smsc™

STANDARD
MICROSYSTEMS
CORPORATION

LAN91C95

# ISA/PCMCIA Full Duplex Single-Chip Ethernet and Modem Controller with RAM

## FEATURES

- ISA/PCMCIA Single Chip Ethernet Controller With Modem Support
- 6 Kbytes Built-In RAM
- Supports IEEE 802.3 (ANSI 8802-3) Ethernet Standards
- Full Duplex Support
- Hardware Memory Management Unit
- Built-In AUI and 10BASE-T Network Interfaces
- Simultasking **J** - Early Transmit and Early Receive Functions
- Advanced Power Management Features/Including Magic Packet Frame Control
- Software Compatible with LAN91C92/ LAN91C94 (in ISA Mode)
- Configuration Registers Implement Cardbus Multi-Function Specification V3.0 with Backward Compatibility to V2.1
- Interfaces Directly to Lucent Technologies and Conexant (formerly Rockwell) Modem Chipsets
- On-Chip Attribute Memory (CIS) of up to 512 Bytes (On Even Addresses) For Card Configuration Information; Expandable Externally
- Option for Serial or Parallel EEPROM for CIS
- Optional External Flash Capability for XIP (Execute in Place)
- Automatic Technology to Detect 10 BASE-T TX/RX Polarity Reversal
- Low Power CMOS Design
- Supports Magic Packet Wakeup
- 128 Pin TQFP Package

### Bus Interface

- Direct Interface to ISA and PCMCIA with No Wait States
- High Impedance Speaker Interface
- Flexible Bus Interface
- 16-Bit Data and Control Paths
- Fast Access Time (40 ns)
- Pipelined Data Path
- Handles Block Word Transfers for Any Alignment
- High Performance Chained ("Back-to-Back") Transmit and Receive
- Flat Memory Structure for Low CPU Overhead
- Dynamic Memory Allocation Between Transmit and Receive
- Buffered Architecture, Insensitive to Bus Latencies (No Overruns/Underruns)
- Supports Boot PROM for Diskless ISA Applications

**Simultasking is a trademark and SMSC is a registered trademark of Standard Microsystems Corporation**

# TABLE OF CONTENTS

## Related Documentation

1. PCMCIA 5.0 standard (for multi-function extensions)
2. AT&T HSM288xCF Modem Chip Set Data Sheet - July 5, 1994, V.35 bis extension 1996
3. Rockwell ACL & ACFL series L39 based modem designer’s guide, 1994/95/97

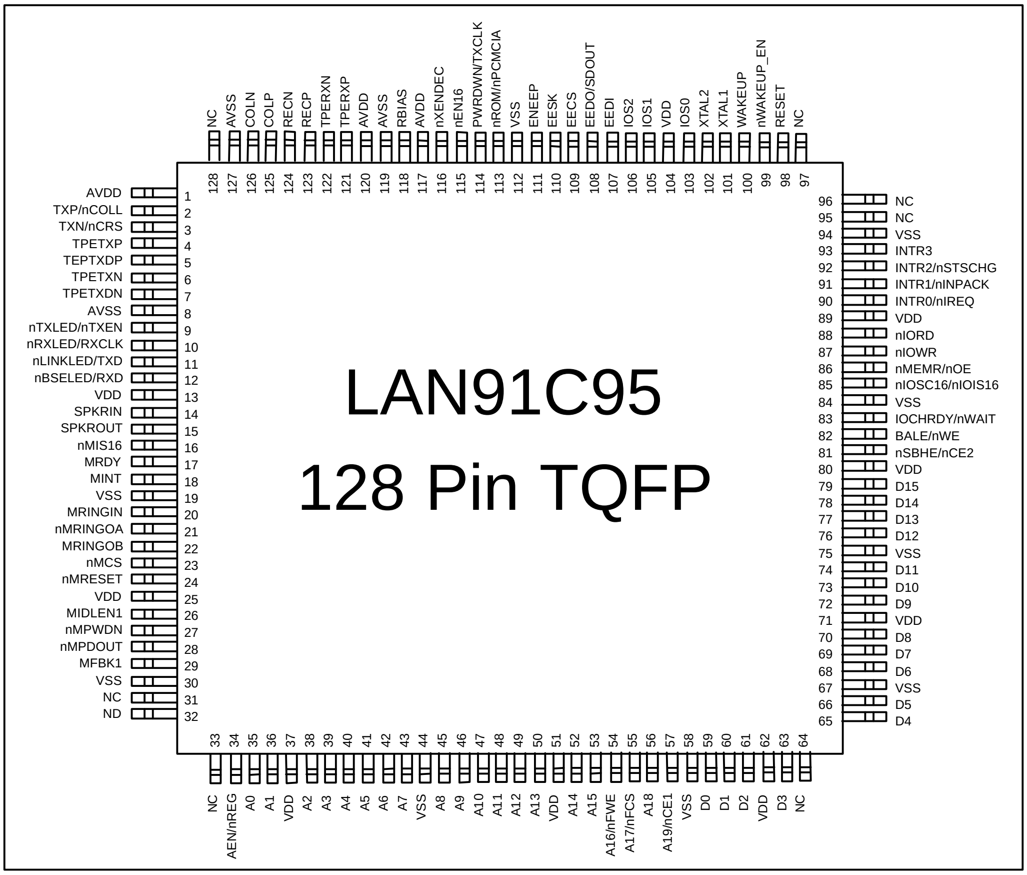
### Network Interface

- Integrated 10BASE-T Transceiver Functions:
  - Driver and Receiver
  - Link Integrity Test
  - Receive Polarity Detection and Correction
- Integrated AUI Interface
- Implements 10 Mbps Manchester Encoding/Decoding and Clock Recovery
- Automatic Retransmission, Bad Packet Rejection, and Transmit Padding
- External and Internal Loopback Modes
- Four Direct Drive LED Outputs for Status/ Diagnostics

### Software Drivers

- Uses Certified LAN9000 Drivers for Major Network Operating Systems
- Software Driver Compatible with LAN91C92, LAN91C94 and LAN91C100 (100 Mbps) and LAN91C110 (100 Mbps) Controllers in ISA Mode
- Software Driver Utilizes Full Capability of 32 Bit Microprocessor

## PIN CONFIGURATION



## GENERAL DESCRIPTION

The LAN91C95 is a VLSI Ethernet Controller that combines ISA and PCMCIA interfaces, as well as an interface to a companion modem chip set, in one chip. The LAN91C95 integrates all the MAC and physical layer functions as well as the packet RAM needed to implement a high performance 10BASE-T (twisted pair) node. For 10BASE5 (thick coax), 10BASE2 (thin coax), and 10BASE-F (fiber) implementations, the LAN91C95 interfaces to external transceivers via its AUI port. Only one additional IC is required on most applications.

The LAN91C95 occupies 16 I/O locations and no memory space except for PCMCIA attribute memory space. The same I/O space is used for both ISA and PCMCIA operations. The LAN91C95 can directly interface the ISA and PCMCIA buses and deliver no wait state operation. Its shared memory is sequentially accessed with 40ns access times to any of its registers, including its packet memory. No DMA services are used by the LAN91C95; virtually decoupling network traffic from local or system bus utilization. For packet memory management, the LAN91C95 integrates a unique hardware Memory Management Unit (MMU) with enhanced performance and decreased software overhead when compared to ring buffer and linked list architectures. The LAN91C95 is portable to different CPU and bus platforms due to its flexible bus interface, flat memory structure (no pointers), and its loosely coupled buffered architecture (not sensitive to latency).

The LAN91C95 interfaces directly with Rockwell International L39/C39 controller-based modems and Lucent Technologies' HSM288xCF modem.

## OVERVIEW

A unique architecture allows the LAN91C95 to combine high performance, flexibility, high integration and simple software interface.

The LAN91C95 incorporates the LAN91C92/4 functionality for ISA environments with several new features, as well as a PCMCIA interface and attribute registers that comply with the PCMCIA Multi-Function specification. Mode selection between ISA and PCMCIA is static and is done only once at the end of power on reset. The LAN91C95 consists of the same logical I/O register structure in ISA and PCMCIA modes. However, some of the signals used to access the PCMCIA differ from the ISA mode. Additional registers exist in the PCMCIA attribute space. The ROM memory space only exists in ISA mode and the attribute space only exists in PCMCIA mode.

I/O decoders are included in the LAN91C95's PCMCIA interface, with independent decoders for the LAN and for the modem functions.

These decoders are used whenever the LAN91C95 is used as a multi-function card, and they can be bypassed when only one function is enabled. The LAN91C95 also merges the LAN's internal interrupt source with the external modem interrupt connected to the LAN91C95.

The MMU (Memory Management Unit) architecture used by the LAN91C95 combines the simplicity and low overhead of fixed areas with the flexibility of linked lists providing improved performance over other methods.

The LAN91C95 is designed to support full duplex switched Ethernet where transmit and receive are fully independent. It has 6 kbytes of internal memory and the MMU manages memory in 256 byte pages. The memory size accommodates the increase in interrupt latency resulting from simultaneous LAN and modem operation as well as the potential for simultaneous transmit and receive traffice in some full duplex applications.

Packet reception and transmission are determined by memory availability. All other resources are always available if memory is available. To complement this flexible architecture, all ISA bus interface functions are incorporated in the LAN91C95, as well as a 6K byte packet RAM and serial EEPROM-based setup. The user can select or modify configuration choices.

The LAN91C95 stores the Configuration Information Structure (CIS) on reset or power-up from the serial EEPROM. This allows the host to access data to allow the setup of the PCMCIA multi-function card.

In ISA mode, the serial EEPROM acts as storage for configuration and IEEE Ethernet address information compatible with the existing LAN9000 family of ISA Ethernet controllers.

In PCMCIA mode, the serial EEPROM stores the CIS, as well as the IEEE address, information, but it does not store any I/O or IRQ information since this information is handled by the host’s socket controller. For CIS requirements above 512 bytes, an optional external parallel EEPROM can be used in conjunction with the internal CIS. This allows additional external, non-volatile storage for applications that require XIP and use the modem function. If the serial EEPROM is not used in PCMCIA mode, the parallel EEPROM must be used. In this case, the parallel EEPROM is selected for the first 512 bytes of storage as well, allowing the CIS to be stored in the parallel EEPROM and, on power-up, to be read directly by the host. The remaining parallel EEPROM can be used for XIP applications, if needed.

The LAN91C95 integrates most of the 802.3 functionality, incorporating the MAC layer protocol, the physical layer encoding and decoding functions with the ability to handle the AUI interface. For twisted pair networks, the LAN91C95 integrates the twisted pair transceiver as well as the link integrity test functions.

The LAN91C95 is a true 10BASE-T single chip able to interface a system or a local bus.

Directly-driven LEDs for installation and run-time diagnostics are provided, as well as 802.3 statistics gathering to facilitate network management.

The LAN91C95 offers:

High integration:

- Single chip adapter including:
  - Packet RAM
  - ISA bus interface
  - PCMCIA interface
  - EEPROM interface
  - Encoder decoder with AUI interface
  - Full duplex, magic packet 10BASE-T transceiver
  - Lucent Technologies and Rockwell International modem interface

High performance:

- Chained (“back-to-back”) packet handling with no CPU intervention:
  - Queues transmit packets
  - Queues receive packets
  - Full duplex operation for higher network throughput
  - Stores results in memory along with packet
  - Queues Ethernet and modem interrupts
  - Optional single interrupt upon completion of transmit chain

Fast block move operation for load/unload:

- CPU sees packet bytes as if stored contiguously
- Handles 16 bit transfers regardless of address alignment
- Access to packet through fixed window

Fast bus interface:

- Compatible with ISA type and faster buses

Flexibility:

- Flexible packet and header processing:
  - Can be set to Simultasking - Early Receive and transmit modes
  - Can access any byte in the packet
  - Can immediately remove undesired packets from queue
  - Can move packets from receive to transmit queue
  - Can alter receive processing order without copying data
  - Can discard or enqueue again a failed transmission

Resource allocation:

- Memory dynamically allocated for transmit and receive
- Can automatically release memory on successful transmission

Configuration:

- ISA:
  - Uses non-volatile jumperless setup via serial EEPROM

- PCMCIA:
  - Uses serial EEPROM for attribute memory storage. PCMCIA I/O ignores address lines A4-A15 and relies on the PCMCIA host, decoding for the slot

nROM/nPCMCIA on the LAN91C95 is left open with a pullup for ISA mode. This pin is sampled at the end of Power On Reset. If found low, the LAN91C95 is configured for PCMCIA mode.

## PIN REQUIREMENTS

| FUNCTION | ISA | PCMCIA | NUMBER OF PINS |
|---|---|---|---|
| SYSTEM ADDRESS BUS | A0-A15<br>A16<br>A17<br>A18<br>A19<br>AEN | A0-A15<br>nFWE<br>nFCS<br><br>nCE1<br>nREG | 21 |
| SYSTEM DATA BUS | D0-D15 | D0-D15 | 16 |
| SYSTEM CONTROL BUS | RESET<br>BALE<br>nIORD<br>nIOWR<br>nMEMR<br>IOCHRDY<br>nIOCS16<br>nSBHE<br>INTR0<br>INTR1<br>INTR2<br>INTR3 | RESET<br>nWE<br>nIORD<br>nIOWR<br>nOE<br>nWAIT<br>nIOIS16<br>nCE2<br>nIREQ<br>nINPACK<br>nSTSCHG | 12 |
| MODEM INTERFACE | nMRESET<br>MINT<br>nMCS<br>MRDY<br>nMPWDN<br>MIDLEN1<br>MRINGIN<br>nMRINGOA<br>MRINGOB<br>SPKRIN<br>SPKROUT<br>nMPDOUT<br>MFBK1<br>nMIS16 | nMRESET<br>MINT<br>nMCS<br>MRDY<br>nMPWDN<br>MIDLEN1<br>MRINGIN<br>nMRINGOA<br>MRINGOB<br>SPKRIN<br>SPKROUT<br>nMPDOUT<br>MFBK1<br>nMIS16 | 14 |
| SERIAL EEPROM | EEDI<br>EEDO<br>EECS<br>EESK<br>ENEEP<br>IOS0<br>IOS1<br>IOS2 | EEDI<br>EEDO<br>EECS<br>EESK<br>ENEEP<br>IOS0<br>IOS1<br>IOS2 | 8 |

| FUNCTION | ISA | PCMCIA | NUMBER OF PINS |
|---|---|---|---|
| CRYSTAL OSC. | XTAL1<br>XTAL2 | XTAL1<br>XTAL2 | 2 |
| POWER | VDD<br>AVDD | VDD<br>AVDD | 12 |
| GROUND | GND<br>AGND | GND<br>AGND | 12 |
| 10BASE-T INTERFACE | TPERXP<br>TPERXN<br>TPETXP<br>TPETXN<br>TPETXDP<br>TPETXDN | TPERXP<br>TPERXN<br>TPETXP<br>TPETXN<br>TPETXDP<br>TPETXDN | 6 |
| AUI INTERFACE | RECP<br>RECN<br>COLP<br>COLN TXP/nCOLL<br>TXN/nCRS | RECP<br>RECN<br>COLP<br>COLN<br>TXP/nCOLL<br>TXN/nCRS | 6 |
| LEDs | nLNKLED/TXD<br>nRXLED/RXCLK<br>nBSELED/RXD<br>nTXLED/nTXEN | nLNKLED/TXD<br>nRXLED/RXCLK<br>nBSELED/RXD<br>nTXLED/nTXEN | 4 |
| MISC. | RBIAS<br>WAKEUP<br>nWAKEUPEN<br>PWRDWN/TXCLK<br>nXENDEC<br>nEN16<br>ROM/nPCMCIA | RBIAS<br>WAKEUP<br>nWAKEUPEN<br>PWRDWN/TXCLK<br>nXENDEC<br>nEN16<br>ROM/nPCMCIA | 7 |
| TOTAL PINS | | | 120 |

## DESCRIPTION OF PIN FUNCTIONS

| PIN NO. | NAME | SYMBOL | TYPE | DESCRIPTION |
|---|---|---|---|---|
| 113 | | nROM/ nPCMCIA | I/O4 with pullup | This pin is sampled at the end of RESET. When this pin is sampled low the LAN91C95 is configured for PCMCIA operation and all pin definitions correspond to the PCMCIA mode. For ISA operation this pin is left open and is used as a ROM chip select output that goes active when nMEMR is low and the address bus contains a valid ROM address. |
| 35, 36, 38-43, 45-50, 52, 53 | Address 0-15 | A0-A15 | I | Input address lines 0 through 15. |
| 54 | Address 16<br>nFlash Memory Write | A16<br>nFWE | I<br>O4 | ISA - Input address line 16.<br>PCMCIA - Output. Flash Memory Write Enable used for programming the attribute memory. Goes active (low) when nWE=0 and WRATTRIB=1 (in ECOR bit 3). |
| 55 | Address 17<br>nFlash Memory Chip Select | A17<br>nFCS | I<br>O4 | ISA - Input address line 17.<br>PCMCIA - Output. Flash Memory Chip Select used to access attribute memory. Goes active (low) when nREG=0, nCE1=0 and A15=0. |
| 56 | Address 18 | A18 | I | Input address line 18. |
| 57 | Address 19<br>Card Enable 1 | A19<br>nCE1 | I with pullup | ISA - Input address line 19.<br>PCMCIA - Card Enable 1 input. Used to select card on even byte accesses. |
| 34 | Address Enable | AEN<br>nREG | I with pullup | ISA - Address enable input. Used as an address qualifier. Address decoding is only enabled when AEN is low.<br>PCMCIA - Attribute memory and IO select input. Asserted when the card attribute space or IO space is being accessed. |
| 81 | nByte High Enable | nSBHE | I with pullup | ISA - Byte High Enable input. Asserted (low) by the system to indicate a data transfer on the upper data byte. |
| | nCard Enable 2 | nCE2 | | PCMCIA - Card Enable 2 input. Used to select card on odd byte accesses. |

## DESCRIPTION OF PIN FUNCTIONS

| PIN NO. | NAME | SYMBOL | TYPE | DESCRIPTION |
|---|---|---|---|---|
| 83 | Ready<br>nWait | IOCHRDY<br>nWAIT | OD24 with pullup | ISA - Output. Optionally used by the LAN91C95 to extend host cycles.<br>PCMCIA - Output. Optionally used by the LAN91C95 to extend host cycles. |
| 59, 60, 61, 63, 65, 66, 68-70, 72-74, 76-79 | Data Bus | D0-D15 | I/O24 | Bidirectional. 16 bit data bus used to access the LAN91C95 internal registers. The data bus has weak internal pullups. Supports direct connection to the system bus without external buffering. |
| 98 | Reset | RESET | IS with pullup | Input. Active high Reset. This input is not considered active unless it is active for at least 100ns to filter narrow glitches. A POR circuit generates an internal reset upon power up for at least 15msec. All hardware reset references in this spec relate to the OR function of the POR and the RESET pin. |
| 82 | Address Latch<br>nWrite Enable | BALE<br>nWE | IS with pullup | ISA - Input. Address strobe. For systems that require address latching, the falling edge of BALE latches address lines and nSBHE.<br>PCMCIA - Write Enable input. Used for writing into COR and CSR registers as well as attribute memory space. |
| 90 | Interrupt | INTR0 | O24 | ISA - Active high interrupt signal. The interrupt line selection is determined by the value of INT SEL1-0 bits in the Configuration Register. This interrupt is tri-stated when not selected. |
|  | nInterrupt Request | nIREQ |  | PCMCIA - Active low interrupt request output. Pin acts as a Ready pin during power-up. The pin should be pulled low within 10us of the application of the VCC or Reset (which ever occurs later). It remains low(0) until the CIS is loaded in the Internal SRAM.  The high(1) state indicates to the host controller that the device is ready. |
| 91 | Interrupt 1 | INTR1 | O24 | ISA - Output. Active high interrupt signal. The interrupt line selection is determined by the value of INT SEL1-0 bits in the Configuration Register. This interrupt is tri-stated when not selected. |

## DESCRIPTION OF PIN FUNCTIONS

| PIN NO. | NAME | SYMBOL | TYPE | DESCRIPTION |
|---|---|---|---|---|
|  |  | nINPACK |  | PCMCIA - Output asserted to acknowledge read cycles for an enabled function. |
| 92 | Interrupt 2 | INTR2 | O24 | ISA - Output. Active high interrupt signal. The interrupt line selection is determined by the value of INT SEL1-0 bits in the Configuration Register. This interrupt is tri-stated when not selected. |
|  | nStatus Changed | nSTSCHG |  | PCMCIA - Status changed bit. Depending on the setting of the RingEn bit (Modem CCSR), this pin either reflects the ringing status (ExCA mode) or the state of the Modem Changed bit. The ringing status is obtained by stretching the MRINGIN to convert a 20Hz toggle rate to a constant level. |
| 93 | Interrupt 3 | INTR3 | O24 | ISA - Output. Active high interrupt signal. The interrupt line selection is determined by the value of INT SEL1-0 bits in the Configuration Register. This interrupt is tri-stated when not selected. |
| 85 | nI/O16 | nIOCS16 | OD24 | ISA - Active low output asserted in 16 bit mode when AEN is low and A4-A15 decode to the LAN91C95 address programmed into the high byte of the Base Address Register. |
|  |  | nIOIS16 |  | PCMCIA - Active low output asserted whenever the LAN91C95 is in 16 bit mode, and “Enable Function” bit in the ECOR register is high, nREG is low and A4-A15 decode to the LAN address specified in I/O Base Registers 0 and 1 in PCMCIA attribute space. |
| 88 | nI/O Read | nIORD | IS with pullup | Input. Active low read strobe used to access the LAN91C95 IO space. |
| 87 | nI/O Write | nIOWR | IS with pullup | Input. Active low write strobe used to access the LAN91C95 IO space. |
| 86 | nMemory Read | nMEMR | IS with pullup | ISA - Active low signal used by the host processor to read from the external ROM. |
|  | nOutput Enable | nOE |  | PCMCIA - Output Enable input used to read from the COR, CSR and attribute memory. |

## DESCRIPTION OF PIN FUNCTIONS

| PIN NO. | NAME | SYMBOL | TYPE | DESCRIPTION |
|---|---|---|---|---|
| 24 | nModem Reset | nMRESET | O4 | Reset output to Modem. Asserted whenever RESET pin is high, internal POR is active, or SRESET bit is high (MCOR bit 7). |
| 18 | Modem Interrupt | MINT | I with pull down | Interrupt input from Modem. Reflected in INTR (CSR bit 1) and asserts the appropriate interrupt pin if enabled. |
| 23 | nModem Chip Select | nMCS | O4 | Chip select output to modem. |
| 17 | Modem Ready | MRDY | I with pullup | Modem ready input. Low indicates the modem is not ready, either after reset or exiting from stop or sleep modes. |
| 27 | nModem Powerdown | nMPWDN | O4 | Powerdown output to modem controller. This pin is active (low) when either the PWRDWN bit (CSR bit 2) is set or the modem is disabled (not configured). |
| 26 |  | MIDLEN1 | O4 | Powerdown output to modem controller. This pin is active (high) when either the PWRDWN bit (CSR bit 2) is set or the modem is disabled (not configured). |
| 20 | Modem Ring Input | MRINGIN | I | Ring input from the modem controller. Toggles when ringing, low when not ringing. |
| 21 | nModem Ring Output A | nMRINGOA | O4 | Ring output signal. When there is no ringing on the MRINGIN pin and the modem is not in Powerdown mode this output is high. During ringing this signal is the inverse of the MRINGIN input. When the PWRDWN bit is set or the function is disabled, this output is low. This signal is activated about 12 msec after removing Powerdown. |
| 22 | Modem Ring Output B | MRINGOB | O4 | Ring output signal. When the modem is not in Powerdown mode (PWRDWN bit is zero and the function is enabled) this output follows the value of the MRINGIN input. When entering Powerdown mode, a rising edge is generated on the pin. A rising edge is also generated when exiting Powerdown mode also. Refer to Figure 2. |
| 14 | Speaker Input | SPKRIN | I with pullup | Speaker Input. This is a digitized (single level) audio input from the modem controller. |

## DESCRIPTION OF PIN FUNCTIONS

| PIN NO. | NAME | SYMBOL | TYPE | DESCRIPTION |
|---|---|---|---|---|
| 15 | Speaker Output | SPKROUT | O4 tri-stable with pullup | Speaker Output. This pin reflects the SPKRIN pin when enabled by the AUDIO bit (Modem CSR bit 3). When disabled this pin is tri-stated. |
| 28 | | nMPDOUT | I Schmitt | Used to control Powerdown mode. Tie to a 180K pull-up and a 0.1uF cap to ground. Tie high when not used. This signal is used in the RC time constant. |
| 29 | | MFBK1 | O4 with pullup | Tie to nMPDOUT through a 5.1M resistor. This signal is used in the RC time constant in conjuction with the nMPDOUT pin. |
| 16 | n16 Bit Modem | nMIS16 | I with pullup | Input. When low, it indicates a 16 bit modem, otherwise the modem is 8 bit wide. Used to determine if nIOIS16 (PCMCIA) and nIOCS16 (ISA) need to be asserted for modem cycles. The value of this pin may change from cycle to cycle. |
| 110 | EEPROM Clock | EESK | O4 | Output. 4µs clock used to shift data in and out of the serial EEPROM. |
| 109 | EEPROM Chip Select | EECS | O4 | Output. Serial EEPROM chip select. |
| 108 | EEPROM Data Out | EEDO/ SDOUT | O4 | Output. Connected to the DI input of the serial EEPROM. |
| 107 | EEPROM Data In | EEDI | I with pull-down | Input. Connected to the DO output of the serial EEPROM. |
| 103, 105, 106 | I/O Base | IOS0-IOS2 | I with pullup | Input. External switches can be connected to these lines to select between predefined EEPROM configurations. The values of these pins are readable. These pins are used in ISA mode only. If in PCMCIA mode, these pins are not used. |
| 9 | nTransmit LED<br>nTransmit Enable | nTXLED<br>nTXEN | OD16<br>O162 | Internal ENDEC - Transmit LED output.<br>External ENDEC - Active low Transmit Enable output. |

## DESCRIPTION OF PIN FUNCTIONS

| PIN NO. | NAME | SYMBOL | TYPE | DESCRIPTION |
|---|---|---|---|---|
| 12 | nBoard Select LED | nBSELED | OD16 | Internal ENDEC - Board Select LED activated by accesses to I/O space (nIORD or nIOWR active with AEN low and valid address decode for ISA, and with nREG low and and “Enable Function” bit in the ECOR register is high for PCMCIA). The pulse is stretched beyond the access duration to make the LED visible. |
| | Receive Data | RXD | I with pullup | External ENDEC - NRZ receive data input. |
| 10 | nReceive LED | nRXLED | OD16 | Internal ENDEC - Receive LED output. |
| | Receive Clock | RXCLK | I with pullup | External ENDEC - Receive clock input. |
| 11 | nLink LED | nLNKLED | OD16 | Internal ENDEC - Link LED output. |
| | Transmit Data | TXD | 0162 | External ENDEC - Transmit Data output. |
| 111 | Enable EEPROM | ENEEP | I with pullup | Input. This active high input enables the EEPROM to be read or written by the LAN91C95. Internally pulled up. Must be connected to ground if no serial EEPROM is used. In PCMCIA mode, a parallel EEPROM is required if no serial EEPROM is used. |
| 115 | nEnable 16 Bit | nEN16 | I with pullup | Input. When low the LAN91C95 is configured for 16 bit bus operation. If left open the LAN91C95 works in 8 bit bus mode. 16 bit configuration can also be programmed via serial EEPROM (In ISA Mode only) or via software initialization of the CONFIGURATION REGISTER. |
| 101<br>102 | Crystal 1<br>Crystal 2 | XTAL1<br>XTAL2 | Iclk | An external parallel resonance 20 MHz crystal should be connected across these pins. If an external clock source is used, it should be connected to XTAL1 and XTAL2 should be left open. |
| 123<br>124 | AUI Receive | RECP<br>RECN | Diff. Input | AUI receive differential inputs. |

## DESCRIPTION OF PIN FUNCTIONS

| PIN NO. | NAME | SYMBOL | TYPE | DESCRIPTION |
|---|---|---|---|---|
| 2<br>3 | AUI Transmit | TXP/nCOLL<br>TXN/nCRS | Diff. Output | Internal ENDEC - (nXENDEC pin open). In this mode TXP and TXN are the AUI transmit differential outputs. They must be externally pulled up using 150 ohm resistors. |
| | | | I | External ENDEC - (nXENDEC pin tied low). In this mode the pins are inputs used for collision and carrier sense functions. |
| 125<br>126 | AUI Collision | COLP<br>COLN | Diff. Input | AUI collision differential inputs. A collision is indicated by a 10 MHz signal at this input pair. |
| 121<br>122 | TPE Receive | TPERXP<br>TPERXN | Diff. Input | 10BASE-T receive differential inputs. |
| 4<br>6 | TPE Transmit | TPETXP<br>TPETXN | Diff. Output | Internal ENDEC - 10BASE-T transmit differential outputs. |
| 5<br>7 | TPE Transmit Delayed | TPETXDP<br>TPETXDN | Diff. Output | 10BASE-T delayed transmit differential outputs. Used in combination with TPETXP and TPETXN to generate the 10BASE-T transmit pre-distortion. |
| 114 | Powerdown | PWRDWN | I with pullup | Internal ENDEC - Powerdown input. It keeps the LAN91C95 in powerdown mode when high (open). Must be low for normal operation. Refer to the Powerdown Matrix following for further details. |
| | Transmit Clock | TXCLK | | External ENDEC - Transmit clock input from external ENDEC |
| 99 | nWakeup Enable | nWAKEUP-EN | I with pullup | Input. When pulled down, the device will enable Magic Packet (MP) logic and put the Ethernet function in powerdown mode. The pin assertion will override the state of WAKEUP_EN and PWRDN bits in CTR, Pwrdwn bit in ECSR, and Enable Function bit ECOR. When deasserted, the WAKEUP_EN and PWRDN bits will be changed to (0), Pwrdwn to (0), and Enable Function to (1). |
| 100 | Wakeup | WAKEUP | O4 | Output. Asserted high if nWAKEUP_EN is asserted and a valid Magic Packet (MP) pattern is detected. The pin remains asserted until nWAKEUP_EN is deasserted. |

## DESCRIPTION OF PIN FUNCTIONS

| PIN NO. | NAME | SYMBOL | TYPE | DESCRIPTION |
|---|---|---|---|---|
| 118 | Bias Resistor | RBIAS | Analog Input | A 39 kohm 1% resistor should be connected between this pin and analog ground. |
| 116 | nExternal ENDEC | nXENDEC | I with pullup | When tied low the LAN91C95 is configured for external ENDEC. When tied high or left open the LAN91C95 will use its internal encoder/decoder. |
| 13, 25, 37, 51, 62, 71, 80, 89, 104 | Power | VDD | | +5V power supply pins |
| 1, 117, 120 | Analog Power | AVDD | | +5V analog power supply pins |
| 19, 30, 44, 58, 67, 75, 84, 94, 112 | Ground | VSS | | Ground pins |
| 8, 119, 127 | Analog Ground | AVSS | | Analog ground pins |

## BUFFER TYPES

| | |
|---|---|
| O4 | Output buffer with 2mA source and 4mA sink |
| O162 | Output buffer with 2mA source and 16mA sink |
| O24 | Output buffer with 12mA source and 24mA sink |
| OD16 | Open drain buffer with 16mA sink |
| OD24 | Open drain buffer with 24mA sink |
| I/O24 | Bidirectional buffer with 12mA source and 24mA sink |
| I | Input buffer with TTL levels |
| IS | Input buffer with Schmitt Trigger Hysteresis |
| Iclk | Clock input buffer |

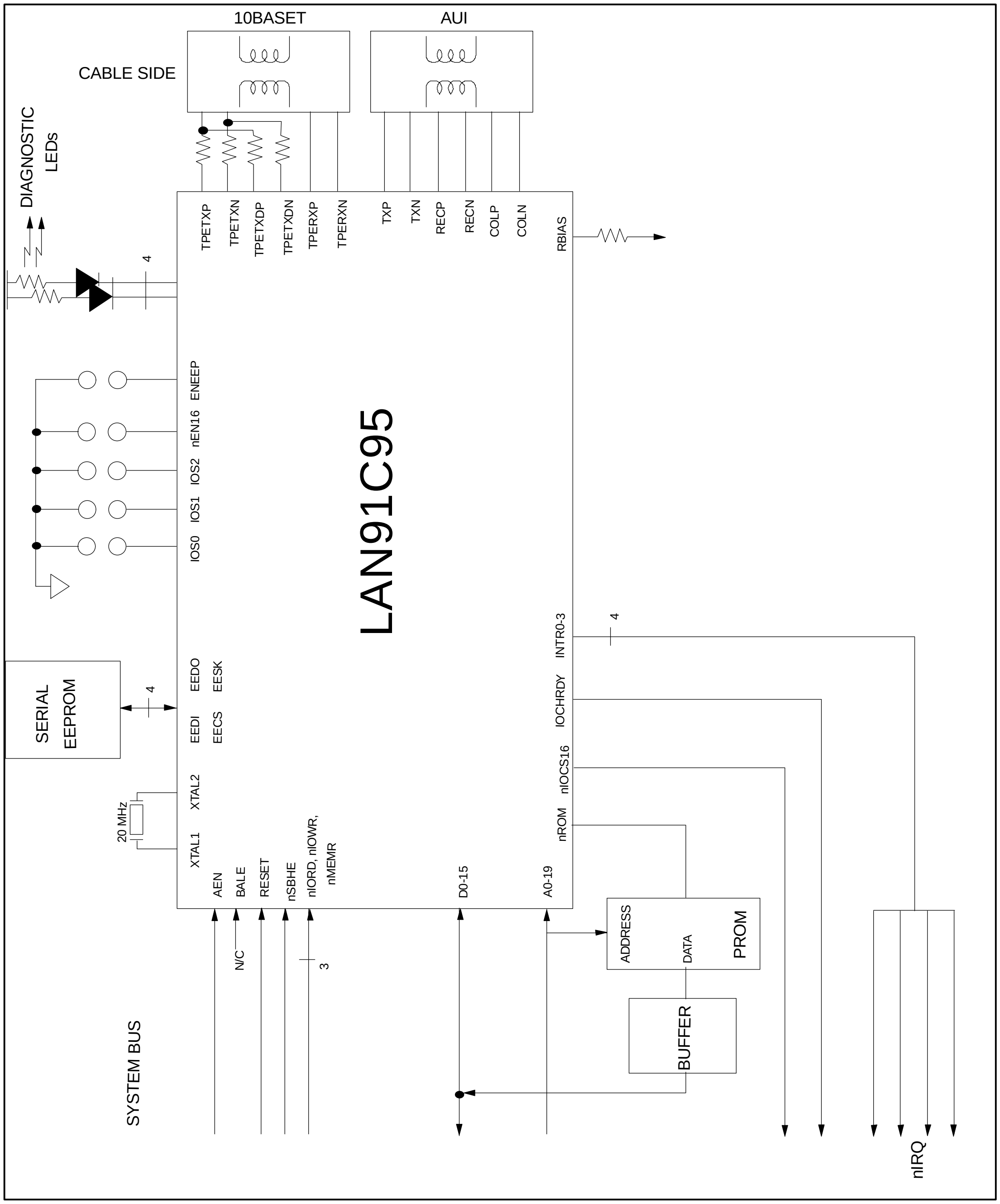


**FIGURE 1 - LAN91C95 SYSTEM BLOCK DIAGRAM FOR ISA BUS WITH BOOT PROM**

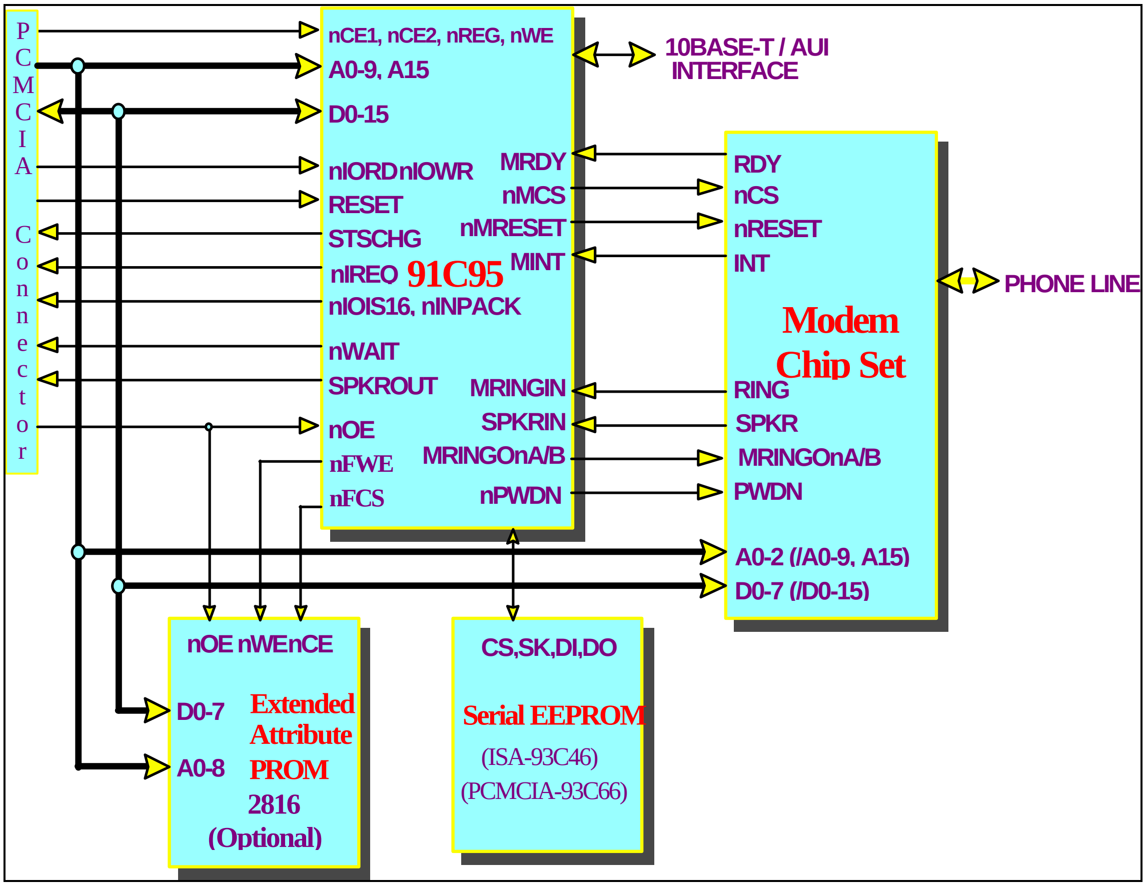


FIGURE 2 - LAN91C95 SYSTEM BLOCK DIAGRAM FOR DUAL FUNCTION PCMCIA CARD

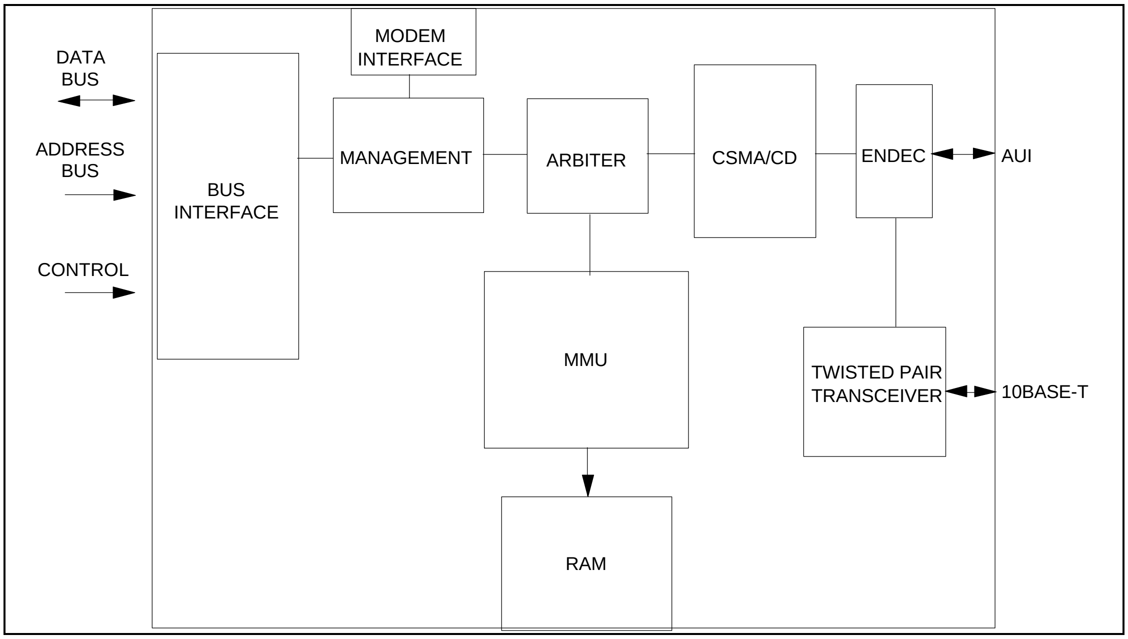


**FIGURE 3 - LAN91C95 INTERNAL BLOCK DIAGRAM**

## FUNCTIONAL DESCRIPTION

The LAN91C95 consists of an integrated Ethernet controller mapped entirely in I/O space, as well as support for an external Modem controller also mapped in I/O space. In addition, PCMCIA attribute memory space is decoded to interface an external CIS ROM, with configuration registers as per PCMCIA 3.X extensions implemented on-chip in attribute space above the CIS ROM decode area. The PCMCIA Configuration Registers are also accessible in I/O space to allow non-PCMCIA dual function designs.

The Ethernet controller function includes a built-in 6kbyte RAM for packet storage. This RAM buffer is accessed by the CPU through two sequential access regions of 3 kbytes each. The RAM access is internally arbitrated by the LAN91C95 and dynamically allocated between transmit and receive packets using 256 byte pages. The Ethernet controller functionality is identical to the LAN91C94 except where indicated otherwise.

**Table 1 - Bus Transactions In Isa Mode**

| | **A0** | **nSBHE** | **D0-D7** | **D8-D15** |
|---|---|---|---|---|
| 8 BIT MODE<br>(nEN16=1) | 0 | X | even byte | - |
| (16 BIT=0) | 1 | X | odd byte | - |
| 16 BIT MODE | 0 | 0 | even byte | odd byte |
| otherwise | 0 | 1 | even byte | - |
| | 1 | 0 | - | odd byte |
| | 1 | 1 | invalid cycle | |

**Table 2 - Bus Transactions In PCMCIA Mode**

| | A0 | nCE1 | nCE2 | D0-D7 | D8-D15 |
|---|---|---|---|---|---|
| 8 BIT MODE<br>((IOis8=1) + (nEN16=1).(16BIT=0)) | 0 | 0 | X | even byte | - |
| | 1 | 0 | X | odd byte | - |
| | X | 1 | X | NO CYCLE | |
| 16 BIT MODE | 0 | 0 | 0 | even byte | odd byte |
| otherwise | 0 | 0 | 1 | even byte | - |
| | 1 | 0 | 1 | odd byte | |
| | X | 1 | 0 | - | odd byte |
| | X | 1 | 1 | NO CYCLE | |

16BIT = CONFIGURATION REGISTER bit 7
IOis8 = ECSR register bit 5
nEN16 = pin nEN16

For the modem function, the transactions are similarxcept that the modem is assumed to be 8 bit wide unless (IOis8=0) and (nMIS16=0).

Note: The IOis8 value should be identical in MCSR and ECSR if both functions are enabled.

8 Bit mode = (IOis8=1)+(nMIS16=1)

**Table 3 - LAN91C95 Address Spaces**

| | SIGNALS USED | ISA | PCMCIA | ON-CHIP | DEPTH | WIDTH |
|---|---|---|---|---|---|---|
| PCMCIA Attribute Memory | nOE/nWE | N | Y | N (external ROM) | Up to 32k locations, only even bytes are usable | 8 bits on even addresses |
| PCMCIA Configuration Registers | nOE/nWE | N | Y | Y | 64 locations, only even bytes are usable | 8 bits |
| Modem I/O Space | nIORD/nIOWR | Y | Y | N | 8 locations | 8 bit |
| Ethernet I/O Space (1) | nIORD/nIOWR | Y | Y | Y | 16 locations | 8 or 16 bits |

(1) This space also allows access to the PCMCIA Configuration Register through Banks 4 and 5 (LAN91C95 only).

Except for the bus interface, the functional behavior of the LAN91C95 after initial configuration is identical for ISA and PCMCIA modes.

The LAN91C95 includes an arbitrated shared memory of 6 kbytes accessed by the CPU. The MMU unit allocates RAM memory to be used for transmit and receive packets, using 256 byte pages.

The arbitration is transparent to the CPU in every sense. There is no speed penalty for ISA type of machines due to arbitration. There are no restrictions on what locations can be accessed at any time. RAM accesses as well as MMU requests are arbitrated.

The RAM is accessed by mapping it into I/O space for sequential access. Except for the RAM accesses and the MMU request/release commands, I/O accesses are not arbitrated.

The I/O space is 16 bits wide. Provisions for 8 bit systems are handled by the bus interface.
In the system memory space, up to 64 kbytes are decoded by the LAN91C95 as expansion ROM. The ROM expansion area is 8 bits wide.

Device configuration is done using a serial EEPROM, with support for modifications to the EEPROM at installation time. A Flash ROM is supported for PCMCIA attribute memory.

The CSMA/CD core implements the 802.3 MAC layer protocol. It has two independent interfaces, the data path and the control path. Both interfaces are 16 bits wide.

The control path provides a set of registers used to configure and control the block. These registers are accessible by the CPU through the LAN91C95's I/O space. The data path is of sequential access nature and typically works in one direction at any given time. An internal DMA type of interface connects the data path to the device RAM through the arbiter and MMU.

The CSMA/CD data path interface is not accessible to the host CPU.

The internal DMA interface can arbitrate for RAM access and request memory from the MMU when necessary.

An encoder/decoder block interfaces the CSMA/CD block on the serial side. The encoder will do the Manchester encoding of the transmit data at 10

Mbps, while the decoder will recover the receive clock and decode received data.

Carrier and Collision detection signals are also handled by this block and relayed to the CSMA/CD block.

The encoder/decoder block can interface the network through the AUI interface pairs or it can be programmed to use the internal 10BASE-T transceiver and connect to a twisted pair network.

The twisted pair interface takes care of the medium dependent signaling for 10BASE-T type of networks. It is responsible for line interface (with external pulse transformers and pre-distortion resistors), collision detection as well as the link integrity test function.

The LAN91C95 provides a 16 bit data path into RAM. The RAM is private and can only be accessed by the system via the arbiter. RAM memory is managed by the MMU. Byte and word accesses to the RAM are supported.

If the system to SRAM bandwidth is insufficient, the LAN91C95 will automatically use its IOCHRDY line for flow control. However, for ISA buses, IOCHRDY will never be negated.

**Modem Interface**

LAN91C95 provides seemless interface to the Lucent HSM288xCF and Rockwell ACL and ACFL series modems. The controller provides all PCMCIA 5.0 registers needed to support the modem function. These registers include support for:

1. Function Enable/Disable
2. Power Down/Up
3. Function Reset
4. Modem Ready/Busy indication
5. Data Bus Access Width
6. I/O Base Select
7. I/O Access Range
8. Interrupt Enable
9. Wake up events ‘Status Change’ information and routing to nSTSCHG pin
10. Audio Path Mute Control

Among other functions, the LAN91C95 also provides Power-On Reset to the modem. The data sheet specifies 0.1mF capacitor and 180KW resistor at the nMPDOUT pin assuming a minimum of 15msec modem reset - required for the modem. The duration of this POR can be changed with the RC value connected with the LAN91C95 nMPDOUT pin.

The LAN91C95 decodes the host address bus and asserts chip select for the specified modem I/O range provided the function is enabled and powered up. The modem chips’ internal registers are accessed via the I/O interface only. PCMCIA registers for the modem, however, can be accessed via the PCMCIA configuration memory space or the LAN91C95 control registers I/O space through Bank 5.

## BUFFER MEMORY

The logical addresses for RAM access are divided into TX area and RX area. Each one of the areas is 1.536 kbytes long and accommodates one maximum size Ethernet packet.

The TX area is seen by the CPU as a window through which packets can be loaded into memory before queuing them in the TX FIFO of packets. The TX area can also be used to examine the transmit completion status after packet transmission.

The RX area is associated to the output of the RX FIFO of packets, and is used to access receive packet data and status information.

The logical address is specified by loading the address pointer register. The pointer can automatically increment on accesses.

All accesses to the RAM are done via I/O space. A bit in the address pointer also specifies if the address refers to the TX or RX area.

In the TX area, the host CPU has access to the next transmit packet being prepared for transmission. In the RX area, it has access to the first receive packet not processed by the CPU yet.

The FIFO of packets, existing beneath the TX and RX areas, is managed by the MMU. The MMU dynamically allocates and releases memory to be used by the transmit and receive functions.

The MMU related parameters for the LAN91C95 are:

| | |
|---|---|
| RAM size | 6 kbytes (internal) |
| Max. number of packets | 24 |
| Max. pages per packet | 6 |
| Max. number of pages | 24 |
| Page size | 256 bytes |

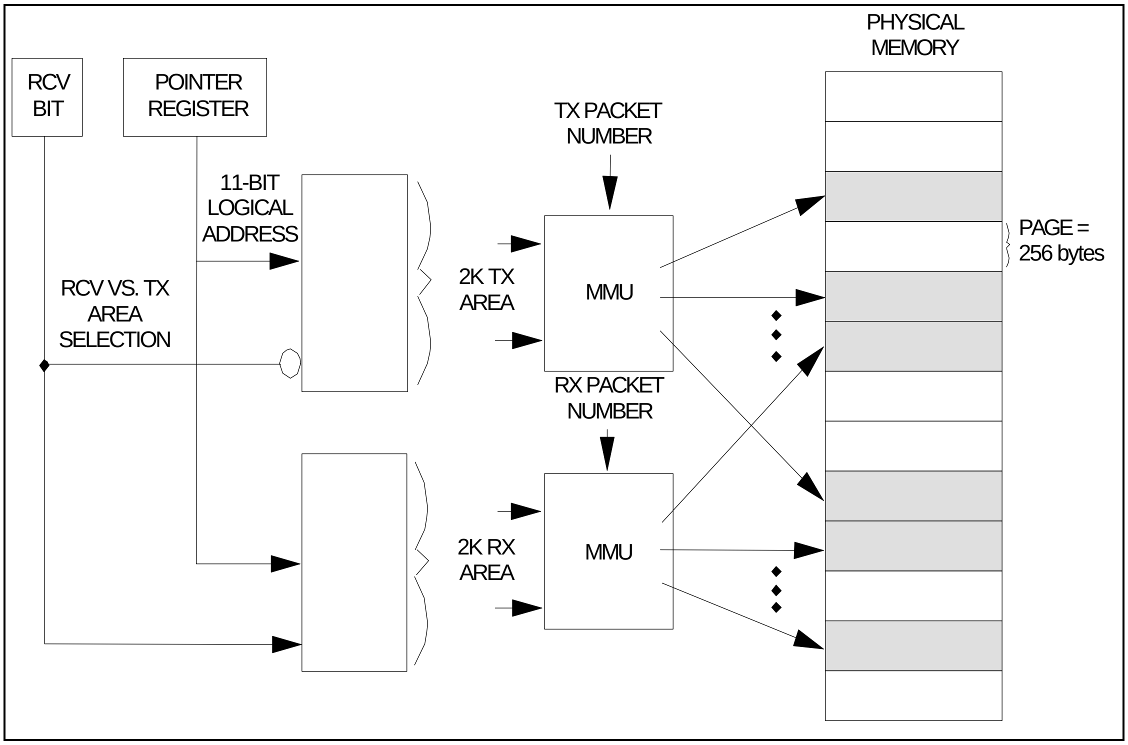


FIGURE 4 - MAPPING AND PAGING VS. RECEIVE AND TX AREA

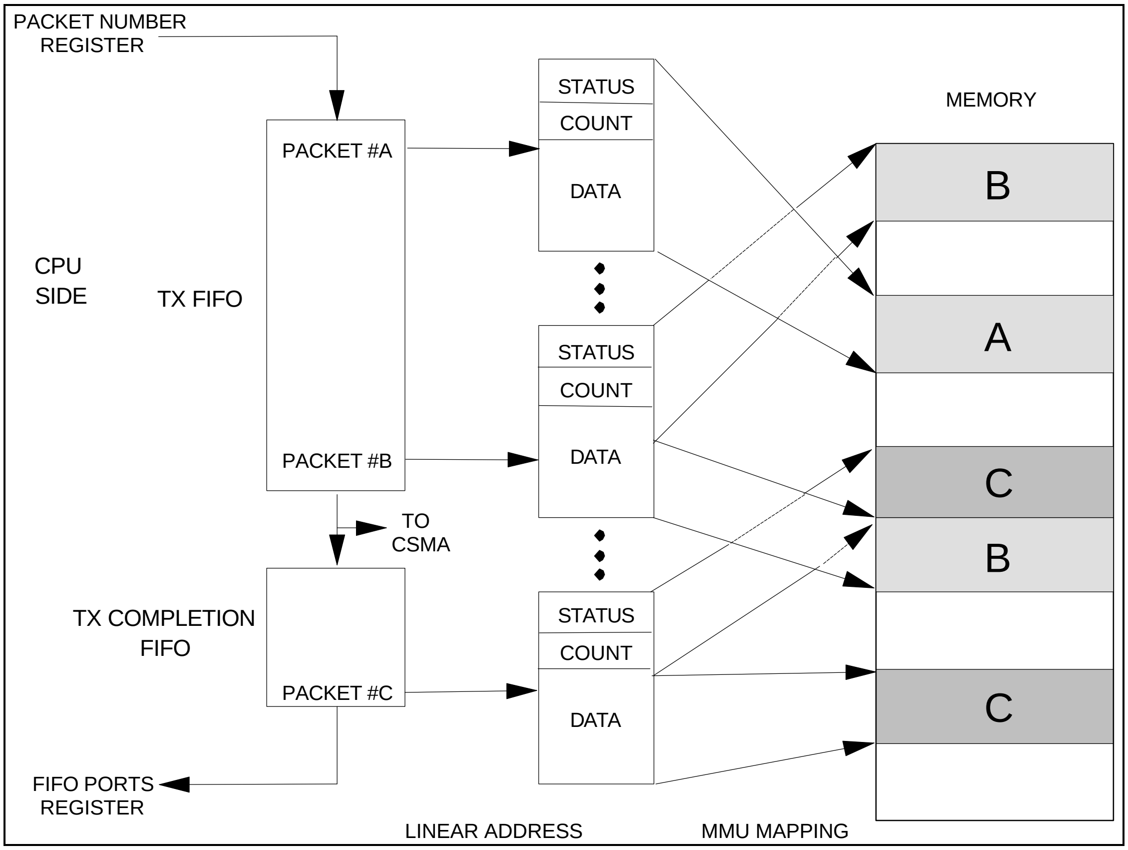


**FIGURE 5 - TRANSMIT QUEUES AND MAPPING**

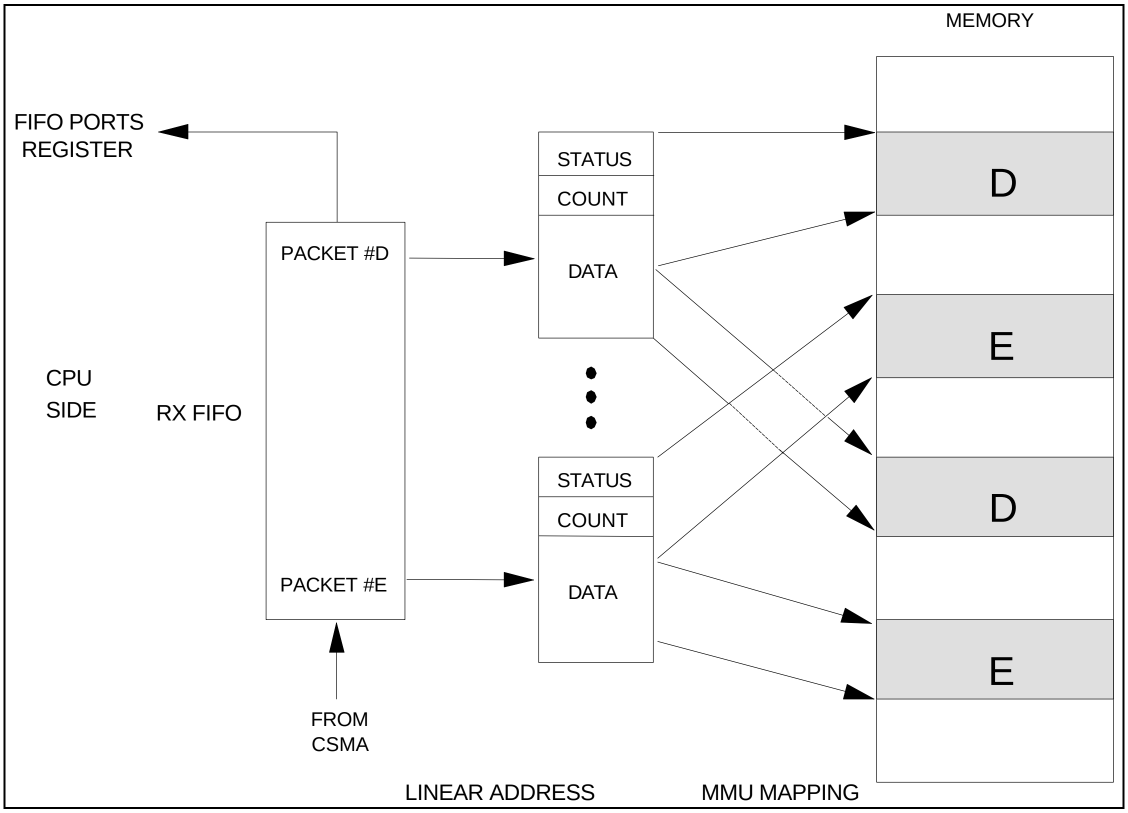


**FIGURE 6 - RECEIVE QUEUE AND MAPPING**

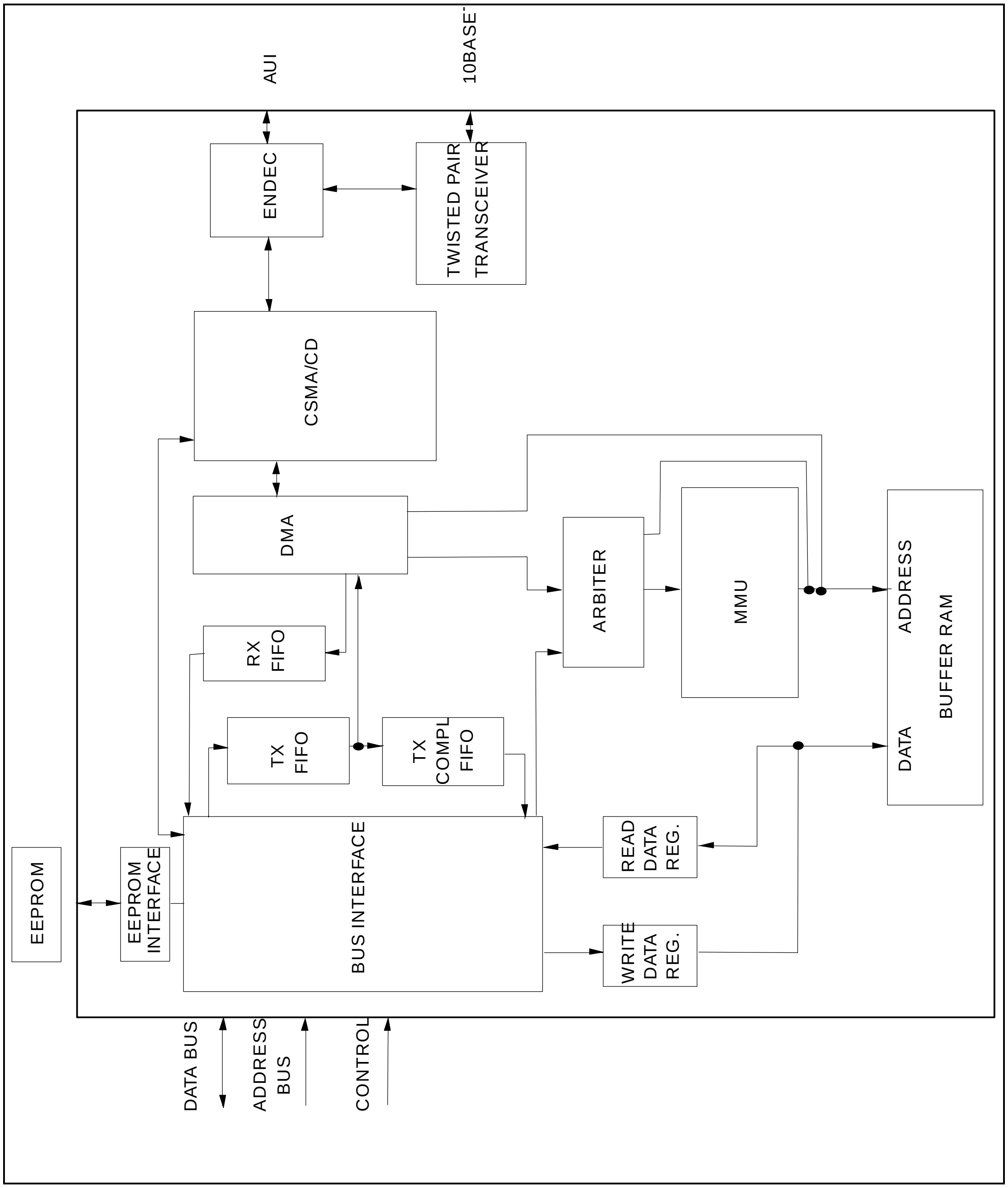


**FIGURE 7 - LAN91C95 INTERNAL BLOCK DIAGRAM WITH DATA PATH**

## PACKET FORMAT IN BUFFER MEMORY

The packet format in memory is similar for the TRANSMIT and RECEIVE areas. The first word is reserved for the status word, the next word is used to specify the total number of bytes, and that in turn is followed by the data area. The data area holds the packet itself, and its length is determined by the byte count. The packet memory format is word oriented.

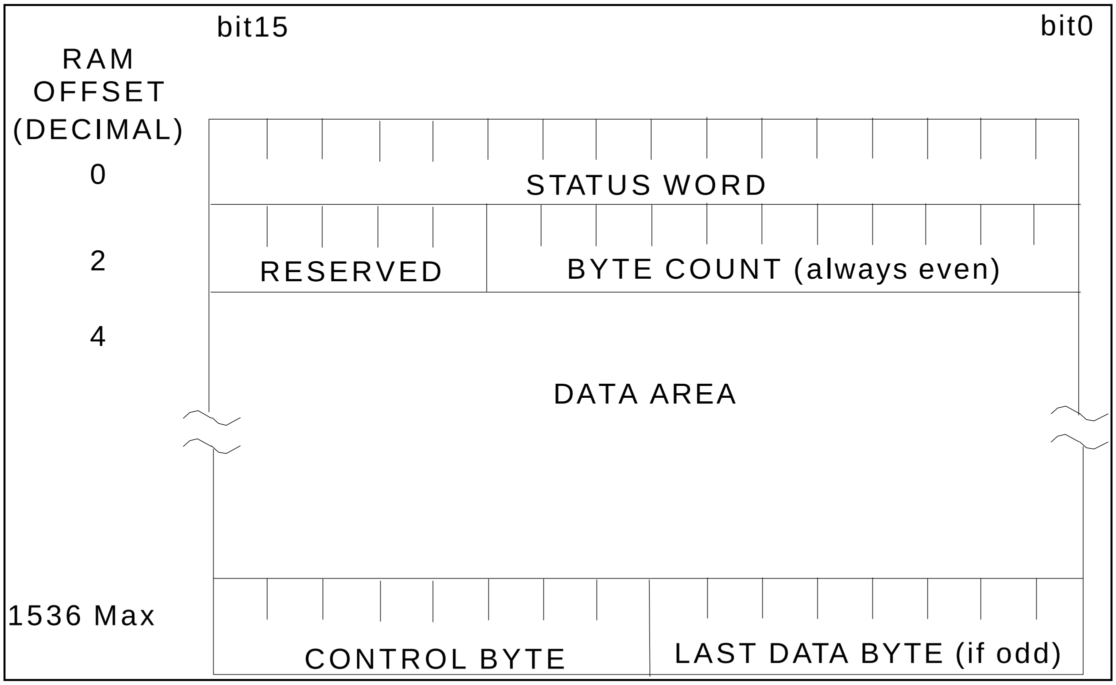


**FIGURE 8 - DATA PACKET FORMAT**

| | **TRANSMIT PACKET** | **RECEIVE PACKET** |
|---|---|---|
| STATUS WORD | Written by CSMA upon transmit completion (see Status Register) | Written by CSMA upon receive completion (see RX Frame Status Word) |
| BYTE COUNT | Written by CPU | Written by CSMA |
| DATA AREA | Written by CPU | Written by CSMA |
| CONTROL BYTE | Written by CPU to control ODD/EVEN data bytes | Written by CSMA. Also has ODD/EVEN bit |

BYTE COUNT - Divided by two, it defines the total number of words including the STATUS WORD, the BYTE COUNT WORD, the DATA AREA and the CONTROL BYTE.

The receive byte count always appears as even; the ODDFRM bit of the receive status word indicates if the low byte of the last word is relevant.

The transmit byte count least significant bit will be assumed 0 by the controller regardless of the value written in memory.

DATA AREA - The data area starts at offset 4 of the packet structure and can extend up to 1536 bytes.

The data area contains six bytes of DESTINATION ADDRESS followed by six bytes of SOURCE ADDRESS, followed by a variable-length number of bytes. On transmit, all bytes are provided by the CPU, including the source address. The LAN91C95 does not insert its own source address. On receive, all bytes are provided by the CSMA side.

The 802.3 Frame Length word (Frame Type in Ethernet) is not interpreted by the LAN91C95. It is treated transparently as data both for transmit and receive operations.

CONTROL BYTE - The CONTROL BYTE always resides on the high byte of the last word. For transmit packets the CONTROL BYTE is written by the CPU as:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | X | ODD | CRC | 0 | 0 | 0 | 0 |

ODD - If set, indicates an odd number of bytes, with the last byte being right before the CONTROL BYTE. If clear, the number of data bytes is even and the byte before the CONTROL BYTE is not transmitted.

CRC - When set, CRC will be appended to the frame. This bit has only meaning if the NOCRC bit in the TCR is set.

For receive packets the CONTROL BYTE is written by the controller as:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | ODD | 0 | 0 | 0 | 0 | 0 |

ODD - If set, indicates an odd number of bytes, with the last byte being right before the CONTROL BYTE. If clear, the number of data bytes is even and the byte before the CONTROL BYTE should be ignored.

**RECEIVE FRAME STATUS WORD**

This word is written at the beginning of each receive frame in memory. It is not available as a register.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | ALGN ERR | BROD CAST | BAD CRC | ODD FRM | TOOLNG | TOO SHORT | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| LOW BYTE | | HASH VALUE | | | | | | MULT CAST |
| | | 5 | 4 | 3 | 2 | 1 | 0 | |

ALGNERR - Frame had alignment error.

BRODCAST - Receive frame was broadcast.

BADCRC - Frame had CRC error.

ODDFRM - This bit when set indicates that the received frame had an odd number of bytes.

TOOLNG - The received frame is longer than 802.3 maximum size (1518 bytes on the cable).

TOOSHORT - The received frame is shorter than 802.3 minimum size (64 bytes on the cable).

HASH VALUE - Provides the hash value used to index the Multicast Registers. Can be used by receive routines to speed up the group address search. The hash value consists of the six most significant bits of the CRC calculated on the Destination Address, and maps into the 64 bit multicast table. Bits 5,4,3 of the hash value select a byte of the multicast table, while bits 2,1,0 determine the bit within the byte selected.

Examples of the address mapping:

| ADDRESS | HASH VALUE 5-0 | MULTICAST TABLE BIT |
|---|---|---|
| ED 00 00 00 00 00 | 000 000 | MT-0 bit 0 |
| 0D 00 00 00 00 00 | 010 000 | MT-2 bit 0 |
| 01 00 00 00 00 00 | 100 111 | MT-4 bit 7 |
| 2F 00 00 00 00 00 | 111 111 | MT-7 bit 7 |

MULTCAST - Receive frame was multicast. If hash value corresponds to a multicast table bit that is set, and the address was a multicast, the packet will pass address filtering regardless of other filtering criteria.

## INTERRUPT STRUCTURE

The LAN91C95 merges two main interrupt sources into a single interrupt line. One source is the Ethernet interrupt and the other is the modem interrupt. The Ethernet interrupt is conceptually equivalent to the LAN91C92 interrupt line; it is the OR function of all enabled interrupts within the Ethernet core. The modem interrupt is an input pin (MINT). The enabling, reporting and clearing of these two sources is controlled by the ECOR, ECSR, MCOR and MCSR registers. The interrupt structure is similar for ISA and PCMCIA modes with the following exceptions:

a) PCMCIA uses a single interrupt pin (nIREQ) while ISA can use any of four INTR0-INTR3 pins.

b) PCMCIA defaults to Ethernet interrupts disabled (Enable IREQ=0 in ECOR), while ISA defaults to Ethernet interrupts enabled.

The following table summarizes the interrupt merging:

**TABLE 4 - LAN91C95 INTERRUPT MERGING**

| FUNCTION | PCMCIA MODE | ISA MODE |
|---|---|---|
| Interrupt Output | nIREQ when either function is enabled. When both functions are disabled the nIREQ is used to report the latched value of MRDY. | INTR0-INTR3 |
| Ethernet Interrupt Source | OR function of all interrupt bits specified in the Interrupt Status Register ANDed with their respective Enable bits. | |
| Modem Interrupt Source | MINT input pin. | |
| Ethernet Interrupt Enable | Enable IREQ bit in ECOR. This bit defaults low in PCMCIA mode. | Enable IREQ bit in ECOR. This bit defaults high in ISA mode. |
| Modem Interrupt Enable | Enable IREQ bit in MCOR. This bit defaults low. | |
| Ethernet Interrupt Status Bit | Intr bit in ECSR. | |
| Modem Interrupt Status Bit | Intr bit in MCSR. | |

## RESET LOGIC

The pins and bits involved in the different reset mechanisms are:

RESET - Input Pin

POR - Internal circuit activated by Power On

nMRESET - Output pin to reset modem

SRESET - Soft Reset bit in ECOR and MCOR, one SRESET bit for each function.

SOFT RST - EPH Soft Reset bit in RCR

**TABLE 5 - RESET FUNCTIONS**

| | **RESETS THE FOLLOWING FUNCTIONS** | **ACTIVATES** | **SAMPLES ISA VS. PCMCIA MODE** | **TRIGGERS EEPROM READ** |
|---|---|---|---|---|
| RESET Pin | All internal logic | nMRESET | Yes | Yes |
| POR Circuit | Generates an internal reset of at least 15 msec, with same effect as the RESET pin | | | |
| ECOR Register SRESET Bit | The Ethernet controller function and Ethernet PCMCIA Configuration Registers except for the bit itself | | No | No |
| MCOR Register SRESET Bit | The Modem controller function and Modem PCMCIA Configuration Registers except for the bit itself | | No | No |
| SOFT RST | The Ethernet Controller itself except for the IA, CONF, and BASE registers. It does not reset any PCMCIA Configuration Register | | No | No |

## POWERDOWN LOGIC

The pins and bits involved in powerdown are:

1. PWRDWN/TXCLK - Input pin valid when XENDEC is not zero (0).
2. Pwrdwn bits in ECSR and MCSR registers - One bit for each function
3. Enable Function bits in ECOR and MCOR registers - One bit for each function
4. PWRDN - Ethernet powerdown bit in Control Register.
5. WAKUP_EN - Magic packet receive enable bit in the Control Register
6. nWAKEUP - Pin for Magic Packet receive + Ethernet function powerdown

**Table 6A - ISA MODE Defined States (Refer to table 6B for next states to wake-up events)**
**(*Rev. C and Higher)**

| CURRENT STATE | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **No.** | **PWRDWN Pin(A= Assrtd)** | **nWAKEUP-EN PIN (A= Assrtd)** | **ECOR FUCNTION ENABLE** | **ECSR POWER DOWN** | **CTR PWRDWN BIT** | **CTR WAKE-UP_EN BIT** | **MCOR FUNC ENABLE** | **MCSR POWER DOWN** | **POWERS DOWN** | **DOES NOT POWER DOWN** |
| 1 | A | X | X | X | X | X | X | X | Everything. Asserts the modem power down pin (nPWDN) also | |
| 2 | nA | A | X | X | X | X | - | - | Ethernet Tx | Ethernet Rx, Link |
| 3 | nA | nA | X | 0 | 0 | 0 | - | - | | Ethernet Tx, Rx, Link |
| 4 | nA | nA | X | 0<br>0 | 0<br>1 | 1 | - | - | Ethernet Tx | Ethernet Rx, Link |
| 5 | nA | nA | X | 0 | 1 | 0 | - | - | Ethernet Tx, Rx, Link | |

Note 1: The chart assumes that ECOR Function Enable bit is meaningless in ISA mode.

Note 2: ECSR Power Down bit must not be set to one(1) in ISA mode

## Table 6B - ISA MODE  (*Rev. C and Higher)

| | NEXT STATE | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **WAKES UP BY** | **PWR DWN Pin (A=Assrtd)** | **nWAKEUP-EN PIN (A=Assrtd)** | **ECOR FUCNTION ENABLE** | **ECSR POWER DOWN** | **CTR PWR-DWN BIT** | **CTR WAKE UP_EN BIT** | **MCOR FUNC ENABLE** | **MCSR POWER DOWN** | **COMMENTS** |
| PWRDWN Pin deassertn | nA | X | No change | No change | No change | No change | No change | No change | ECOR Function Enable Bit value is meaningless in ISA mode |
| nWAKEUP-EN Pin deassertn -AND-, ECSR PWRDWN = 0, CTR PWRDWN bit = 0 | nA | nA | No change | 0 | 0 | No change | - | - | |
| | nA | nA | X | 0 | 0 | 0 | - | - | Fully Awake |

Note 1: The chart assumes that ECOR Function Enable bit is meaningless in ISA mode.

Note 2:  ECSR Power Down bit must not be set to one (1) in ISA mode

## Table 7A - PCMCIA MODE (Refer to table 7B for next states to wake-up events) (*Rev. C and Higher)

| CURRENT STATE | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| No. | PWRDWN Pin(A=Assrtd) | nWAKEUP-EN PIN (A=Assrtd) | ECOR FUNC ENABLE | ECSR PWR DOWN | CTR PWR DWN BIT | CTR WAKE UP_EN BIT | MCOR FUNC ENABLE | MCSR PWR DOWN | POWERS DOWN | DOES NOT POWER DOWN |
| 1 | A | X | X | X | X | X | X | X | Everything. Asserts the modem power down pin(nPWDN) also | |
| 2 | nA | A | X | X | X | X | - | - | Ethernet Tx | Ethernet Rx, Link; PCMCIA Att/Config Mem |
| 3 | nA | nA | 1 | 0 | 0 | 0 | - | - | | Ethernet Tx, Rx, Link; PCMCIA Att/Config Mem |
| 4 | nA | nA | 1 | 0 | 0 | 1 | - | - | Ethernet Tx | Ethernet Rx, Link; PCMCIA Att/Config Mem |
| 5 | nA | nA | 0 | X | X | 0 | - | - | Ethernet Tx, Rx, Link | PCMCIA Att/Conf Memory |
| 6 | nA | nA | 0 | 1<br>0<br>1 | 0<br>1<br>1 | 1 | - | - | Ethernet Tx | Ethernet Rx, Link; PCMCIA Att/Conf Memory |
| 7 | nA | nA | 0 | X | X | 1[1] | - | - | Ethernet Tx | Ethernet Rx, Link; PCMCIA Att/Conf Memory |
| 8 | nA | nA | 1 | 1 | 0 | 0 | - | - | Ethernet Tx, Rx, Link | PCMCIA Att/Config Mem |
| 8S | nA | nA | 1 | 1 | 1 | 0 | - | - | Ethernet Tx, Rx, Link | PCMCIA Att/Config Mem |
| 9 | nA | nA | 1 | 1 | 0 | 1 | - | - | Ethernet Tx | Ethernet Rx, Link; PCMCIA Att/Config Mem |
| 9S | nA | nA | 1 | 1 | 1 | 1 | - | - | Ethernet Tx | Ethernet Rx, Link; PCMCIA Att/Config Mem |

Note[1]: Implementation is such that the IREQ signal is not generated for a Magic Packet detect while the function is disabled.

## Table 7B - PCMCIA MODE (*Rev. C and Higher)

| | NEXT STATE | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **WAKES UP BY** | **PWR DWN Pin (A=Assrtd)** | **nWAKEUP-EN PIN (A=Assrtd)** | **ECOR FUNC ENABLE** | **ECSR PWR DOWN** | **CTR PWRDWN BIT** | **CTR WAKEUP_EN BIT** | **MCOR FUNC ENABLE** | **MCSR PWR DOWN** | **COMMENTS** |
| PWRDWN Pin deassertn | nA | X | No change | No change | No change | No change | No change | No change | Pin deassertion will make the Att/Conf Mem accessible entirely |
| nWAKEUP-EN Pin deassertn -AND- ECOR Func. En. = 1, ECSR PWRDWN = 0, CTR PWRDWN bit = 0 | nA | nA | No change | No change | No change | No change | - | - | |
| | nA | nA | 1 | 0 | 0 | 0 | - | - | Fully Awake |
| By writing a 0 to CTR WAKEUP_EN bit | nA | nA | 1 | 0 | 0 | 0 | - | - | MP precedence |
| | nA | nA | 0 | X | X | 0 | - | - | |
| By writing 1 to 0 to ECSR Power Down, a 0 to CTR PWRDWN, and a 0 to WAKEUP_EN bit | nA | nA | 1 | 0 | 0 | 0 | - | - | NOTE: Both Power down bits need to be written as 0 only if both were set to 1 |
| By writing 1 to ECOR Func Enable, 0 to ECSR Power Down, 0 to CTR PWRDWN, and 0 to WAKEUP_EN bit | nA | nA | 1 | 0 | 0 | 0 | - | - | NOTE: Both Power down bits need to be written as 0 only if both were set to 1 |
| By writing 0 to ECSR Power Down bit* | nA | nA | 1 | 0 | 0 | 0 | - | - | |
| By writing 0 to ECSR Power Down and a 0 to CTR PWRDWNbit | nA | nA | 1 | 0 | 0 | 0 | - | - | NOTE: Both Power down bits need to be written as 0 only if both were set to 1 |

|  | NEXT STATE |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| **WAKES UP BY** | **PWR DWN Pin (A=Ass rtd)** | **nWAKE UP-EN PIN (A=Ass rtd)** | **ECOR FUNC ENABLE** | **ECSR PWR DOWN** | **CTR PWRDWN BIT** | **CTR WAKE UP_EN BIT** | **MCOR FUNC ENABLE** | **MCSR PWR DOWN** | **COMMENTS** |
| By writing 0 to ECSR Power Down taking the chip out of MP mode by writing WAKEUP_EN = 0 | nA | nA | 1 | 0 | 0 | 0 | - | - |  |
| By writing 0 to ECSR Power Down, a 0 to CTR PWRDWN, and a 0 to WAKEUP_EN bit | nA | nA | 1 | 0 | 0 | 0 | - | - |  |

### Table 8A - Effect of STSCHG_SLCT Bit in CR/Refer to table 8B for next states to wake-up events (*Rev. C and Higher)

| CURRENT STATE | | | | | | |
|---|---|---|---|---|---|---|
| No | MCOR FUNC ENABLE BIT | MCSR PWR DOWN BIT | CR STSCHG_ SLCT BIT (New) | MODEM IN POWER DOWN | Ring Event (Assuming Enabled) on nSTSCHG Pin (A=Assrtd) | Ring Event (Assuming Disabled) on nSTSCHG Pin (A=Assrtd) |
| 1 | 0 | X | 0 | Y | nA | nA |
| 2 | 0 | X | 1 | Y | A | nA |
| 3 | 1 | 0 | X | N | A | nA |
| 4 | 1 | 1 | X | Y | A | nA |

### Table 8B (*Rev. C and Higher)

| NEXT STATE | | | |
|---|---|---|---|
| MCOR FUNCTION ENABLE BIT | MCSR POWER DOWN BIT | CR STSCHG_SLCT BIT (New) | MODEM IN POWER DOWN |
| No Change | No Change | No Change | No Change |
| No Change | No Change | No Change | No Change |
| No Change | No Change | No Change | No Change |
| No Change | No Change | No Change | No Change |

### PCMCIA Attribute Memory: Address 0-7FFEh

The Attribute Memory is implemented using a combination of interrnal SRAM and external parallel EEPROM, ROM or Flash ROM. The internal SRAM is initialized during power up using the serial EEPROM. This serial EEPROM in PCMCIA mode is used for the CIS (Card Information Structure). If no serial EEPROM is used, the parallel EEPROM must be used. Internal CIS RAM address space is replaced by part of the external parallel EEPROM in this case.

In ISA mode, the serial EEPROM is used for configuration and IEEE Node address making it software compatible to the LAN9000 family of Ethernet LAN Controllers. In ISA mode, the EEPROM is optional requiring a minimum size of 64 X 16 bit word addresses. In PCMCIA mode, the minimum serial EEPROM (if used) size can be 64 X 16 up to 256 X 16.

This combination of internal and external attribute memory allows the designer to reduce costs by using a serial EEPROM device when using up to 512 bytes of "Card Information" and, if additional memory is needed, an external EEPROM may be used. When the LAN91C95 goes into powerdown mode, the internal CIS data buffer RAM is re-initialized.

The LAN91C95 generates the appropriate control lines (nFCS and nFWE) to read and write the Attribute memory, and it tri-states the data bus during external Attribute Memory accesses. Note that the parallel EEPROM is selected for the first 512 byte CIS information also in the absence of the serial EEPROM in PCMCIA mode. Only even locations are used.

### Internal VS External Attribute Memory Map

The Internal VS External EPROM attribute memory decodes are shown below. This allows the designer to not require an external EPROM device if the single or multi-function PCMCIA card needs less than 512 bytes of configuration information. As can be seen in the map, if 512 bytes of CIS or less is required, the nFCS and nFWE output pins of the LAN91C95 need not be used (if serial EEPROM is being used). Internal to the LAN91C95, the memory addressing logic will allow byte or word on odd byte address access (A0=1), the LAN91C95 will generate an arbitrary value of zero (0) since the PCMCIA specification states that the high byte of a word access in attribute memory is a don't care. This allows backward compatibility to 8 bit hosts.

**TABLE 9 - ATTRIBUTE MEMORY ADDRESS USING SERIAL EEPROM**

| ATTRIBUTE MEMORYADDRESS | EXTERNAL EPROM STORE | INTERNAL SRAM STORE (512 BYTES) | CONFIGURATION REGISTERS |
|---|---|---|---|
| 0 - 3FEh | | X | |
| 400h-7FFEh | X | | |
| 8000h - 803Eh | | | X |

**TABLE 10 - ATTRIBUTE MEMORY ADDRESS WITHOUT SERIAL EEPROM**

| ATTRIBUTE MEMORYADDRESS | EXTERNAL EPROM STORE | INTERNAL SRAM STORE (512 BYTES) | CONFIGURATION REGISTERS |
|---|---|---|---|
| 0 - 7FFEh | X | | |
| 8000h - 803Eh | | | X |

### PCMCIA Configuration Registers: Address 8000-803Eh

The PCMCIA Configuration Registers are stored inside the LAN91C95 above the external Attribute Memory address space. These registers are used to configure and control the PCMCIA related functionality of the Ethernet and Modem functions.

These registers are eight bits wide and reside on even locations. The LAN91C95 ignores odd accesses to this area (ignore writes, do not drive the bus on reads). This address offset has been changed from prior LAN9000 PCMCIA designs to allow a larger address range for other attribute memory data. This data could be a larger card information structure or a XIP data image.

## PCMCIA CONFIGURATION REGISTERS DESCRIPTION

Ethernet Function (Base Address 8000h)

8000h - Ethernet Configuration Option Register (ECOR)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SRESET | LevIREQ | | | WRATTRIB | Enable IREQ | Enable Base and Limit | Enable Function |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT 7 - SRESET: This bit when set will clear all internal registers associated with the Ethernet function except itself.

BIT 6 - LevIREQ: This bit is read only and reads as a one to indicate level mode interrupts are used. Pulse mode interrupts are not supported.

BIT 5 - Not defined

BIT 4 - Not defined

BIT 3 - WRATTRIB: This bit when when set (1), allows writting into the external attribute memory space.

BIT 2 - Enable IREQ Routing: This bit enables (1) or disables (0) the Ethernet function from asserting nIREQ. The nIREQ pin on power up and RESET is in a high (1) de-asserted state.

BIT 1 - Enable Base and Limit: This bit enables the on board I/O base decoder. If set, the on-board decoder is used to select the function. If cleared, the decoder is disabled and it is assumed that the host provides for the decoding. When the decode is disabled, the function is enabled, and is configured for 16 bits (and not in power-down mode), the signal nIOIS16 is always asserted. It is then up to the host to qualify the usage of nIOIS16. **For multifunction PCMCIA functionality, this bit must be set.**

BIT 0 - Enable Function: This bit enables (1) or disables (0) the Ethernet function. While the Ethernet function is disabled it remains in powerdown mode, no access to the Ethernet I/O space is allowed. (ie. The bank register is not accessable). IREQ is not generated for this function and nINPACK is not returned for accesses to the Ethernet registers.

If the “Magic Packet” function is enabled, the device is not completley powered down. The MAC controller is still enabled to receive.

**8002h - Ethernet Configuration and Status Register (ECSR)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | IOIs8 | | | Pwrdwn | Intr | IntrACK |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT 7 - Not defined

BIT 6 - Not defined

BIT 5 - IOIs8: This bit when set, indicates that the Host can only do 8 bit cycles (on D7-D0). The Ethernet function is forced in this case to eight bit mode regardless of the nEN16 pin and 16 BIT value. This bit also disables (floats) the -IOIs16 signal.

BIT 4 - Not defined

BIT 3 - Not defined

BIT 2 - PwrDwn: When set (1), this bit puts the LAN91C95 Ethernet function into powerdown mode. The Ethernet function is also put into powerdown mode when the Enable Function bit (ECOR bit 0) is cleared. Refer to the Powerdown Logic section for additional details as to what logic is powered down.

BIT 1 - Intr: This bit is read/set to a one when this function is requesting interrupt service. It is cleared depending upon the setting of IntrACK. When this bit and Enable IREQ Routing are set, IREQ Out is asserted.

All setting and resetting of this bit is edge triggered with exception of the internally generated reset signal for the modem Ethernet related PC card registers. The Intr bit can be reset the following ways and priority ranging from 1=highest to 3=lowest:

1) A hardware reset/power up
2) The function ie. interrupt source can only reset this field to zero (0) if the IntrACK field is reset to zero (0)
3) The host system can only reset this field to a zero (0) if the IntrACK bit is set to a one (1)

BIT 0 - IntrACK: This bit controls the clearing of the Intr bit. When this bit is cleared, Intr reflects the function's interrupt status. When this bit is set, the Intr bit must be cleared by the host writing a 0 into it. If the function requires additional service the Intr bit will remain asserted when the host writes the 0.

### I/O Base Register 0 & 1 (I/O Base 0 & 1) Address 800Ah & 800Ch

#### 800Ah - Ethernet I/O BASE Register 0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| A7 | A6 | A5 | A4 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

#### 800Ch - Ethernet I/O BASE Register 1

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

The I/O Base registers determine the base address of the I/O range used to access function specific registers. These registers allow the function's registers to be placed anywhere in the host's I/O space. I/O Base 0 contains the low order byte (A7-A0) and I/O Base 1 contains the high order byte (A15-A8). Since the Ethernet function requires 16 I/O locations, bits 3-0 of I/O Base 0 are always 0.

Only A15 to A4 are decoded by the controller (64K address space), it is up to the host to resolve any conflicts with addressing above 64K. The default decode value is 300h (A9=A8=1, others=0). NOTE: These registers are ignored in ISA mode. These registers are accessable even if the “Enable Base and Limit) bit in the MCOR is cleared (0).

## Modem Function (Base Address 8020h)

### 8020h - Modem Configuration Option Register (MCOR)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SRESET | LevIREQ | | | | Enable IREQ | Enable Base and Limit | Enable Function |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT 7 - SRESET: This bit when set clears all internal registers associated with the Modem function (except itself) and asserts the nMRESET pin .

BIT 6 - LevIREQ: This bit is read only and reads as a one to indicate level mode interrupts are used. Pulse mode interrupts are not supported.

BIT 5 - Not defined

BIT 4 - Not defined

BIT 3 - Not defined

BIT 2 - Enable IREQ Routing: This bit enables (1) or disables (0) the Modem function from asserting nIREQ.

BIT 1 - Enable Base and Limit: This bit enables the on board modem I/O base decoder. If set, the on-board decoder is used to select the function. If cleared, the decoder is disabled and it is assumed that the host provides for the decoding. When the decode is disabled, the function is enabled, and is configured for 16 bits (and not in power-down mode), the signal nIOIS16 is always asserted. **For multifunction PCMCIA functionality, this bit must be set.**

BIT 0 - Enable Function: This bit enables (1) or disables (0) the Modem function. While the Modem is disabled the LAN91C95 inhibits nMCS, IREQ is not generated for the Modem function and nINPACK is not returned for accesses to the Modem registers.

**8022h - Modem Configuration and Status Register (MCSR) Address 8022h**

| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|
| Changed | SigChg | IOIs8 | Reserved | Audio | Pwrdwn | Intr | IntrACK |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit 7 - Changed: This bit is the logical OR of the CREADY/-Bsy and (RINGEVENT bit logically anded with RINGENABLE) states.

Bit 6 - SigChg: If this bit is a one, the function is enabled (configured), the Changed bit controls the nSTSCHG pin. If this bit is low or the function is disabled, the nSTSCHG pin is set to a high.

Bit 5 - IOIs8: This bit when set, indicates that the Host can only do 8 bit cycles (on D7-D0). In the case of the IOIs8 bit being cleared (0), during Reset and Power up for example, the PIN MIS16 will override the IOIs8 default, setting the bit (1).

Bit 4 - ResurvedBit 3 - Audio: This bit controls the audio pass through of the digital audio. When cleared, the SPKROUT pin is three-stated. When set, the SPKRIN pin is passed to the SPKROUT pin.

Bit 2 - PwrDwn: When set (1), this bit puts the modem into powerdown mode. The modem is also put into powerdown mode when the Enable Function bit is cleared. When in powerdown, the MRINGIN signal is blocked from going to the ringing output signal. When taken out of powerdown (when PwrDwn is cleared (1) and Enable Function in the MCOR (bit 0) is set (1)), the modem is awakened by a pulse on the output ringing signals as appropriate. The pulse duration is determined by the input signal MRINGIN. MRINGIN is then passed to the outputs.

Bit 1 - Intr: This bit is read/set to a one when this function is requesting interrupt service. It is cleared depending upon the setting of IntrACK. When this bit and Enable IREQ Routing are set, -IREQOut is asserted.

The Intr bit can be reset the following ways and priority ranging from 1=highest to 3=lowest:

1) A hardware reset/power up
2) The function ie. interrupt source can only reset this field to zero (0) if the IntrACK field is reset to zero (0)
3) The host system can only reset this field to a zero (0) if the IntrACK bit is set to a one (1)

Bit 0 - IntrACK: This bit controls the clearing of the Intr bit. When this bit is cleared, Intr reflects the function's interrupt status. When this bit is set, the Intr bit must be cleared by the host writing a 0 into it. If the function requires additional service (indicated by leaving MINT active) the Intr bit will remain asserted when the host writes the 0.

### 8024h - Pin Replacement Register (PRR)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
|  |  | Cready/<br>-Bsy |  |  |  | Rready/<br>-Bsy |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Cready/-Bsy: This bit is set to a one when the bit Rready/-Bsy bit changes state from zero(0) to one (1) or one (1) to zero (0) with the source of the change of state is a change in the modem ready (MRDY) signal. The Cready/-Bsy bit can be written by the CPU also. The CPU attempt to write to this bit is masked by the value for Rready/Bsy bit. A CPU write to this bit is successful only if the Rready/-Bsy bit is being written as one (1). Note that the h/w represented value of the Rready/-Bsy bit itself is not affected by the write attempt as shown in the following illustration:

| CURRENT VALUE |  | VALUE WRITTEN |  | NEW VALUE |  |
|---|---|---|---|---|---|
| Cready/-Bsy | Rready/-Bsy | Cready/-Bsy | Rready/-Bsy | Cready/-Bsy | Rready/-Bsy |
| 0 | 0 | 1 | 0 | 0 | 0 |
| X | 0 | 1 | 1 | 1 | 0 |
| X | 0 | 0 | 1 | 0 | 0 |
| X | 1 | 1 | 1 | 1 | 1 |

In the unlikely event that the MRDY changes state at the same time that the nWE signal changes from low to high (ie. writing the register), the value written by the host will have priority over the new state of the MRDY input pin.

Rready/-Bsy: When read, this bit represents the current state of the modem Ready/-Busy (MRDY) signal. Attempts of CPU writes to this bit are ignored.

**8028h - Extended Status Register(ESR)**

| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|
| | | | RINGEVENT | | | | RINGENABLE |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

RINGEVENT: This bit is latched to a one at the start of each ring frequency cycle (input from ring input from modem, the MRINGIN signal going high). When this bit and RINGENABLE are both set to a one (1), the Changed bit in the MCSR is set to a one(1). This bit is reset if the host writes a one (1) and the change bit in the MCSR register is unaffected if the host writes a zero (0).

RINGENABLE: When set, this bit gates the RINGEVENT into the Changed bit.

**802Ah - Modem I/O BASE Register 0**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| A7 | A6 | A5 | A4 | A3 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**802Ch - Modem I/O BASE Register 1**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

The Modem I/O Base registers determine the base address of the I/O range used to access function specific registers. These registers allow the function's registers to be placed anywhere in the host's I/O space. I/O Base 0 contains the low order byte (A7-A0) and I/O Base 1 contains the high order byte (A15-A8). Since the modem function requires 8 I/O locations, bits 2-0 of I/O Base 0 are always 0. Since only A15 to A4 are decoded by the controller (64K address space), it is up to the host to resolve any conflicts with addressing above 64K.

These registers are still accessable even if the “Enable Base and Limit” bit in the MCOR is cleared (0).

**8032h - Modem I/O Size Register**

| Modem I/O Size Mask | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

The I/O Size Register holds a bit mask used to specify the number of address lines decoded by the Modem function. Each bit in the register represents an I/O address line. A value of one means that the LAN91C95 will not decode the corresponding address line for the Modem function, a value of zero means that the address line will be decoded. If a bit in the register is set to one, all bits of lesser significance in the register must also be set to one.

This register defaults to 7h, that is A0, A1 and A2 are not decoded resulting in 8 address locations for the modem function.

**Modem I/O Space:** Address determined by Modem I/O Base Registers

The Modem I/O space is external to the LAN91C95. The LAN91C95 decodes the address bus ( A15 through A3) via a internal comparator and generates nMCS for I/O cycles that decode to the Modem area as defined by the Modem I/O Base Registers. The I/O address space consists of eight (8) 8 bit locations. The LAN91C95 will tri-state the data bus during Modem I/O space accesses. In ISA mode, nMCS is disabled (1). No address decodeing for the modem will be done.

**Ethernet I/O Space**: Address determined by Ethernet I/O Base Registers

The Ethernet I/O space consists of sixteen locations whose base address is determined in PCMCIA mode by the Ethernet I/O Base Registers, and in ISA mode by the default value of the Base Register (either reset or serial EEPROM default in ISA mode only).

The Ethernet I/O space can be configured as an 8 or 16 bit space, and is similar to the LAN91C95, LAN91C92, etc. I/O space. To limit the I/O space requirements to 16 locations, the registers are split into six banks. The last word of the I/O area is shared by all banks and can be used to change the bank in use. Banks 0 through 3 functionally correspond to the LAN91C95 banks, while Banks 4 and 5 allow access to the multifunction registers in ISA mode.

## I/O SPACE (ISA and PCMCIA Mode)

In ISA mode, the base I/O space is determined by the IOS0-IOS2 inputs and the EEPROM contents. A4-A15 are compared against the base I/O address for I/O space accesses.

In PCMCIA mode nREG (along with nIORD or nIOWR) defines an I/O access regardless of the A4-A15 value.

To limit the I/O space requirements to 16 locations, the registers are assigned to different banks. The last word of the I/O area is shared by all banks and can be used to change the bank in use.

Registers are described using the following convention:

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
| | X | X | X | X | X | X | X | X |
| LOW BYTE | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
| | X | X | X | X | X | X | X | X |

OFFSET - Defines the address offset within the IOBASE where the register can be accessed at, provided the bank select has the appropriate value.

The offset specifies the address of the even byte (bits 0-7) or the address of the complete word.

The odd byte can be accessed using address (offset + 1).

Some registers (like the Interrupt Ack., or like Interrupt Mask) are functionally described as two eight bit registers; in that case the offset of each one is independently specified.

Regardless of the functional description, when the LAN91C95 is in 16 bit mode, all registers can be accessed as words or bytes.

The default bit values upon hard reset are highlighted below each register.

**TABLE 11 -**

**Internal I/O Space Map**

| | **BANK0** | **BANK1** | **BANK2** | **BANK3** | **BANK4** | **BANK5** |
|---|---|---|---|---|---|---|
| 0 | TCR | CONFIG | MMU COMMAND | MT0-MT1 | ECOR (low byte)<br>ECSR (high byte) | MCOR (low byte)<br>MCSR (high byte) |
| 2 | EPH STATUS | BASE | PNR ARR | MT2-MT3 | | PRR (low byte) |
| 4 | RCR | IA0-IA1 | FIFO PORTS | MT4-MT5 | EBASE0 (high byte) | IOEIR (low byte)<br>MBASE0 (high byte) |
| 6 | COUNTER | IA2-IA3 | POINTER | MT6-MT7 | EBASE1 (low byte) | MBASE1 (low byte) |
| 8 | MIR | IA4-IA5 | DATA | MGMT | | Msize (high byte) |
| A | MCR | GENERAL PURPOSE | DATA | REVISION | | |
| C | RESERVED (0) | CONTROL | INTERRUPT | ERCV | | |
| E | BANK SELECT | BANK SELECT | BANK SELECT | BANK SELECT | | |

**BANK SELECT REGISTER**

**I/O REGISTERS DESCRIPTION**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| E | BANK SELECT REGISTER | READ/WRITE | BSR |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| LOW BYTE | Res. | Res. | Res. | Res. | Res. | BS2 | BS1 | BS0 |
| | X | X | X | X | X | 0 | 0 | 0 |

BS2, BS1, BS0 - Determine the bank presently in use.

The BANK SELECT REGISTER is always accessible except in PCMCIA powerdown mode and is used to select the register bank in use.

The upper byte always reads as 33h and can be used to help determine the I/O location of the LAN91C95.

The BANK SELECT REGISTER is always accessible regardless of the value of BS0-BS2.

The LAN91C95 implements only four banks in ISA mode, therefore, accesses to non-existing banks (BS2=1) are ignored.  All six banks are accessible in PCMCIA mode.

| BS2 | BS1 | BS0 | BANK# |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | X | X | None |

**I/O SPACE - BANK0**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| 0 | TRANSMIT CONTROL REGISTER | READ/WRITE | TCR |

This register holds bits programmed by the CPU to control some of the protocol transmit options.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | FDSE | Res | EPH LOOP | STP SQET | FDUPLX | MON_ CSN | Res | NOCRC |
| | 0 | X | 0 | 0 | 0 | 0 | X | 0 |
| LOW BYTE | PAD_EN | Res | Res | Res | Res | FORCOL | LOOP | TXENA |
| | 0 | X | X | X | X | 0 | 0 | 0 |

FDSE - Full Duplex Switched Ethernet. When set, the LAN91C95 is configured for Full Duplex Switched Ethernet, it defaults clear to normal CSMA/CD protocol. In FDSE mode the LAN91C95 transmit and receive processes are fully independent, namely no deferral and no collision detection are implemented. When FDSE is set, FDUPLX is internally assumed high and MON_CSN is assumed low regardless of their actual values.

EPH_LOOP - Internal loopback at the EPH block. Does not exercise the encoder decoder. Serial data is looped back when set. Defaults low.

NOTE: After exiting the loopback test, an SRESET in the ECOR or the SOFT_RST in the RCR must be set before returning to normal operation.

STP_SQET - Stop transmission on SQET error. If set, stops and disables transmitter on SQE test error. Does not stop on SQET error and transmits next frame if clear. Defaults low.

FDUPLX - When set, it enables full duplex operation. This will cause frames to be received if they pass the address filter regardless of the source for the frame. When clear the node will not receive a frame sourced by itself. Clearing this bit (Normal Operation) in promiscuous mode allows it to not receive its own packet.

MON_CSN - When set, the LAN91C95 monitors carrier while transmitting. It must see its own carrier by the end of the preamble. If it is not seen, or if carrier is lost during transmission, the transmitter aborts the frame without CRC and turns itself off. When this bit is clear the transmitter ignores its own carrier. Defaults low.

NOCRC - Does not append CRC to transmitted frames when set; allows software to insert the desired CRC. Defaults to zero, namely CRC inserted.

PAD_EN - When set, the LAN91C95 will pad transmit frames shorter than 64 bytes with 00. Does not pad frames when reset.

FORCOL - When set, the transmitter will force a collision by deliberately not deferring. After the collision, this bit is automatically reset. This bit defaults low for normal operation. Note: The LATCOL bit in EPHSR, setting up as a result of FORCOL, will reset TXENA to 0. In order to force another collision, TXENA must be set to 1 again.

LOOP - Local Loopback. When set, transmit frames are internally looped to the receiver after the encoder/decoder. Collision and Carrier Sense are ignored. No data is sent out. Defaults low for normal mode.

TXENA - Transmit enabled when set. Transmit is disabled if clear. When the bit is cleared, the LAN91C95 will complete the current transmission before stopping. When stopping due to an error, this bit is automatically cleared.

| **AUI** | **FDSE** | **FDUPLX** | **EPH_LOOP** | **LOOP** | **LOOPS AT** | **TRANSMITS TO NETWORK** |
|---|---|---|---|---|---|---|
| X | X | X | 1 | X | EPH Block | No |
| X | X | 1 | 0 | 1 | ENDEC | No |
| 1 | 0 | 1 | 0 | 0 | Cable | Yes |
| 0 | 0 | 1 | 0 | 0 | 10BASE-T Driver | Yes |
| X | 0 | 0 | 0 | 0 | Normal CSMA/CD - No Loopback | Yes |
| X | 1 | 1 | 0 | 0 | Full Duplex Switched Ethernet - No loopback and no SQET | Yes |

**I/O SPACE - BANK0**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **2** | **EPH STATUS REGISTER** | **READ ONLY** | **EPHSR** |

This register stores the status of the last transmitted frame. This register value, upon individual transmit packet completion, is stored as the first word in the memory area allocated to the packet. Packet interrupt processing should use the copy in memory as the register itself will be updated by subsequent packet transmissions. The register can be used for real time values (like TXENA and LINK OK). If TXENA is cleared the register holds the last packet completion status.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH<br>BYTE | TX<br>UNRN | LINK_<br>OK | RX_<br>OVRN | CTR<br>_ROL | EXC<br>_DEF | LOST<br>CARR | LATCOL | WAKEUP |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW<br>BYTE | TX<br>DEFR | LTX<br>BRD | SQET | 16COL | LTX<br>MULT | MUL<br>COL | SNGL<br>COL | TX_SUC |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TXUNRN - Transmit Underrun. Set if underrun occurs, it also clears TXENA bit in TCR. Cleared by setting TXENA high. This bit may only be set if early TX is being used.

LINK_OK - State of the 10BASE-T Link Integrity Test. A transition on the value of this bit generates an interrupt when the LE ENABLE bit in the Control Register is set.

RX_OVRN - Upon FIFO overrun, the receiver asserts this bit and clears the FIFO. The receiver stays enabled. After a valid preamble has been detected on a subsequent frame, RX_OVRN is de-asserted. The RX_OVRN INT bit in the Interrupt Status Register will also be set and stay set until cleared by the CPU. Note that receive overruns could occur only if receive memory allocations fail.

CTR_ROL - Counter Roll over. When set one or more 4 bit counters have reached maximum count (15). Cleared by reading the ECR register.

EXC_DEF - Excessive deferral. When set last/current transmit was deferred for more than 1518 * 2 byte times. Cleared at the end of every packet sent.

LOST_CARR - Lost carrier sense. When set indicates that Carrier Sense was not present at end of preamble. Valid only if MON_CSN is enabled. This condition causes TXENA bit in TCR to be reset. Cleared by setting TXENA bit in TCR.

LATCOL - Late collision detected on last transmit frame. If set a late collision was detected (later than 64 byte times into the frame). When detected the transmitter jams and turns itself off clearing the TXENA bit in TCR. Cleared by setting TXENA in TCR.

WAKEUP - When this bit is set, it indicates that a receive packet was received that had the “Magic Packet” (MP) signature of the node’s own IP address repetitions in it. This bit indicates a valid detection for magic packet - enabled by nWAKEUPEN pin (92 QFP) or WAKEUP_EN in CTR.

Note: If the MP mode is activated using the nWAKEUPEN pin, the pin must be deasserted to exit the mode.

TX_DEFR - Transmit Deferred. When set, carrier was detected during the first 6.4 µsec of the inter frame gap. Cleared at the end of every packet sent.

LTX_BRD - Last transmit frame was a broadcast. Set if frame was broadcast. Cleared at the start of every transmit frame.

SQET - Signal Quality Error Test. The transmitter opens a 1.6 µsec window 0.8 µsec after transmission is completed and the receiver returns inactive. During this window, the transmitter expects to see the SQET signal from the transceiver. The absence of this signal is a 'Signal Quality Error' and is reported in this status bit. Transmission stops and EPH INT is set if STP_SQET is in the TCR is also set when SQET is set. This bit is cleared by setting TXENA high.

16COL- 16 collisions reached. Set when 16 collisions are detected for a transmit frame. TXENA bit in TCR is reset. Cleared when TXENA is set high.

LTX_MULT - Last transmit frame was a multicast. Set if frame was a multicast. Cleared at the start of every transmit frame.

MULCOL - Multiple collision detected for the last transmit frame. Set when more than one collision was experienced. Cleared when TX_SUC is high at the end of the packet being sent.

SNGLCOL - Single collision detected for the last transmit frame. Set when a collision is detected. Cleared when TX_SUC is high at the end of the packet being sent.

TX_SUC - Last transmit was successful. Set if transmit completes without a fatal error. This bit is cleared by the start of a new frame transmission or when TXENA is set high.

Fatal errors are:

16 collisions
SQET fail and STP_SQET = 1
FIFO Underrun
Carrier lost and MON_CSN = 1
Late collision

**I/O SPACE - BANK0**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **4** | **RECEIVE CONTROL REGISTER** | **READ/WRITE** | **RCR** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | SOFT_ RST | FILT_ CAR | Res | Res | Res | Res | STRIP_ CRC | RXEN |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW BYTE | Res | Res | Res | Res | Res | ALMUL | PRMS | RX_ ABORT |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SOFT_RST - Software activated Reset. Active high. Initiated by writing this bit high and terminated by writing the bit low. The LAN91C95 configuration is not preserved, except for Configuration, Base, and IA0-IA5 Registers. The EEPROM in both ISA and PCMCIA mode is not reloaded after software reset.

FILT_CAR - Filter Carrier. When set filters leading edge of carrier sense for 12 bit times. Otherwise recognizes a receive frame as soon as carrier sense is active.

STRIP_CRC - When set it strips the CRC on received frames. When clear the CRC is stored in memory following the packet. Defaults low.

RXEN - Enables the receiver when set. If cleared, completes receiving current frame and then goes idle. Defaults low on reset.

ALMUL - When set accepts all multicast frames (frames in which the first bit of DA is '1'). When clear accepts only the multicast frames that match the multicast table setting. Defaults low.

PRMS - Promiscuous mode. When set receives all frames. Does not receive its own transmission unless it is in full duplex.

RX_ABORT - This bit is set if a receive frame is aborted due to length longer than 1532 bytes. The frame will not be received. The bit is cleared by RESET or by the CPU writing it low.

| | **RX_ABORT** | **RX_OVRN_INT** |
|---|---|---|
| Packet Too Long | 1 | 0 |
| Run out of Memory During Receive | 1 | 1 |

**I/O SPACE - BANK0**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **6** | **COUNTER REGISTER** | **READ ONLY** | **ECR** |

Counts four parameters for MAC statistics. When any counter reaches 15 an interrupt is issued. All counters are cleared when reading the register and do not wrap around beyond 15.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | NUMBER OF EXC. DEFFERED TX | | | | NUMBER OF DEFFERED TX | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW BYTE | MULTIPLE COLLISION COUNT | | | | SINGLE COLLISION COUNT | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Each four bit counter is incremented every time the corresponding event, as defined in the EPH STATUS REGISTER bit description, occurs. Note that the counters can only increment once per enqueued transmit packet, never faster, limiting the rate of interrupts that can be generated by the counters. For example if a packet is successfully transmitted after one collision the SINGLE COLLISION COUNT field is incremented by one. If a packet experiences between 2 to 16 collisions, the MULTIPLE COLLISION COUNT field is incremented by one. If a packet experiences deferral the NUMBER OF DEFERRED TX field is incremented by one, even if the packet experienced multiple deferrals during its collision retries.

The COUNTER REGISTER facilitates maintaining statistics in the AUTO RELEASE mode where no transmit interrupts are generated on successful transmissions.

Reading the register in the transmit service routine will be enough to maintain statistics.

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| 8 | MEMORY INFORMATION REGISTER | READ ONLY | MIR |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | FREE MEMORY AVAILABLE (IN BYTES * 256 * M) | | | | | | | |
| | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| LOW BYTE | MEMORY SIZE (IN BYTES *256 * M) | | | | | | | |
| | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

FREE MEMORY AVAILABLE - This register can be read at any time to determine the amount of free memory. The register defaults to the MEMORY SIZE upon reset or upon the RESET MMU command.

MEMORY SIZE - This register can be read to determine the total memory size, and will always read 18H (6144 bytes) for the LAN91C95.

| | MEMORY SIZE REGISTER | M | ACTUAL MEMORY |
|---|---|---|---|
| LAN91C90 | FFH | 1 | 64 kbytes |
| LAN91C90 | 40H | 1 | 16 kbytes |
| LAN91C92/4 | 12H | 1 | 4608 bytes |
| LAN91C95 | 18H | 1 | 6144 bytes |
| LAN91C100 | FFH | 2 | 128 kbytes |

**I/O SPACE - BANK0**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| A | MEMORY CONFIGURATION REGISTER | Lower Byte - READ/WRITE<br>Upper Byte - READ ONLY | MCR |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | | | | | MEMORY SIZE MULTIPLIER | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| LOW BYTE | MEMORY RESERVED FOR TRANSMIT (IN BYTES * 256 * M) | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

MEMORY RESERVED FOR TRANSMIT - Programming this value allows the host CPU to reserve memory to be used later for transmit, limiting the amount of memory that receive packets can use up.

When programmed for zero, the memory allocation between transmit and receive is completely dynamic.

When programmed for a non-zero value, the allocation is dynamic if the free memory exceeds the programmed value, while receive allocation requests are denied if the free memory is less or equal to the programmed value.

This register defaults to zero upon reset. It is not affected by the RESET MMU command.

The value written to the MCR is a reserved memory space IN ADDITION TO ANY MEMORY CURRENTLY IN USE. If the memory allocated for transmit plus the reserved space for transmit is required to be constant (rather than grow with transmit allocations) the CPU should update the value of this register after allocating or releasing memory.

The contents of MIR as well as the low byte of MCR are specified in 256 * M bytes. The multiplier M is determined by bits 11,10, and 9 as follows. Bits 11,10 and 9 are read only bits used by the software driver to transparently run on different controllers of the LAN9000 family.

| DEVICE | BIT 11 | BIT 10 | BIT 9 | M | MAX MEMORY SIZE |
|---|---|---|---|---|---|
| FEAST | 0 | 1 | 0 | 2 | 256*256*2=128k |
| LAN91C90 | 0 | 0 | 1 | 1 | 256*256*1=64k |
| FUTURE | 0 | 1 | 1 | 4 | 256k |
| FUTURE | 1 | 0 | 0 | 8 | 512k |
| FUTURE | 1 | 0 | 1 | 16 | 1M |

**I/O SPACE - BANK1**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **0** | **CONFIGURATION REGISTER** | **READ/WRITE** | **CR** |

The Configuration Register holds bits that define the device configuration and are not expected to change during run-time. This register is part of the EEPROM saved setup in ISA mode only. In PCMCIA mode, this register is initialized to the state as defined below the corresponding bits as if no EEPROM was present in ISA mode (i.e. ENEEP pin is a don't care in PCMCIA mode).

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | 0 | | | NO WAIT | *STSCHG _SLCT* | FULL STEP | SET SQLCH | AUI SELECT |
| | 0 | X | X | 0 | *0* | 0 | 0 | 0 |
| LOW BYTE | 16 BIT | DIS LINK | Reserved | | | INT SEL1 | INT SEL0 | |
| | Function of EN16 pin | 0 | 1 | 1 | 0 | 0 | 0 | X |

NO WAIT - When set, does not request additional wait states. An exception to this are accesses to the Data Register if not ready for a transfer. When clear, negates IOCHRDY for two to three 20 MHz clocks on any cycle to the LAN91C95.

*STSCHG_SLCT(*) - PCMCIA mode only. With STSCHG_SLCT low (0), Modem Function Enable bit = 0, in MCOR, it does not allow any events (such as Ring In) to be reported to the host via the nSTSCHG Pin assertion. With STSCHG_SLCT high (1), Functon Enable = 0, and does not prohibit any events (such as Ring In) to be reported to the host via the nSTSCHG Pin assertion. Look for further event enabling and individual enable bits (such as RINGENABLE in the MESR) information in the PCMCIA Registers description.*

*Note *: Rev. C or higher*
*Note: SRESET bit in MCOR, SRESET bit in ECOR, or SOFT_RST bit in RCR assertion have no effect on the STSCHG_SLCT bit. POR or chip reset with the RESET pin will reset the bit to zero (0).*

FULL STEP - This bit is used to select the signaling mode for the AUI port. When set the AUI port uses full step signaling. Defaults low to half step signaling. This bit is only meaningful when AUI SELECT is high.

SET SQLCH - When set, the squelch level used for the 10BASE-T receive signal is 240mV. When clear the receive squelch level is 400mV. Defaults low.

AUI SELECT - When set the AUI interface is used, when clear the 10BASE-T interface is used. Defaults low.

16 BIT - Used in conjunction with nEN16 and IOis8 (in PCMCIA mode only) to define the width of the system bus. If the nEN16 pin is low, this bit is forced high. Otherwise the bit defaults low and can be programmed by the host CPU.

DIS LINK - This bit is used to disable the 10BASE-T link test functions. When this bit is high the

LAN91C95 disables link test functions by not generating nor monitoring the network for link pulses. In this mode the LAN91C95 will transmit packets regardless of the link test, the EPHSR LINK_OK bit will be set and the LINK LED will stay on. When low the link test functions are enabled. If the link status indicates FAIL, the EPHSR LINK_OK bit will be low, while transmit packets enqueued will be processed by the LAN91C95, transmit data will not be sent out to the cable.

INT SEL1-0 - In ISA mode, used to select one out of four interrupt pins. The three unused interrupts are tristated.

| INT SEL1 | INT SEL0 | INTERRUPT PIN USED |
|---|---|---|
| 0 | 0 | INTR0 |
| 0 | 1 | INTR1 |
| 1 | 0 | INTR2 |
| 1 | 1 | INTR3 |

## I/O SPACE - BANK1

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| 2 | BASE ADDRESS REGISTER | READ/WRITE | BAR |

For ISA mode only, this register holds the I/O address decode option chosen for the I/O and ROM space. It is part of the EEPROM saved setup, and is not usually modified during run-time. NOTE: This register should ONLY be used in ISA mode. In PCMCIA mode, this register is read only.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | A15 | A14 | A13 | A9 | A8 | A7 | A6 | A5 |
| | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| LOW BYTE | ROM SIZE | | RA18 | RA17 | RA16 | RA15 | RA14 | |
| | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

A15 - A13 and A9 - A5 - These bits are compared in ISA mode against the I/O address on the bus to determine the IOBASE for LAN91C95 registers. The 64k I/O space is fully decoded by the LAN91C95 down to a 16 location space, therefore the unspecified address lines A4, A10, A11 and A12 must be all zeros.

ROM SIZE - Determines the ROM decode area in ISA mode memory space as follows:

00 = ROM disable
01 = 16k: RA14-18 define ROM select
10 = 32k: RA15-18 define ROM select
11 = 64k: RA16-18 define ROM select

RA18-RA14 - These bits are compared in ISA mode against the memory address on the bus to determine if the ROM is being accessed, as a function of the ROM SIZE. ROM accesses are read only memory accesses defined by nMEMRD going low.

For a full decode of the address space unspecified upper address lines have to be:

> A19 = "1", A20-A23 lines are not directly decoded, however ISA systems will only activate nMEMRD only when A20-A23=0.

All bits in this register are loaded from the serial EEPROM in ISA Mode only. In PCMCIA mode, the I/O base is set to the default value (as in ISA mode) as defined below.

The I/O base decode defaults to 300h (namely, the high byte defaults to 18h). ROM SIZE defaults to 01. ROM decode defaults to CC000 (namely the low byte defaults to 67h).

**I/O SPACE - BANK1**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **4 THROUGH 9** | **INDIVIDUAL ADDRESS REGISTERS** | **READ/WRITE** | **IAR** |

These registers are loaded starting at word location 20h of the EEPROM upon hardware reset or EEPROM reload in ISA mode only. The registers can be modified by the software driver, but a STORE operation will not modify (in ISA mode only) the EEPROM Individual Address contents.

In PCMCIA mode, the IEEE Individual Address is stored in the EEPROM, but is stored in PCMCIA Tuple format as defined in the Metaformat specification. Refer to the PCMCIA v3.0 card specification on the Metaformat.

The LAN91C95 in PCMCIA mode knows nothing about the location or structure of the IEEE Ethernet Address stored in the EEPROM. Once this data is stored in the CIS SRAM data buffer in the LAN91C95, it is parsed by the host to extract the IEEE Address information and stored manualy by the LAN Driver.

Bit 0 of Individual Address 0 register corresponds to the first bit of the address on the cable.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | ADDRESS 0 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW BYTE | ADDRESS 1 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HIGH BYTE | ADDRESS 2 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW BYTE | ADDRESS 3 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HIGH BYTE | ADDRESS 4 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW BYTE | ADDRESS 5 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| A | GENERAL PURPOSE REGISTER | READ/WRITE | GPR |

<table>
<tr><td rowspan="2">HIGH<br>BYTE</td><td colspan="8">HIGH DATA BYTE</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td rowspan="2">LOW<br>BYTE</td><td colspan="8">LOW DATA BYTE</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

This register can be used as a way of storing and retrieving non-volatile information in the EEPROM to be used by the software driver. The storage is word oriented, and the EEPROM word address to be read or written is specified using the six lowest bits of the Pointer Register. In PCMCIA mode, Bits 0 to 10 of the pointer register are used.

This register can also be used to sequentially program the Individual Address area of the EEPROM, that is normally protected from accidental Store operations.

This register will be used for EEPROM read and write only when the EEPROM SELECT bit in the Control Register is set. This allows generic EEPROM read and write routines that do not affect the basic setup of the LAN91C95.

**I/O SPACE - BANK1**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| C | CONTROL REGISTER | READ/WRITE | CTR |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | 0 | RCV_ BAD | PWRDWN | WAKE UP_EN | AUTO RELEASE | | | 1 |
| | 0 | 0 | 0 | 0 | 0 | X | X | 1 |
| LOW BYTE | LE ENABLE | CR ENABLE | TE ENABLE | | | EEPROM SELECT | RELOAD | STORE |
| | 0 | 0 | 0 | X | X | 0 | 0 | 0 |

RCV_BAD - When set, bad CRC packets are received. When clear bad CRC packets do not generate interrupts and their memory is released.

PWRDN - Active high bit used to put the Ethernet function in powerdown mode.

Cleared by:

1. A write to any register in the LAN91C95 I/O space
2. Hardware reset
3. “Magic Packet” was received

This bit is combined with the Pwrdwn bit in ECSR and with the powerdown bit to determine when the function is powered down.

WAKUP_EN - Active high bit used to enable the controller in any of the powerdown modes to power up if a “Magic Packet” is detected and set the WAKEUP bit in the EPHSR to generate an EPH interrupt (if not masked). When clear (0), no “Magic Packet” scanning is done on receive packets.

NOTE: If “Magic Packet” is enabled using this bit only the notification is done on the WAKEUP bit in the EPHSR only.

AUTO RELEASE - When set, transmit pages are released by transmit completion if the transmission was successful (when TX_SUC is set). In that case there is no status word associated with its packet number, and successful packet numbers are not even written into the TX COMPLETION FIFO. A sequence of transmit packets will only generate an interrupt when the sequence is completely transmitted (TX EMPTY INT will be set), or when a packet in the sequence experiences a fatal error (TX INT will be set).

Upon a fatal error TXENA is cleared and the transmission sequence stops. The packet number that failed is the present in the FIFO PORTS register, and its pages are not released, allowing the CPU to restart the sequence after corrective action is taken.

LE ENABLE - Link Error Enable. When set it enables the LINK_OK bit transition as one of the interrupts merged into the EPH INT bit. Defaults low (disabled). Writing this bit also serves as the acknowledge by clearing previous LINK interrupt conditions.

CR ENABLE - Counter Roll over Enable. When set it enables the CTR_ROL bit as one of the interrupts merged into the EPH INT bit. Defaults low (disabled).

TE ENABLE - Transmit Error Enable. When set it enables Transmit Error as one of the interrupts merged into the EPH INT bit. Defaults low (disabled). Transmit Error is any condition that clears TXENA with TX_SUC staying low as described in the EPHSR register.

EEPROM SELECT - This bit allows the CPU to specify which registers the EEPROM RELOAD or STORE refers to. When high, the General Purpose Register is the only register read or written. When low, the RELOAD and STORE functions are enabled.

RELOAD

In ISA Mode: The LAN91C95 reads the Configuration, Base and Individual Address, and STORE writes the Configuration and Base registers. Also when set it will read the EEPROM and update relevant registers with its contents. This bit then Clears upon completing the operation.

In PCMCIA Mode: The LAN91C95 reads the contents of the EEPROM and stores the contents in the LAN91C95 CIS SRAM as defined in Table 12.

STORE

In ISA Mode: The STORE bit when set, stores the contents of all relevant registers in the serial EEPROM. This bit is cleard upon completing the operation.

In PCMCIA Mode: The LAN91C95 performs no function.

NOTE: When an EEPROM access is in progress the STORE and RELOAD bits will be read back as high. The remaining 14 bits of this register will be invalid. During this time, attempted read/write operations, other than polling the EEPROM status, will NOT have any effect on the internal registers. The CPU can resume accesses to the LAN91C95 after both bits are low. A worst case RELOAD operation initiated by RESET or by software takes less than 750 µsec.

**PCMCIA EEPROM to SRAM Memory Map**

As defined in the PCMCIA specification, Odd byte attribute memory locations are a don’t care. In order to utilize the serial EEPROM and internal LAN91C95 SRAM, the data to memory mapping is shown in Table 12.

**Table 12 -**
**PCMCIA - EEPROM to SRAM Memory Map**

| ATTRIBUTE MEMORY HOST ADDRESS (HEX) | ATTRIBUTE DATA | EEPROM ADDRESS IN WORDS | LAN91C95 SRAM IN BYTES |
|---|---|---|---|
| 000 | Data Byte 0 | Word 1 - Low Byte | Byte 0 |
| 001 | Don't Care | | |
| 002 | Data Byte 1 | Word 1 - High Byte | Byte1 |
| 003 | Don't Care | | |
| 004 | Data Byte 2 | Word 2 - Low Byte | Byte2 |
| 005 | Don't Care | | |
| 006 | Data Byte 3 | Word 2 - High Byte | Byte3 |
| .. | .. | .. | .. |
| 3F8 | Data Byte 1FC | Word FE - Low Byte | Byte FC |
| 3F9 | Don't Care | | |
| 3FA | Data Byte 1FD | Word FE- High Byte | Byte1FD |
| 3FB | Don't Care | | |
| 3FC | Data Byte 1FE | Word FF - Low Byte | Byte1FE |
| 3FD | Don't Care | | |
| 3FE | Data Byte 1FF | Word FF - High Byte | Byte1FF |
| 3FF | Don't Care | | |

Note: This memory map assumes a 4096 bit Serial EEPROM in PCMCIA mode.

**I/O SPACE - BANK2**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| 0 | MMU COMMAND REGISTER | WRITE ONLY<br>BUSY Bit Readable | MMUCR |

This register is used by the CPU to control the memory allocation, de-allocation, TX FIFO and RX FIFO control. The three command bits determine the command issued as described below:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | | | | | | | | |
| | | | | | | | | |
| LOW BYTE | COMMAND | | | 0 | 0 | N2 | N1 | N0/BUSY |
| | x | y | z | | | | | |
| | | | | | | | | 0 |

COMMAND SET:

xyz

000 0) NOOP - NO OPERATION

001 1) ALLOCATE MEMORY FOR TX - N2,N1,N0 defines the amount of memory requested as (value + 1) * 256 bytes. Namely N2,N1,N0 = 1 will request 2 * 256 = 512 bytes. Valid range for N2,N1,N0 is 0 through 5. A shift-based divide by 256 of the packet length yields the appropriate value to be used as N2,N1,N0. Immediately generates a completion code at the ALLOCATION RESULT REGISTER. Can optionally generate an interrupt on successful completion. The allocation time can take worst case (N2, N1, N0 + 2) * 200ns.

010 2) RESET MMU TO INITIAL STATE - Frees all memory allocations, clears relevant interrupts, resets packet FIFO pointers.

011 3) REMOVE FRAME FROM TOP OF RX FIFO - To be issued after CPU has completed processing of present receive frame. This command removes the receive packet number from the RX FIFO and brings the next receive frame (if any) to the RX area (output of RX FIFO).

100 4) REMOVE AND RELEASE TOP OF RX FIFO - Like 3) but also releases all memory used by the packet presently at the RX FIFO output.

101 5) RELEASE SPECIFIC PACKET - Frees all pages allocated to the packet specified in the PACKET NUMBER REGISTER. Should not be used for frames pending transmission. Typically used to remove transmitted frames, after reading their completion status. Can be used following 3) to release receive packet memory in a more flexible way than 4).

110 6) ENQUEUE PACKET NUMBER INTO TX FIFO - This is the normal method of transmitting a packet just loaded into RAM. The packet number to be enqueued is taken from the PACKET NUMBER REGISTER.

111 7) RESET TX FIFOs - This command will reset both TX FIFOs: the TX FIFO holding the packet numbers awaiting transmission and the TX Completion FIFO. This command provides a mechanism for canceling packet transmissions, and reordering or bypassing the transmit queue. The RESET TX FIFOs command should only be used when the transmitter is disabled. Unlike the RESET MMU command, the RESET TX FIFOs does not release any memory.

Note 1: Only command 1) uses N2,N1,N0.

Note 2: When using the RESET TX FIFOS command, the CPU is responsible for releasing the memory associated with outstanding packets, or re-enqueuing them. Packet numbers in the completion FIFO can be read via the FIFO ports register before issuing the command.

Note 3: MMU commands releasing memory (commands 4 and 5) should only be issued if the corresponding packet number has memory allocated to it.

**COMMAND SEQUENCING**

A second allocate command (command 1) should not be issued until the present one has completed. Completion is determined by reading the FAILED bit of the allocation result register or through the allocation interrupt.

A second release command (commands 4, 5) should not be issued if the previous one is still being processed. The BUSY bit indicates that a release command is in progress. After issuing command 5, the contents of the PNR should not be changed until BUSY goes low. After issuing command 4, command 3 should not be issued until BUSY goes low.

BUSY BIT - Readable at bit 0 of the MMU command register address. When set indicates that MMU is still processing a release command. When clear, MMU has already completed last release command. BUSY and FAILED bits are set upon the trailing edge of command.

## I/O SPACE - BANK2

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **2** | **PACKET NUMBER REGISTER** | **READ/WRITE** | **PNR** |

|  |  |  | PACKET NUMBER AT TX AREA |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

PACKET NUMBER AT TX AREA - The value written into this register determines which packet number is accessible through the TX area. Some MMU commands use the number stored in this register as the packet number parameter. This register is cleared by a RESET or a RESET MMU Command.

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **3** | **ALLOCATION RESULT REGISTER** | **READ ONLY** | **ARR** |

This register is updated upon an ALLOCATE MEMORY MMU command.

| FAILED |  |  | ALLOCATED PACKET NUMBER |  |  |  |  |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

FAILED - A zero indicates a successful allocation completion. If the allocation fails the bit is set and only cleared when the pending allocation is satisfied. Defaults high upon reset and reset MMU command. For polling purposes, the ALLOC_INT in the Interrupt Status Register should be used because it is synchronized to the read operation. Sequence:

1) Allocate Command
2) Poll ALLOC_INT bit until set
3) Read Allocation Result Register

ALLOCATED PACKET NUMBER - Packet number associated with the last memory allocation request. The value is only valid if the FAILED bit is clear.

Note: For software compatibility with future versions, the value read from the ARR after an allocation request is intended to be written into the PNR as is, without masking higher bits (provided FAILED = 0).

**I/O SPACE - BANK2**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| 4 | FIFO PORTS REGISTER | READ ONLY | FIFO |

This register provides access to the read ports of the Receive FIFO and the Transmit completion FIFO. The packet numbers to be processed by the interrupt service routines are read from this register.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | REMPTY | | | RX FIFO PACKET NUMBER | | | | |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW BYTE | TEMPTY | | | TX DONE PACKET NUMBER | | | | |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

REMPTY - No receive packets queued in the RX FIFO. For polling purposes, uses the RCV_INT bit in the Interrupt Status Register.

TOP OF RX FIFO PACKET NUMBER - Packet number presently at the output of the RX FIFO. Only valid if REMPTY is clear. The packet is removed from the RX FIFO using MMU Commands 3) or 4).

TEMPTY - No transmit packets in completion queue. For polling purposes, uses the TX_INT bit in the Interrupt Status Register.

TX DONE PACKET NUMBER - Packet number presently at the output of the TX Completion FIFO. Only valid if TEMPTY is clear. The packet is removed when a TX INT acknowledge is issued.

NOTE: For software compatibility with future versions, the value read from each FIFO register is intended to be written into the PNR as is, without masking higher bits (provided TEMPTY and REMPTY = 0 respectively).

**I/O SPACE - BANK2**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| 6 | POINTER REGISTER | READ/WRITE | PTR |

<table>
<tr><td>HIGH<br>BYTE</td><td>RCV</td><td>AUTO<br>INCR.</td><td>READ</td><td>ETEN</td><td>0</td><td colspan="3">POINTER HIGH</td></tr>
<tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>LOW<br>BYTE</td><td colspan="8">POINTER LOW</td></tr>
<tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

POINTER REGISTER - The value of this register determines the address to be accessed within the transmit or receive areas. It will auto-increment on accesses to the data register when AUTO INCR. is set. The increment is by one for every byte access, and by two for every word access.

RCV - When RCV is set the address refers to the receive area and uses the output of RX FIFO as the packet number, when RCV is clear the address refers to the transmit area and uses the packet number at the Packet Number Register.

READ - Determines the type of access to follow. If the READ bit is high the operation intended is a read. If the READ bit is low the operation is a write. Loading a new pointer value, with the READ bit high, generates a pre-fetch into the Data Register for read purposes.

Readback of the pointer will indicate the value of the address last accessed by the CPU (rather than the last pre-fetched). This allows any interrupt routine that uses the pointer, to save it and restore it without affecting the process being interrupted.

The Pointer Register should not be loaded until 400ns after the last write operation to the Data Register to ensure that the Data Register FIFO is empty. If the pointer is loaded using 8 bit writes, the low byte should be loaded first and the high byte last.

ETEN bit - When set enables EARLY Transmit underrun detection. Normal operation when clear. For Underrun detection purposes the RAM logical address and packet numbers of the packet being loaded are compared against the logical address and packet numbers of the packet being transmitted. If the packet numbers match and the logical address of the packet being transmitted exceeds the address being loaded the packet transmission is aborted and Underrun is reported in the transmit status word.

NOTE: If AUTO INCR. is not set and 16 bits are used, the pointer must be loaded with an even value. If AUTO INCR. is not set and 8 bit (allows even and odd values) writes are used, consecutive writes to the data register for the same pointer (same memory address) value is not allowed; even and odd pointer values are always allowed.

**I/O SPACE - BANK2**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **8 THROUGH Ah** | **DATA REGISTER** | **READ/WRITE** | **DATA** |

| |
|---|
| DATA HIGH |
| |
| DATA LOW |
| |

DATA REGISTER - Used to read or write the data buffer byte/word presently addressed by the pointer register.

This register is mapped into two uni-directional FIFOs that allow moving words to and from the LAN91C95 regardless of whether the pointer address is even or odd. Data goes through the write FIFO into memory, and is pre-fetched from memory into the read FIFO. If byte accesses are used, the appropriate (next) byte can be accessed through the Data Low or Data High registers. The order to and from the FIFO is preserved. Byte and word accesses can be mixed on the fly in any order.

This register is mapped into two consecutive word locations to facilitate the usage of double word move instructions. The DATA register is accessible at any address in the 8 through Ah range, while the number of bytes being transferred are determined by A0 and nSBHE in ISA mode, and by A0, nCE1 and nCE2 in PCMCIA mode.

**I/O SPACE - BANK2**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **C** | **INTERRUPT STATUS REGISTER** | **READ ONLY** | **IST** |

| Res. | ERCV INT | EPH INT | RX_OVRN INT | ALLOC INT | TX EMPTY INT | TX INT | RCV INT |
|---|---|---|---|---|---|---|---|
| X | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **C** | **INTERRUPT ACKNOWLEDGE REGISTER** | **WRITE ONLY** | **ACK** |

| Res. | ERCV INT | Res. | RX_OVRN INT | Res. | TX EMPTY INT | TX INT | Res. |
|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **D** | **INTERRUPT MASK REGISTER** | **READ/WRITE** | **MSK** |

| Res. | ERCV INT | EPH INT | RX_OVRN INT | ALLOC INT | TX EMPTY INT | TX INT | RCV INT |
|---|---|---|---|---|---|---|---|
| X | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register can be read and written as a word or as two individual bytes.

The Interrupt Mask Register bits enable the appropriate bits when high and disable them when low. An enabled bit being set will cause a hardware interrupt.

EPH INT - Set when the Ethernet Protocol Handler section indicates one out of various possible special conditions. This bit merges exception type of interrupt sources, whose service time is not critical to the execution speed of the low level drivers. The exact nature of the interrupt can be obtained from the EPH Status Register (EPHSR), and enabling of these sources can be done via the Control Register. The possible sources are:

LINK_OK transition
CTR_ROL - Statistics counter roll over.
TXENA cleared - A fatal transmit error occurred forcing TXENA to be cleared. TX_SUC will be low and the specific reason will be reflected by the bits:

- TXUNRN - Transmit underrun
- SQET - SQE Error
- LOST CARR - Lost Carrier
- LATCOL - Late Collision
- 16COL - 16 collisions
- WAKE_UP - “Magic Packet” is received if enabled

RX_OVRN INT - Set when the receiver overruns due to a failed memory allocation. The RX_OVRN bit of the EPHSR will also be set, but if a new packet is received it will be cleared. The RX_OVRN INT bit, however, latches the overrun condition for the purpose of being polled or generating an interrupt, and will only be cleared by writing the acknowledge register with the RX_OVRN INT bit set.

ALLOC INT - Set when an MMU request for TX pages allocation is completed. This bit is the complement of the FAILED bit in the ALLOCATION RESULT register. The ALLOC INT ENABLE bit should only be set following an allocation command, and cleared upon servicing the interrupt.

TX EMPTY INT - Set if the TX FIFO goes empty, can be used to generate a single interrupt at the end of a sequence of packets enqueued for transmission. This bit latches the empty condition, and the bit will stay set until it is specifically cleared by writing the acknowledge register with the TX EMPTY INT bit set. If a real time reading of the FIFO empty is desired, the bit should be first cleared and then read. The TX EMPTY INT ENABLE should only be set after the following steps:

a) a packet is enqueued for transmission
b) the previous empty condition is cleared (acknowledged)

TX INT - Set when at least one packet transmission was completed. The first packet number to be serviced can be read from the FIFO PORTS register. The TX INT bit is always the logic complement of the TEMPTY bit in the FIFO PORTS register. After servicing a packet number, its TX INT interrupt is removed by writing the Interrupt Acknowledge Register with the TX INT bit set.

RCV INT - Set when a receive interrupt is generated. The first packet number to be serviced can be read from the FIFO PORTS register. The RCV INT bit is always the logic complement of the REMPTY bit in the FIFO PORTS register.

ERCV INT - Early receive interrupt. Set whenever a receive packet is being received, and the number of bytes received into memory exceeds the value programmed as ERCV THRESHOLD (Bank 3, Offset Ch). ERCV INT stays set until acknowledged by writing the INTERRUPT ACKNOWLEDGE REGISTER with the ERCV INT bit set.

NOTE: If the driver uses AUTO RELEASE mode it should enable TX EMPTY INT as well as TX INT. TX EMPTY INT will be set when the complete sequence of packets is transmitted. TX INT will be set if the sequence stops due to a fatal error on any of the packets in the sequence.

NOTE: For edge triggered systems, the Interrupt Service Routine should clear the Interrupt Mask Register, and only enable the appropriate interrupts after the interrupt source is serviced (acknowledged).

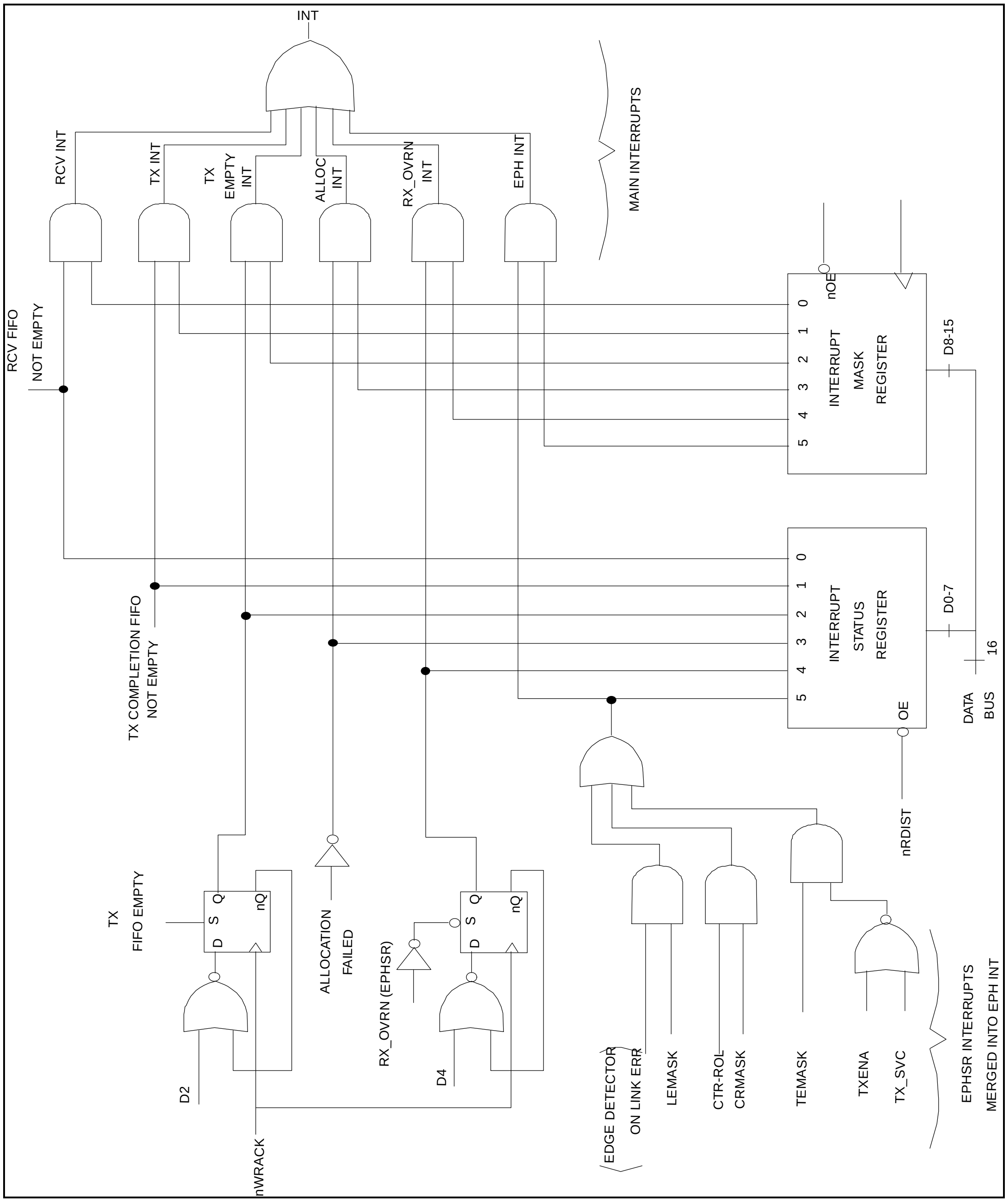


**FIGURE 9 - INTERRUPT STRUCTURE**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **0 THROUGH 7** | **MULTICAST TABLE** | **READ/WRITE** | **MT** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| LOW BYTE | MULTICAST TABLE 0 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HIGH BYTE | MULTICAST TABLE 1 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW BYTE | MULTICAST TABLE 2 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HIGH BYTE | MULTICAST TABLE 3 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW BYTE | MULTICAST TABLE 4 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HIGH BYTE | MULTICAST TABLE 5 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LOW BYTE | MULTICAST TABLE 6 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HIGH BYTE | MULTICAST TABLE 7 | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The 64 bit multicast table is used for group address filtering. The hash value is defined as the six most significant bits of the CRC of the destination addresses. The three msb's determine the register to be used (MT0-MT7), while the other three determine the bit within the register. If the appropriate bit in the table is set, the packet is received. If the ALMUL bit in the RCR register is set, all multicast addresses are received regardless of the multicast table values.

Hashing is only a partial group addressing filtering scheme, but being the hash value available as part of the receive status word, the receive routine can reduce the search time significantly. With the proper memory structure, the search is limited to comparing only the multicast addresses that have the actual hash value in question.

### I/O SPACE - BANK3

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **8** | **MANAGEMENT INTERFACE** | **READ/WRITE** | **MGMT** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | | | | | nXNDEC | IOS2 | IOS1 | IOS0 |
| | 0 | 0 | 1 | 1 | | | | |
| LOW BYTE | | | | | MDOE | MCLK | MDI | MDO |
| | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

nXNDEC - Read only bit reflecting the status of the nXENDEC pin.

IOS0-IOS2 - Read only bits reflecting the status of the IOS0-IOS2 pins.

MDO - The value of this bit drives the EEDO pin when MDOE=1.

MDI - Reads the value of the EEDI pin.

MDCLK - The value of this bit drives the EESK pin when MDOE=1.

MDOE - When this bit is high pins EEDO EECS and EESK will be used for transceiver management functions, otherwise the pins assume the EEPROM values.

| | **MDOE = 0** | **MDOE = 1** |
|---|---|---|
| EEDO | Serial EEPROM Data Out | Bit MDO |
| EESK | Serial EEPROM Clock | Bit MCLK |
| EECS | Serial EEPROM Chip Select | 0 |

**I/O SPACE - BANK3**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| A | REVISION REGISTER | READ ONLY | REV |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | | | | | | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| LOW BYTE | CHIP | | | | REV | | | |
| | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

CHIP - Chip ID. Can be used by software drivers to identify the device used.

REV - Revision ID. Incremented for each revision of a given device.

| CHIP ID VALUE | DEVICE |
|---|---|
| 3 | LAN91C90/LAN91C92 |
| 4 | LAN91C94 |
| 5 | LAN91C95 |
| 7 | FEAST |

**I/O SPACE - BANK 3**

| OFFSET | NAME | TYPE | SYMBOL |
|---|---|---|---|
| **C** | **EARLY RCV REGISTER** | **READ/WRITE** | **ERCV** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HIGH BYTE | | | | | | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| LOW BYTE | RCV DISCRD | | | ERCV THRESHOLD | | | | |
| | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

RCV DISCRD - Set to discard a packet being received. This bit can be used in conjunction with ERCV THRESHOLD and ERCV INT to process a packet header while it is being received and sicard it if the packet is not desired. Setting this bit will only discard packets that are still in the process of being received.

If the RCV DISCRD bit is set prior to the end of a receive packet, RXOVRN bit in the Interrupt Status Register will be set to indicate that the packet was discarded and its memory released. If the receive packet is complete prior to the RCV DISCRD bit being set, the packet is received normally and RCV INT bit is set in the Interrupt Status Register. The RCV DISCRD bit is self-clearing.

ERCV THRESHOLD - Threshold for ERCV interrupt. Specified in 64 byte multiples. Whenever the number of bytes written in memory for the presently received packet exceeds the ERCV THRESHOLD, ERCV INT bit of the INTERRUPT STATUS REGISTER is set.

Refer to the PCMCIA Configuration Registers Description for Banks 4 and 5.

PC Card 5.0 treats the individual functions of a multifunction PCMCIA application independently. Card and Socket Services (C&SS) 5.0 is designed to provide the support for determining the function generating the interrupt and informing relevant drivers. The registers for the two functions are treated as independent sets. One of the only requirements is to set the functions' I/O base registers with different values to avoid any access conflict.

### MEMORY ARCHITECTURE

The concept of presenting the shared RAM as FIFO of packets, with a memory management unit allocating memory on a per packet basis responds to the following needs:

Memory allocation for receive vs. transmit - A fixed partition between receive and transmit area would not be efficient. Being able to dynamically allocate it to transmit and receive represents almost the equivalent of duplicating the memory size for some workstation type of drivers.

Software overhead - By presenting a FIFO of packets, the software driver does not have to waste any time in calculating pointers for the different buffers that make up different packets. The driver usually deals with one packet at a time. With this approach, packets are always accessible at the same fixed address, and access is provided to any byte of the packet.

Headers can be analyzed without reading the entire packet. The packet can be read or written with a block move operation.

Multiple upper layer support - The LAN91C95 facilitates interfacing to multiple upper layer protocols because of the receive packet processing flexibility. A receive lookahead scheme like ODI or NDIS drivers are supported by copying a small part of the received packet and letting the upper layer provide a pointer for the rest of the data. If the upper layer indicates it does not want a specific part of the packet, a block move operation starting at any particular offset can be done. Out of order receive processing is also supported: if memory for one packet is not yet available, receive packet processing can continue.

Efficiency - Lacking any level of indirection or linked lists of pointers, virtually all the memory is used for data. There are no descriptors, forward links and pointers at all. This simplicity and memory efficiency is accomplished without giving up the benefits of linked lists which is unlimited back-to-back transmission and reception without CPU intervention for as long as memory is available.

### FULL DUPLEX ETHERNET SUPPORT

Full Duplex Ethernet operation refers to the ability of the network (or parts of it) to simultaneously transmit and receive packets. The CSMA/CD protocol used by Ethernet for accessing a shared medium is inherently half duplex , and so is the 10BASE-T physical layer where simultaneous transmit and receive activity is interpreted as a collision.

The LAN91C95 supports two types of Full Duplex operation:

1. Full Duplex mode for diagnostic purposes only, where the received packet is the transmit packet being looped back. This mode is enabled using the FDUPLX bit in the TCR. In this mode the CSMA/CD algorithm is used to gain access to the media.

2. FDSE (Full Duplex Switched Ethernet). Enabled by FDSE bit in TCR bit. When the LAN91C95 is configured for FDSE, its transmit and receive paths will operate independently and some CSMA/CD functions are disabled such as Carrier Sense.

### Behavior in FDSE Mode

The main 802.3 section affected by FDSE is 4.2.8 where the Frame Transmission procedural model is presented. The changes are:

1. No deferral - The transmit channel is dedicated and always available - The device will transmit whenever it has a packet ready to be sent, while respecting the interframe spacing between transmit packets.

2. No collision detection - There are no collisions in a switched environment.

   The EPH implementation of the procedural model uses as 'collisionDetect' the MAC collision input, sourced from the 10BASE-T front end, AUI front end, or External Endec interface. This collision input is observed by the Transmit State Machine while 'transmitting' is true, that is during Preamble, Data, Pad, and CRC states. If collision is active during any of these states the state machine transitions to JAM and BACKOFF states.

3. 10BASE-T loopback - Typically 10BASE-T drivers are internally looped back to the differential receivers.

### Magic Packet Support

If the WAKEUP_EN bit in the Control Register (Bank1, Offset C) is set, the controller will come out of any powerdown mode. If this bit is not set, this functionality is disabled.

When a Magic Packet is received, the ethernet controller will generate an interrupt causing the host to initiate a service routine to find the source of the event. The Interrupt bit in the ECSR is also set if the host plans on polling the controller for Wakeup status.

## TYPICAL FLOW OF EVENTS FOR TRANSMIT

| | S/W DRIVER | CSMA/CD SIDE |
|---|---|---|
| 1 | ISSUE ALLOCATE MEMORY FOR TX - N BYTES - the MMU attempts to allocate N bytes of RAM. | |
| 2 | WAIT FOR SUCCESSFUL COMPLETION CODE - Poll until the ALLOC INT bit is set or enable its mask bit and wait for the interrupt. The TX packet number is now at the Allocation Result Register. | |
| 3 | LOAD TRANSMIT DATA - Copy the TX packet number into the Packet Number Register. Write the Pointer Register, then use a block move operation from the upper layer transmit queue into the Data Register. | |
| 4 | ISSUE "ENQUEUE PACKET NUMBER TO TX FIFO" - This command writes the number present in the Packet Number Register into the TX FIFO. The transmission is now enqueued. No further CPU intervention is needed until a transmit interrupt is generated. | |
| 5 | | The enqueued packet will be transferred to the CSMA/CD block as a function of TXENA (n TCR) bit and of the deferral process (in half duplex mode) state. |
| 6 | | Upon transmit completion the first word in memory is written with the status word. The packet number is moved from the TX FIFO into the TX completion FIFO. Interrupt is generated by the TX completion FIFO being not empty. |
| 7 | SERVICE INTERRUPT - Read Interrupt Status Register. If it is a transmit interrupt, read the TX Done Packet Number from the FIFO Ports Register. Write the packet number into the Packet Number Register. The corresponding status word is now readable from memory. If the status word shows successful transmission, issue RELEASE packet number command to free up the memory used by this packet. Remove packet number from completion FIFO by writing TX INT Acknowledge Register. | |

## TYPICAL FLOW OF EVENTS FOR RECEIVE

| | S/W DRIVER | CSMA/CD SIDE |
|---|---|---|
| 1 | ENABLE RECEPTION - By setting the RXEN bit. | |
| 2 | | A packet is received with matching address. Memory is requested from MMU. A packet number is assigned to it. Additional memory is requested if more pages are needed. |
| 3 | | The internal DMA logic generates sequential addresses and writes the receive words into memory. The MMU does the sequential to physical address translation. If overrun, packet is dropped and memory is released. |
| 4 | | When the end of packet is detected, the status word is placed at the beginning of the receive packet in memory. The Byte count is placed at the second word. If the CRC is correct, the packet number is written into the RX FIFO. The RX FIFO being not empty causes RCV INT (interrupt) to be set. If CRC is incorrect the packet memory is released and no interrupt will occur. |
| 5 | SERVICE INTERRUPT - Read the Interrupt Status Register and determine if RCV INT is set. The next receive packet is at the receive area. (Its packet number can be read from the FIFO Ports Register). The software driver can process the packet by accessing the RX area, and can move it out to system memory if desired. When processing is complete the CPU issues the REMOVE AND RELEASE FROM TOP OF RX command to have the MMU free up the used memory and packet number. | |

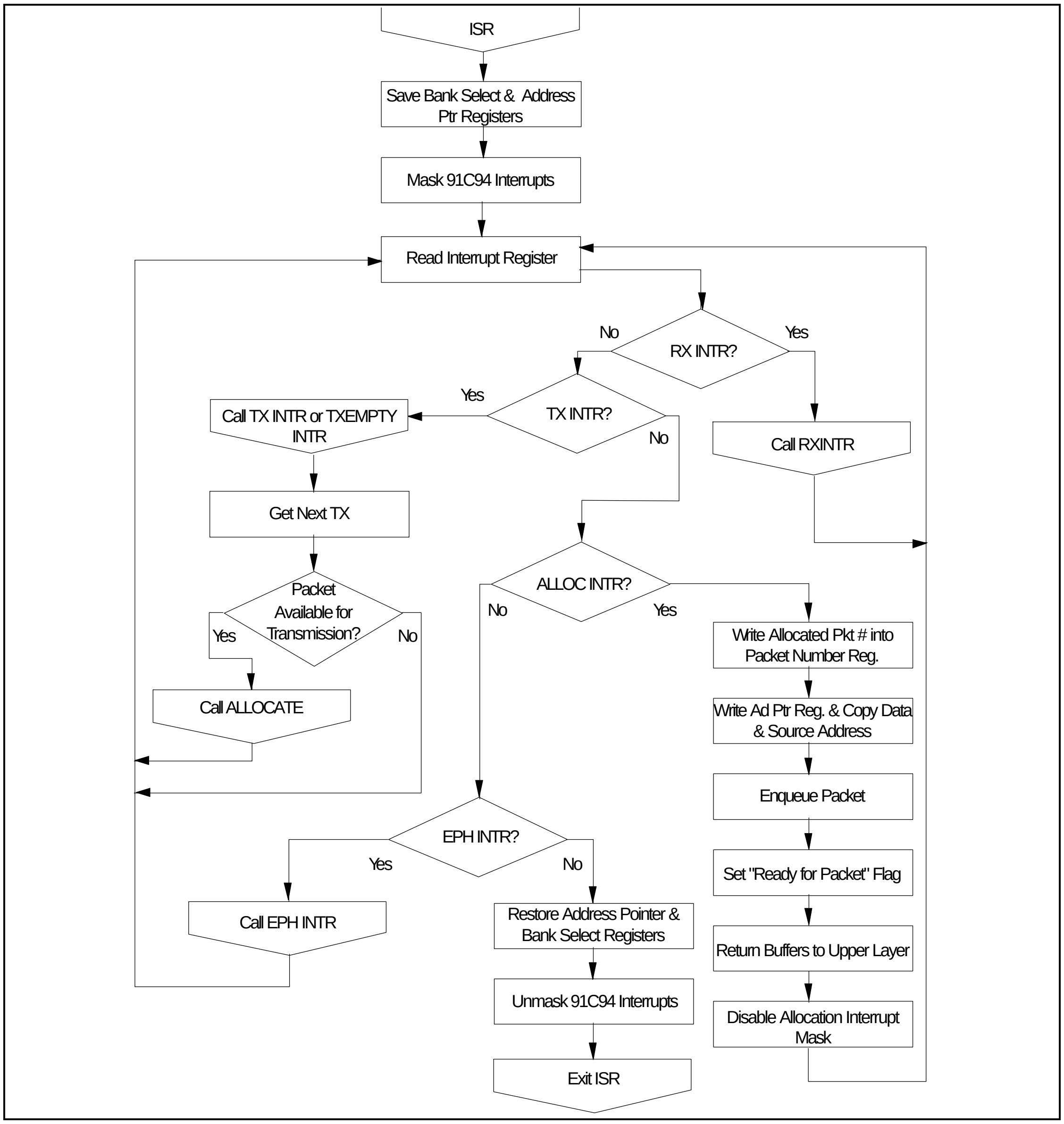

ISR
Save Bank Select & Address Ptr Registers
Mask 91C94 Interrupts
Read Interrupt Register
RX INTR?
No
Yes
TX INTR?
Yes
No
Call TX INTR or TXEMPTY INTR
Call RXINTR
Get Next TX
Packet Available for Transmission?
Yes
No
ALLOC INTR?
No
Yes
Call ALLOCATE
Write Allocated Pkt # into Packet Number Reg.
Write Ad Ptr Reg. & Copy Data & Source Address
Enqueue Packet
EPH INTR?
Yes
No
Set "Ready for Packet" Flag
Call EPH INTR
Restore Address Pointer & Bank Select Registers
Return Buffers to Upper Layer
Unmask 91C94 Interrupts
Disable Allocation Interrupt Mask
Exit ISR

**FIGURE 10 - ETHERNET INTERRUPT SERVICE ROUTINE**

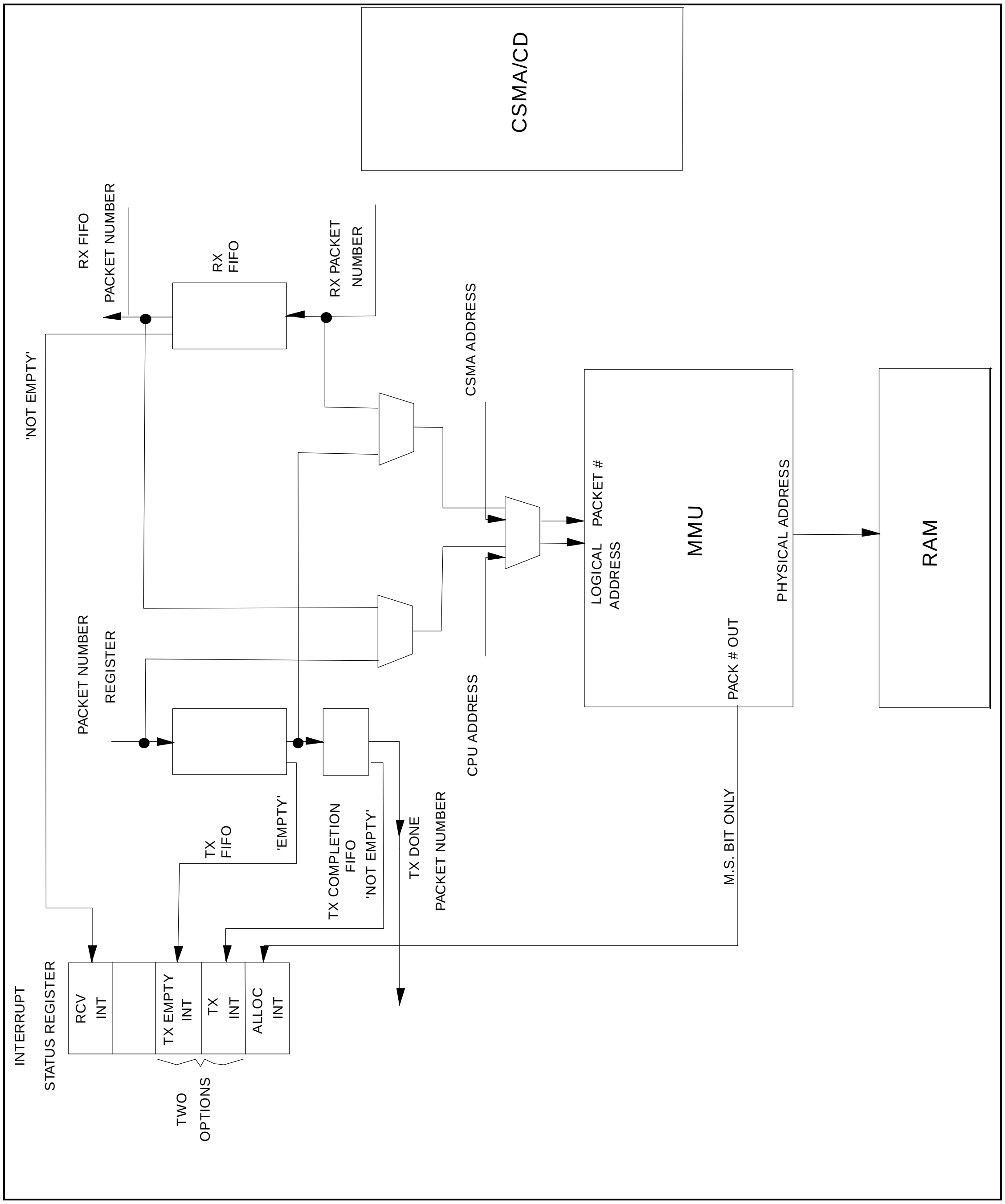


**FIGURE 11 - INTERRUPT GENERATION FOR TRANSMIT, RECEIVE, MMU**

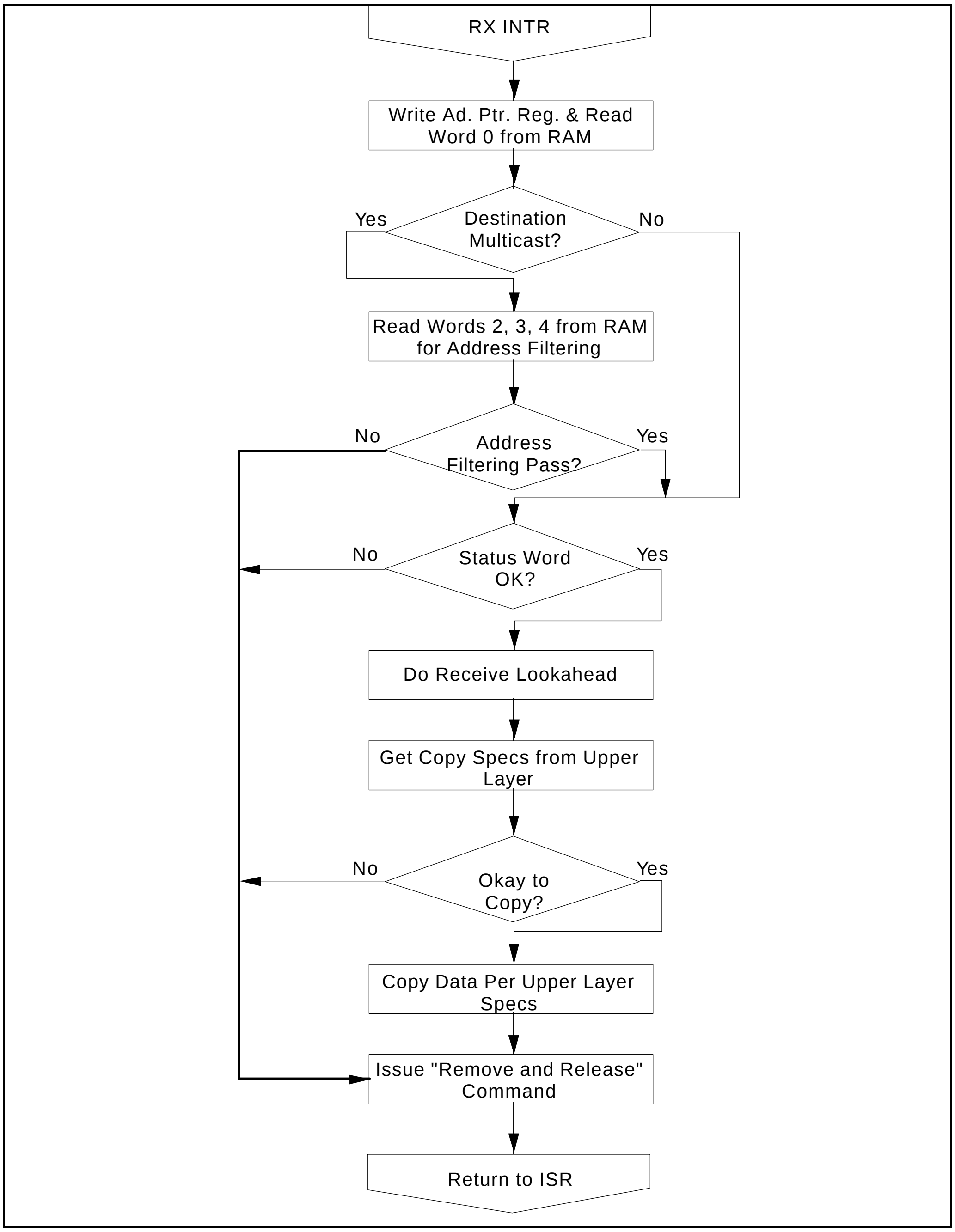

RX INTR
Write Ad. Ptr. Reg. & Read Word 0 from RAM
Destination Multicast?
Yes
No
Read Words 2, 3, 4 from RAM for Address Filtering
Address Filtering Pass?
No
Yes
Status Word OK?
No
Yes
Do Receive Lookahead
Get Copy Specs from Upper Layer
Okay to Copy?
No
Yes
Copy Data Per Upper Layer Specs
Issue "Remove and Release" Command
Return to ISR

**FIGURE 12 - RX INTR**

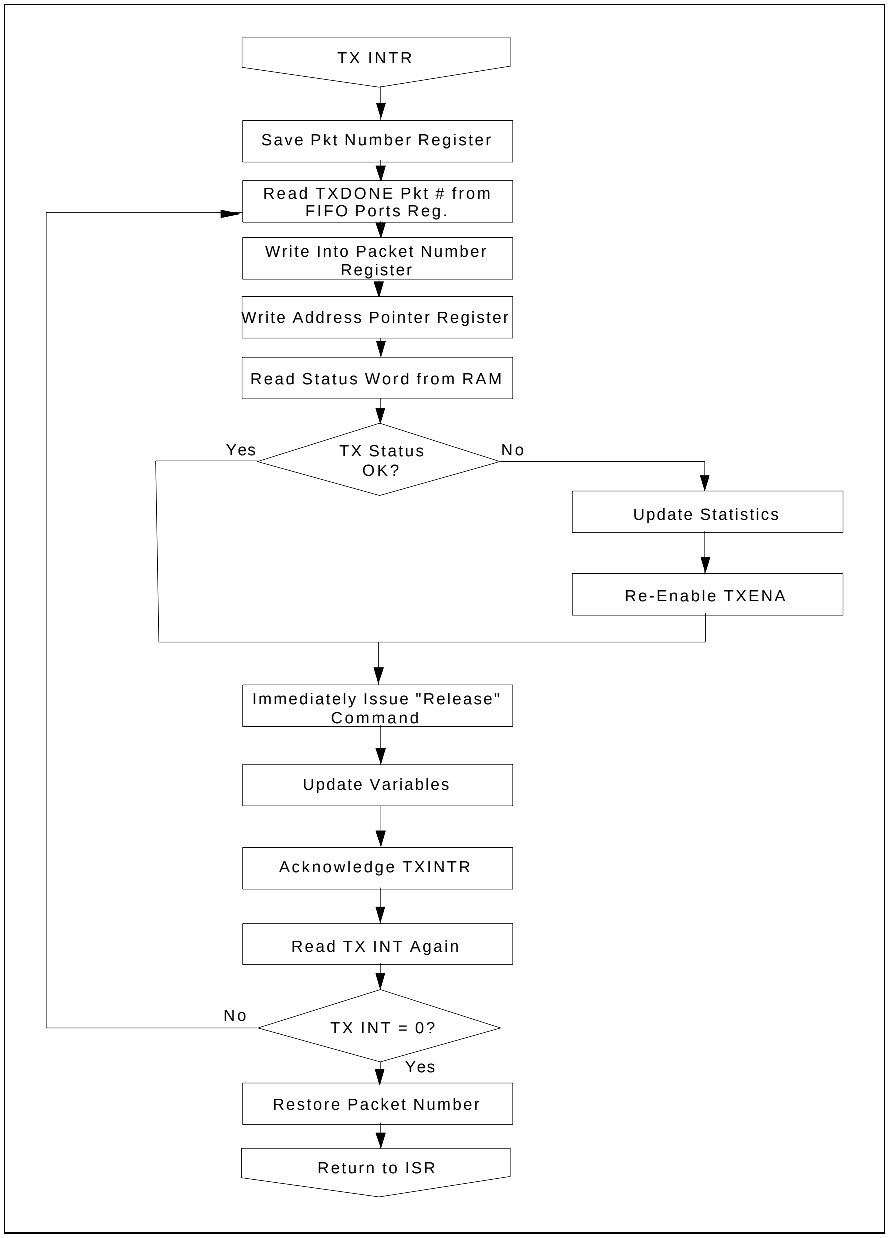


**FIGURE 13 - TX INTR**
**(Assumes Auto Release Option Selected)**

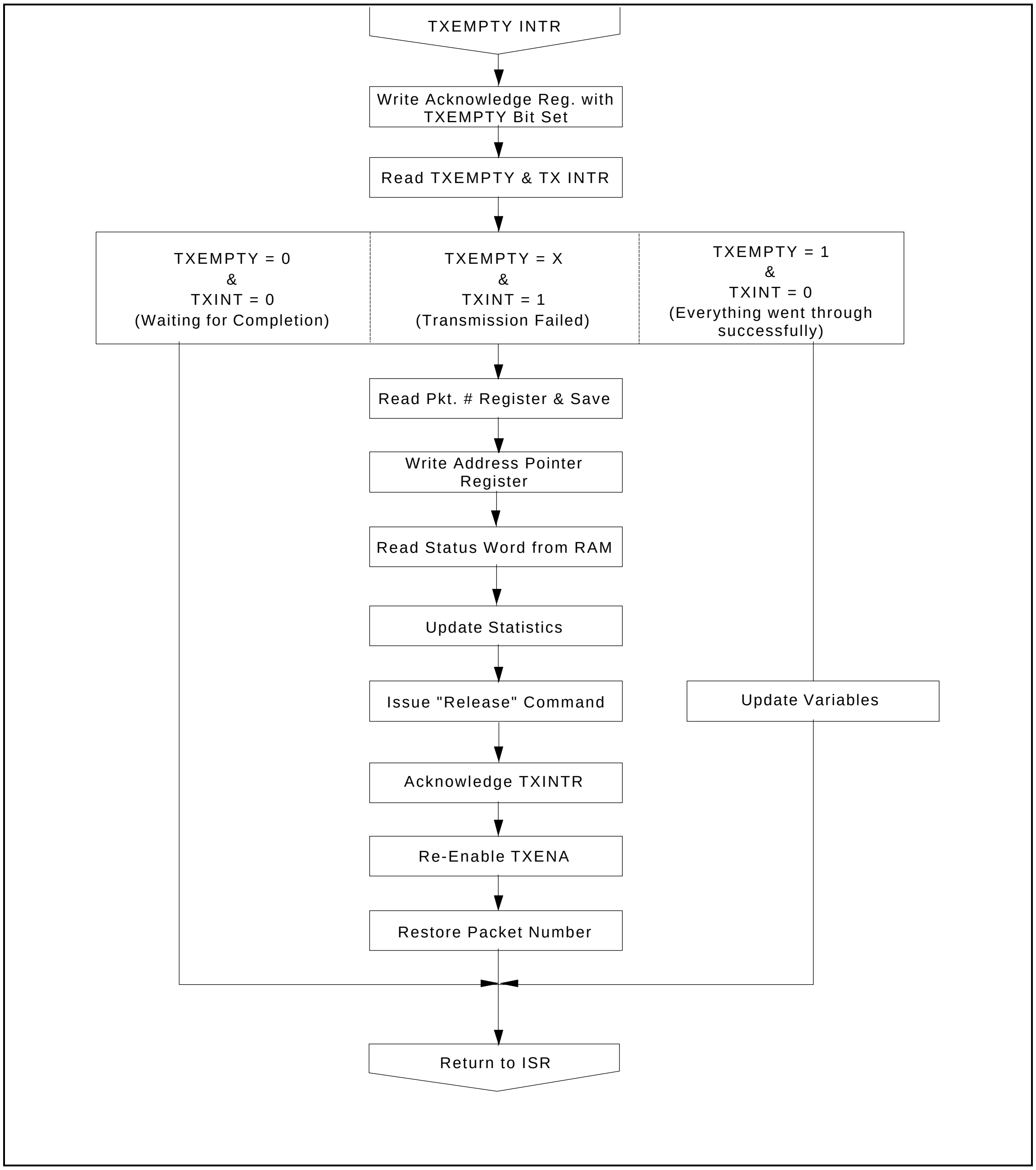
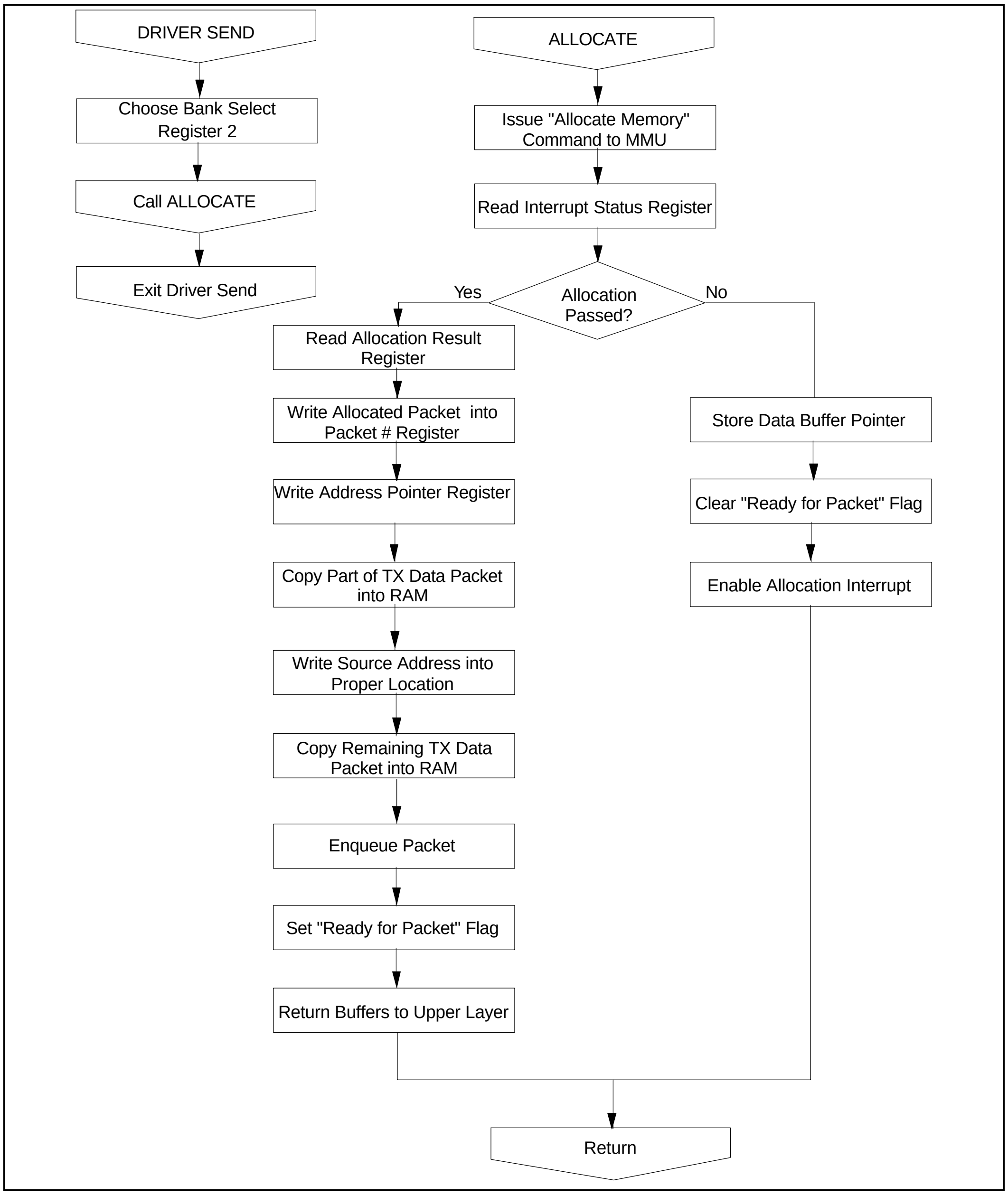

TXEMPTY INTR
Write Acknowledge Reg. with TXEMPTY Bit Set
Read TXEMPTY & TX INTR
TXEMPTY = 0 & TXINT = 0 (Waiting for Completion)
TXEMPTY = X & TXINT = 1 (Transmission Failed)
TXEMPTY = 1 & TXINT = 0 (Everything went through successfully)
Read Pkt. # Register & Save
Write Address Pointer Register
Read Status Word from RAM
Update Statistics
Issue "Release" Command
Update Variables
Acknowledge TXINTR
Re-Enable TXENA
Restore Packet Number
Return to ISR

**FIGURE 14 - TXEMPTY INTR**



DRIVER SEND
Choose Bank Select Register 2
Call ALLOCATE
Exit Driver Send
ALLOCATE
Issue "Allocate Memory" Command to MMU
Read Interrupt Status Register
Allocation Passed?
Yes
No
Read Allocation Result Register
Write Allocated Packet into Packet # Register
Write Address Pointer Register
Copy Part of TX Data Packet into RAM
Write Source Address into Proper Location
Copy Remaining TX Data Packet into RAM
Enqueue Packet
Set "Ready for Packet" Flag
Return Buffers to Upper Layer
Store Data Buffer Pointer
Clear "Ready for Packet" Flag
Enable Allocation Interrupt
Return

**FIGURE 15 - DRIVER SEND AND ALLOCATE ROUTINES**

## MEMORY PARTITIONING

Unlike other controllers, the LAN91C95 does not require a fixed memory partitioning between transmit and receive resources. The MMU allocates and de-allocates memory upon different events. An additional mechanism allows the CPU to prevent the receive process from starving the transmit memory allocation.

Memory is always requested by the side that needs to write into it, that is: the CPU for transmit or the CSMA/CD for receive. The CPU can control the number of bytes it requests for transmit but it cannot determine the number of bytes the receive process is going to demand. Furthermore, the receive process requests will be dependent on network traffic, in particular on the arrival of broadcast and multicast packets that might not be for the node, and that are not subject to upper layer software flow control.

In order to prevent unwanted traffic from using too much memory, the CPU can program a "memory reserved for transmit" parameter. If the free memory falls below the "memory reserved for transmit" value, MMU requests from the CSMA/CD block will fail and the packets will overrun and be ignored. Whenever enough memory is released, packets can be received again. If the reserved value is too large, the node might lose data which is an abnormal condition. If the value is kept at zero, memory allocation is handled on first-come first-served basis for the entire memory capacity.

Note that with the memory management built into the LAN91C95, the CPU can dynamically program this parameter. For instance, when the driver does not need to enqueue transmissions, it can allow more memory to be allocated for receive (by reducing the value of the reserved memory). Whenever the driver needs to burst transmissions it can reduce the receive memory allocation. The driver program the parameter as a function of the following variables:

1) Free memory (read only register)
2) Memory size (read only register)

The reserved memory value can be changed on the fly. If the MEMORY RESERVED FOR TX value is increased above the FREE MEMORY, receive packets in progress are still received, but no new packets are accepted until the FREE MEMORY increases above the MEMORY RESERVED value.

## INTERNAL VS. EXTERNAL ATTRIBUTE MEMORY MAP

The Internal vs. External EPROM attribute memory decodes are shown in Table 13 and Table 14. This allows the designer to not require an external EPROM device if the single or multi-function PCMCIA card needs less than 512 bytes of configuration information. As can be seen in the map, if 512 bytes of CIS or less is required, the nFCS and nFWE output pins of the LAN91C95 need not be used (if serial EEPROM is being used). Internal to the LAN91C95, the memory addressing logic will allow byte or word access on even byte boundaries. This implies that on odd byte address access (A0=1), the LAN91C95 will generate a arbitrary value of Zero (0) since the PCMCIA specification states that the high byte of a word access in attribute memory is a don’t care. This allows backward compatibility to 8 bit hosts.

TABLE 13 - ATTRIBUTE MEMORY DECODES USING SERIAL EPROM

| ATTRIBUTE MEMORYADDRESS | EXTERNAL EPROM STORE | INTERNAL SRAM STORE (512 BYTES) | CONFIGURATION REGISTERS |
|---|---|---|---|
| 0 - 3FEh | | X | |
| 400h-7FFEh | X | | |
| 8000h - 803Eh | | | X |

TABLE 14 - ATTRIBUTE MEMORY DECODES WITHOUT SERIAL EPROM

| ATTRIBUTE MEMORYADDRESS | EXTERNAL EPROM STORE | INTERNAL SRAM STORE (512 BYTES) | CONFIGURATION REGISTERS |
|---|---|---|---|
| 0 - 7FFEh | X | | |
| 8000h - 803Eh | | | X |

**INTERRUPT GENERATION**

The interrupt strategy for the transmit and receive processes is such that it does not represent the bottleneck in the transmit and receive queue management between the software driver and the controller. For that purpose there is no register reading necessary before the next element in the queue (namely transmit or receive packet) can be handled by the controller. The transmit and receive results are placed in memory.

The receive interrupt will be generated when the receive queue (FIFO of packets) is not empty and receive interrupts are enabled. This allows the interrupt service routine to process many receive packets without exiting, or one at a time if the ISR just returns after processing and removing one.

There are two types of transmit interrupt strategies:

1) One interrupt per packet
2) One interrupt per sequence of packets

The strategy is determined by how the transmit interrupt bits and the AUTO RELEASE bit are used.

TX INT bit - Set whenever the TX completion FIFO is not empty.

TX EMPTY INT bit - Set whenever the TX FIFO is empty.

AUTO RELEASE - When set, successful transmit packets are not written into completion FIFO, and their memory is released automatically.

1) One interrupt per packet: enable TX INT, set AUTO RELEASE=0. The software driver can find the completion result in memory and process the interrupt one packet at a time. Depending on the completion code the driver will take different actions. Note that the transmit process is working in parallel and other transmissions might be taking place. The LAN91C95 is virtually queuing the packet numbers and their status words.

In this case, the transmit interrupt service routine can find the next packet number to be serviced by reading the TX DONE PACKET NUMBER at the FIFO PORTS register. This eliminates the need for the driver to keep a list of packet numbers being transmitted. The numbers are queued by the LAN91C95 and provided back to the CPU as their transmission completes.

2) One interrupt per sequence of packets: Enable TX EMPTY INT and TX INT, set AUTO RELEASE=1. TX EMPTY INT is generated only after transmitting the last packet in the FIFO.

TX INT will be set on a fatal transmit error allowing the CPU to know that the transmit process has stopped and therefore the FIFO will not be emptied.

This mode has the advantage of a smaller CPU overhead, and faster memory de-allocation. Note that when AUTO RELEASE=1 the CPU is not provided with the packet numbers that completed successfully.

Note: The pointer register is shared by any process accessing the LAN91C95 memory. In order to allow processes to be interruptable, the interrupting process is responsible for reading the pointer value before modifying it, saving it, and restoring it before returning from the interrupt.

Typically there would be three processes using the pointer:

1) Transmit loading (sometimes interrupt driven)
2) Receive unloading (interrupt driven)
3) Transmit Status reading (interrupt driven)

1) and 3) also share the usage of the Packet Number Register. Therefore saving and restoring the PNR is also required from interrupt service routines.

# FUNCTIONAL DESCRIPTION OF THE BLOCKS

## MEMORY MANAGEMENT UNIT

The MMU interfaces the on-chip RAM on one side and the arbiter on the other for address and data flow purposes. For allocation and de-allocation, it interfaces the arbiter only.

The MMU deals with a single ported memory and is not aware of the fact that there are two entities requesting allocation and actually accessing memory. The mapping function done by the MMU is only a function of the packet number accessed and of the offset within that packet being accessed. It is not a function of who is requesting the access or the direction of the access.

To accomplish that, memory accesses as well as MMU allocation and de-allocation requests are arbitrated by the arbiter block before reaching the MMU.

Memory allocation could take some time, but the ALLOC INT bit in Interrupt Status Register is negated immediately upon allocation request, allowing the system to poll that register at any time. Memory de-allocation command completion indication is provided via the BUSY bit, readable through the MMU command register.

The mapping and queuing functions of the MMU rely on the uniqueness of the packet number assigned to the requester. For that purpose the packet number assignment is centralized at the MMU, and a number will not be reused until the memory associated with it is released. It is clear that a packet number should not be released while the number is in the TX or RX packet queue.

The TX and RCV FIFOs are deep enough to handle the total number of packets the MMU can allocate, therefore there is no need for the programmer or the hardware to check FIFO full conditions.

## ARBITER

The function of the arbiter is to sequence packet RAM accesses as well as MMU requests in such a way that the on-chip single ported RAM and a single MMU can be shared by two parties. One party is the host CPU and the other party is the CSMA/CD block.

The arbiter is address transparent, namely, any address can be accessed at any time. In order to exploit the sequential nature of the access, and minimize the access time on the system side, the CPU cycle is buffered by the Data Register rather than go directly to and from memory. Whenever a write cycle is performed, the data is written into the Data Register and will be written into memory as a result of that operation, allowing the CPU cycle to complete before the arbitration and memory cycle are complete. Whenever a read cycle is performed, the data is provided immediately from the Data Register, without having to arbitrate and complete a memory cycle. The present cycle results in an arbitration request for the next data location. Loading the pointer causes a similar pre-fetch request.

This type of read-ahead and write-behind arbitration allows the controller to have a very fast access time, and would work without wait states for as long as the cycle time spec. is satisfied. The values are 40 ns access time, and 185ns cycle time.

By the same token, CSMA/CD cycles might be postponed. The worst case CSMA/CD latency for arbiter service is one memory cycle.

The arbiter uses the pointer register as the CPU provided address, and the internal DMA address from the CSMA/CD side as the addresses to be provided to the MMU.

The data path routed by the arbiter goes between memory (the data path does not go through the MMU) on one side and either the CPU side bus or the data path of the CSMA/CD core.

The data path between memory and the Data Register is in fact buffered by a small FIFO in each direction. The FIFOs beneath the Data Register can be read and written as bytes or words, in any sequential combination. The presence of these FIFOs makes sure that word transfers are possible on the system bus even if the address loaded into the pointer is odd.

## BUS INTERFACE

The bus interface handles the data, address and control interfaces as a superset of the ISA and PCMCIA specifications and allows 8 or 16 bit adapters to be designed with the LAN91C95 with no glue to interface to the ISA or PCMCIA bus.

The functions done in this block are address decoding for I/O and ROM memory (including address relocation support) for ISA, data path routing, sequential memory address support, optional wait state generation, boot ROM support, EEPROM setup function, bus transceiver control, and interrupt generation/selection.

For ISA, I/O address decoding is done by comparing A15-A4 to the I/O BASE address determined in part by the upper byte of the BASE ADDRESS REGISTER, and also requiring that AEN be low. If the above address comparison is satisfied and the LAN91C95 is in 16 bit mode, nIOCS16 will be asserted (low).

A valid comparison does not yet indicate a valid I/O cycle is in progress, as the addresses could be used for a memory cycle, or could even glitch through a valid value. Only when nIORD or nIOWR are activated the I/O cycle begins.

In PCMCIA mode, A4-A15 are ignored for I/O decodes, which rely on the PCMCIA host decoding for the slot. Input A10 for ISA is used as an output (nFWE) for PCMCIA to enable Flash Memory Write for programming the attribute memory. It is valid only when nWE is 0 and COR2 is 1. nA11/nFCS is used to select the Flash Memory Chip.

## WAIT STATE POLICY

The LAN91C95 can work on most system buses without having to add wait states. The two parameters that determine the memory access profile are the read access time and the cycle time into the Data Register.

The read access time is 40ns and the cycle time is 185ns. If any one of them does not satisfy the application requirements, wait states should be added.

If the access time is the problem, IOCHRDY should be negated for all accesses to the LAN91C95. This can be achieved by programming the NO WAIT ST bit in the configuration register to 0. The LAN91C95 will negate IOCHRDY for 100ns to 150ns on every access to any register.

If the cycle time is the problem, programming NO WAIT ST as described before will solve it but at the expense of slowing down all accesses. The alternative is to let the LAN91C95 negate IOCHRDY only when the Data Register FIFOs require so. Namely, if NO WAIT ST is set, IOCHRDY will only be negated if a Data Register read cycle starts and there is less than a full word in the read FIFO, or if a write cycle starts and there is more than two bytes in the write FIFO.

The cycle time is defined as the time between leading edges of read from the Data Register, or equivalently between trailing edges of write to the Data Register. For example, in an ISA system the cycle time of a 16 bit transfer will be at least 2 clocks for the I/O access to the LAN91C95 + one clock for the memory cycle) = 3 clocks. In absolute time it means 375ns for a 8MHz bus, and 240ns for a 12.5 MHz bus.

The cycle time will not increase when configured for full duplex mode, because the CSMA/CD memory arbitration requests are sequenced by the DMA logic and never overlap.

**DMA BLOCK**

The DMA block resides between the CSMA/CD block and the arbiter. It can interface both the data path and the control path of the CSMA/CD block for different operations.

Its functions include the following:

- Start transmission process into the CSMA/CD block.
- Generate CSMA/CD side addresses for accessing memory during transmit and receive operations.
- Generate MMU memory requests and verify success.
- Compute byte count and write it in first locations of receive packet.
- Write transmit status word in first locations of transmit packet.
- Determine if enough memory is available for reception.
- De-allocate transmit memory after suitable completion.
- De-allocate receive memory upon error conditions.
- Initiate retransmissions upon collisions (if less than 16 retries).
- Terminate reception and release memory if packet is too long.

The specific nature of each operation and its trigger event are:

1) TX operations will begin if TXENA is set and TX FIFO is not empty. The DMA logic does not need to use the TX PACKET NUMBER, it goes directly from the FIFO to the MMU. However the DMA logic controls the removal of the PACKET NUMBER from the FIFO.

2) Generation of CSMA/CD side addresses into memory: Independent 11 bit counters are kept for transmit and receive in order to allow full-duplex operation.

3) MMU requests for allocation are generated by the DMA logic upon reception. The initial allocation request is issued when the CSMA block indicates an active reception. If allocation succeeds, the DMA block stores the packet number assigned to it, and generates write arbitration requests for as long as the CSMA/CD FIFO is not empty. In parallel the CSMA/CD completes the address filtering and notifies the DMA of an address match. If there is no address match, the DMA logic will release the allocated memory and stop reception.

4) When the CSMA/CD block notifies the DMA logic that a receive packet was completed, if the CRC is OK, the DMA will either write the previously stored packet number into the RX PACKET NUMBER FIFO (to be processed by the CPU), or if the CRC is bad the DMA will just issue a release command to the MMU (and the CPU will never see that packet).

   Packets with bad CRC can be received if the RCV_BAD bit in the configuration register is set.

5) If AUTO_RELEASE is set, a release is issued by the DMA block to the MMU after a successful transmission (TX_SUCC set), and the TX completion FIFO is clocked together with the TX FIFO preventing the packet number from moving into the TX completion FIFO.
6) Based on the RX counter value, if a receive packet exceeds 1532 bytes, reception is stopped by the DMA and the RX ABORT bit in the Receive Control Register is set. The memory allocated to the packet is automatically released.

7) If an allocation fails, the CSMA/CD block will activate RX_OVRN upon detecting a FIFO full condition. RXEN will stay active to allow reception of subsequent packets if memory becomes available. The CSMA/CD block will flush the FIFO upon the new frame arrival.

**PACKET NUMBER FIFOS**

The transmit packet FIFO stores the packet numbers awaiting transmission, in the order they were enqueued. The FIFO is advanced (written) when the CPU issues the "enqueue packet number command", the packet number to be written is provided by the CPU via the Packet Number Register. The number was previously obtained by requesting memory allocation from the MMU. The FIFO is read by the DMA block when the CSMA/CD block is ready to proceed on to the next transmission. By reading the TX EMPTY INT bit the CPU can determine if this FIFO is empty.

The transmit completion FIFO stores the packet numbers that were already transmitted but not yet acknowledged by the CPU. The CPU can read the next packet number in this FIFO from the Fifo Ports Register. The CPU can remove a packet number from this FIFO by issuing a TX INT acknowledge. The CPU can determine if this FIFO is empty by reading the TX INT bit or the FIFO Ports Register.

The receive packet FIFO stores the packet numbers already received into memory, in the order they were received. The FIFO is advanced (written) by the DMA block upon reception of a complete valid packet into memory. The number is determined the moment the DMA block first requests memory from the MMU for that packet. The first receive packet number in the FIFO can be read via the Fifo Ports Register, and the data associated with it can be accessed through the receive area. The packet number can be removed from the FIFO with or without an automatic release of its associated memory.

The FIFO is read out upon CPU command (remove packet from top of RX FIFO, or remove and release command) after processing the receive packet in the receive area.

The width of each FIFO is 5 bits per packet number. The depth of each FIFO equals the number of packets the LAN91C95 can handle (18).

The guideline is software transparency; the software driver should not be aware of different devices or FIFO depths. If the MMU memory allocation succeeded, there will be room in the transmit FIFO for enqueuing the packet. Conversely if there is free memory for receive, there should be room in the receive FIFO for storing the packet number.

Note that the CPU can enqueue a transmit command with a packet number that does not follow the sequence in which the MMU assigned packet numbers. For example, when a transmission failed and it is retried in software, or when a receive packet is modified and sent back to the network.

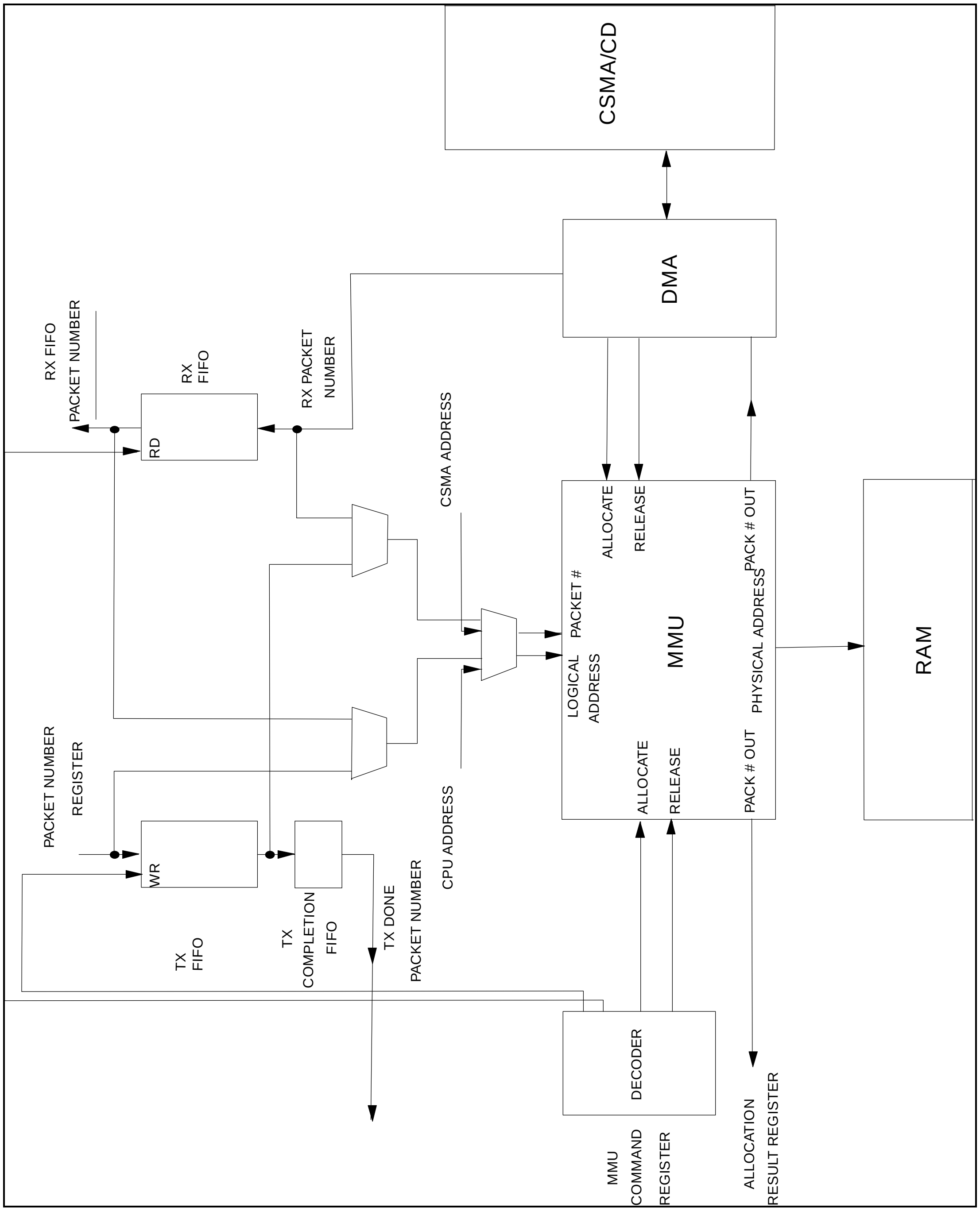


**FIGURE 16 - MMU PACKET NUMBER FLOW AND RELEVANT REGISTERS**

**CSMA BLOCK**

The CSMA/CD block is first interfaced via its control registers in order to define its operational configuration.  From then on, the DMA interface between the CSMA/CD block and memory is used to transfer data to and from its data path interface.

For transmit, the CSMA/CD block will be asked to transmit frames as soon as they are ready in memory.  It will continue transmissions until any of the following transmit error occurs:

a) 16 collisions on same frame (half duplex mode)
b) Late collision (half duplex mode)
c) Lost Carrier sense and MON_CSN set
d) Transmit Under run
e) SQET error and STP_SQET set

In that case TXENA will be cleared and the CPU should restart the transmission by setting it again.  If a transmission is successful, TXENA stays set and the CSMA/CD is provided by the DMA block with the next packet to be transmitted.

For receive, the CPU sets RXEN as a way of starting the CSMA/CD block receive process. The CSMA/CD block will send data after address filtering through the data path to the DMA block.  Data is transferred into memory as it is received, and the final check on data acceptance is the CRC checking done by the CSMA/CD block.  In any case, the DMA takes care of requesting/releasing memory for receive packets, as well as generating the byte count.

The receive status word is provided by the CSMA/CD block and written in the first location of the receive structure by the DMA block.  If configured for storing CRC in memory, the CSMA/CD unit will transfer the CRC bytes through the DMA interface, and then will be treated like regular data bytes.

Note that the receive status word of any packet is available only through memory and is not readable through any other register.  In order to let the CPU know about receive overruns, the RX_OVRN bit is latched into the Interrupt Status Register, which is readable by the CPU at any time.

The address filtering is done inside the CSMA/CD block.  A packet will be received if the destination address is broadcast, or if it is addressed to the individual address of the LAN91C95, or if it is a multicast address and ALMUL bit is set, or if it is a multicast address matching one of the multicast table entries.  If the PRMS bit is set, all packets are received.

The CSMA/CD block is a full duplex machine, and when working in full duplex mode, the CSMA/CD block will be simultaneously using its data path transmit and receive interfaces.

Statistical counters are kept by the CSMA/CD block, and are readable through the appropriate register.  The counters are four bits each, and can generate an interrupt when reaching their maximum values.  Software can use that interrupt to update statistics in memory, or it can keep the counter interrupt disabled, while relying on the transmit interrupt routine reading the counters.  Given that the counters can increment only once per transmit, this technique is a good complement for the single interrupt per sequence strategy.

The interface between the CSMA/CD block and memory is word oriented.  Two bi-directional FIFOs make the data path interface.

Whenever a normal collision occurs (less than 16 retries - half duplex mode), the CSMA/CD will trigger the backoff logic and will indicate the DMA logic of the collision.  The DMA is responsible for restarting the data transfer into the CSMA/CD block regardless of whether the collision happened on the preamble or not.

Only when 16 retries are reached, the CSMA/CD block will clear the TXENA bit, and CPU intervention is required. The DMA will not automatically restart data transfer in this case, nor will it transmit the next enqueued packet until TXENA is set by the CPU. The DMA will move the packet number in question from the TX FIFO into the TX completion FIFO.

## NETWORK INTERFACE

The LAN91C95 includes both an AUI interface for thick and thin coax applications and a 10BASE-T interface for twisted pair applications. Functions common to both are:

1. Manchester encoder/decoder to convert NRZ data to Manchester encoded data and back.
2. A 32 ms jabber timer to prevent inadvertently long transmissions. When 'jabbing' occurs, the transmitter is disabled, automatic loopback is disabled (in 10BASE-T mode), and a collision indication is given to the controller. The interface 'unjabs' when the transmitter has been idle for a minimum of 256 ms.
3. A phase-lock loop to recover data and clock from the Manchester data stream with up to plus or minus 18ns of jitter.
4. Diagnostic loopback capability.
5. LED drivers for collision, transmission, reception, and jabber.

## 10BASE-T

The 10BASE-T interface conforms to the twisted pair MAU addendum to the 802.3 specification. On the transmission side, it converts the NRZ data from the controller to Manchester data and provides the appropriate signal level for driving the media. Signal are predistorted before transmission to minimize ISI. In half duplex mode, the collision detection circuitry monitors the simultaneous occurrence of received signals and transmitted data on the media. During transmission, data is automatically looped back to the receiver except during collision periods, in which case the input to the receiver is network data. During collisions, should the receive input go idle prior to the transmitter going idle, input to the receiver switches back to the transmitter within 9 bit times. Following transmission, the transmitter performs a SQE test. This test exercises the collision detection circuitry within the 10BASE-T interface.

In full duplex mode, carrier sense is asserted during receive activity only. The receiver monitors the media at all times. It recovers the clock and data and passes it along to the controller. In the absence of any receive activity, the transmitter is looped back to the receiver. In addition, the receiver performs automatic polarity correction. The 10BASE-T interface performs link integrity tests per section 14.2.1.7 of 802.3, using the following values:

1. Link_loss_timer: 64 ms
2. Link_test_min_timer: 4 ms
3. Link_count: 2
4. Link_test_max_timer: 64 ms

The state of the link is reflected in the EPHSR.

## AUI

The LAN91C95 also provides a standard 6 wire AUI interface to a coax transceiver.

## PHYSICAL INTERFACE

The internal physical interface (PHY) consists of an encoder/decoder (ENDEC) and an internal 10BASE-T transceiver. The ENDEC also provides a standard 6-pin AUI interface to an external coax transceiver for 10BASE-T and 10BASE-5 applications. The signals between MAC and the PHY can be routed to pins by asserting the nXENDEC pin low. This feature allows the interface to an external ENDEC and transceiver. The PHY functions can be divided into transmit and receive functions.

### Transmit Functions

Manchester Encoding

The PHY encodes the transmit data received from the MAC. The encoded data is directed internally to the selected output driver for transmission over the twisted-pair network or the AUI cable. Data transmission and encoding is initiated by the Transmit Enable input, TXE, going low.

Transmit Drivers

The encoded transmit data passes through to the transmit driver pair, TPETXP(N), and its complement, TPETXDP(N). Each output of the transmit driver pair has a source resistance of 10 ohms maximum and a current rating of 25 mA maximum. The degree of predistortion is determined by the termination resistors; the equivalent resistance should be 100 ohms.

Jabber Function

This integrated function prevents the DTE from locking into a continuous transmit state. In 10BASE-T mode, if transmission continues beyond the specified time limit, the jabber function inhibits further transmission and asserts the collision indicator nCOLL. The limits for jabber transmission are 20 to 15 ms in 10BASE-T mode. In the AUI mode, the jabber function is performed by the external transceiver.

SQE Function

In the 10BASE-T mode, the PHY supports the signal quality error (SQE) function. At the end of a transmission, the PHY asserts the nCOLL signal for 10+/-5 bit times beginning 0.6 to 1.6ms after the last positive transition of a transmitted frame. In the AUI mode, the SQE function is performed by the external transceiver.

### Receive Functions

Receive Drivers

Differential signals received off the twisted-pair network or AUI cable are directed to the internal clock recovery circuit prior to being decoded for the MAC.

Manchester Decoder and Clock Recovery

The PHY performs timing recovery and Manchester decoding of incoming differential signals in 10BASE-T or AUI modes, with its built-in phase-lock loop (PLL). The decoded (NRZ) data, RXD, and the recovered clock, RXCLK, becomes available to the MAC, typically within 9 bit times (5 for AUI) after the assertion of nCRS. The receive clock, RXCLK, is phase-locked to the transmit clock in the absence of a received signal (idle).

Squelch Function

The integrated smart squelch circuit employs a combination of amplitude and timing measurements to determine the validity of data received off the network. It prevents noise at the differential inputs from falsely triggering the decoder in the absence of valid data or link test pulses. Signal levels below 300mV (180mV for AUI) or pulse widths less than 15ns at the differential inputs are rejected. Signals above 585mV (300mV for AUI) and pulse widths greater than 30ns will be accepted. When using the extended cable mode with 10BASE-T media which extends beyond the standard limit of 100 meters, the squelch level can optionally be set to reject signals below 180mV and accept signals above 300mV. If the input signal exceeds the squelch requirements, the carrier sense output, nCRS, is asserted.

Reverse Polarity Function

In the 10BASE-T mode, the PHY monitors for receiver polarity reversal due to crossed wires and corrects by reversing the signal internally.

Collision Detection Function

In the 10BASE-T mode, while in half duplex, a collision state is indicated when there are simultaneous transmissions and receptions on the twisted pair link. During a collision state, the nCOLL signal is asserted. If the received data ends and the transmit control signal is still active, the transmit data is sent to the MAC within 9 bit times. The nnCOLL signal is de-asserted within 9 bit times after the collision terminates. In the AUI mode, the external transceiver sends a 10MHz signal to the PHY upon detection of a collision.

Link Integrity

The PHY test for a faulty twisted-pair link. In the absence of transmit data, link test pulses are tranmsitted every 16+/-18ms after the end of the last transmission or link pulse on the twisted pair medium. If neither valid data nor link test pulses are received within 10 to 150ms, the link is declared bad and both data transmission as well as the operational loopback function are disabled. The Link Integrity function can be disabled for pre-10BASE-T twisted-pair networks.

# BOARD SETUP INFORMATION

## ISA MODE

The following parameters are obtained from the EEPROM as board setup information:

ETHERNET INDIVIDUAL ADDRESS
I/O BASE ADDRESS
ROM BASE ADDRESS
8/16 BIT ADAPTER
10BASE-T or AUI INTERFACE
INTERRUPT LINE SELECTION

All the above mentioned values are read from the EEPROM upon hardware reset. Except for the INDIVIDUAL ADDRESS, the value of the IOS switches determines the offset within the EEPROM for these parameters, in such a way that many identical boards can be plugged into the same system by just changing the IOS jumpers.

In order to support a software utility based installation, even if the EEPROM was never programmed, the EEPROM can be written using the LAN91C95. One of the IOS combination is associated with a fixed default value for the key parameters (I/O BASE, ROM BASE, INTERRUPT) that can always be used regardless of the EEPROM based value being programmed. This value will be used if all IOS pins are left open or pulled high.

The EEPROM is arranged as a 64 x 16 array. The specific target device is the 9346 1024-bit Serial EEPROM. All EEPROM accesses are done in words. All EEPROM addresses shown are specified as word addresses.

| REGISTER | EEPROM WORD ADDRESS |
|---|---|
| Configuration Register | IOS Value * 4 |
| Base Register | (IOS Value *4) + 1 |

INDIVIDUAL ADDRESS 20-22 hex

If IOS2-IOS0=7, only the INDIVIDUAL ADDRESS is read from the EEPROM. Currently assigned values are assumed for the other registers. These values are default if the EEPROM read operation follows hardware reset.

The EEPROM SELECT bit is used to determine the type of EEPROM operation: a) normal or b) general purpose register.

a) NORMAL EEPROM OPERATION - EEPROM SELECT bit = 0

On EEPROM read operations (after reset or after setting RELOAD high) the CONFIGURATION REGISTER and BASE REGISTER are updated with the EEPROM values at locations defined by the IOS2-0 pins. The INDIVIDUAL ADDRESS registers are updated with the values stored in the INDIVIDUAL ADDRESS area of the EEPROM.

On EEPROM write operations (after setting the STORE bit) the values of the CONFIGURATION REGISTER and BASE REGISTER are written in the EEPROM locations defined by the IOS2-0 pins.

The three least significant bits of the CONTROL REGISTER (EEPROM SELECT, RELOAD and STORE) are used to control the EEPROM. Their values are not stored nor loaded from the EEPROM.

b) GENERAL PURPOSE REGISTER - EEPROM SELECT bit = 1

On EEPROM read operations (after setting RELOAD high) the EEPROM word address defined by the POINTER REGISTER 6 least significant bits is read into the GENERAL PURPOSE REGISTER.

On EEPROM write operations (after setting the STORE bit) the value of the GENERAL PURPOSE REGISTER is written at the EEPROM word address defined by the POINTER REGISTER 6 least significant bits.

RELOAD and STORE are set by the user to initiate read and write operations respectively. Polling the value until read low is used to determine completion.  When an EEPROM access is in progress the STORE and RELOAD bits of CTR will readback as both bits high.  No other bits of the LAN91C95 can be read or written until the EEPROM operation completes and both bits are clear.  This mechanism is also valid for reset initiated reloads.

NOTE: If no EEPROM is connected to the LAN91C95, for example for some embedded applications, the ENEEP pin should be grounded and no accesses to the EEPROM will be attempted. Configuration, Base, and Individual Address assume their default values upon hardware reset and the CPU is responsible for programming them for their final value.

**DIAGNOSTIC LEDs**

The following LED drive signals are available for diagnostic and installation aid purposes:

nTXLED - Activated by transmit activity.

nBSELED - Board select LED. Activated when the board space is accessed, namely on accesses to the LAN91C95 register space or the ROM area decoded by the LAN91C95. The signal is stretched to 125 msec.

nRXLED - Activated by receive activity.

nLINKLED - Reflects the link integrity status.

**ARBITRATION CONSIDERATIONS**

The arbiter exploits the sequential nature of the CPU accesses to provide a very fast access time.  Memory bandwidth considerations will have an effect on the CPU cycle time but no effect on access time.

For normal 8 MHz, 10 MHz and 12.5 MHz ISA buses as well as EISA normal cycles the LAN91C95 can be accessed without negating ready.

When write operations occur, the data is written into a FIFO.  The CPU cycle can complete immediately, and the buffered data will be written into memory later.  The memory arbitration request is generated as a function of that FIFO being not empty. The nature of the cycle requested (byte/word) is determined by the lsb of the pointer and the number of bytes in the FIFO.

When read operations occur, words are pre-fetched upon pointer loading in order to have at least a word ready in the FIFO to be read. New pre-fetch cycles are requested as a function of the number of bytes in the FIFO.

For example, if an odd pointer value is loaded, first a byte is pre-fetched into the FIFO, and immediately a full word is pre-fetched completing three bytes into the FIFO. If the CPU reads a word, one byte will be left again a new word is pre-fetched.

In the case of write, if an odd pointer value is loaded, and a full word is written, the FIFO holds two bytes, the first of which is immediately written into an odd memory location. If by that time another byte or word was written, there will be two or three bytes in the FIFO and a full word can be written into the now even memory address.

When a CSMA/CD cycle begins, the arbiter will route the CSMA/CD DMA addresses to the MMU as well as the packet number associated with the operation in progress. In full-duplex mode, receive and transmit requests are alternated in such a way that the CPU arbitration cycle time is not affected.

| IOS2-0 | WORD ADDRESS | 16 BITS |
|---|---|---|
| 000 | 0h | CONFIGURATION REG. |
|  | 1h | BASE REG. |
|  |  |  |
| 001 | 4h | CONFIGURATION REG. |
|  | 5h | BASE REG. |
|  |  |  |
| 010 | 8h | CONFIGURATION REG. |
|  | 9h | BASE REG. |
|  |  |  |
| 011 | Ch | CONFIGURATION REG. |
|  | Dh | BASE REG. |
|  |  |  |
| 100 | 10h | CONFIGURATION REG. |
|  | 11h | BASE REG. |
|  |  |  |
| 101 | 14h | CONFIGURATION REG. |
|  | 15h | BASE REG. |
|  |  |  |
| 110 | 18h | CONFIGURATION REG. |
|  | 19h | BASE REG. |
|  |  |  |
| XXX | 20h | IA0-1 |
|  | 21h | IA2-3 |
|  | 22h | IA4-5 |
|  |  |  |

**FIGURE 17 - 64 X 16 SERIAL EEPROM MAP FOR ISA MODE**

**Table 15 -**
**PCMCIA - EEPROM to SRAM Memory Map**

| ATTRIBUTE MEMORY HOST ADDRESS (HEX) | ATTRIBUTE DATA | EEPROM ADDRESS IN WORDS | LAN91C95 SRAM IN BYTES |
|---|---|---|---|
| 000 | Data Byte 0 | Word 1 - Low Byte | Byte 0 |
| 001 | Don't Care | | |
| 002 | Data Byte 1 | Word 1 - High Byte | Byte1 |
| 003 | Don't Care | | |
| 004 | Data Byte 2 | Word 2 - Low Byte | Byte2 |
| 005 | Don't Care | | |
| 006 | Data Byte 3 | Word 2 - High Byte | Byte3 |
| .. | .. | .. | .. |
| 3F8 | Data Byte 1FC | Word FE - Low Byte | Byte FC |
| 3F9 | Don't Care | | |
| 3FA | Data Byte 1FD | Word FE- High Byte | Byte1FD |
| 3FB | Don't Care | | |
| 3FC | Data Byte 1FE | Word FF - Low Byte | Byte1FE |
| 3FD | Don't Care | | |
| 3FE | Data Byte 1FF | Word FF - High Byte | Byte1FF |
| 3FF | Don't Care | | |

Note: This memory map assumes a 4096 bit Serial EEPROM in PCMCIA mode.

## OPERATIONAL DESCRIPTION

### MAXIMUM GUARANTEED RATINGS*

Operating Temperature Range........................................................................................0$^{o}$ C to +70$^{o}$ C
Storage Temperature Range ......................................................................................-55$^{o}$ C to +150$^{o}$ C
Lead Temperature Range (soldering, 10 seconds) ..................................................................+325$^{o}$ C
Positive Voltage on any pin, with respect to Ground ...........................................................Vcc + 0.3V
Negative Voltage on any pin, with respect to Ground ................................................................... -0.3V
Maximum Vcc ........................................................................................................................... +7V

*Stresses above those listed above could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

Note: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested that a clamp circuit be used.

**DC ELECTRICAL CHARACTERISTICS** ($T_A$ = 0$^{o}$ C - 70$^{o}$ C, Vcc = +5.0 V ± 10%)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|
| ***Current Supply*** | | | | | | |
| *Active* | $I_{CC}$ | | *TBD* | *TBD* | | |
| ***Current Supply*** | | | | | | *Magic Packet Scanning Disabled* |
| *Powerdown (Ethernet)* | $I_{CCPDN}$ | | *TBD* | *TBD* | | |
| ***Current Supply*** | | | | | | |
| *Magic Packet Mode* | $I_{CCMP}$ | | *TBD* | *TBD* | | |
| **I Type Input Buffer** | | | | | | |
| Low Input Level | $V_{ILI}$ | | | 0.8 | V | TTL Levels |
| High Input Level | $V_{IHI}$ | 2.0 | | | V | |
| **IS Type Input Buffer** | | | | | | |
| Low Input Level | $V_{ILIS}$ | | | 0.8 | V | Schmitt Trigger |
| High Input Level | $V_{IHIS}$ | 2.2 | | | V | Schmitt Trigger |
| Schmitt Trigger Hysteresis | $V_{HYS}$ | | 250 | | mV | |

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|
| **$I_{CLK}$ Input Buffer** | | | | | | |
| Low Input Level | $V_{ILCK}$ | | | 0.4 | V | |
| High Input Level | $V_{IHCK}$ | 3.0 | | | V | |
| **Input Leakage**<br>(All I and IS buffers except pins with pullups/pulldowns) | | | | | | |
| Low Input Leakage | $I_{IL}$ | -10 | | +10 | µA | $V_{IN} = 0$ |
| High Input Leakage | $I_{IH}$ | -10 | | +10 | µA | $V_{IN} = V_{CC}$ |
| **IP Type Buffers** | | | | | | |
| Input Current | $I_{IL}$ | -150 | -75 | | µA | $V_{IN} = 0$ |
| **ID Type Buffers** | | | | | | |
| Input Current | $I_{IH}$ | | +75 | +150 | µA | $V_{IN} = V_{CC}$ |
| **I/04 Type Buffer** | | | | | | |
| Low Output Level | $V_{OL}$ | | | 0.4 | V | $I_{OL}$ = 4 mA |
| High Output Level | $V_{OH}$ | 2.4 | | | V | $I_{OH}$ = -2 mA |
| Output Leakage | $I_{OL}$ | -10 | | +10 | µA | $V_{IN}$ = 0 to $V_{CC}$ |
| **I/024 Type Buffer** | | | | | | |
| Low Output Level | $V_{OL}$ | | | 0.5 | V | $I_{OL}$ = 24 mA |
| High Output Level | $V_{OH}$ | 2.4 | | | V | $I_{OH}$ = -12 mA |
| Output Leakage | $I_{OL}$ | -10 | | +10 | µA | $V_{IN}$ = 0 to $V_{CC}$ |
| **024 Type Buffer** | | | | | | |
| Low Output Level | $V_{OL}$ | | | 0.5 | V | $I_{OL}$ = 24 mA |
| High Output Level | $V_{OH}$ | 2.4 | | | V | $I_{OH}$ = -12 mA |
| Output Leakage | $I_{OL}$ | -10 | | +10 | µA | $V_{IN}$ = 0 to $V_{CC}$ |
| **04 Type Buffer** | | | | | | |
| Low Output Level | $V_{OL}$ | | | 0.4 | V | $I_{OL}$ = 4 mA |
| High Output Level | $V_{OH}$ | 2.4 | | | V | $I_{OH}$ = -2 mA |
| Output Leakage | $I_{OL}$ | -10 | | +10 | µA | $V_{IN}$ = 0 to $V_{CC}$ |

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | COMMENTS |
|---|---|---|---|---|---|---|
| **OD16 Type Buffer** | | | | | | |
| Low Output Level | $V_{OL}$ | | | 0.5 | V | $I_{OL}$ = 16 mA |
| Output Leakage | $I_{OL}$ | -10 | | +10 | µA | $V_{IN}$ = 0 to $V_{CC}$ |
| **OD162 Type Buffer** | | | | | | |
| Low Output Level | $V_{OL}$ | | | 0.5 | V | $I_{OL}$ = 16 mA |
| High Output Level | $V_{OH}$ | 2.4 | | | V | $I_{OH}$ = -2 mA |
| Output Leakage | $I_{OL}$ | -10 | | +10 | µA | $V_{IN}$ = 0 to $V_{CC}$ |
| **OD24 Type Buffer** | | | | | | |
| Low Output Level | $V_{OL}$ | | | 0.5 | V | $I_{OL}$ = 24 mA |
| Output Leakage | $I_{OL}$ | -10 | | +10 | µA | $V_{IN}$ = 0 to $V_{CC}$ |
| Supply Current Active | $I_{CC}$ | | 60 | 95 | mA | All outputs open |
| Supply Current Standby | $I_{CSBY}$ | | 8 | | mA | |

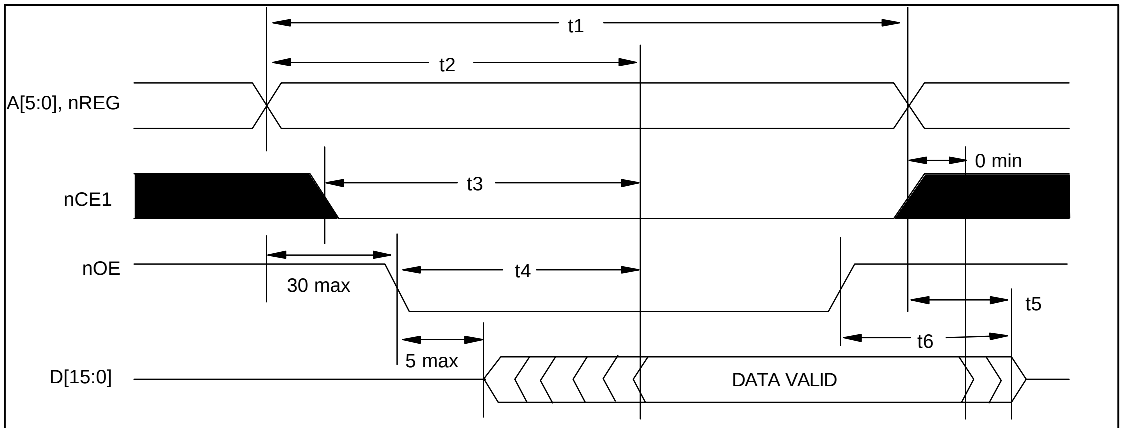
CAPACITANCE $T_A$ = 25$^o$C; fc = 1MHz; $V_{CC}$ = 5V

| | | LIMITS | | | | |
|---|---|---|---|---|---|---|
| **PARAMETER** | **SYMBOL** | **MIN** | **TYP** | **MAX** | **UNIT** | **TEST CONDITION** |
| Clock Input Capacitance | $C_{IN}$ | | | 20 | pF | All pins except pin under test tied to AC ground |
| Input Capacitance | $C_{IN}$ | | | 10 | pF | |
| Output Capacitance | $C_{OUT}$ | | | 20 | pF | |

| PARAMETER | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|
| **10BASE-T** | | | | |
| Receiver Threshold Voltage | | 100 | | mV |
| Receiver Squelch | 300 | 400 | 585 | mV |
| Receiver Common Mode Range | 0 | | $V_{DD}$ | |
| Transmitter Output: Voltage<br>Source Resistance | ±2 | ±2.5 | ±3<br>10 | V<br>ohms |
| Transmitter Output DC Offset | | | 50 | mV |
| Transmitter Backswing Voltage to Idle | | | 100 | mV |
| Differential Input Voltage | ±0.585 | | ±3 | V |
| **AUI** | | | | |
| Receiver Threshold Voltage | | 60 | | mV |
| Receiver Squelch | 180 | 240 | 300 | mV |
| Receiver Common Mode Range | 0 | | $V_{DD}$ | |
| Transmitter Output Voltage ($R=78\Omega$) | ±0.45 | ±0.85 | ±1.2 | V |
| Transmitter Backswing Voltage to Idle | | | 100 | mV |
| Input Differential Voltage | ±0.3 | | ±1.2 | V |
| Output Short Circuit (to $V_{CC}$ or GND) Current | | | ±150 | mV |
| Differential Idle Voltage (measured 8.0 µs after last positive transition of data frame) | | | ±40 | mV |

**CAPACITIVE LOAD ON OUTPUTS**

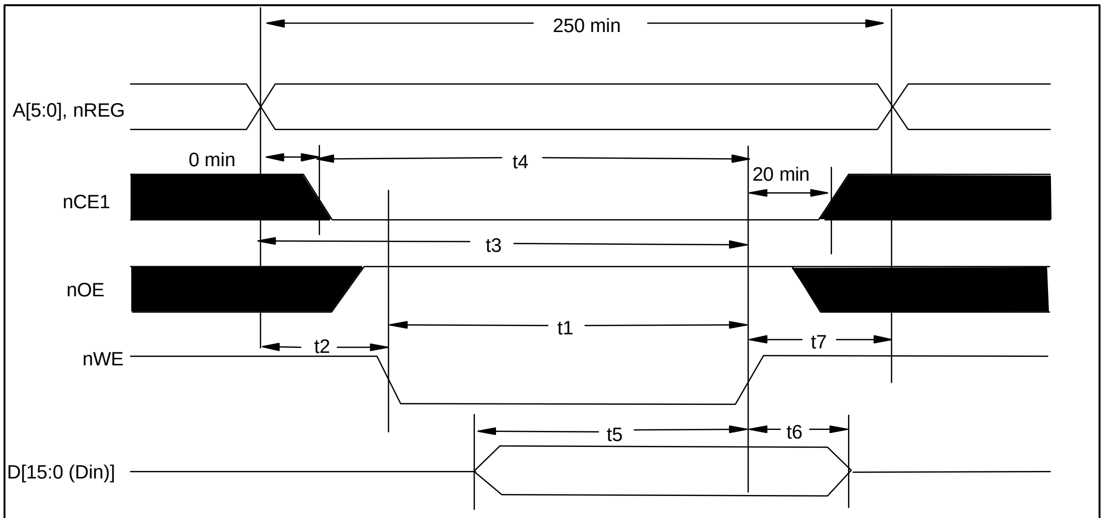
| | |
|---|---|
| nIOCS16, IOCHRDY | 240 pF |
| INTRO-INTR3 | 120 pF |
| All other outputs | 45 pF |



| | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| t1 | Address Access Time | | | 300 | ns |
| t2 | nREG Access Time | | | 300 | ns |
| t3 | nCE1 Access Time | | | 300 | ns |
| t4 | nOE Access Time | | | 150 | ns |
| t5 | Output Disable Time from nCE1 high | | | 100 | ns |
| t6 | Output disable Time from nOE high | | | 100 | ns |

NOTE:  Applies only when nWAIT is asserted by the LAN91C95.

**FIGURE 18 - PCMCIA MEMORY READ TIMING**



| | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| t1 | nWE Pulse Width | 150 | | | ns |
| t2 | Address/nREG Setup Time to nWE Low | 30 | | | ns |
| t3 | Address/nREG Setup Time to nWE High | 180 | | | ns |
| t4 | nCE1 Low to nWE High Setup Time | 180 | | | ns |
| t5 | Data to nWE High Setup Time | 80 | | | ns |
| t6 | Data Hold Time from nWE High | 30 | | | ns |
| t7 | Write Recovery Time (Address, nREG Hold from nWE High) | 30 | | | ns |

NOTE: Minimum write pulse width must be met whether or not nWAIT is asserted by the LAN91C95

**FIGURE 19 - PCMCIA MEMORY WRITE TIMING**

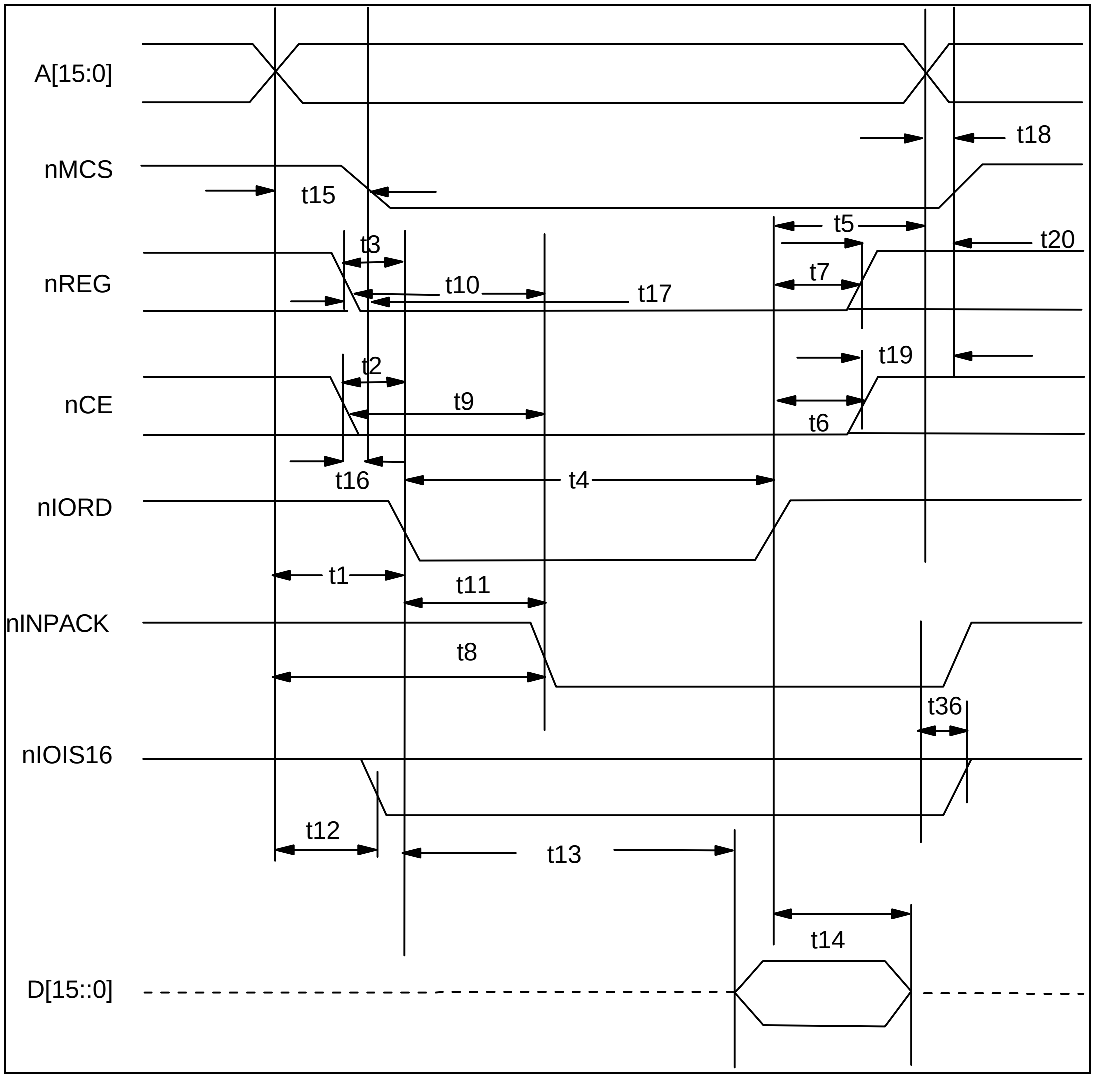


**FIGURE 20 - I/O READ TIMING**
**(Table on the following page)**

|  | PARAMETER | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| t1 | Address setup before nIORD low | 70 |  |  | ns |
| t2 | nCEl, nCE2 setup before nIORD low | 5 |  |  | ns |
| t3 | nREG setup before nIORD low | 5 |  |  | ns |
| t4 | nIORD low width | 165 |  |  | ns |
| t5 | Address hold from nIORD high | 20 |  |  | ns |
| t6 | nCE1, nCE2 hold following nIORD high | 20 |  |  | ns |
| t7 | nREG hold following nIORD high | 0 |  |  | ns |
| t8 | Address valid to nINPACK low |  |  | 115 | ns |
| t9 | nCE1, nCE2 low to nINPACK low |  |  | 50 | ns |
| t10 | nREG low to nINPACK low |  |  | 50 | ns |
| t11 | nIORD low to nINPACK low |  |  | 45 | ns |
| t12 | Address valid to nIOIs16 valid |  |  | 35 | ns |
| t13 | nIORD low to data valid |  |  | 100 | ns |
| t14 | Data hold following nIORD | 0 |  |  | ns |
| t36 | nIOIS16 delay from address |  |  | 35 | ns |
| t15 | Address valid to nMCS low |  |  | 20 | ns |
| t16 | nCE1 low to nMCS low |  |  | 40 | ns |
| t17 | nREG low to nMCS low |  |  | 40 | ns |
| t18 | Address invalid to nMCS high |  |  | 100 | ns |
| t19 | nCE1 high to nMCS high |  |  | 40 | ns |
| t20 | nREG high to nMCS high |  |  | 40 | ns |

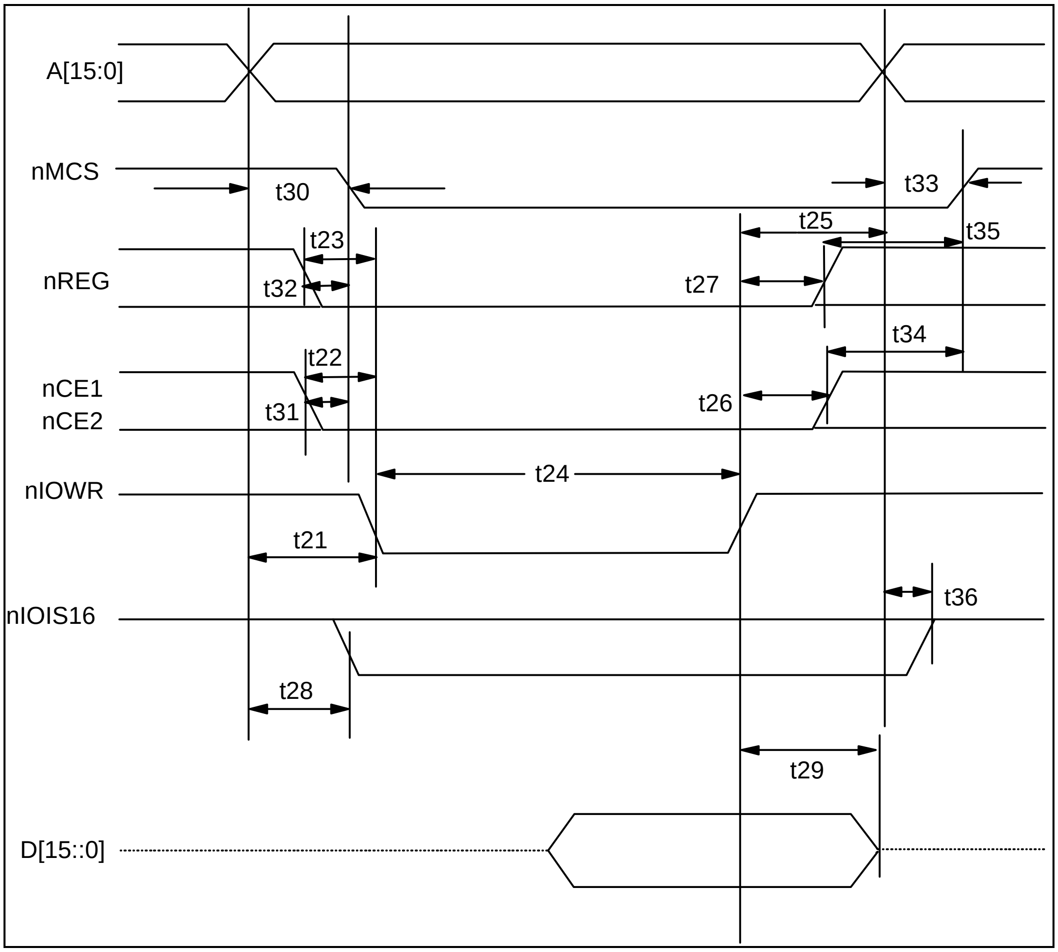
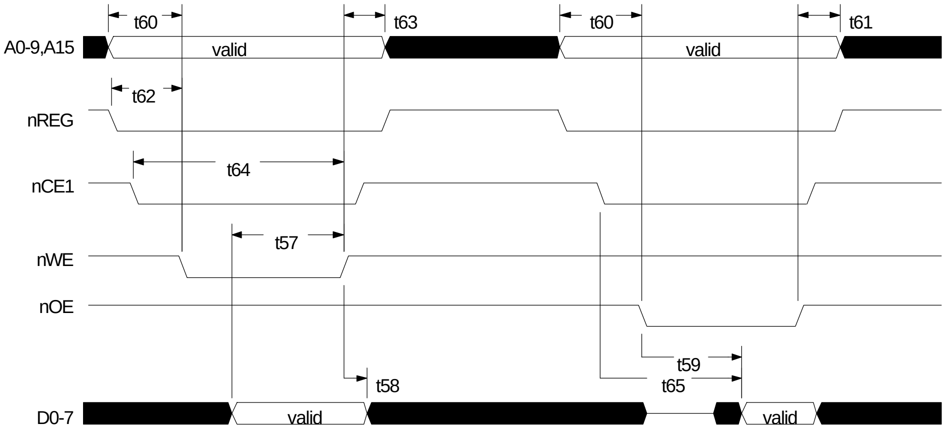


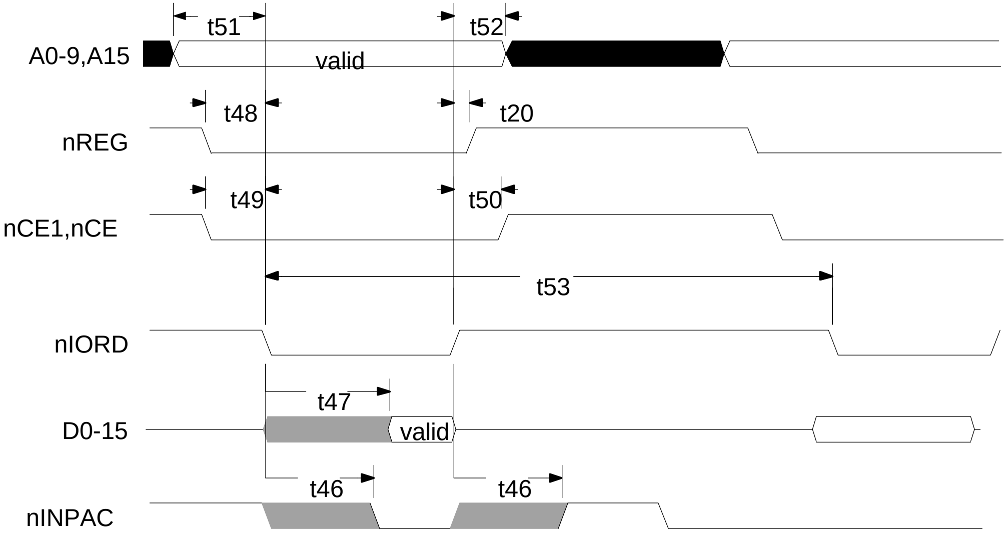
**FIGURE 21 - (I/O WRITE TIMING)**
**(Table on the following page)**

|  | PARAMETER | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| t21 | Address setup before nIOWR low | 70 |  |  | ns |
| t22 | nCE1, nCE2 setup before nIOWR low | 5 |  |  | ns |
| t23 | nREG setup before nIOWR low | 5 |  |  | ns |
| t24 | nIOWR low width | 165 |  |  | ns |
| t25 | Address hold from nIOWR high | 20 |  |  | ns |
| t26 | nCE1, nCE2, hold following nIOWR high | 20 |  |  | ns |
| t27 | nREG hold following nIOWR high | 0 |  |  | ns |
| t28 | Address valid to nIOIS16 valid |  |  | 35 | ns |
| t29 | Data hold following nIOWR | 30 |  |  | ns |
| t30 | Address valid to nMCS low |  |  | 100 | ns |
| t31 | nCE1 low to nMCS low |  |  | 40 | ns |
| t32 | nREG low to nMCS low |  |  | 40 | ns |
| t33 | Address invalid to nMCS high |  |  | 100 | ns |
| t34 | nCE1 high to nMCS high |  |  | 40 | ns |
| t35 | nREG high to nMCS high |  |  | 40 | ns |
| t36 | nIOIS16 delay from address |  |  | 35 | ns |



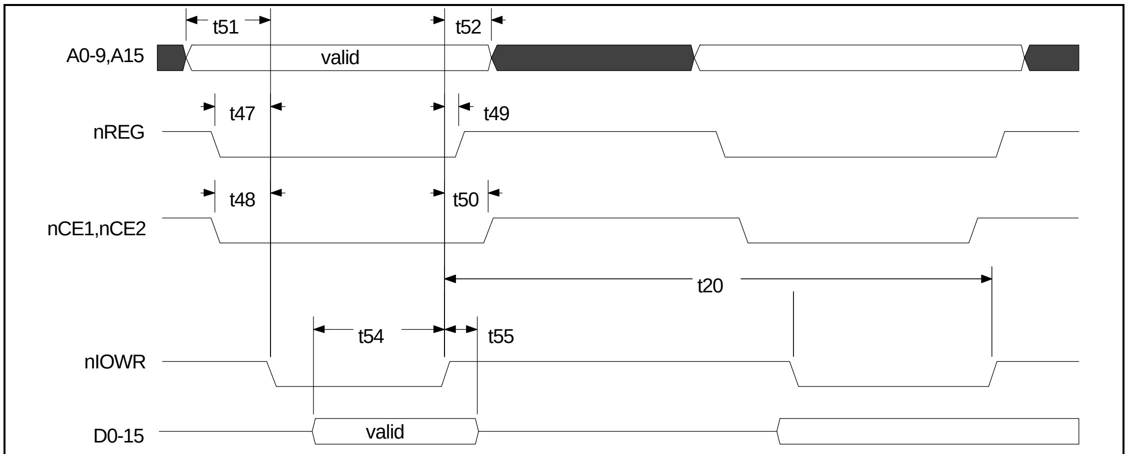
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t57 | Write Data Setup to nWE Rising | 30 | | | ns |
| t58 | Write Data Hold after nWE Rising | 9 | | | ns |
| t59 | nOE Low to Valid Data | | | 40 | ns |
| t60 | Address, nREG Setup to nWE Active | 25 | | | ns |
| t61 | Address, nREG Hold after nOE Inactive | 15 | | | ns |
| t62 | Address, nREG Setup to nOE Active | 25 | | | ns |
| t63 | Address, nREG Hold after Control Inactive | 15 | | | ns |
| t64 | nCE1 Setup to nWE Rising | 60 | | | ns |
| t65 | nCE1 Low to Valid Data | 0 | | 50 | ns |

**FIGURE 22 - CARD CONFIGURATION REGISTERS - READ/WRITE PCMCIA MODE (A15=1)**



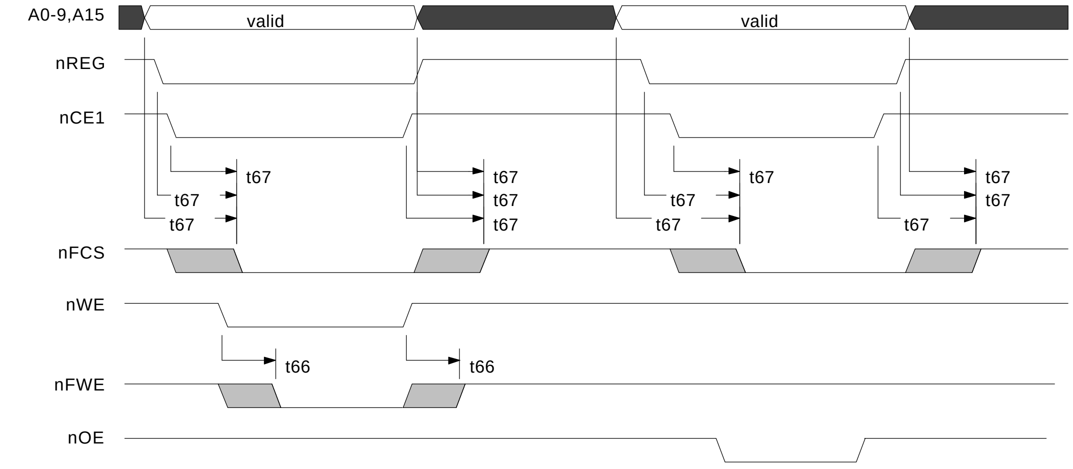
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t46 | nIORD Delay to | 0 | | 35 | ns |
| t47 | nREG Low to Control | 5 | | | ns |
| t48 | nCE1,nCE2 Setup to Control | 5 | | | ns |
| t20 | Cycle Time (No Wait | 185 | | | ns |
| t49 | nREG Hold after Control | 0 | | | ns |
| t50 | nCE1,nCE2 Hold after Control | 15 | | | ns |
| t51 | Address Setup to Control | 25 | | | ns |
| t52 | Address Hold after Control | 15 | | | ns |
| t53 | nIORD Active to Data | 0 | | 40 | ns |

**FIGURE 23 - PCMCIA CONSECUTIVE READ CYCLES**



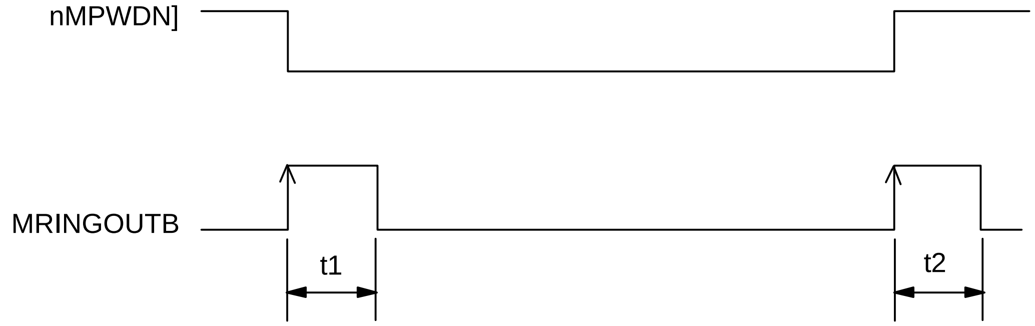
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t47 | nREG Low Setup to Control Active | 5 | | | ns |
| t48 | nCE1,nCE2 Setup to Control Active | 5 | | | ns |
| t49 | nREG Hold after Control Inactive | 0 | | | ns |
| t50 | nCE1,nCE2 Hold after Control Inactive | 15 | | | ns |
| t51 | Address Setup to Control Active | 25 | | | ns |
| t52 | Address Hold after Control Inactive | 15 | | | ns |
| t20 | Cycle Time (No Wait States) | 185 | | | ns |
| t54 | Write Data Setup to nIOWR Rising | 30 | | | ns |
| t55 | Write Data Hold after nIOWR Rising | 9 | | | ns |

**FIGURE 24 - CONSECUTIVE PCMCIA WRITE CYCLES**



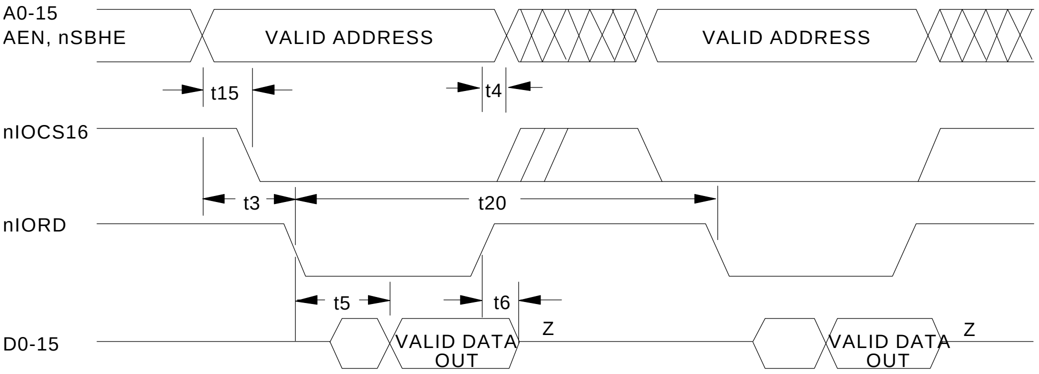
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t66 | nWE to nFWE Delay | 0 | | 20 | ns |
| t67 | Address, nREG, nCE1 Delay to nFCS | 0 | | 25 | ns |

**FIGURE 25 - PCMCIA ATTRIBUTE MEMORY READ/WRITE (A15=0)**



| | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| t1 | MRINGOUTB Pulse Entering Powerdown | | | 7.5 | ms |
| t2 | MRINGOUTB Pulse Exiting Powerdown | | | 7.5 | ms |

**FIGURE 26 - RINGOUT FOR L39/C39 ROCKWELL MODEMS ENTERING/EXITING POWERDOWN**



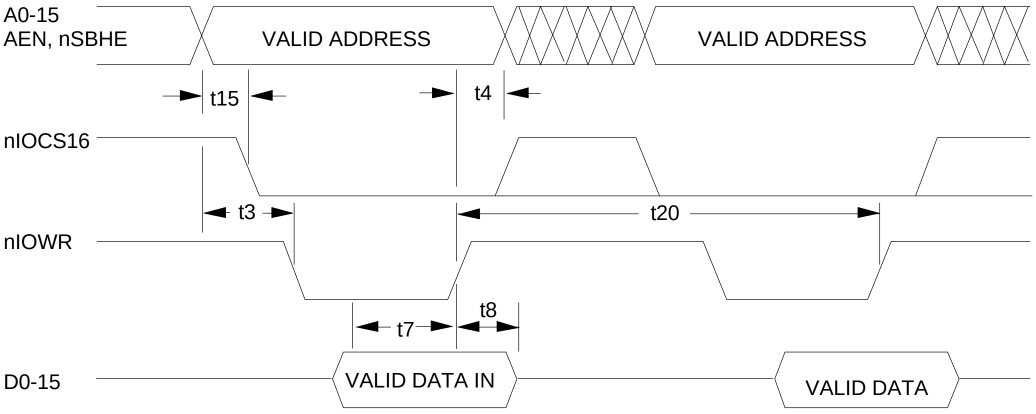
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t3 | Address, nSBHE, AEN Setup to Control Active | 25 | | | ns |
| t4 | Address, nSBHE, AEN Hold after Control Inactive | 20 | | | ns |
| t5 | nIORD Low to Valid Data | | | 40 | ns |
| t6 | nIORD High to Data Floating | | | 30 | ns |
| t15 | A4-A15, AEN Low, BALE High to nIOCS16 Low | | | 25 | ns |
| t20 | Cycle time* | 185 | | | ns |

BALE Tied High

IOCHRDY not used - t20 has to be met

*Note: The cycle time is defined only for consecutive accesses to the Data Register. These values assume that IOCHRDY is not used.

**FIGURE 27 - ISA CONSECUTIVE READ CYCLES**



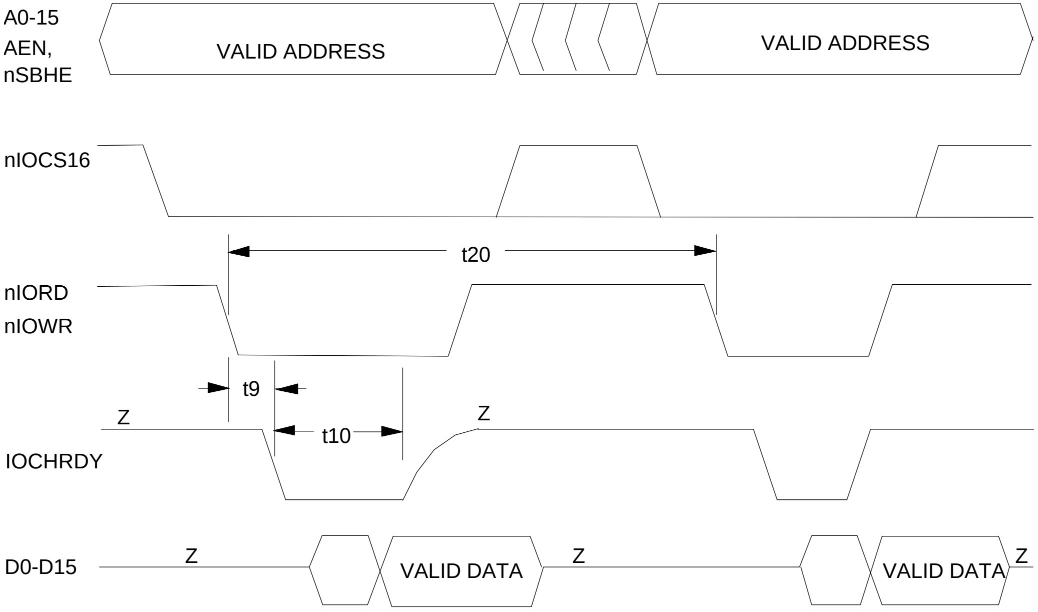
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t3 | Address, nSBHE, AEN Setup to Control Active | 25 | | | ns |
| t4 | Address, nSBHE, AEN Hold after Control Inactive | 20 | | | ns |
| t7 | Data Setup to nIOWR Rising | 30 | | | ns |
| t8 | Data Hold after nIOWR Rising | 9 | | | ns |
| t15 | A4-A15, AEN Low, BALE High to nIOCS16 Low | | | 25 | ns |
| t20 | Cycle time* | 185 | | | ns |

BALE Tied High

IOCHRDY not used - t20 has to be met

*Note: The cycle time is defined only for consecutive accesses to the Data Register. These values assume that IOCHRDY is not used.

**FIGURE 28 - ISA CONSECUTIVE WRITE CYCLES**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t9 | Control Active to IOCHRDY Low | | | 15 | ns |
| t10 | IOCHRDY Low Pulse Width* | 100 | | 150 | ns |
| t20 | Cycle time** | 185 | | | ns |

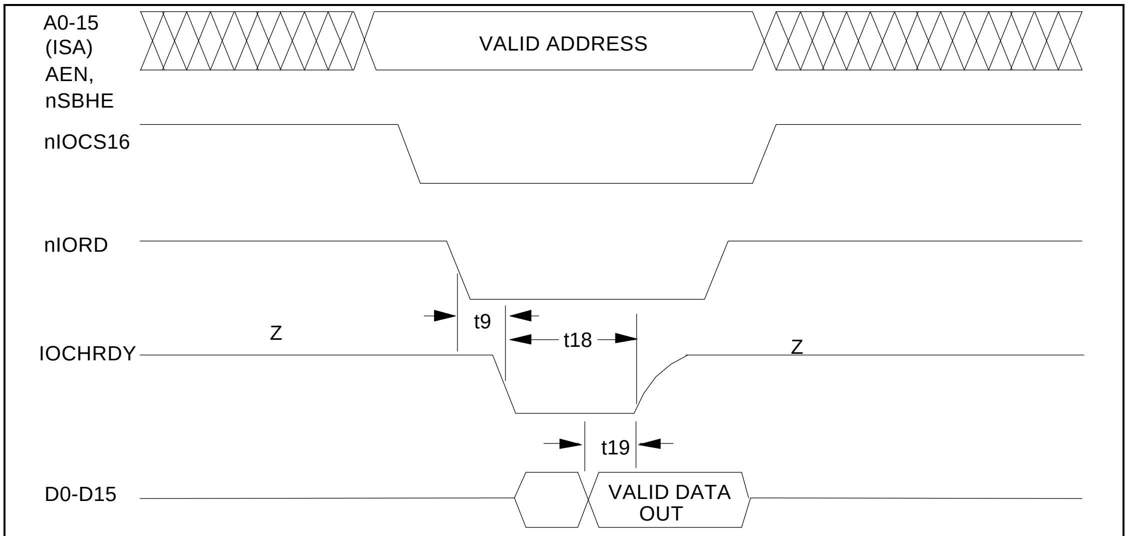
*Note: Assuming NO WAIT ST = 0 in configuration register and cycle time observed.

**Note: The cycle time is defined only for accesses to the Data Register as follows:

For Data Register Read - From nIORD falling to next nIORD falling

For Data Register Write - From nIOWR rising to next nIOWR rising

**FIGURE 29 - ISA CONSECUTIVE READ AND WRITE CYCLES**

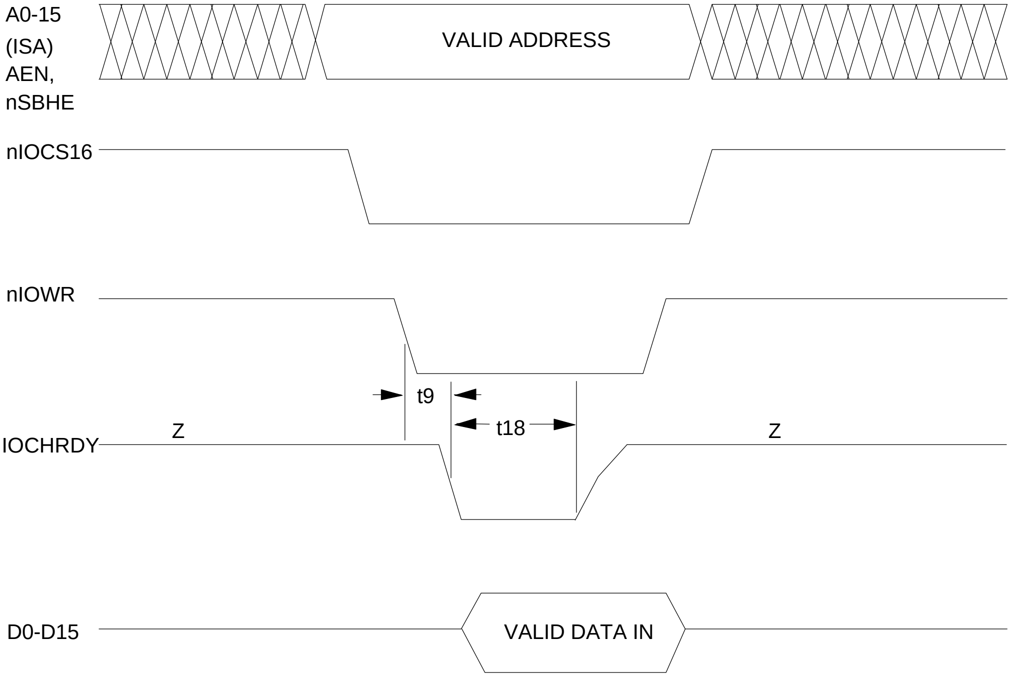


| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t9 | Control Active to IOCHRDY Low | | | 15 | ns |
| t18 | IOCHRDY Width when Data is Unavailable at Data Register | | | 575 | ns |
| t19 | Valid Data to IOCHRDY Inactive | | | 225 | ns |

IOCHRDY is used instead of meeting t20 and t43.

"No Wait St' bit is 1 - IOCHRDY only negated if needed and only for Data Register access.

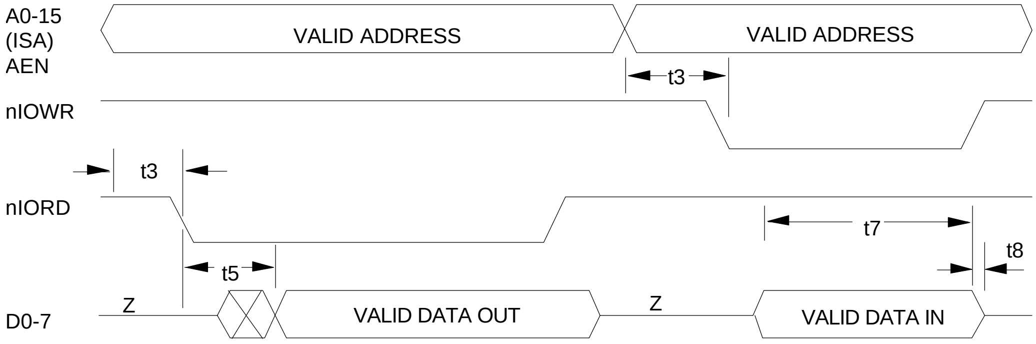
**FIGURE 30 - DATA REGISTER SPECIAL READ ACCESS**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t9 | Control Active to IOCHRDY Low | | | 15 | ns |
| t18 | IOCHRDY Width when Data Register is Full | | | 425 | ns |

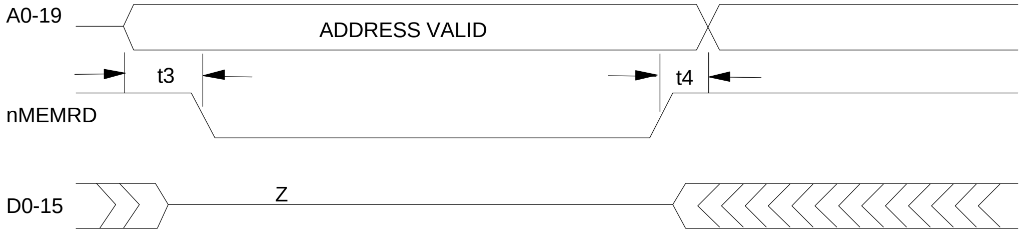
IOCHRDY is used instead of  meeting t20 and t44.
'No Wait St' bit  is 1 - IOCHRDY only negated if needed and only for Data Register access.

**FIGURE 31 - DATA REGISTER SPECIAL WRITE ACCESS**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t3 | Address, nSBHE, AEN Setup to Control Active | 25 | | | ns |
| t5 | nIORD Low to Valid Data | | | 40 | ns |
| t7 | Data Setup to nIOWR Rising | 30 | | | ns |
| t8 | Data Hold after nIOWR Rising | 9 | | | ns |

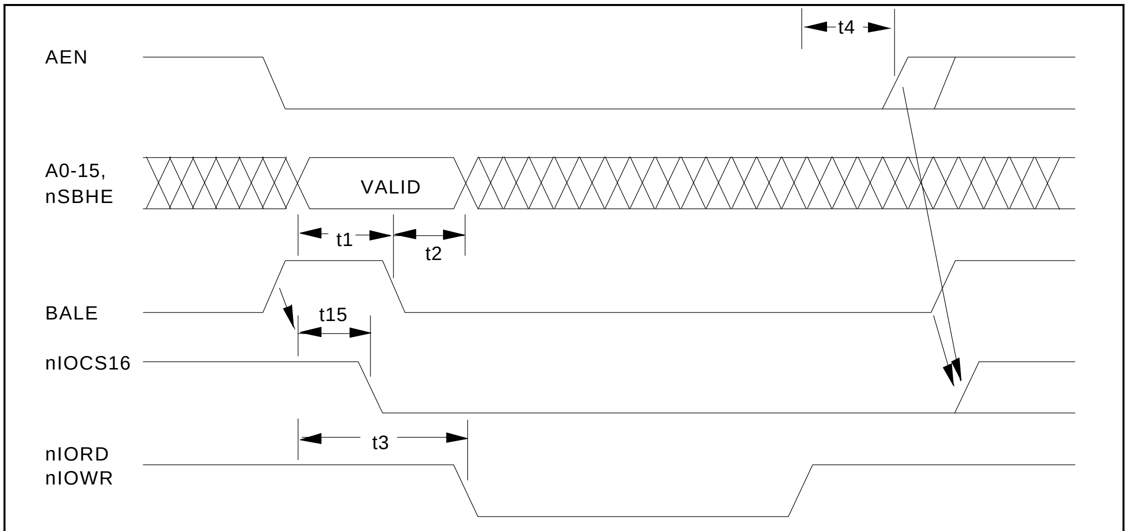
**FIGURE 32 - 8 BIT MODE REGISTER CYCLES**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t3 | Address Setup to Control Active | 25 | | | ns |
| t4 | Address Hold after Control Inactive | 20 | | | ns |
| t16 | nMEMRD Low to nROM Low | 0 | | 25 | ns |
| t17 | nMEMRD High to nROM High | 0 | | 30 | ns |

BALE tied high

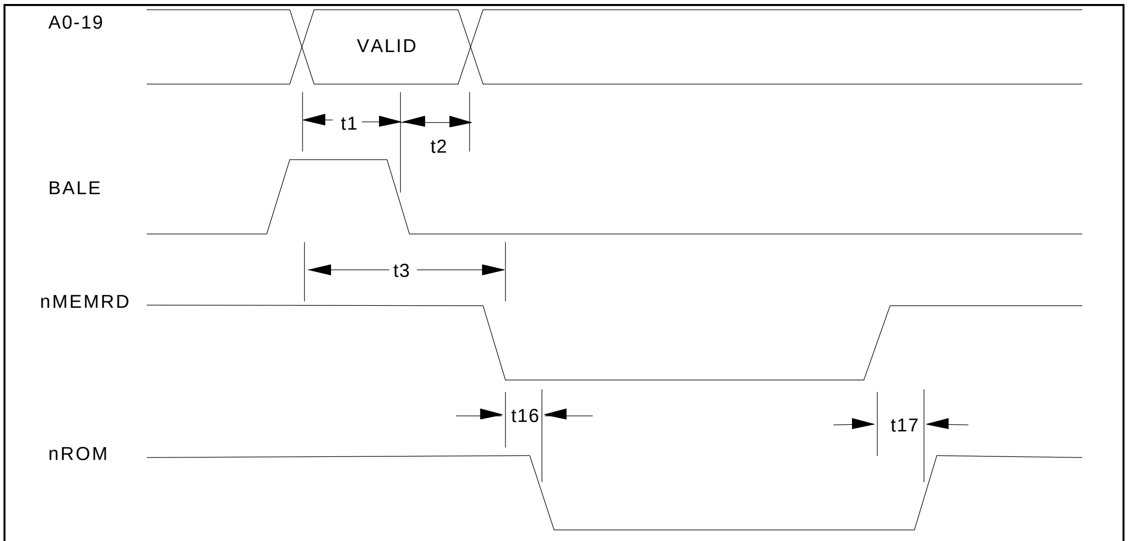
**FIGURE 33 - EXTERNAL ROM READ ACCESS**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t1 | Address, nSBHE Setup to BALE Falling | 20 | | | ns |
| t2 | Address, nSBHE Hold after BALE Falling | 20 | | | ns |
| t3 | Address, nSBHE, AEN Setup to Control Active | 25 | | | ns |
| t4 | AEN Hold after Control Inactive | 20 | | | ns |
| t15 | A4-A15, AEN Low, BALE High to nIOCS16 Low | | | 25 | ns |

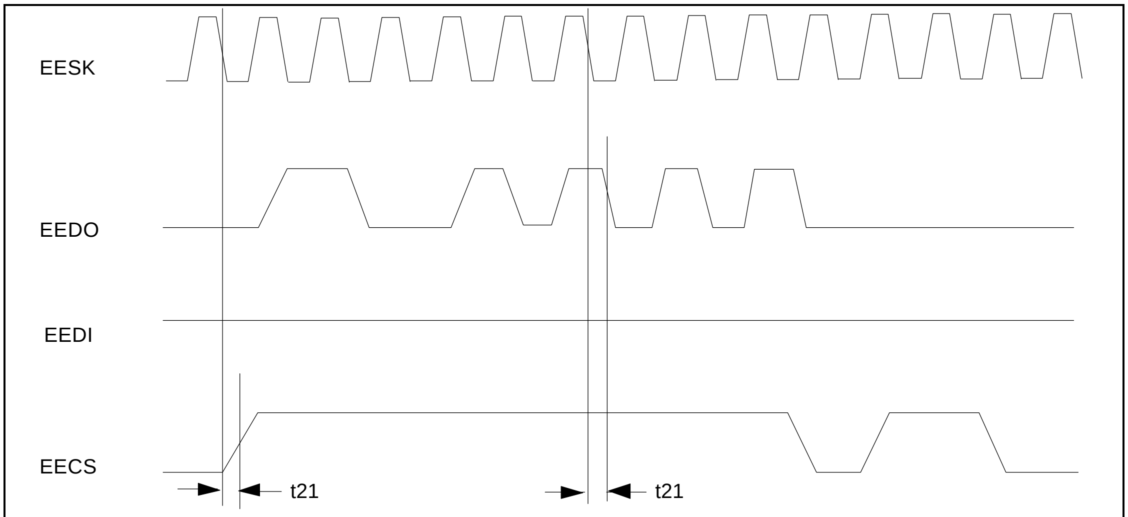
t4 not needed.  nIOCS16 not relevant in 8-bit mode.

**FIGURE 34 - ISA REGISTER ACCESS WHEN USING BALE**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t1 | Address Setup to BALE Falling | 20 | | | ns |
| t2 | Address Hold after BALE Falling | 20 | | | ns |
| t3 | Address Setup to Control Active | 25 | | | ns |
| t16 | nMEMRD Low to nROM Low | | | 25 | ns |
| t17 | nMEMRD High to nROM High | | | 30 | ns |

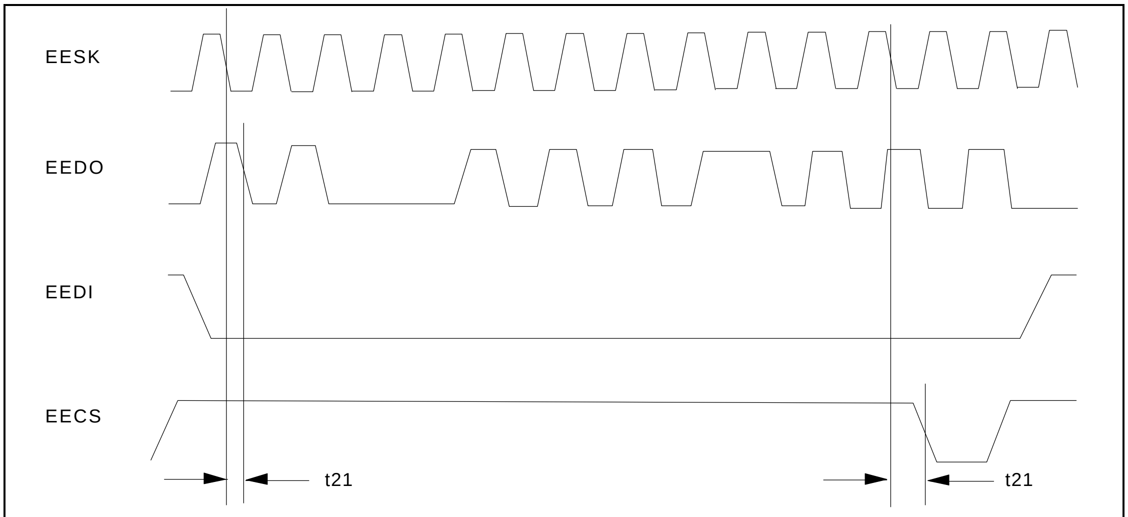
**FIGURE 35 - EXTERNAL ROM READ ACCESS USING BALE**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t21 | EESK Falling to EEDO, EECS Changing | 0 | | 100 | ns |

9346 is typically the serial EEPROM used.

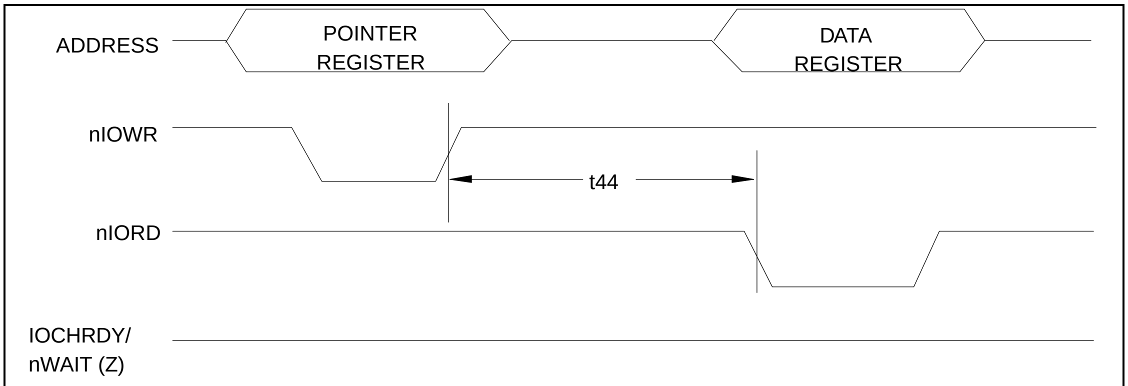
**FIGURE 36 - EEPROM READ**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t21 | EESK Falling to EEDO, EECS Changing | | | 100 | ns |

9346 is typically the serial EEPROM used.

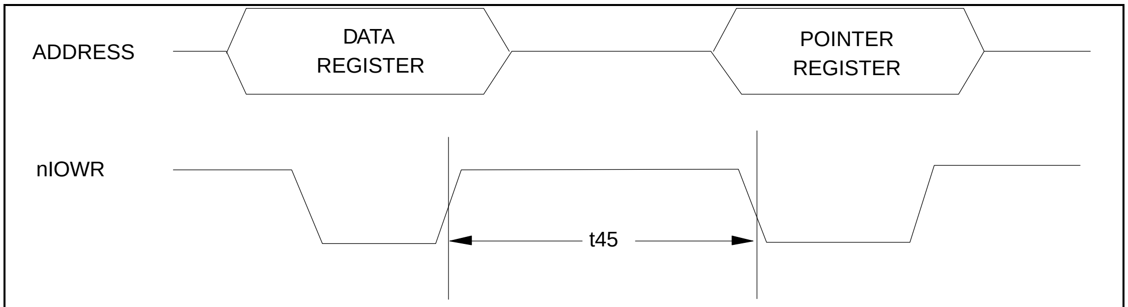
**FIGURE 37 - EEPROM WRITE**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t44 | Pointer Register Reloaded to a Word of Data Prefetched into Data Register | 2 * t20 | | | ns |

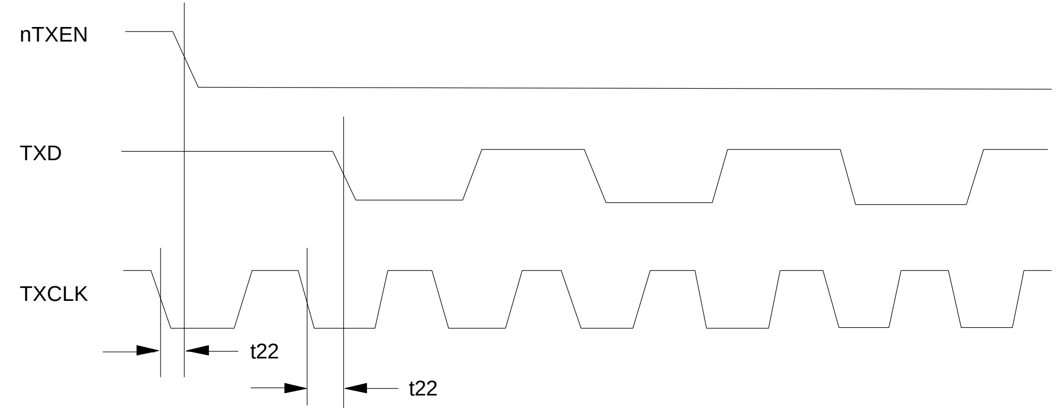
Note: If t44 is not met, IOCHRDY will be negated for the required time. This parameter can be ignored if IOCHRDY is connected to the system.

**FIGURE 38 - MEMORY READ TIMING**



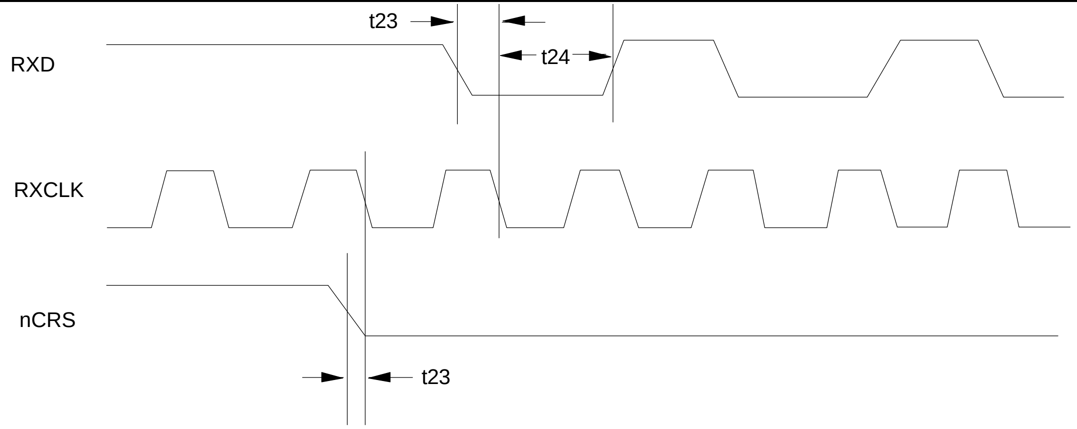
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t45 | Last Access to Data Register to Pointer Reloaded | 2 * t20 | | | ns |

**FIGURE 39 - MEMORY WRITE TIMING**



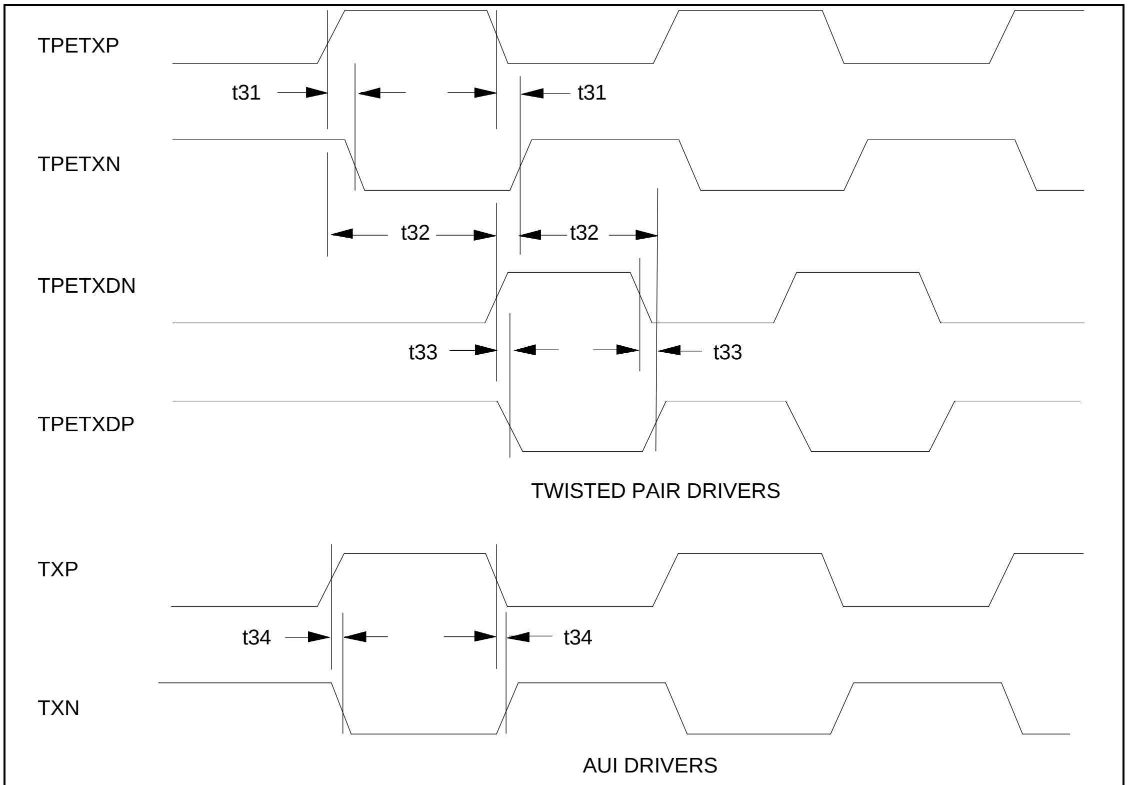
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t22 | TXD, nTXEN Delay from TXCLK Falling | 0 | | 40 | ns |

**FIGURE 40 - EXTERNAL ENDEC INTERFACE - START OF TRANSMIT**



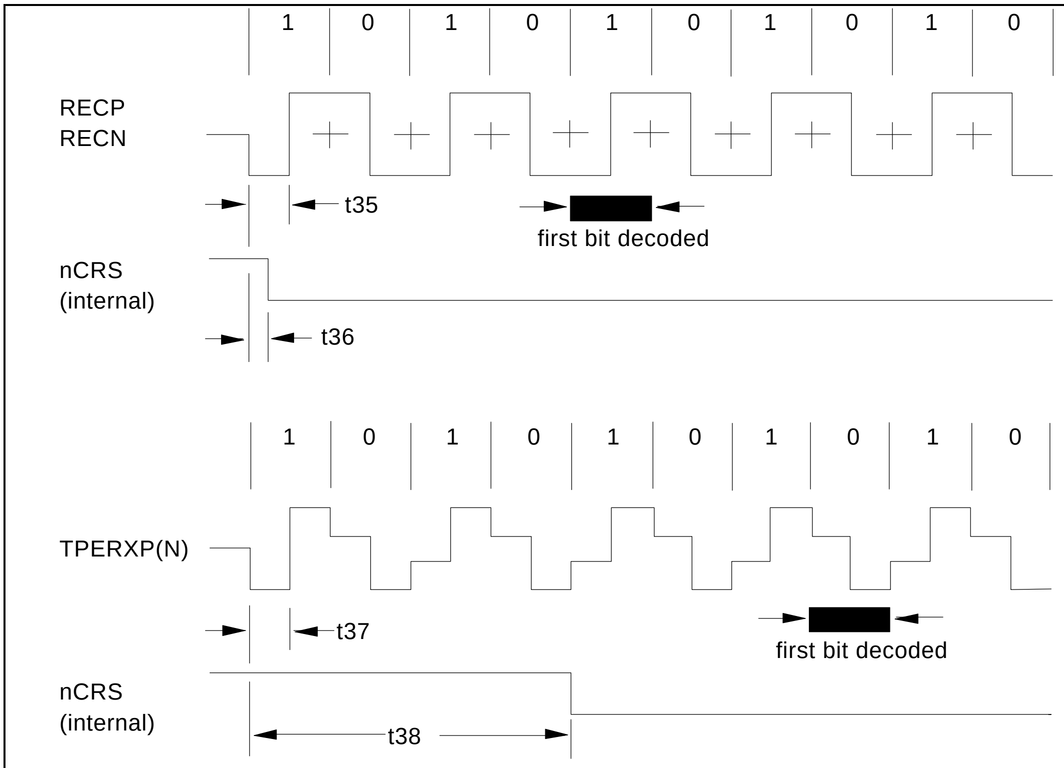
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t23<br>t24 | nCRS, RXD Setup to RXCLK Falling<br>nCRS, RXD Hold after RXCLK Falling | 10<br>30 | | | ns<br>ns |

**FIGURE 41 - EXTERNAL ENDEC INTERFACE - RECEIVE DATA (RXD SAMPLED BY FALLING RXCLK)**



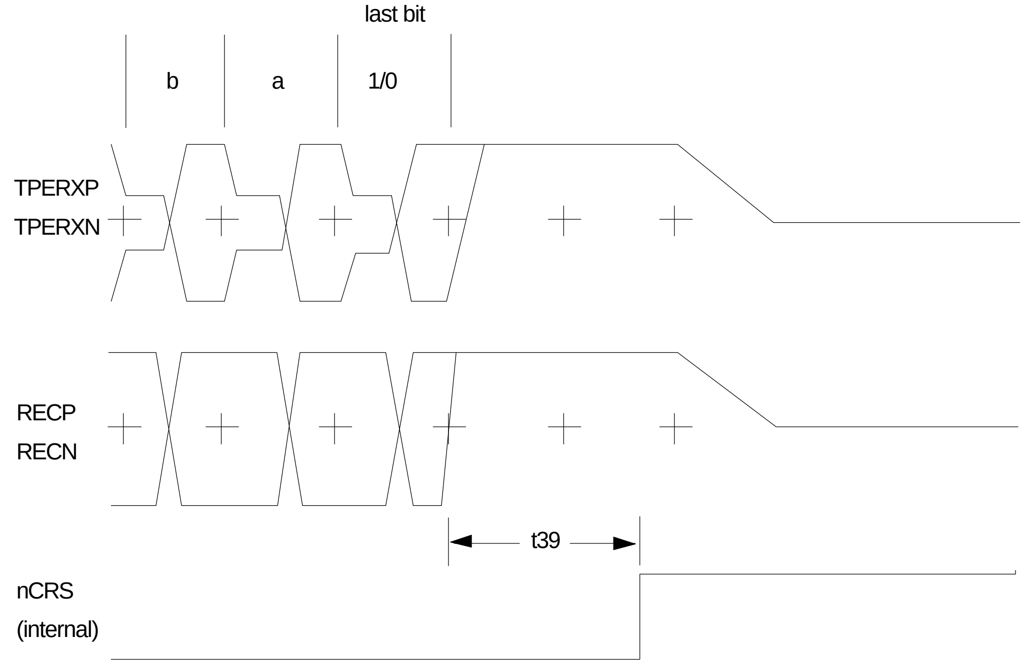
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t31 | TPETXP to TPETXN Skew | -1 | | +1 | ns |
| t32 | TPETXP(N) to TPETXDP(N) Delay | 47 | | 53 | ns |
| t33 | TPETXDN to TPETXDP Skew | -1 | | +1 | ns |
| t34 | TXP to TXN Skew | -1.5 | | 1.5 | ns |

**FIGURE 42 - DIFFERENTIAL OUTPUT SIGNAL TIMING (10BASE-T AND AUI)**



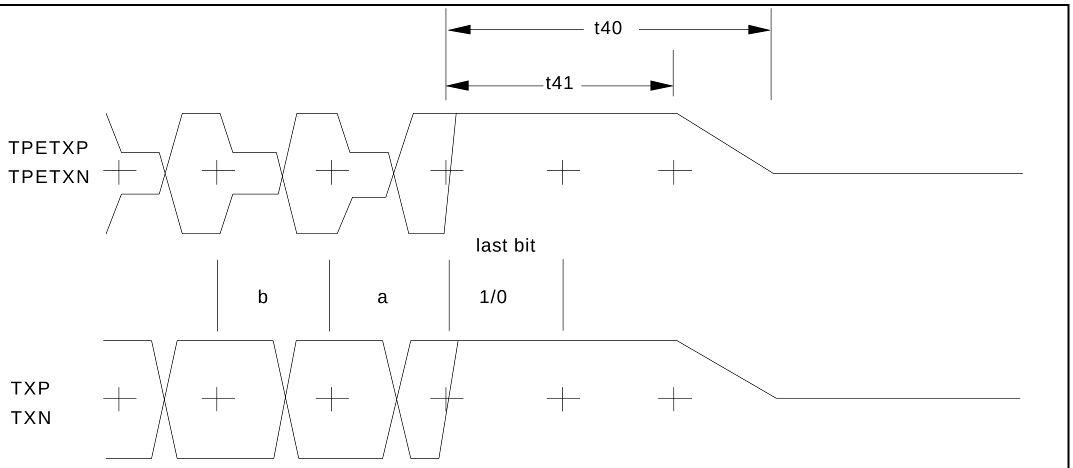
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t35 | Noise Pulse Width Reject (AUI) | 15 | 25 | 30 | ns |
| t36 | Carrier Sense Turn On Delay (AUI) | 50 | 70 | 100 | ns |
| t37 | Noise Sense Pulse Width Reject (10BASE-T) | 15 | 25 | 30 | ns |
| t38 | Carrier Sense Turn On Delay (10BASE-T) | 450 | 500 | 550 | ns |

**FIGURE 43 - RECEIVE TIMING - START OF FRAME (AUI AND 10BASE-T)**



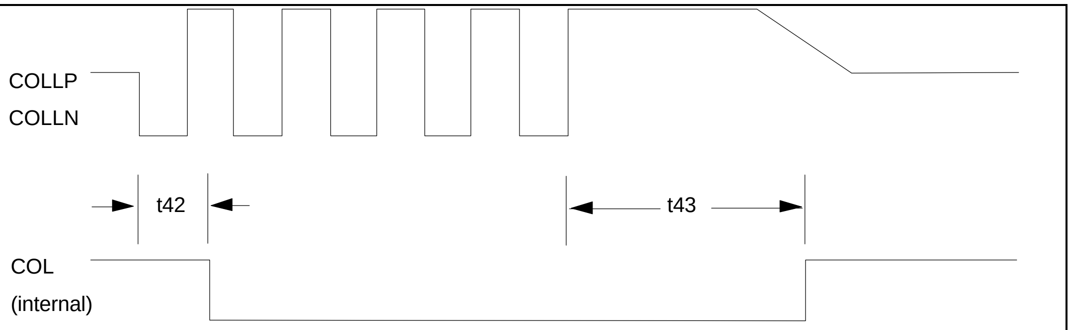
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t39 | Receiver Turn Off Delay | 200 | | 300 | ns |

**FIGURE 44 - RECEIVE TIMING - END OF FRAME (AUI AND 10BASE-T)**



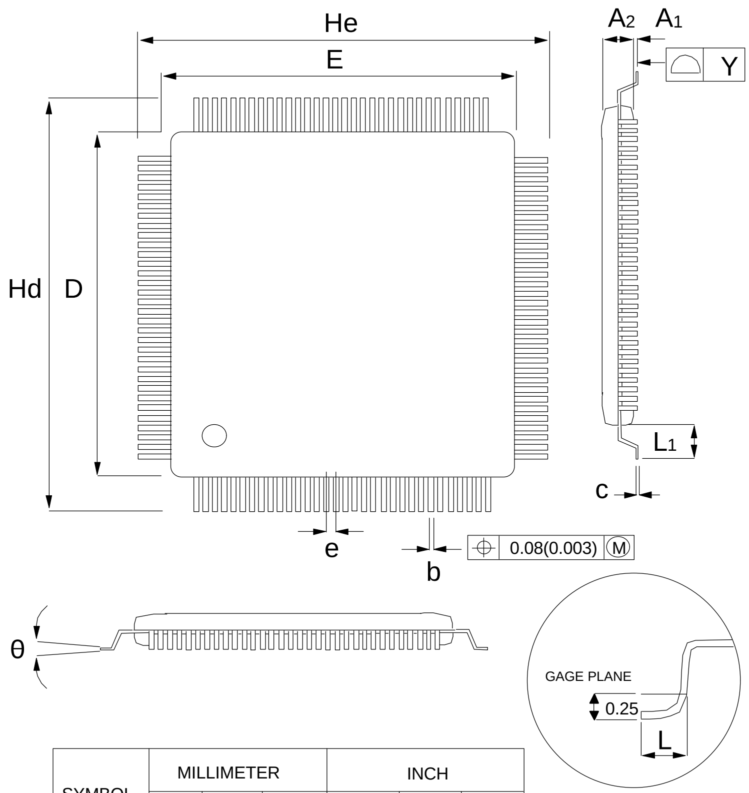
| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t40 | Transmit Output High to Idle in Half-Step Mode | | | 800 | ns |
| t41 | Transmit Output High before Idle in Half-Step Mode | 200 | | | ns |

**FIGURE 45 - TRANSMIT TIMING - END OF FRAME (AUI AND 10BASE-T)**



| | Parameter | min | typ | max | units |
|---|---|---|---|---|---|
| t42 | Collision Turn On Delay | | | 50 | ns |
| t43 | Collision Turn Off Delay | | | 350 | ns |

**FIGURE 46 - COLLISION TIMING (AUI)**



| SYMBOL | MILLIMETER | | | INCH | | |
|---|---|---|---|---|---|---|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| $A_1$ | 0.05 | 0.10 | 0.15 | 0.002 | 0.004 | 0.006 |
| $A_2$ | 0.95 | 1.00 | 1.05 | 0.037 | 0.039 | 0.041 |
| b | 0.13 | 0.18 | 0.23 | 0.005 | 0.007 | 0.009 |
| c | 0.09 | | 0.20 | 0.004 | | 0.008 |
| D | 13.90 | 14.00 | 14.10 | 0.547 | 0.551 | 0.555 |
| E | 13.90 | 14.00 | 14.10 | 0.547 | 0.551 | 0.555 |
| e | | 0.40 | | | 0.016 | |
| $H_d$ | 15.90 | 16.00 | 16.10 | 0.626 | 0.630 | 0.634 |
| $H_e$ | 15.90 | 16.00 | 16.10 | 0.626 | 0.630 | 0.634 |
| L | 0.45 | 0.60 | 0.75 | 0.018 | 0.024 | 0.030 |
| $L_1$ | | 1.00 | | | 0.039 | |
| Y | | | 0.08 | | | 0.003 |
| θ | 0 | | 7 | 0 | | 7 |

**FIGURE 47 - TQFP PACKAGE OUTLINE**

# LAN91C95 ERRATA SHEET

| PAGE | SECTION/FIGURE/ENTRY | CORRECTION | DATE REVISED |
|---|---|---|---|
| 24 | Modem Interface | Section added to data sheet | 7/4/97 |
| 35-40 | Tables 6A - 8B | Tables added to data sheet | 7/4/97 |
| 63 | Refer to table/STSCHG_SLCT Paragraph | See Italicized Text | 7/4/97 |
| 112 | DC Electrical Characteristics | See Italicized Text | 7/4/97 |

smsc™

Circuit diagrams utilizing SMSC products are included as a means of illustrating typical applications; consequently complete information sufficient for construction purposes is not necessarily given. The information has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any licenses under the patent rights of SMSC or others. SMSC reserves the right to make changes at any time in order to improve design and supply the best product possible. SMSC products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of SMSC and further testing and/or modification will be fully at the risk of the customer.