)**

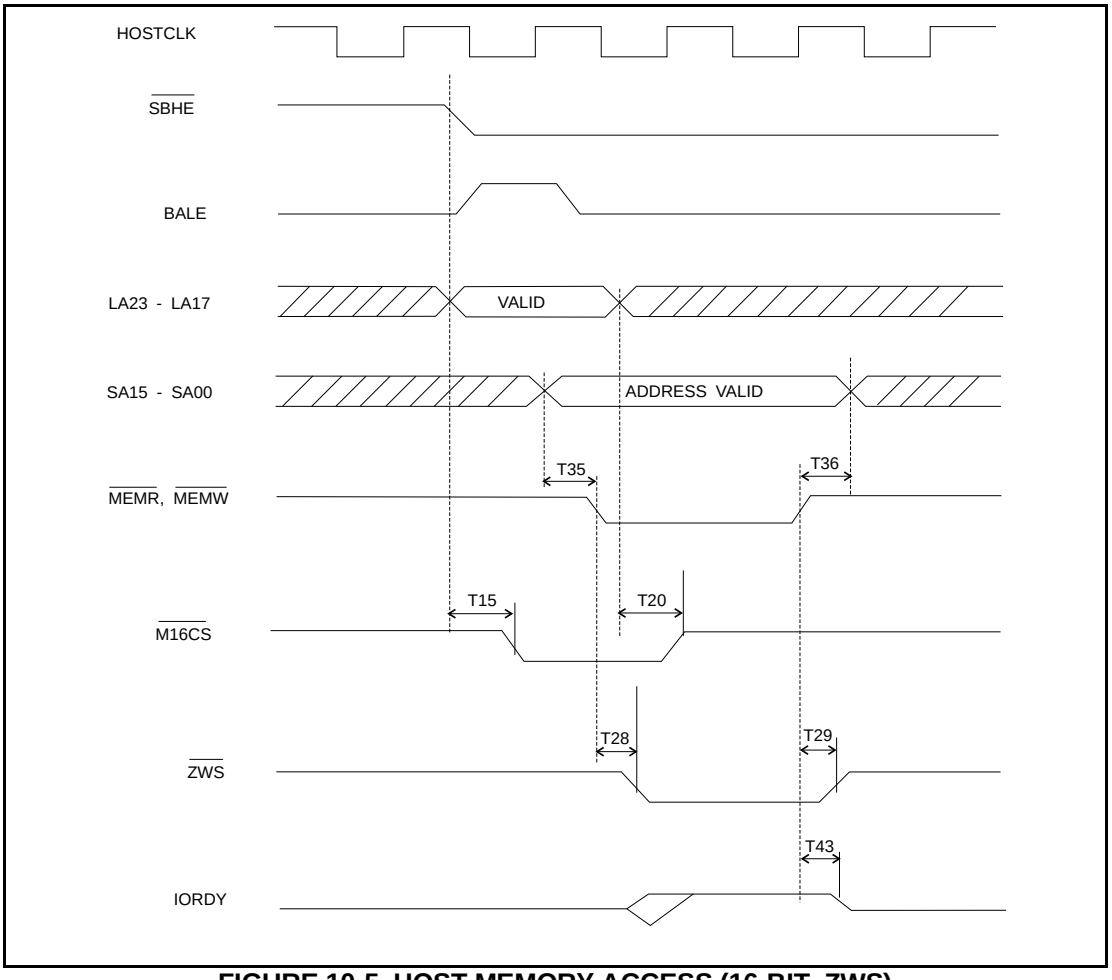


**FIGURE 10-5. HOST MEMORY ACCESS (16-BIT, ZWS)**

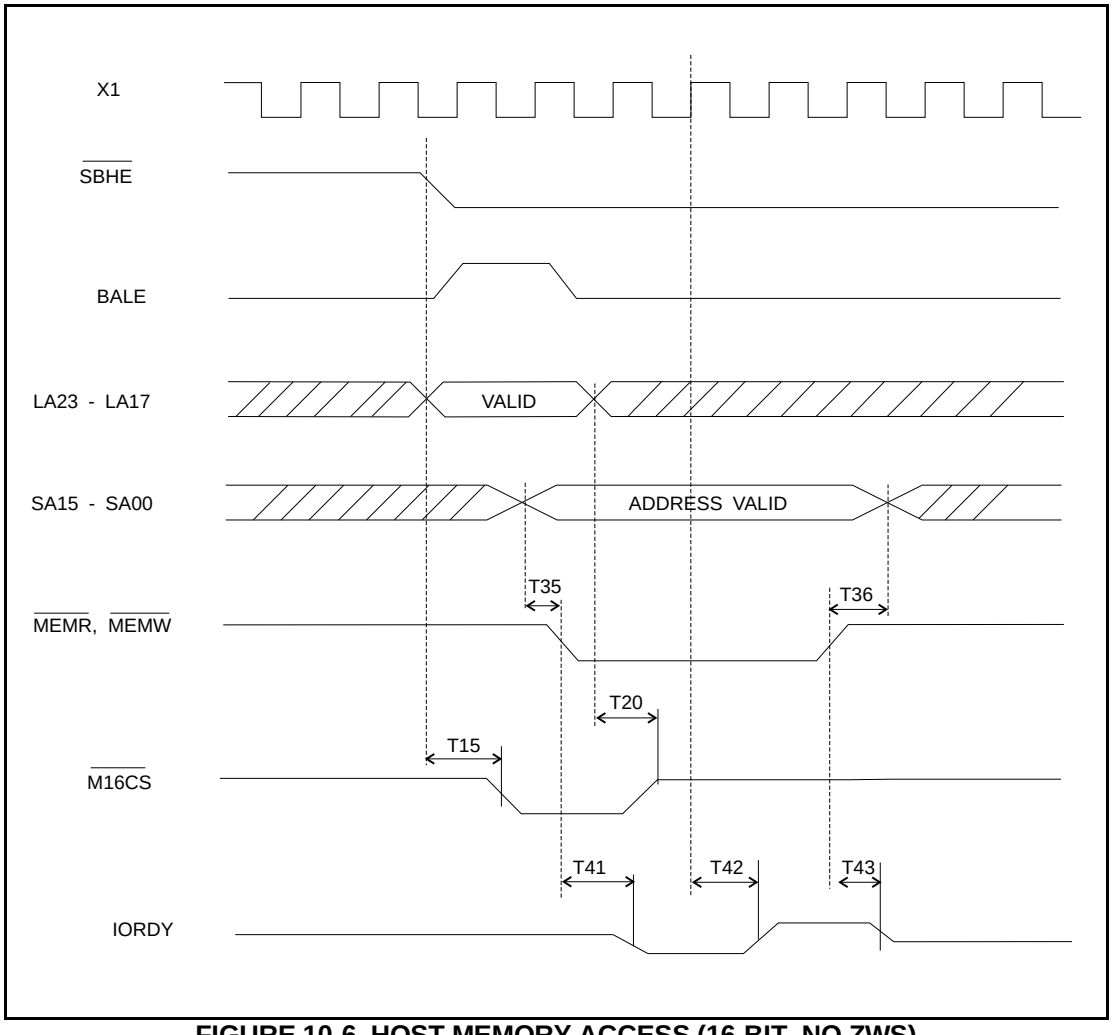


**FIGURE 10-6. HOST MEMORY ACCESS (16-BIT, NO ZWS)**

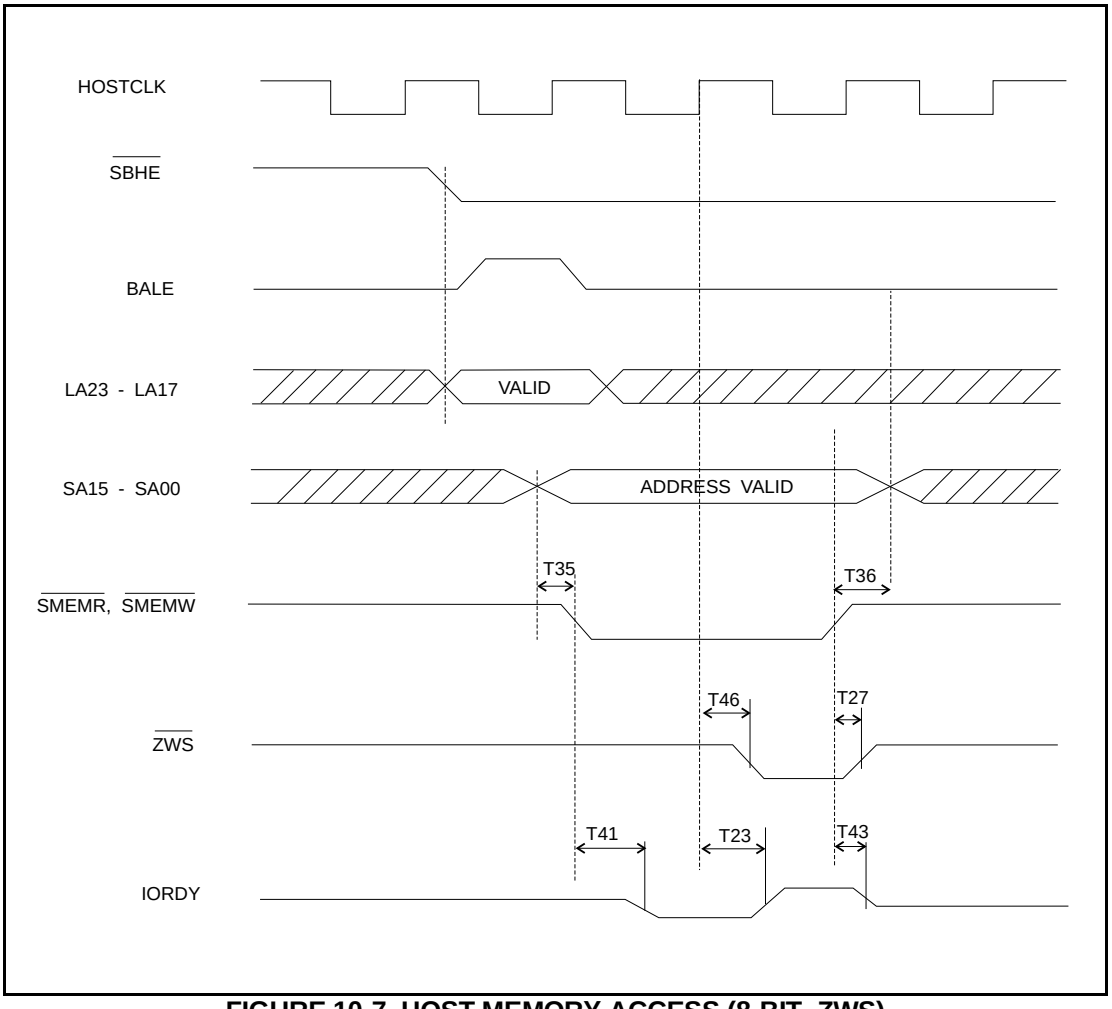


**FIGURE 10-7. HOST MEMORY ACCESS (8-BIT, ZWS)**

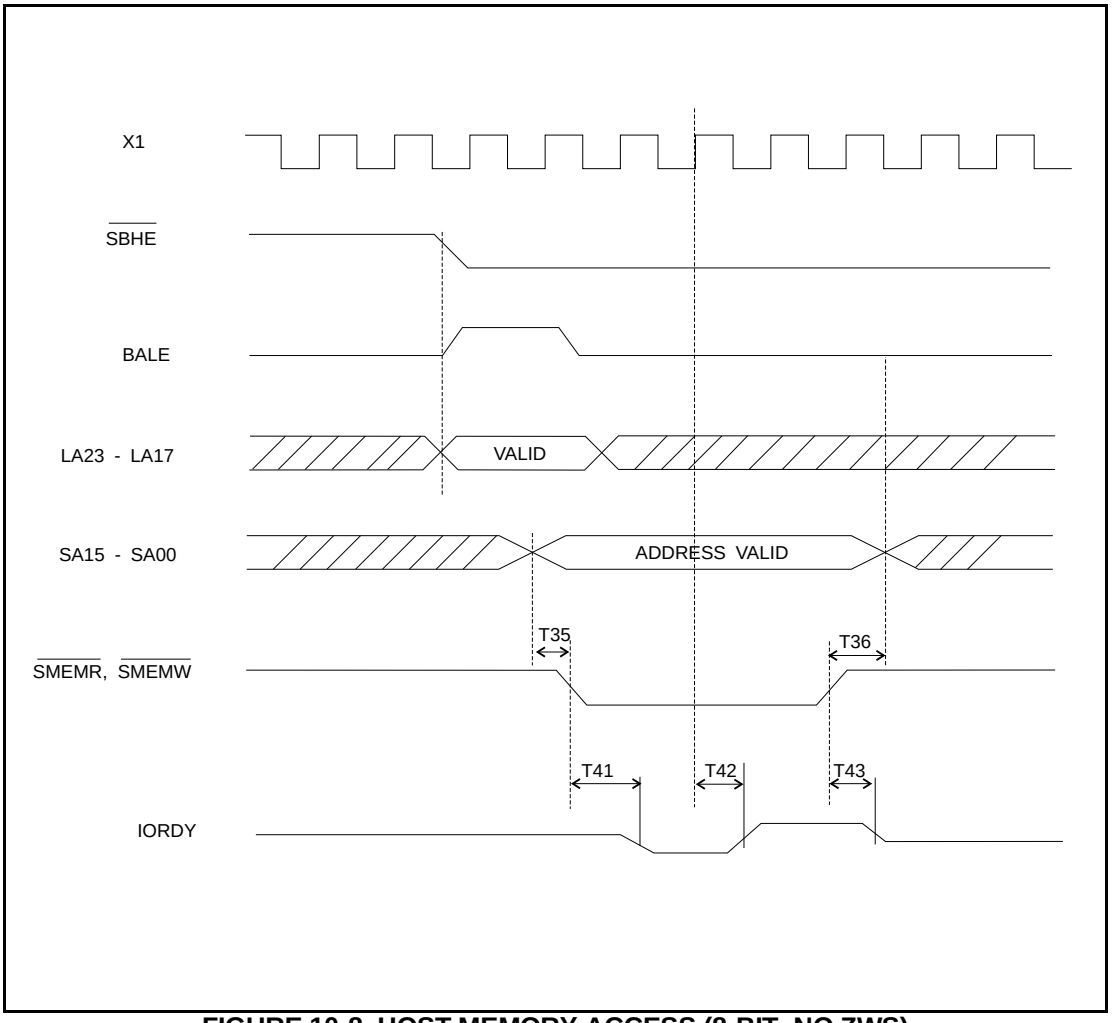


**FIGURE 10-8. HOST MEMORY ACCESS (8-BIT, NO ZWS)**

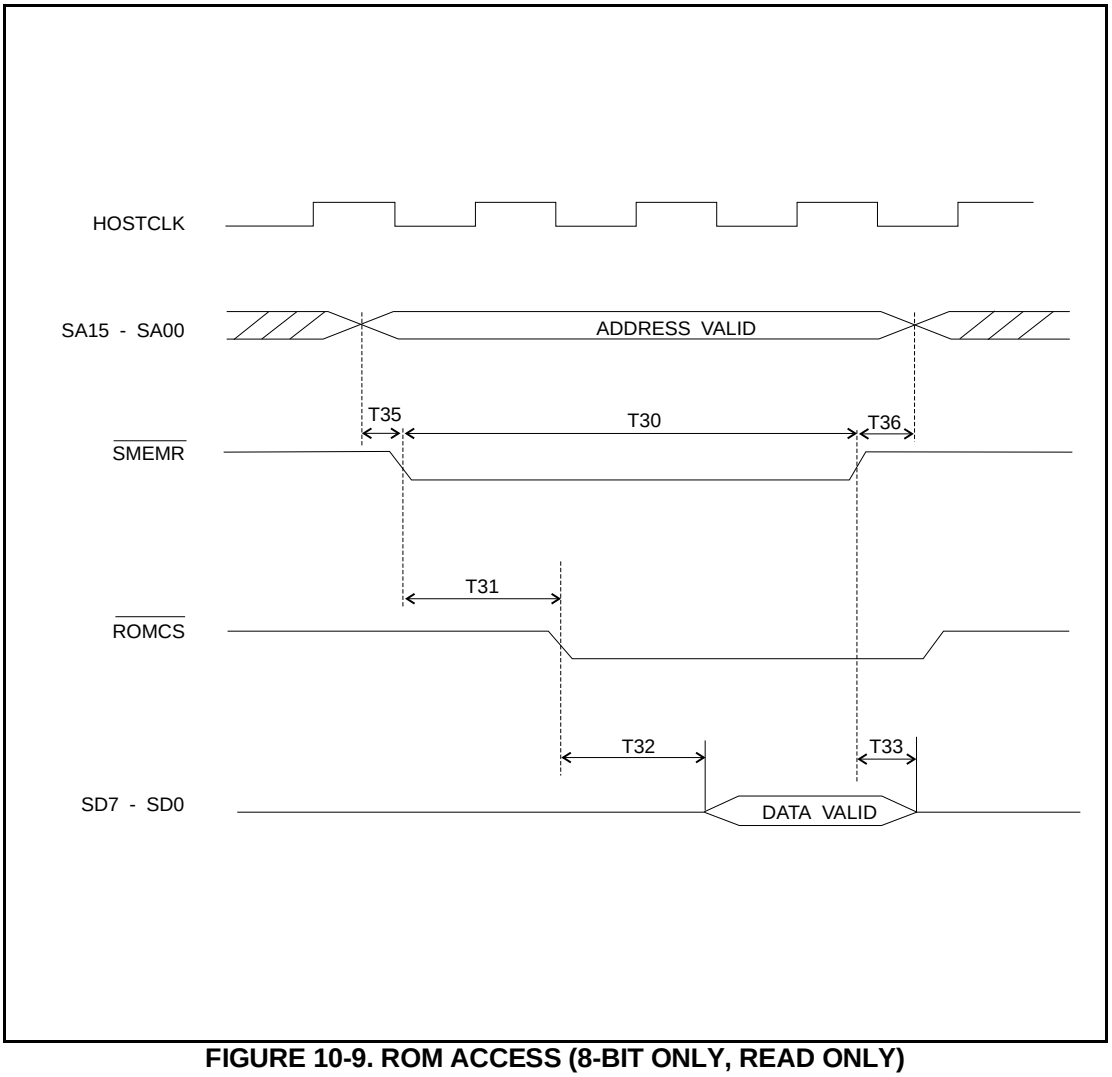


**FIGURE 10-9. ROM ACCESS (8-BIT ONLY, READ ONLY)**

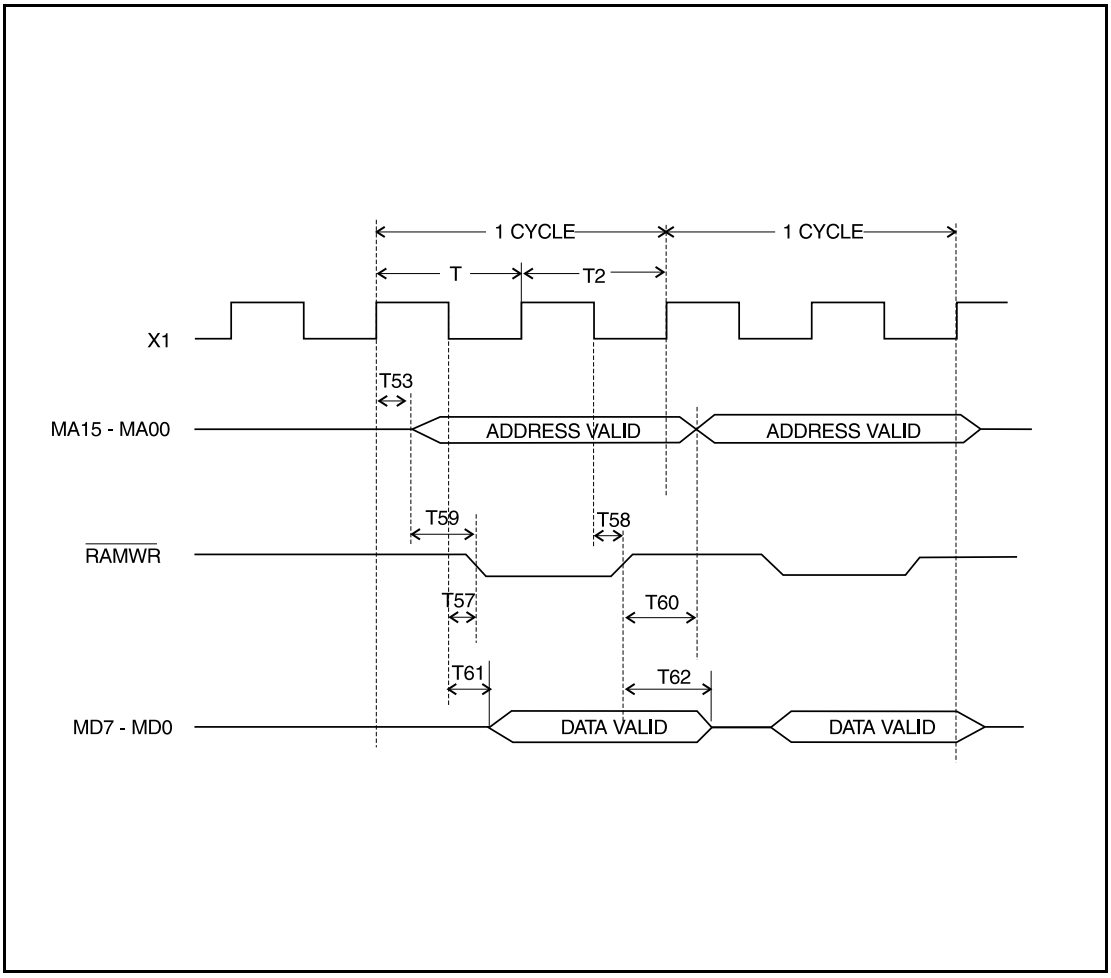


**FIGURE 10-10. DMA OR MEMORY CACHE WRITES**

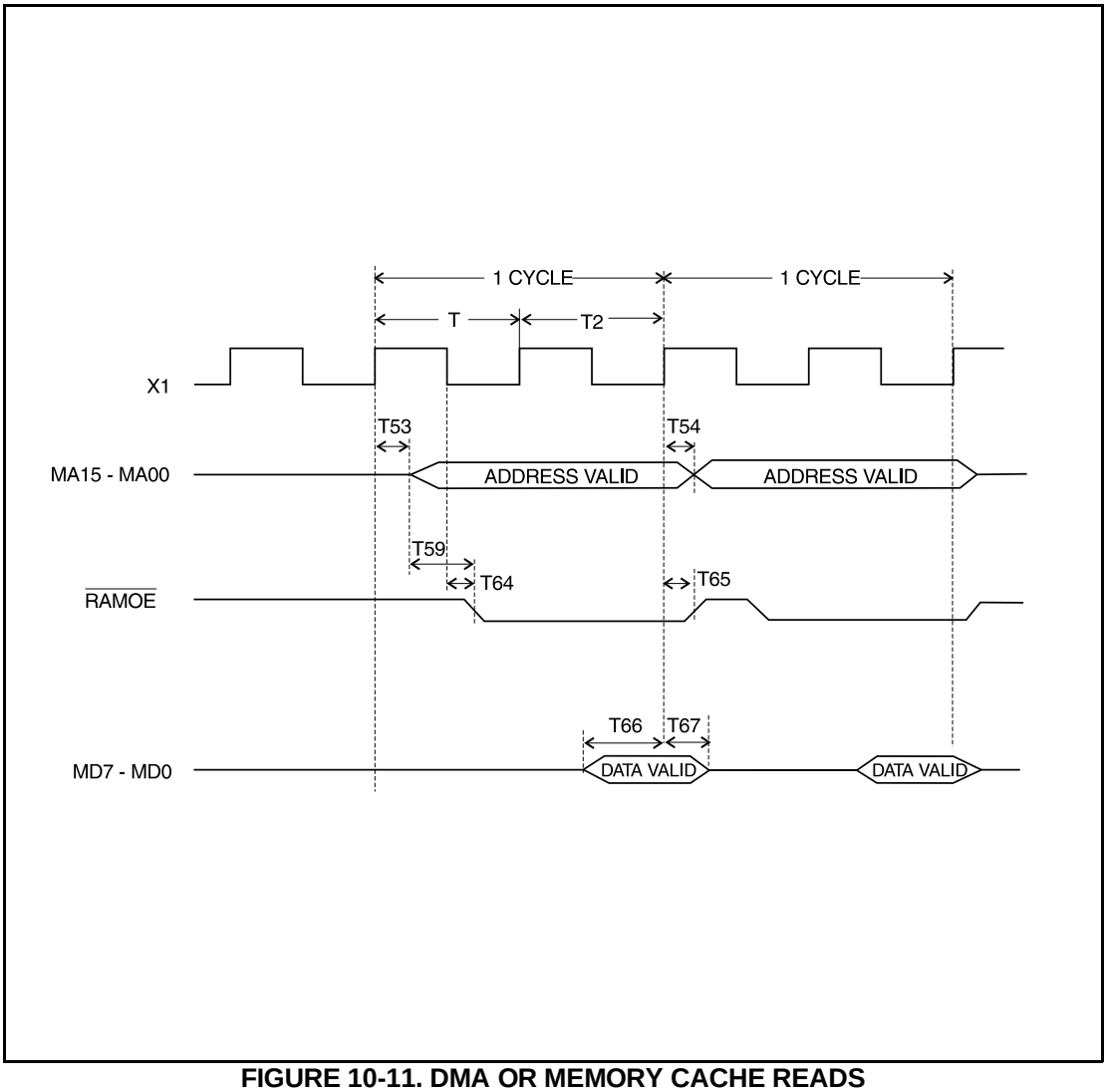


**FIGURE 10-11. DMA OR MEMORY CACHE READS**

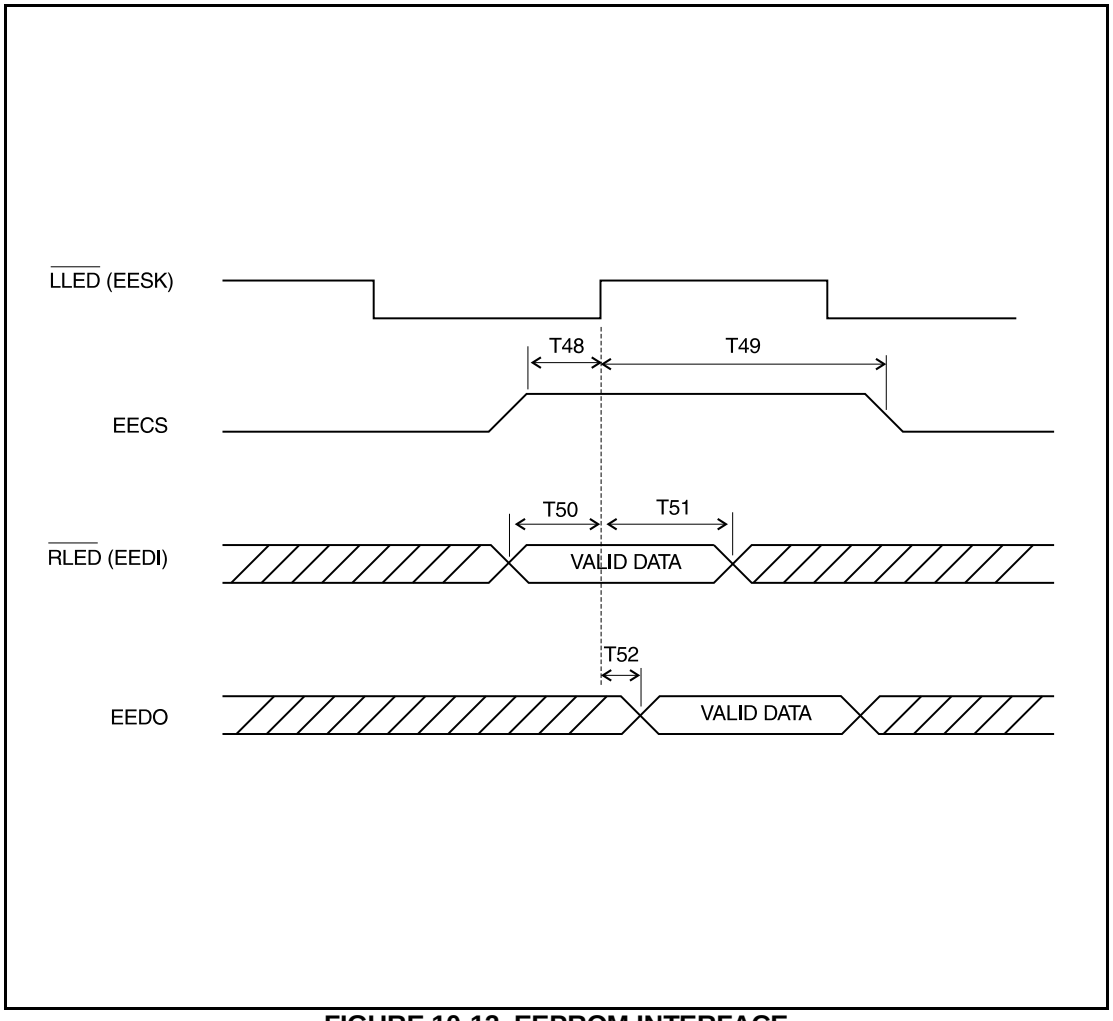


**FIGURE 10-12. EEPROM INTERFACE**

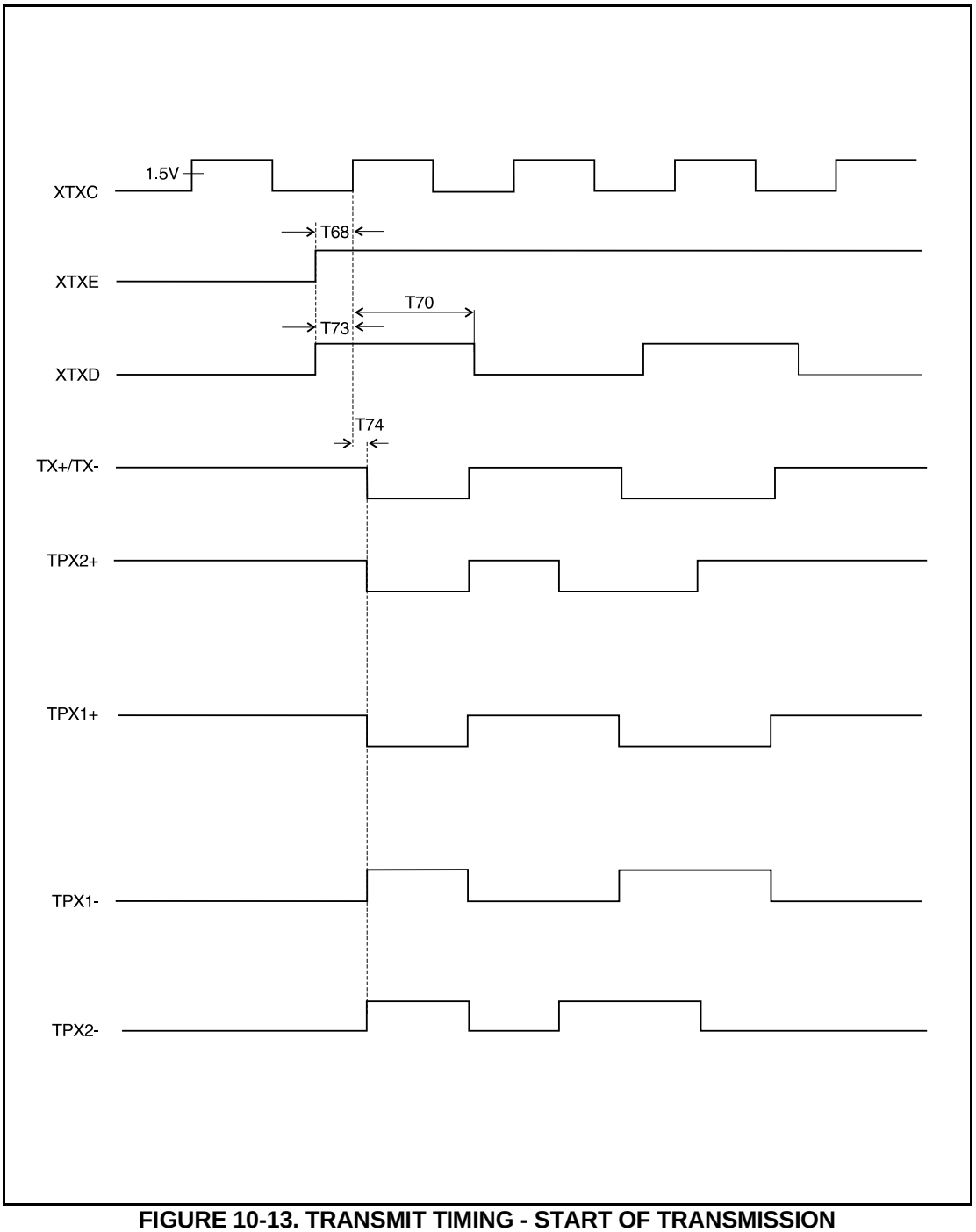


**FIGURE 10-13. TRANSMIT TIMING - START OF TRANSMISSION**

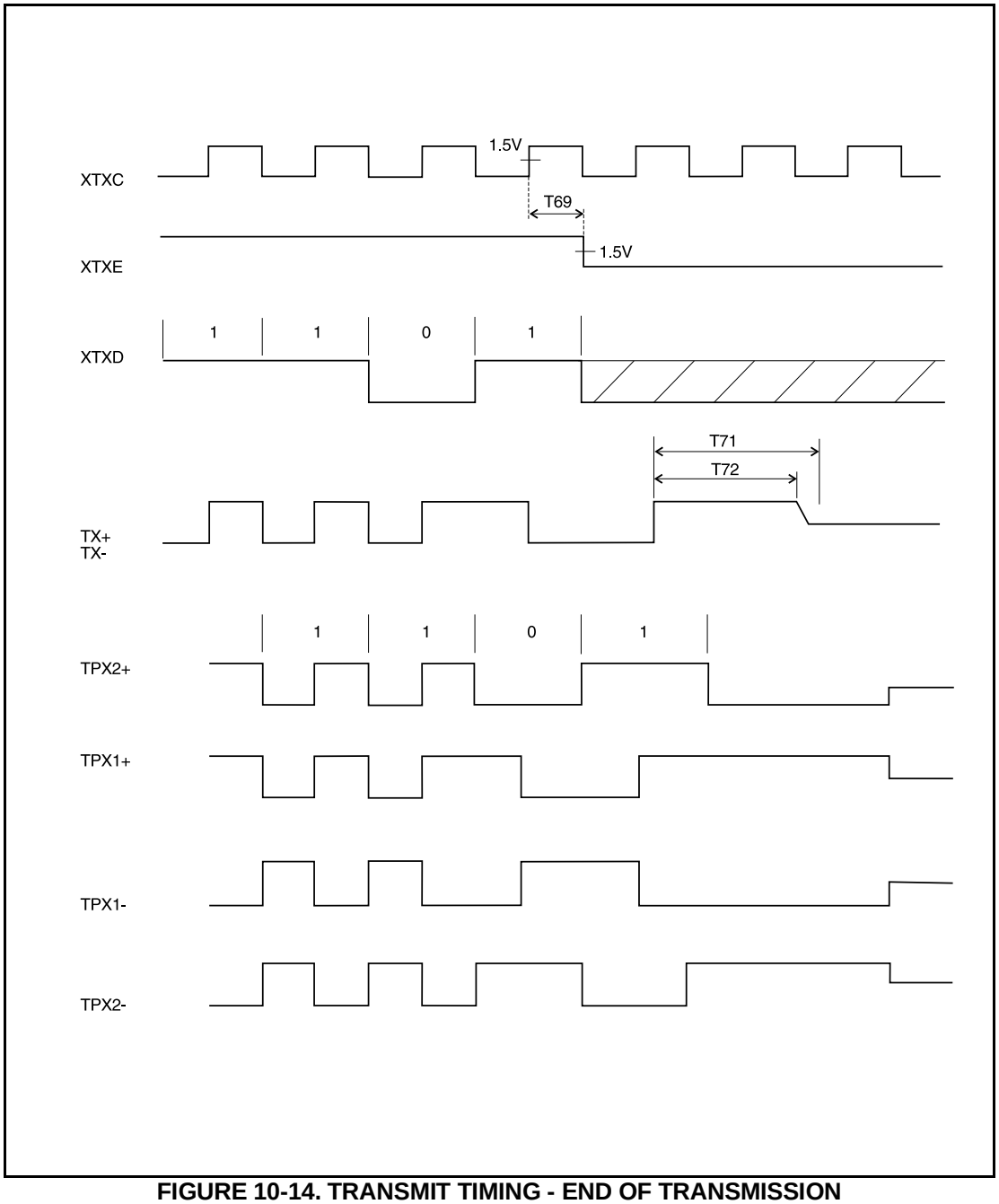


**FIGURE 10-14. TRANSMIT TIMING - END OF TRANSMISSION (LAST BIT = 1)**

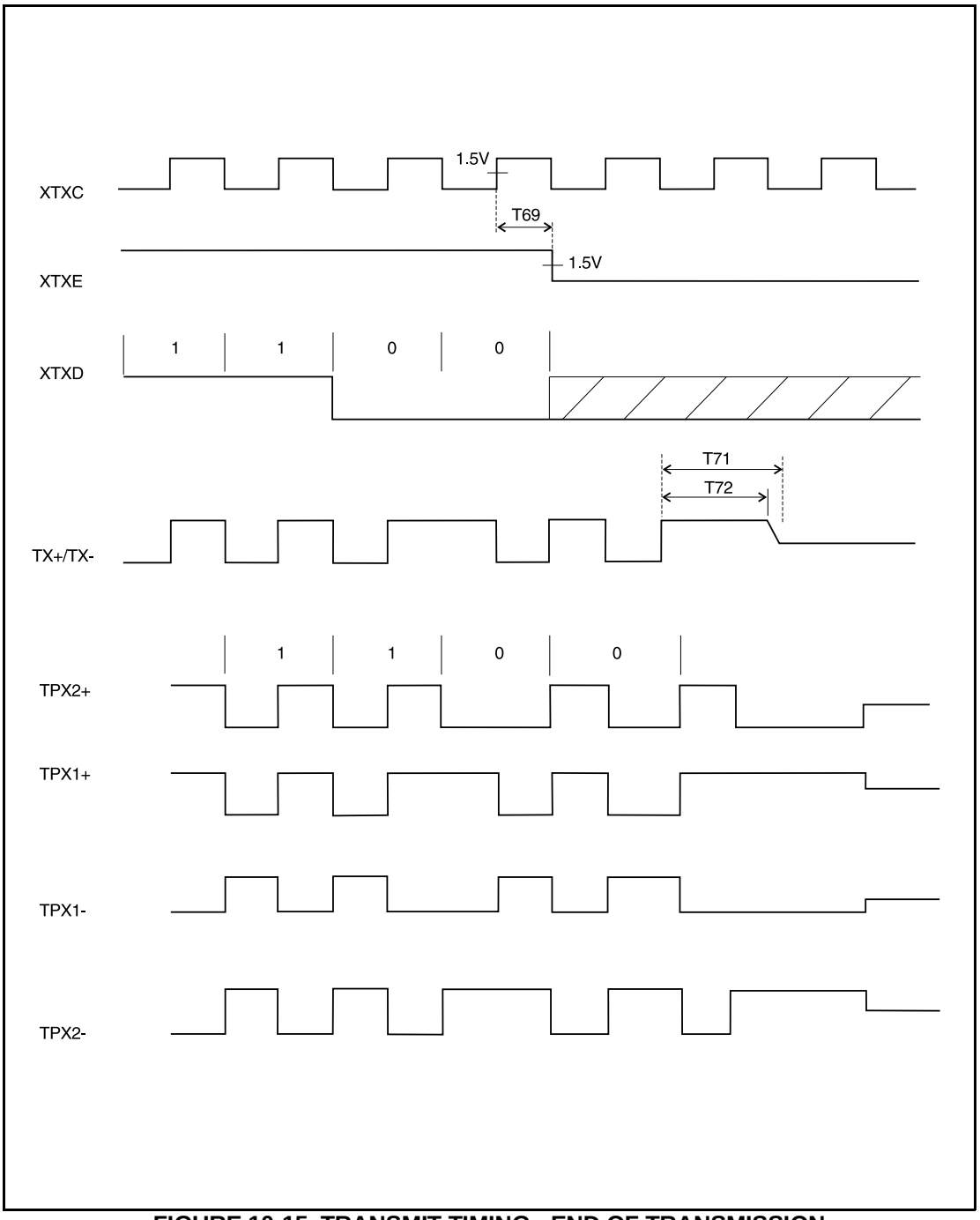


**FIGURE 10-15. TRANSMIT TIMING - END OF TRANSMISSION (LAST BIT = 0)**

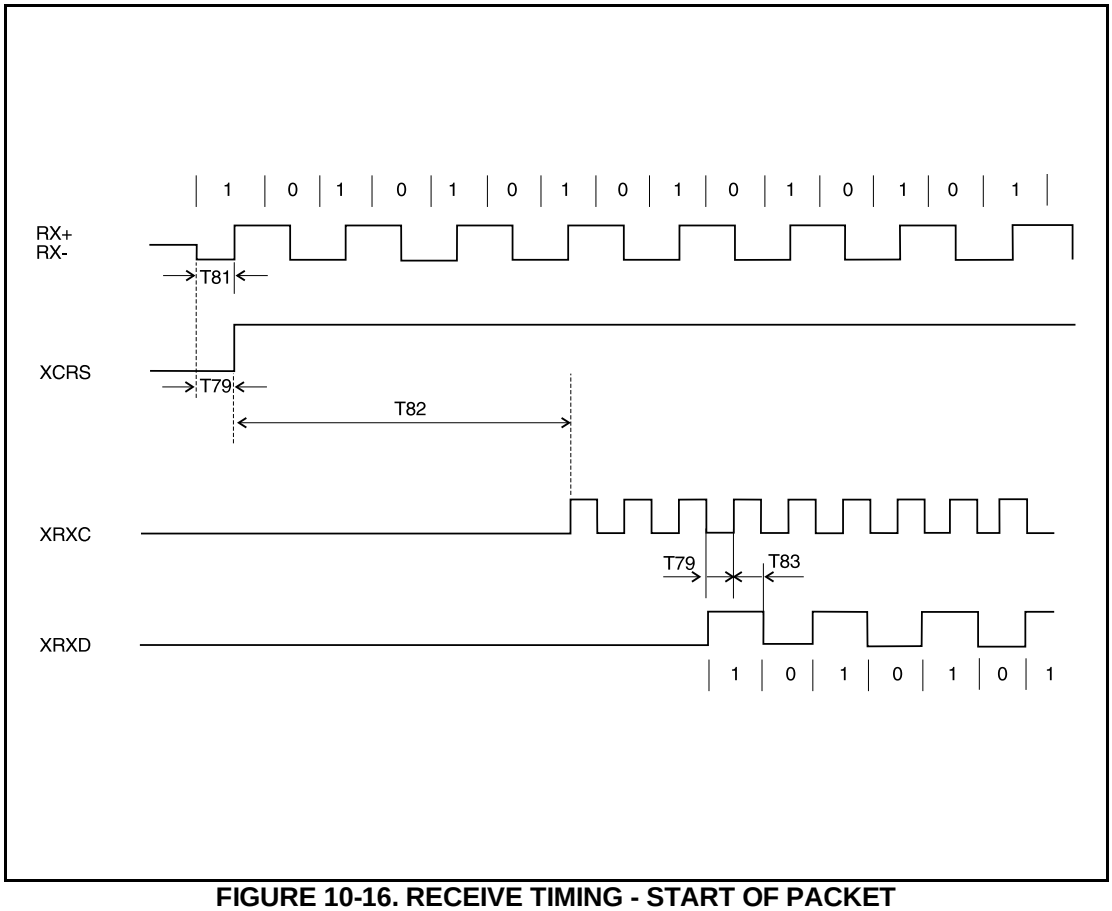


**FIGURE 10-16. RECEIVE TIMING - START OF PACKET**

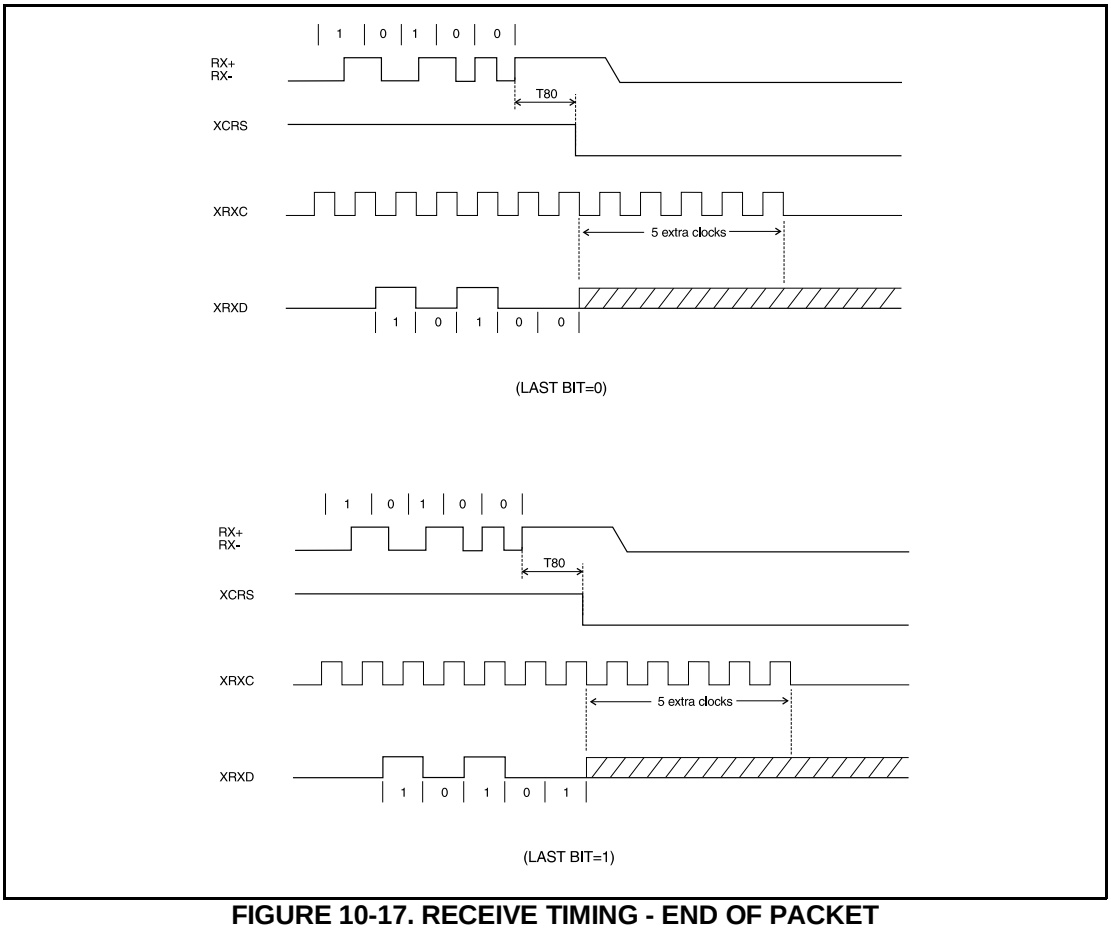


**FIGURE 10-17. RECEIVE TIMING - END OF PACKET**

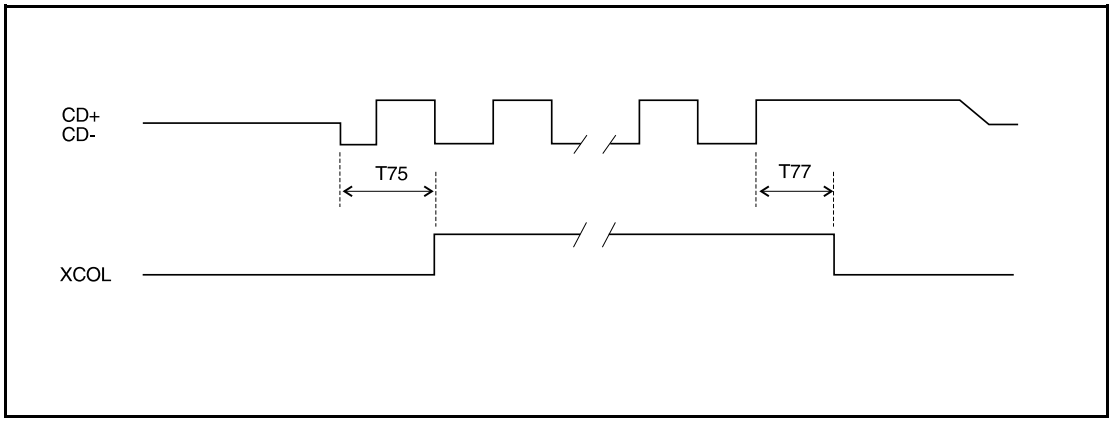


**FIGURE 10-18. COLLISION TIMING - AUI**

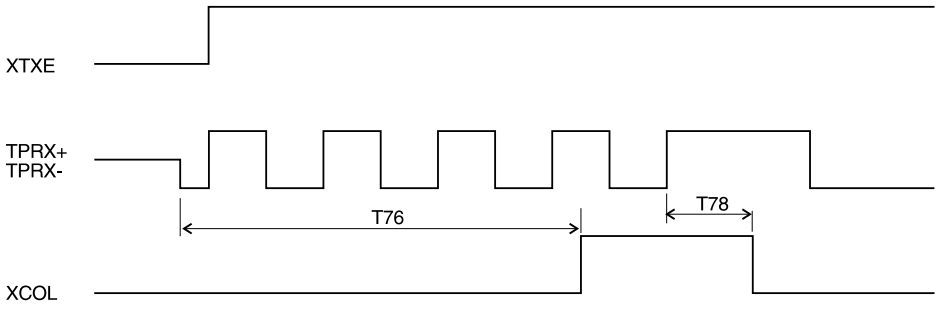


**FIGURE 10-19. COLLISION - TP**

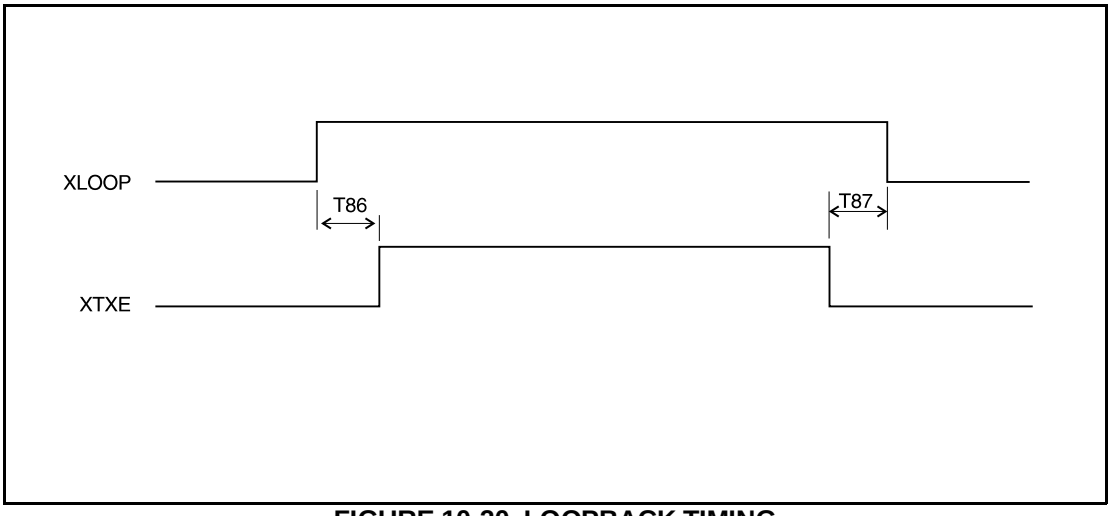


**FIGURE 10-20. LOOPBACK TIMING**

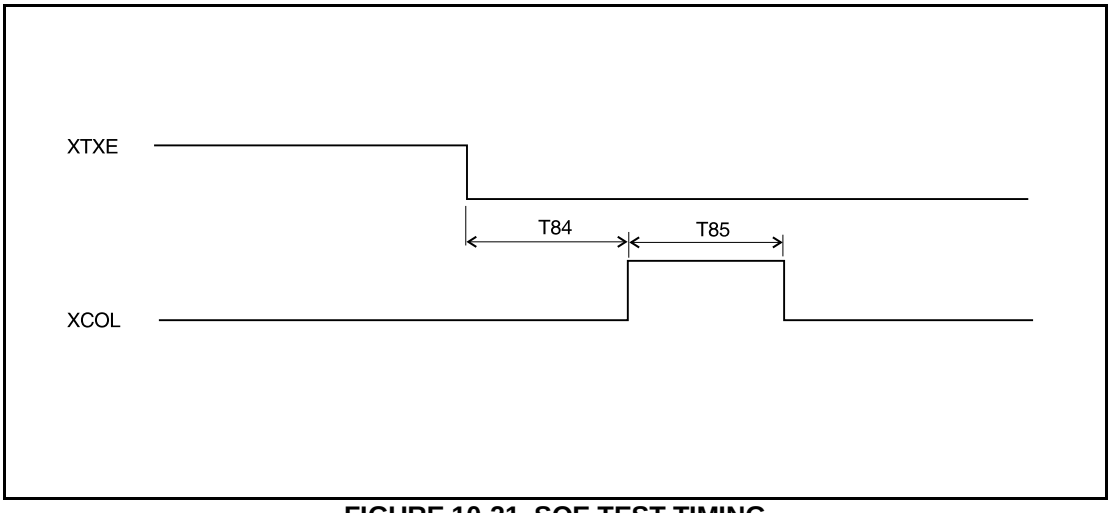


**FIGURE 10-21. SQE TEST TIMING**

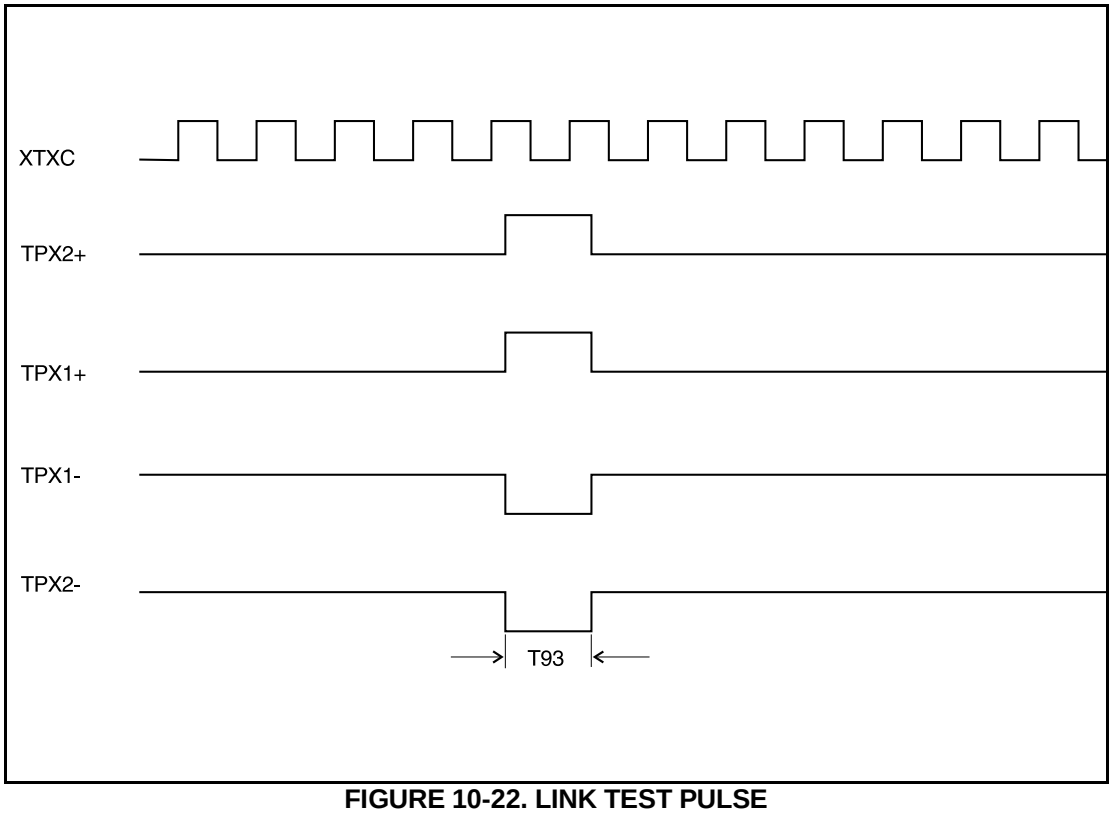


**FIGURE 10-22. LINK TEST PULSE**

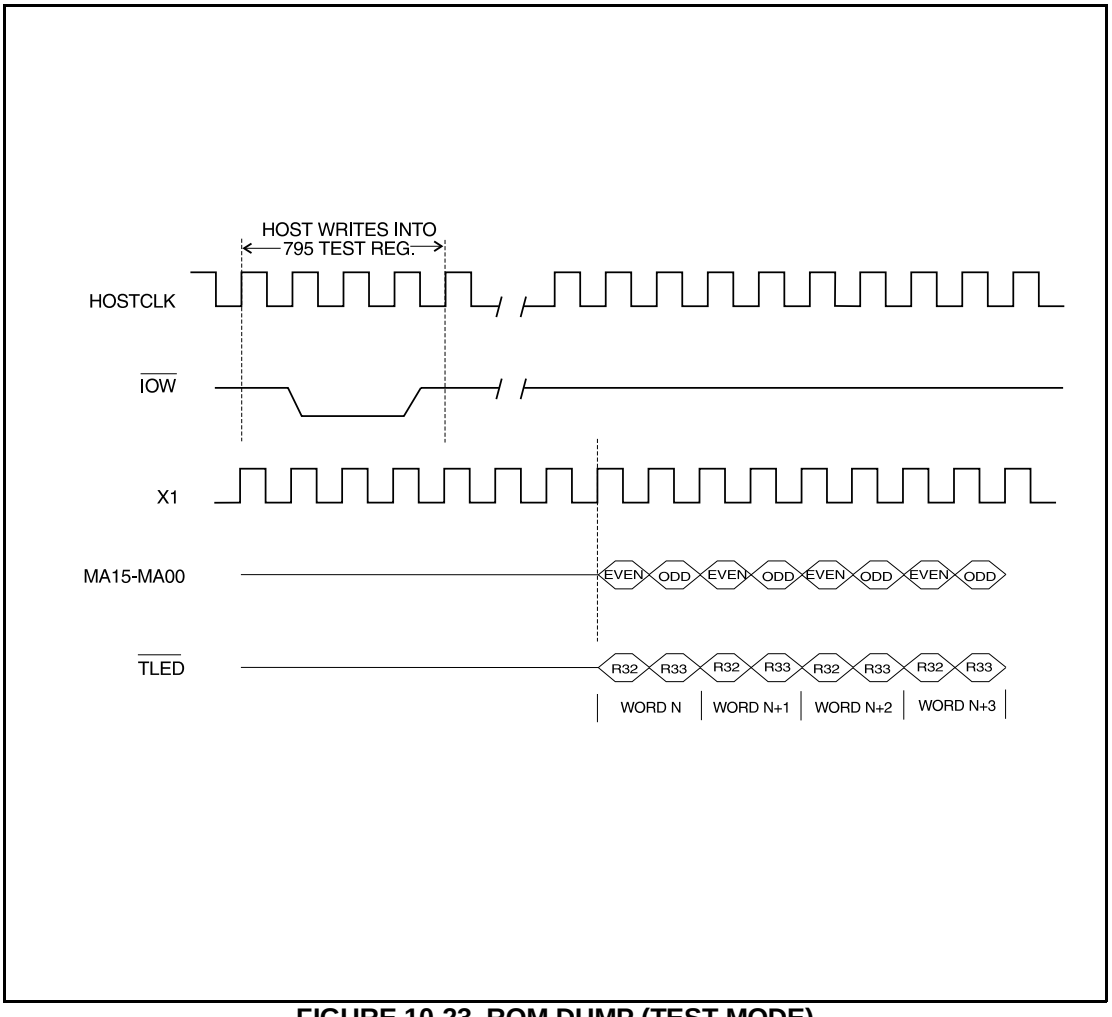


**FIGURE 10-23. ROM DUMP (TEST MODE)**

## 11.0 PACKAGE DIMENSIONS

Figure 11-1 illustrates the 160-pin PQFP package.

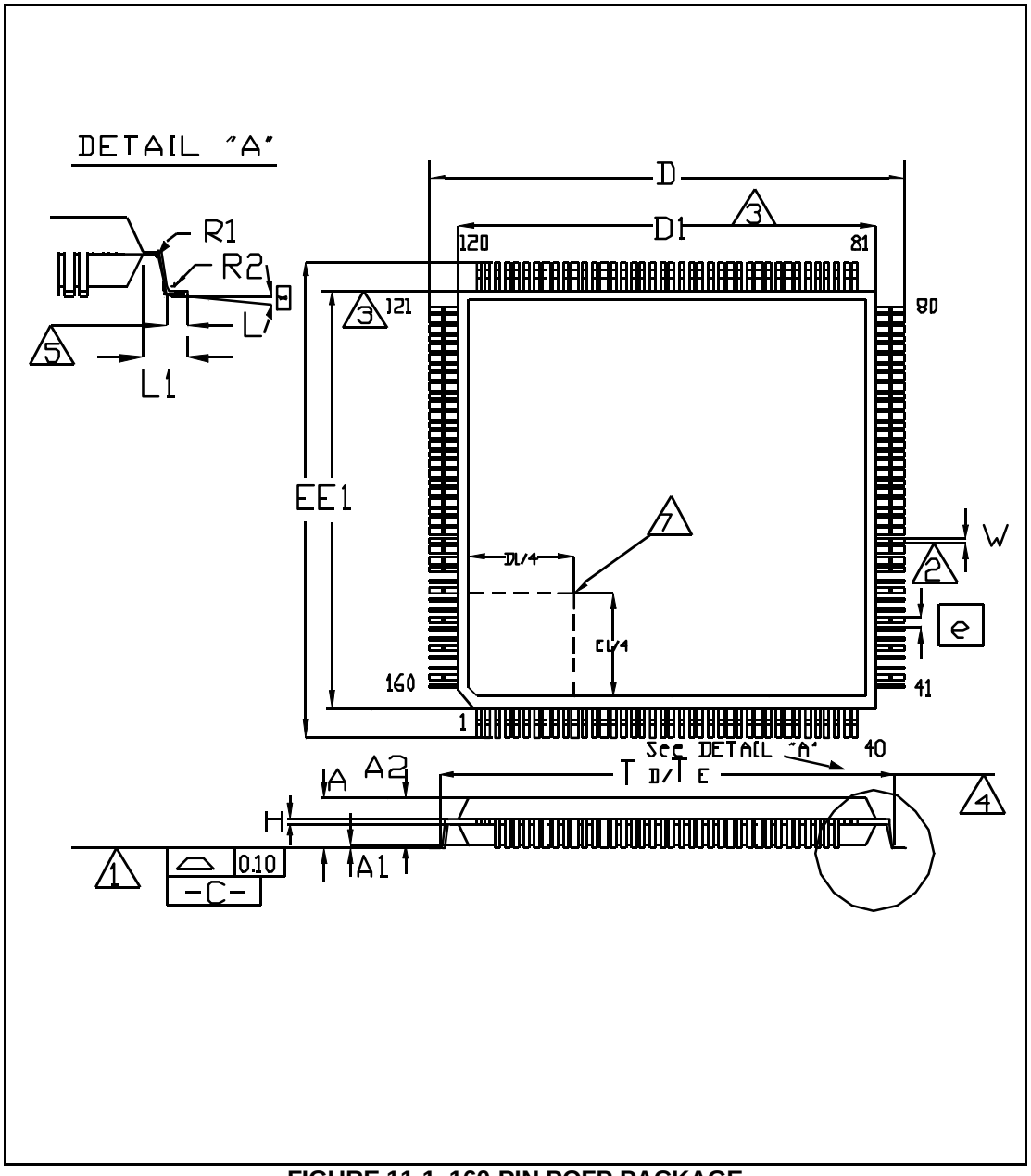


**FIGURE 11-1. 160-PIN PQFP PACKAGE**

| LETTER | MIN | NOM | MAX | | MIN | NOM | MAX |
|---|---|---|---|---|---|---|---|
| A | | | 4.07 | L | | 1.60 | |
| A1 | 0.05 | | 0.5 | P | | 0.65BSC | |
| A2 | 3.10 | | 3.67 | Θ | 0° | | 7° |
| D | 30.95 | 31.20 | 31.45 | W | 0.20 | | 0.40 |
| D1 | 27.90 | 28.00 | 28.10 | R1 | | 0.20 | |
| E3 | 30.95 | 31.20 | 31.45 | R2 | | 0.30 | |
| E1 | 27.90 | 28.00 | 28.10 | $T_D$ | | 30.45 | |
| H | 0.10 | | 2.20 | $T_E$ | | 30.45 | |
| L | 0.65 | 0.80 | 0.95 | | | | |

**TABLE 11-1. PACKAGE DIMENSIONS**

**Notes:**

1. Coplanarity is 0.100 mm. maximum.
2. Tolerance on the position of the leads is 0.120 mm maximum.
3. Package body dimensions D1 and E1 do not include the mold protrusion. Maximum mold protrusion is 0.25 mm.
4. Dimensions $T_D$ and $T_E$ are important for testing by robotic handler.
5. Dimensions for foot length L when measured at the centerline of the leads are given at the table. Dimension for foot length L when measured at the gauge plane 0.25 mm above the seating plane, is 0.78-1.03 mm.
6. Controlling dimension is millimeter.
7. Details of pin 1 identifier are optional but must be located within the zone indicated.

## INDEX