National Semiconductor

**PRELIMINARY**

March 1998

# DP83953 (RIC2A)
# Repeater Interface Controller with Security Features, Internal Drivers and Integrated Filters

## General Description

The DP83953 Repeater Interface Controller with Security Features and Integrated Transmit Filters (RIC2A) is an enhanced version of the DP83952 Repeater Interface Controller with Security Features (RIC II). The RIC2A integrates driver and filter circuitry into the RIC II design.

The functionality of the RIC2A is essentially similar to the RIC II, but the pin definitions have been modified to reflect the added integrated drivers and filters. Additionally, the power and ground pin locations have been rearranged. Therefore, the RIC2A is not a drop in replacement for the RIC II.

The RIC2A is National Semiconductor's managed repeater solution designed to comply with IEEE 802.3 Repeater Specifications. Segment partition and jabber lockup protection state machines are implemented in accordance with this standard. The RIC2A has thirteen network interface ports available, including an AUI compatible port. The AUI port incorporates drivers to connect an external MAU using maximum length cable. Similarly, the other twelve interface ports integrate 10BASE-T transceivers with supporting driver and transmit filter circuitry. (continued)

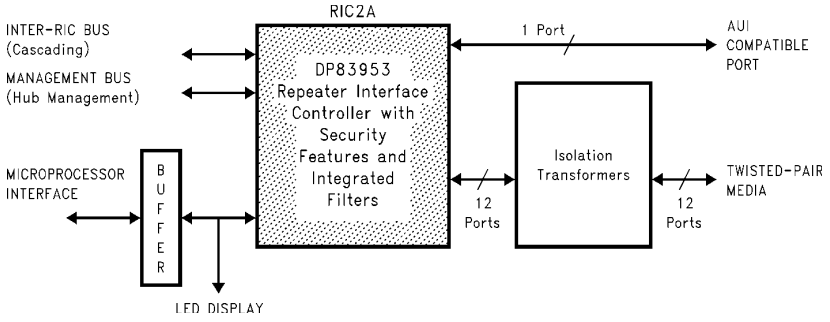
## Features

- Fully compliant with the IEEE 802.3 Repeater Specification
- 12 IEEE 802.3 10BASE-T compatible ports with built-in drivers and analog transmit filters; additional external isolation transformers are required to implement hubs
- 1 IEEE 802.3 compatible AUI port
- Cascadable for larger hub applications
- On chip Elasticity Buffer, Manchester encoder and decoder
- Separate Partition state machines for each port
- Compatible with 802.3k Hub Management requirements
- LED displays to provide port status information, including receive, collision, partition, jabber and link status,
- Power-up configuration options
- Repeater and Partition Specifications, Status Display, Processor Operations
- Simple processor interface for repeater management and port disable.
- On-chip Event Counters and Event Flag Arrays
- Serial Management Bus Interface to combine packet and repeater status information
- Single 5V supply

## The Security Features

- Prevents unauthorized eavesdropping and/or intrusion on a per port basis
- 58 On Chip CAMs (Content Addressable Memory) allow storage of acceptable addresses
- Learn mode automatically records addresses of attached node

## System Diagram



## General Description (continued)

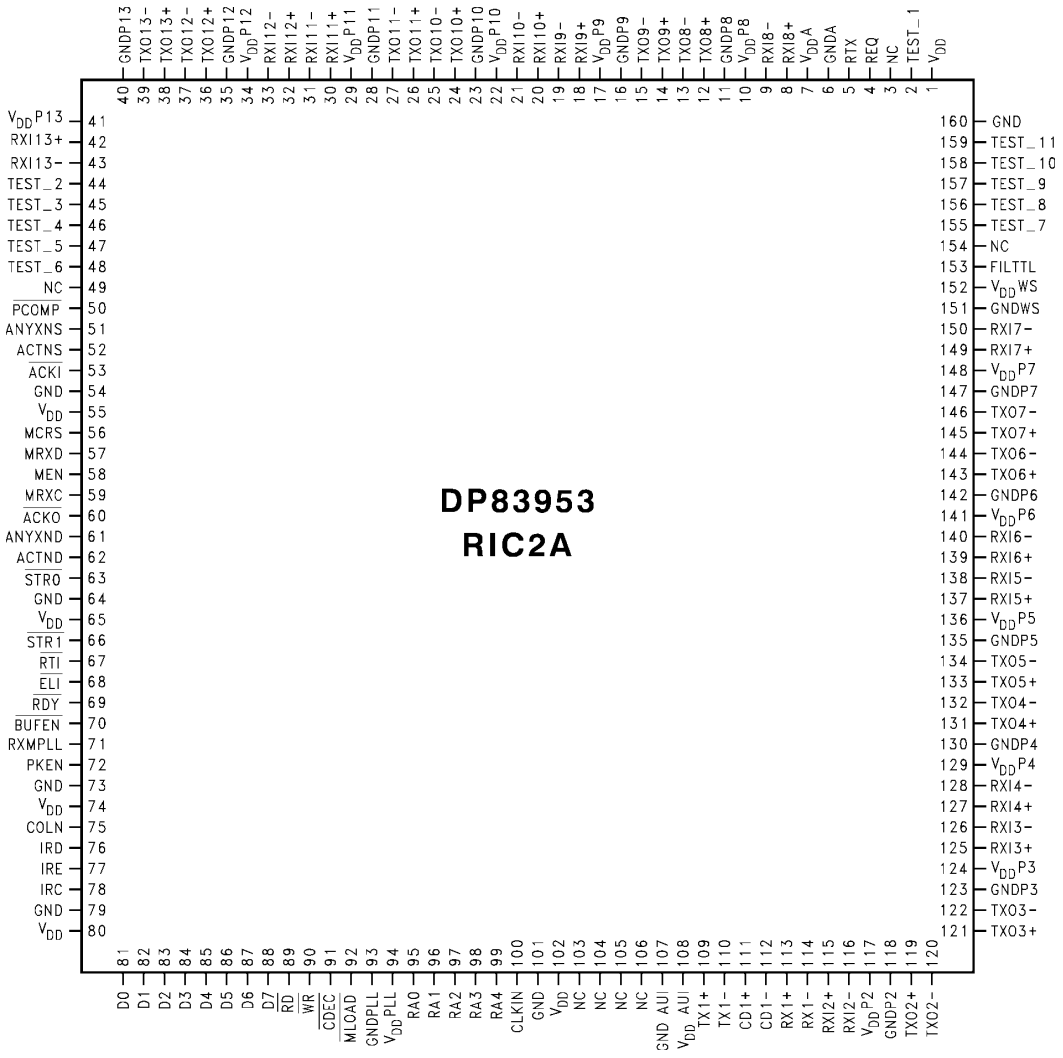
The RIC2A repeater consists of two major functional blocks: The segment specific block and the shared functional blocks. The segment specific block incorporates relevant IEEE specifications on a per port basis. The shared functional blocks incorporate core logic for the entire IEEE repeater unit. The core logic blocks consist of a repeater receive multiplexor, a phase locked loop (PLL), a Manchester decoder, an elasticity buffer, a transmit encoder and a demultiplexor.

A larger repeater system may be constructed by cascading several RIC2A devices via the Inter-RIC bus. This method of cascading allows the RIC2A system to function as a single repeater unit without introducing additional repeater hops.

The RIC2A is configurable for specific applications. It provides port status information for LED array displays and a simple interface for system processors. The RIC2A possesses multifunctional counters and status flag arrays to facilitate network statistics gathering, as well as a serial Hub Management Interface Bus for collecting, event data in managed hub applications.

## Table of Contents

## 1.0 Connection Diagram




**Order Number DP83953VUL**
**NS Package Number VUL160A**

## 1.0 Connection Diagram (Continued)

**12 T.P. Ports + 1 AUI**

| PIN NAME | PIN NO. | PIN NAME | PIN NO. | PIN NAME | PIN NO. | PIN NAME | PIN NO. |
|---|---|---|---|---|---|---|---|
| GNDP13 | 40 | $V_{DD}$ | 80 | TXO2- | 120 | GND | 160 |
| TXO13- | 39 | GND | 79 | TXO2+ | 119 | TEST_11 | 159 |
| TXO13+ | 38 | IRC | 78 | GNDP2 | 118 | TEST_10 | 158 |
| TXO12- | 37 | IRE | 77 | $V_{DD}$P2 | 117 | TEST_9 | 157 |
| TXO12+ | 36 | IRD | 76 | RXI2- | 116 | TEST_8 | 156 |
| GNDP12 | 35 | COLN | 75 | RXI2+ | 115 | TEST_7 | 155 |
| $V_{DD}$P12 | 34 | $V_{DD}$ | 74 | RX1- | 114 | NC | 154 |
| RXI12- | 33 | GND | 73 | RX1+ | 113 | FILTTL | 153 |
| RXI12+ | 32 | PKEN | 72 | CD1- | 112 | $V_{DD}$WS | 152 |
| RXI11- | 31 | RXMPLL | 71 | CD1+ | 111 | GNDWS | 151 |
| RXI11+ | 30 | $\overline{\text{BUFEN}}$ | 70 | TX1- | 110 | RXI7- | 150 |
| $V_{DD}$P11 | 29 | $\overline{\text{RDY}}$ | 69 | TX1+ | 109 | RXI7+ | 149 |
| GNDP11 | 28 | $\overline{\text{ELI}}$ | 68 | $V_{DD}$ AUI | 108 | $V_{DD}$P7 | 148 |
| TXO11- | 27 | $\overline{\text{RTI}}$ | 67 | GND AUI | 107 | GNDP7 | 147 |
| TXO11+ | 26 | $\overline{\text{STR0}}$ | 66 | NC | 106 | TXO7- | 146 |
| TXO10- | 25 | $V_{DD}$ | 65 | NC | 105 | TXO7+ | 145 |
| TXO10+ | 24 | GND | 64 | NC | 104 | TXO6- | 144 |
| GNDP10 | 23 | $\overline{\text{STR0}}$ | 63 | NC | 103 | TXO6+ | 143 |
| $V_{DD}$P10 | 22 | ACTND | 62 | $V_{DD}$ | 102 | GNDP6 | 142 |
| RXI10- | 21 | ANYXND | 61 | GND | 101 | $V_{DD}$P6 | 141 |
| RXI10+ | 20 | $\overline{\text{ACKO}}$ | 60 | CLKIN | 100 | RXI6- | 140 |
| RXI9- | 19 | MRXC | 59 | RA4 | 99 | RXI6+ | 139 |
| RXI9+ | 18 | MEN | 58 | RA3 | 98 | RXI5- | 138 |
| $V_{DD}$P9 | 17 | MRXD | 57 | RA2 | 97 | RXI5+ | 137 |
| GNDP9 | 16 | MCRS | 56 | RA1 | 96 | $V_{DD}$P5 | 136 |
| TXO9- | 15 | $V_{DD}$ | 55 | RA0 | 95 | GNDP5 | 135 |
| TXO9+ | 14 | GND | 54 | $V_{DD}$PLL | 94 | TXO5- | 134 |
| TXO8- | 13 | $\overline{\text{ACKI}}$ | 53 | GNDPLL | 93 | TXO5+ | 133 |
| TXO8+ | 12 | ACTNS | 52 | $\overline{\text{MLOAD}}$ | 92 | TXO4- | 132 |
| GNDP8 | 11 | ANYXNS | 51 | $\overline{\text{CDEC}}$ | 91 | TXO4+ | 131 |
| $V_{DD}$P8 | 10 | $\overline{\text{PCOMP}}$ | 50 | $\overline{\text{WR}}$ | 90 | GNDP4 | 130 |
| RXI8- | 9 | NC | 49 | $\overline{\text{RD}}$ | 89 | $V_{DD}$P4 | 129 |
| RXI8+ | 8 | TEST_6 | 48 | D7 | 88 | RXI4- | 128 |
| $V_{DD}$A | 7 | TEST_5 | 47 | D6 | 87 | RXI4+ | 127 |
| GNDA | 6 | TEST_4 | 46 | D5 | 86 | RXI3- | 126 |
| RTX | 5 | TEST_3 | 45 | D4 | 85 | RXI3+ | 125 |
| REQ | 4 | TEST_2 | 44 | D3 | 84 | $V_{DD}$P3 | 124 |
| NC | 3 | RXI13- | 43 | D2 | 83 | GNDP3 | 123 |
| TEST_1 | 2 | RXI13+ | 42 | D1 | 82 | TXO3- | 122 |
| $V_{DD}$ | 1 | $V_{DD}$P13 | 41 | D0 | 81 | TXO3+ | 121 |

Note 1: NC = No Connect

Note 2: Port $V_{DD}$ and GND are denoted as $V_{DD}$Px and GNDPx, where x=2-13 for all twisted pair ports.

Note 3: The path to each port $V_{DD}$ and GND must have a very low impedance.

## 2.0 Pin Descriptions

| Pin Name | Pin No. | Driver Type | I/O | Description |
|---|---|---|---|---|
| **Network Interface Pins** | | | | |
| RXI2- to RXI13- | | TP | I | Twisted Pair **R**eceive **I**nput Negative |
| RXI2+ to RXI13+ | | TP | I | Twisted Pair **R**eceive **I**nput Positive |
| TXO2- to TXO13- | | TP | O | Twisted Pair **T**ransmit **O**utput Negative |
| TXO2+ to TXO13+ | | TP | O | Twisted Pair **T**ransmit **O**utput Positive |
| FILTTL | 153 | TT | I | **FIL**ter/**TTL**: This pin can be utilized for the PCB diagnostic purposes.<br>**0:** Normal repeater operation<br>**1:** Differential transmit signals change to TTL level +TX and delayed +TX. |
| REQ | 4 | Analog | I | Equalization Resistor: A resistor connected between this pin and GND or $V_{DD}$ adjusts the equalization step amplitude on the Manchester encoded transmit data. Care must be taken to ensure system timing integrity when using cable lengths greater than 100m. The value here is dependent upon board layout. |
| RTX | 5 | Analog | I | Extended Cable Resistor: A resistor connected between this pin and GND or $V_{DD}$ adjusts the amplitude of the differential transmit outputs. Care must be taken to ensure system timing integrity when using cable lengths greater than 100m. The value here is dependent upon board layout. |
| **AUI Port** | | | | |
| CD1+ | 111 | AL | I | AUI **C**ollision **D**etect Input Positive |
| CD1- | 112 | AL | I | AUI **C**ollision **D**etect Input Negative |
| RX1+ | 113 | AL | I | AUI **R**eceive Input Positive |
| RX1- | 114 | AL | I | AUI **R**eceive Input Negative |
| TX1+ | 109 | AD | O | AUI **T**ransmit Output Positive |
| TX1- | 110 | AD | O | AUI **T**ransmit Output Negative |

**TP** = **T**wisted **P**air interface compatible, **TT** = **TT**L compatible, **I** = **I**nput, **O** = **O**utput, Analog = current dependent effect, **AL** = **A**UI **L**evel, **AD** = **A**UI **D**rive

## 2.0 Pin Descriptions (Continued)

| Pin Name | Pin No. | Driver Type | I/O | Description |
|---|---|---|---|---|
| **Processor Bus Pins** | | | | |
| RA0 - RA4 | | TT | I | **R**EGISTER **A**DDRESS INPUTS: These five pins are used to select a register to be read or written. The state of these inputs is ignored when the read, write and Mode Load input strobes are high. (Even under these conditions these inputs must not be allowed to float at an undefined logic state). See text and table for proper Mode Load Operation strapping. |
| $\overline{\text{STR0}}$ | 63 | C | O | DISPLAY UPDATE **STR**OBE 0<br>**Maximum Display Mode:** This signal controls the latching of display data for network ports 1 to 7 into the off chip display latches.<br>**Minimum Display Mode:** This signal controls the latching of display data for the RIC2A into the off chip display latch.<br>During processor access cycles (read or write is asserted) this signal is inactive (high). |
| $\overline{\text{STR1}}$ | 66 | C | O | DISPLAY UPDATE **STR**OBE 1<br>**Maximum Display Mode:** This signal controls the latching of display data for network ports 8 to 13 into the off chip display latches.<br>**Minimum Display Mode:** No operation<br>During processor access cycles (read or write is asserted) this signal is inactive (high). |
| D0 - D7 | | TT | B, Z | **D**ata Bus<br>**Display Update Cycles:** These pins become outputs providing display data and port address information.<br>**Processor Access Cycles:** Data input or output is performed via these pins. The read, write and mode load inputs control the direction of the signals.See text and table for proper Mode Load Operation strapping.<br>Note: The data pins remain in their display update function, i.e., asserted as outputs unless either the read or write strobe is asserted. |
| $\overline{\text{BUFEN}}$ | 70 | C | O | **BUF**FER **EN**ABLE: This output controls the TRI-STATE® operation of the bus transceiver which provides the interface between the RIC2A's data pins and the processor's data bus.<br>Note: The buffer enable output indicates the function of the data pins. When it is high they are performing display update cycles, when it is low a processor access or mode load cycle is occurring. |
| $\overline{\text{RDY}}$ | 69 | C | O | DATA **R**EA**DY** STROBE: The falling edge of this signal during a read cycle indicates that data is stable and valid for sampling. In write cycles the falling edge of $\overline{\text{RDY}}$ denotes that the write data has been latched by the RIC2A. Therefore data must have been available and stable for this operation to be successful. |
| $\overline{\text{ELI}}$ | 68 | C | O | **E**VENT **L**OGGING **I**NTERRUPT: A low level on the $\overline{\text{ELI}}$ output indicates the RIC2A's hub management logic requires CPU attention. The interrupt is cleared by accessing the Port Event Recording register or Event Counter that produced it. All interrupt sources may be masked. |

## 2.0 Pin Descriptions (Continued)

| | | | | |
|---|---|---|---|---|
| $\overline{RTI}$ | 67 | C | O | **R**EAL **T**IME **I**NTERRUPT: A low level on the $\overline{RTI}$ output indicates the RIC2A's real time (packet specific) interrupt logic requires CPU attention. The interrupt is cleared by reading the Real Time Interrupt Status register. All interrupt sources may be masked. |
| $\overline{CDEC}$ | 91 | TT | I | **C**OUNTER **DEC**REMENT: A rising edge on the $\overline{CDEC}$ input strobe decrements all of the RIC2A's Port Event Counters by one. This input is internally synchronized and if necessary the operation of the signal is delayed if there is a simultaneous internally generated counting operation. |
| $\overline{WR}$ | 90 | TT | I | **WR**ITE STROBE: Strobe from the CPU used to write an internal register defined by the RA0 - RA4 inputs. |
| $\overline{RD}$ | 89 | TT | I | **R**EA**D** STROBE: Strobe from the CPU used to read an internal register defined by the RA0 - RA4 inputs. |
| $\overline{MLOAD}$ | 92 | TT | I | DEVICE RESET AND **M**ODE **LOAD:** When this input cycles back up from low to high, all of the RIC2A's state machines, counters and network ports are reset and held inactive. On the rising edge of $\overline{MLOAD}$ the logic levels present on the D0 - 7 pins and RA0 - RA4 inputs are latched into the RIC2A's configuration registers. The rising edge of $\overline{MLOAD}$ also signals the beginning of the display test operation. The clock signal must be present on the CLKIN pin during $\overline{MLOAD}$ assertion and de-assertion. |

**TT** = **TT**L compatible, **B** = **B**i-directional, **C** = **C**MOS compatible, **OD** = **O**pen **D**rain, **I** = **I**nput, **O** = **O**utput,
**Z** = high impedance

## 2.0 Pin Descriptions (Continued)

| Pin Name | Pin No. | Driver Type | I/O | Description |
|---|---|---|---|---|
| **Inter-RIC Bus Pins** | | | | |
| $\overline{\text{ACKI}}$ | 53 | TT | I | **ACK**NOWLEDGE **I**NPUT: Input to the network ports' arbitration chain. |
| $\overline{\text{ACKO}}$ | 60 | TT | O | **ACK**NOWLEDGE **O**UTPUT: Output from the network ports' arbitration chain. |
| IRD | 76 | TT | B,Z | **I**NTER-**RIC D**ATA: When asserted as an output this signal provides a serial data stream in NRZ format. The signal is asserted by a RIC2A when it is receiving data from one of its network segments. The default condition of this signal is to be an input. In this state it may be driven by other devices on the Inter-RIC bus. |
| IRE | 77 | TT | B,Z | **I**NTER-**RIC E**NABLE: When asserted as an output this signal provides an activity framing enable for the serial data stream. The signal is asserted by a RIC2A when it is receiving data from one of its network segments. The default condition of this signal is to be an input. In this state it may be driven by other devices on the Inter-RIC bus. |
| IRC | 78 | TT | B,Z | **I**NTER-**RIC C**LOCK: When asserted as an output this signal provides a clock signal for the serial data stream. Data (IRD) is changed on the falling edge of the clock. The signal is asserted by a RIC2A when it is receiving data from one of its network segments. The default condition of this signal is to be an input. When an input, IRD is sampled on the rising edge of the clock. In this state it may be driven by other devices on the Inter-RIC bus. |
| COLN | 75 | TT | B,Z | **COL**LISION ON PORT **N:** This denotes that a collision is occurring on the port receiving the data packet. The default condition of this signal is to be an input. In this state it may be driven by other devices on the Inter-RIC bus. |
| PKEN | 72 | C | O | **P**AC**K**ET **EN**ABLE: This output acts as an active high enable for an external bus transceiver (if required) for the IRE, IRC IRD and COLN signals. When high the bus transceiver should be transmitting on to the bus, i.e. this RIC2A is driving the IRD, IRE, IRC, and COLN bus lines. When low the bus transceiver should receive from the bus. |
| CLKIN | 100 | TT | I | 40 MHz **CL**OC**K IN**PUT: This input is used to generate the RIC2A's timing reference for the state machines, and phase lock loop decoder. |
| ACTND | 62 | OD | O | **ACT**IVITY ON PORT **N D**RIVE: This output is active when the RIC2A is receiving data or collision information from one of its network segments. |
| ACTNS | 52 | TT | I | **ACT**IVITY ON PORT **N S**ENSE: This input senses when this or another RIC2A in a multi-RIC2A system is receiving data or collision information. |
| ANYXND | 61 | OD | O | **A**CTIVITY ON A**NY** PORT E**X**CLUDING PORT **N D**RIVE: This output is active when a RIC2A is experiencing a transmit collision or multiple ports have active collisions on their network segments. |
| ANYXNS | 51 | TT | I | **A**CTIVITY ON A**NY** PORT E**X**CLUDING PORT **N S**ENSE: This input senses when this RIC2A or other RIC2As in a multi-RIC2A system are experiencing transmit collisions or multiple ports have active collisions on their network segments. |

**TT** = **TT**L compatible, **B** = **B**i-directional, **C** = **C**MOS compatible, **OD** = **O**pen **D**rain, **I** = **I**nput, **O** = **O**utput,
**Z** = high impedance,

## 2.0 Pin Descriptions (Continued)

| Pin Name | Pin No. | Driver Type | I/O | Description |
|---|---|---|---|---|
| **Management Bus Pins** | | | | |
| MRXC | 59 | TT | O,Z | **M**ANAGEMENT **R**ECEIVE **C**LOCK: When asserted this signal provides a clock signal for the MRXD serial data stream. The MRXD signal is changed on the falling edge of this clock. The signal is asserted when a RIC2A is receiving data from one of its network segments. Otherwise the signal is inactive. |
| MCRS | 56 | TT | B,Z | **M**ANAGEMENT **C**ARRIE**R S**ENSE: When asserted this signal provides an activity framing enable for the serial data stream. The signal is asserted when a RIC2A is receiving data from one of its network segments. Otherwise the signal is an input. |
| MRXD | 57 | TT | O,Z | **M**ANAGEMENT **R**ECEIVE **D**ATA: When asserted this signal provides a serial data stream in NRZ format. The data stream is made up of the data packet and RIC2A status information. The signal is asserted when a RIC2A is receiving data from one of its network segments. Otherwise the signal is inactive. |
| MEN | 58 | C | O | **M**ANAGEMENT BUS OUTPUT **EN**ABLE: This output acts as an active high enable for an external bus transceiver (if required) for the MRXC, $\overline{\text{MCRS}}$ and MRXD signals. When high the bus transceiver should be transmitting on to the bus. |
| $\overline{\text{PCOMP}}$ | 50 | TT | I | **P**ACKET **COMP**RESS: This input is used to activate the RIC2A's packet compress logic. A low level on this signal when $\overline{\text{MCRS}}$ is active will cause that packet to be compressed. If $\overline{\text{PCOMP}}$ is tied low all packets are compressed, if $\overline{\text{PCOMP}}$ is tied high packet compression is inhibited. |
| **External Decoder Pins** | | | | |
| RXMPLL | 71 | TT | O | **R**ECEVE DATA MANCHESTER FORMAT: This output makes the data, in Manchester format, received by port N available for test purposes. If not used for testing this pin should be left open. |
| **Test Pins** | | | | |
| TEST_(12:7) | 154-159 | TT | I | Factory test control pins - this pin should be connected to GND for proper operation of the repeater. |
| TEST_(6:2) | 44-48 | TT | I | Factory test control pins - this pin should be connected to GND for proper operation of the repeater. |
| TEST_1 | 2 | TT | O | Factory test control pins - this pin should be left unconnected for proper operation of the repeater. |
| **Power and Ground Pins** | | | | |
| $V_{DD}$ | 1, 55, 65, 74, 80, 102 | | | Positive Supply |
| GND | 54, 64, 73, 79, 101, 160 | | | Negative Supply |
| $V_{DD}A$ | 7 | | | Positive Supply for Analog circuitry |
| GNDA | 6 | | | Negative Supply for Analog circuitry |
| $V_{DD}PLL$ | 94 | | | Positive Supply for Phase Lock Loop |
| GNDPLL | 93 | | | Negative Supply for Phase Lock Loop |
| $V_{DD}WS$ | 152 | | | Positive Supply for Wave Shape circuitry |
| GNDWS | 151 | | | Negative Supply for Wave Shape circuitry |
| $V_{DD}$ $P_n$ | 10, 17, 22, 29, 34, 41, 117, 124, 129, 136, 141, 148 | | | Positive supply for port N. Connect for all ports. |
| GND $P_n$ | 11, 16, 23, 28, 35, 40, 118, 123, 130, 135, 142, 147 | | | Negative supply for port N. Connect for all ports. |
| $V_{DD}$ AUI | 108 | | | Positive supply for AUI port. |
| GND AUI | 107 | | | Negative supply for AUI port. |

**TT** = **TT**L compatible, **B** = **B**i-directional, **C** = **C**MOS compatible, **OD** = **O**pen **D**rain, **I** = **I**nput, **O** = **O**utput,
**Z** = high impedance

## 3.0 Block Diagrams

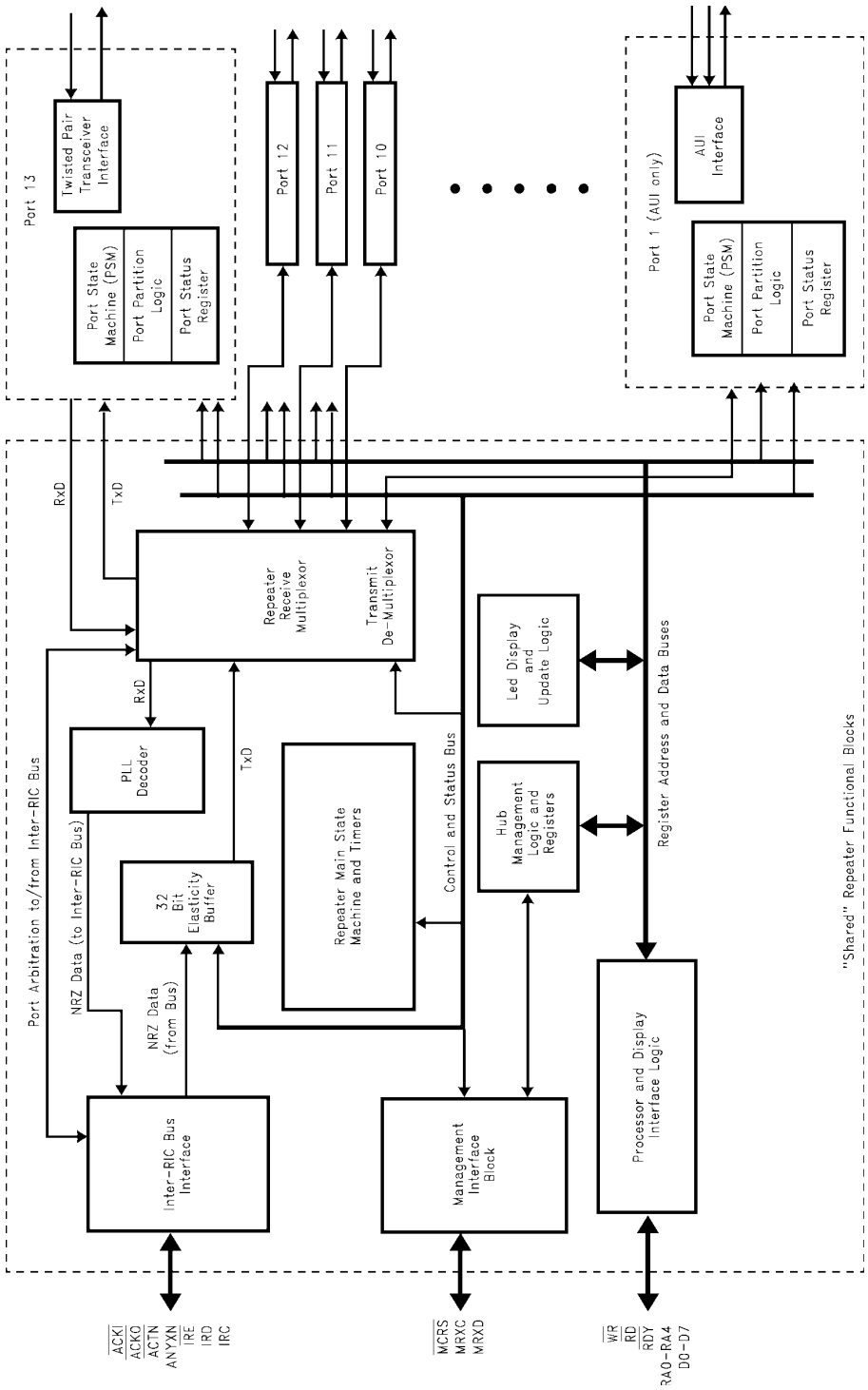



**Figure 1. Shared Repeater and Segment Functional Blocks**

## 3.0 Block Diagrams (Continued)

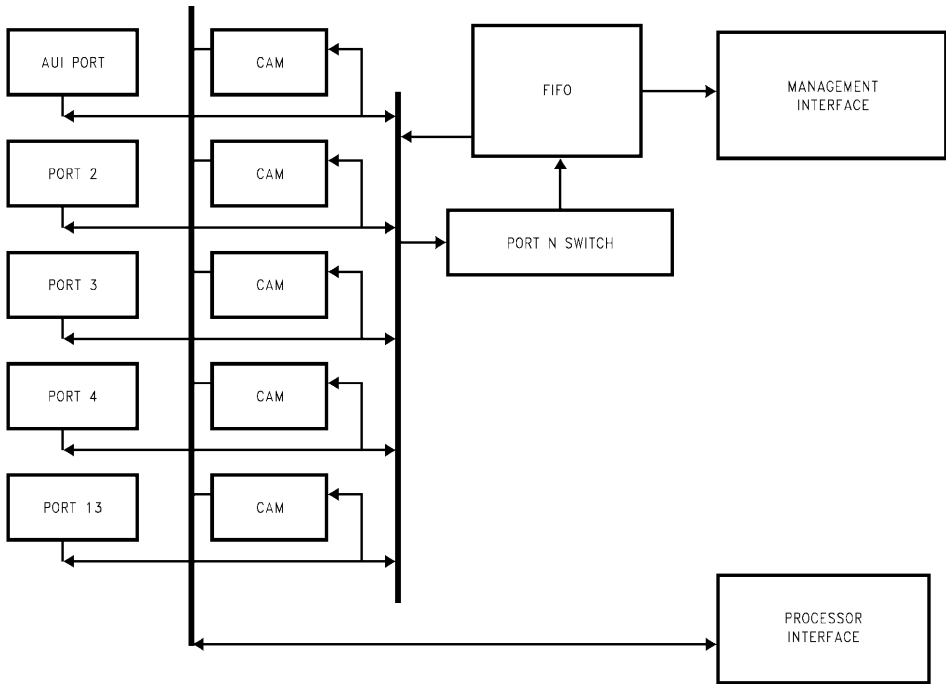



**Figure 2. RIC2A Port Architecture Security Block Diagram**

Note: The block diagram for the RIC2A, when used in the non-secure mode, is identical to the “shared” repeater functional block diagram.(*Figure 1*). But, in secure mode, additional security logic is used when operating the device (*Figure 2*)

# 4.0 Functional Description

The IEEE 802.3 repeater specification delineates the functional criteria that all compliant repeater systems must adhere to. An implementation of these requirements strongly suggest a multiport modular design style. In such a design, functionality is split between those tasks common to all data channels and those exclusive to each individual channel. The RIC2A follows this approach. Certain functional blocks are replicated for each network attachment, (also known as a repeater port), and others are shared.

The following subsections provide an overview of the RIC2A architecture. First, RIC2A feature enhancements from the RIC II is discussed. Then, the RIC2A functional blocks are described.

## 4.1 Summary of DP83953 RIC2A Feature Enhancements from DP83952 RIC II

1. The DP83953 RIC2A integrates transmit filters and drivers on a per port basis. These additional features allow a system developer to add little more than external isolation transformers in order to build fully secured/managed hub products.
2. The functionality of the DP83953 is essentially similar to DP83952, but some of the pin definitions have been modified to reflect the new integrated drivers and transmit filters. Therefore, the RIC2A is *not a drop in replacement for the RICII*. Additionally, power and ground pin locations have been rearranged to accommodate additional pins.
3. Integrated network port drivers provide controlled rise and fall time output signals. These port drivers will facilitate EMI compliance without procuring additional components.
4. Compared to DP83952, the DP83953 requires additional current drive. The additional current is required to implement the new integrated drivers and transmit filters.

## 4.2 Overview Of RIC2A Functions

**Segment Specific Block: Network Port**

As shown in the block diagram, the segment specific blocks consist of

A logic section and a physical layer interface section. The logic block is required for performing repeater operations upon that particular segment. It is known as the “port” logic since it is the access “port” the segment has to the rest of the network.

This function is repeated 13 times in the RIC2A (one for each port) and is shown on the right side of the block diagram, Figure 1.

The physical layer interface depends upon the port. Port 1 has an AUI compliant interface for use with AUI compatible transceiver boxes and cable. Ports 2 to 13 are twisted pair ports.

The four distinct functions inside the port logic block are:

1. The Port State Machine "PSM" performs data and collision repetition as described by the IEEE repeater specification. For example, it may determine if this port should be receiving from or transmitting to a particular network segment.
2. The Port Partition Logic implements the segment partitioning algorithm. This algorithm is defined by the IEEE specification and is used to protect the network from malfunctioning segments.
3. The Port Status Register reflects the current status of the port. The system processor may access this register to obtain port status information or to configure certain port options, such as port disable.
4. The Port security configuration logic determines if the transmitted or received packet will contain intact or pseudo random data. This logic consists of two dedicated CAM locations per port for learning, storing, and comparing port source addresses.

**Shared Functional Blocks: Repeater Core Logic**

The shared functional blocks consists of the repeater Main State Machine (MSM), Timers, a 32 bit Elasticity Buffer, PLL Decoder, Receive and Transmit Multiplexors, and Security Logic with 32 shareable CAM locations. These blocks perform the majority of the operations needed to fulfill the requirements of the IEEE repeater specification.

When a packet is received by a port it is then sent via the Receive Multiplexor to the PLL Decoder. Notification of the data and collision status is sent to the MSM through the receive multiplexor and collision activity status signals. This enables the MSM to determine the source and type of data to be transmitted and eventually repeated to all ports. This information may be valid data or the jam pattern.

According to the IEEE repeater specifications, after a collision has been determined, the transmit data will be replaced with a jam pattern consisting of a alternating ones and zeros. (e.g. 1010...) for at least 96 bit times. If a collision occurs during the preamble, the address field, the type field, or the data field the RIC2A will immediately switch to the jam pattern to be transmitted to all ports.

If the RIC2A is configured for the "non-secure" mode, the valid received data is transmitted to all of the other ports, except the port which is receiving the packet.

If the RIC2A is configured for the "secure" mode, the source and destination addresses within each packet are first checked against the addresses of the local and shared CAMs assigned to the port. Based on this comparison, and the port configuration will be either:

1. A pseudo random bit pattern may be generated in the data field of the designated “secure” packet and then transmitted to their respective port(s). Or,
2. The received data may be transmitted intact.

The data always remains intact on the Inter-RIC bus (IRB to be described later) to allow any cascaded repeaters to compare all destination addresses with their local CAMs. In

## 4.0 Functional Description (Continued)

the case of a source address mismatch, the RIC2A will immediately switch to a random bit pattern on both the local transmitting ports and the IRB.

The main state machine operates in conjunction with a series of counter timers. These timers will ensure that all associated IEEE specification times (referred to as the TW1 to TW6 times) are met.

An IEEE repeater system must meet the same signal jitter performance as any other receiving node attached to a network segment. Consequently, a phase locked loop Manchester decoder is required so that the packet may be decoded, and the jitter accumulated over the receiving segment recovered. The decode logic outputs data in non return to zero (NRZ) format with an associated clock and enable. This format allows the packet to be conveniently transferred to other attached devices, such as network controllers and other repeaters through the Inter-RIC bus (IRB). The data may then be re-encoded into Manchester data and transmitted.

During reception and/or transmission through the physical layer transceivers a loss of bits in the preamble field of a packet may occur. This loss must be replaced according to the IEEE repeater specification. To accomplish this, an elasticity buffer is employed to restore a full length preamble upon transmission.

### The Sequence of Operation

Soon after the network segment receiving a packet has been identified, the RIC2A will transmit the preamble pattern (1010...) to all other network segments. While the preamble is being transmitted, the elasticity buffer will monitor the decoded received clock and data signals via the Inter-RIC bus (IRB). When the start of frame delimiter (SFD) is detected, the received data stream will be written into the elasticity buffer. The removal of stored data from the elasticity buffer for re-transmission is not allowed until a valid length preamble pattern has been transmitted.

### Internal CAMs

To implement the security features, the RIC2A uses two sets of Content Addressable Memory (CAMs) for address comparison: port CAMs, and shared CAMs.

#### Port CAMs

The RIC2A provides two CAM locations (48 bits wide) per port for comparison. The two CAM locations contain source address(es) for incoming packets on their respective ports. The addresses can be stored (CPU access) or learned (Learn Mode). While in learning mode, LME=1, external processor access is not advised or allowed, since the contents of the two CAM registers may not be valid. Once the addresses are learned, they are used to make comparisons between the source and destination addresses. An address can only be learned when a packet has been received with a valid CRC. External processor/logic access to these registers is fine while learning is not in progress, LME=0 in the port security configuration register.

#### Shared CAMs

The RIC2A provides thirty-two shareable CAM locations (48 bits wide) to store Ethernet addresses associated with the ports. The Ethernet addresses are stored by writing to these CAM locations where the addresses could be shared among the thirteen ports. By using shared CAMs, multiple Ethernet addresses can be associated with a single port, or multiple ports can be allocated to a single Ethernet address. After the destination address of the received packet is completely buffered, the RIC2A will compare this address with the stored addresses in the CAM locations. The source address is compared in a similar fashion. These shared CAM locations are user defined only, and will not be filled in learning mode.

A CAM entry could be shared among the thirteen local ports. This is done through a 16-bit CAM Location Mask Register (CLMR). For each CAM entry there is only one CLMR, therefore there are 32 registers for the 32 CAM entries.

Since register access is performed on a byte basis, six write cycles must be completed to program the Ethernet address into the CAM. The upper three bits of the CAM Location Mask Register (CLMR) act as a pointer indicating which byte of the 6-byte address will be accessed next. This pointer will increment every time a read or write cycle is completed to the CAM entry. The pointer starts at 1, indicating the least significant byte of the address.

Four additional registers are provided to validate the 32 shared CAM entries and are referred to as the Shared CAM Validation Registers 1-4 (SCVR 1-4, Page 9H, Address 16-19H). Each bit of the SCVR is mapped to one CAM location. An address in the CAM location will only be valid when a corresponding bit Address Valid (ADV bit) has been set in this register. The RIC2A will include only valid CAM locations for address comparison.

The contents of all CAM locations are unknown at power up. This is not a problem since corresponding Address Valid (ADV) bits are not set for each CAM. Therefore, comparisons will not take place with the CAM contents.

#### Inter-RIC Bus (IRB) Interface

A RIC2A based repeater system may be constructed to support many more network attachments than those available through a single chip. The split functions described earlier, allow data packets and collision status to be transferred between multiple RIC2As while allowing the system to function as a single logical repeater. Since all RIC2As in a multiple RIC2A system are identical and capable of performing all repetition functions, the failure of one RIC2A will not cause a failure of the entire system. This is an important issue, especially with respect to large multi-port repeaters.

In a multi-RIC2A system, the RIC2As can communicate through a specialized interface known as the Inter-RIC bus(IRB). This bus allows the data packet to be transferred from the receiving RIC2A to other RIC2As in the system. Each RIC2A then transmits the datastream to its segments.

## 4.0 Functional Description (Continued)

The notification of collisions occurring across the network is just as important as data transfers. The Inter-RIC bus has a set of status lines capable of conveying collision information between RIC2As in order to ensure that their main state machines operate in the appropriate manner.

### LED Interface and Hub Management Function

Repeater systems usually possess optical displays indicating network activity and the status of specific repeater operations. The display update block of the RIC2A can provide the system designer with a wide variety of indicators. The updates are completely autonomous and merely require SSI logic to drive the display devices, usually made up of light emitting diodes (LEDs). The status display is very flexible and allows the user to choose appropriate indicators for the specification of his equipment.

The RIC2A was designed for those interested in implementing large repeaters with hub management capabilities. Hub management uses the unique position of repeaters in a network to gather statistics about the network segments they are attached to. Important events are gathered by the management block from logic blocks throughout the chip. These events are then stored in on-chip latches, or counted in on-chip counters according to the developer's supplied latching and counting masks.

The fundamental task of a hub management system implementation is to associate the current packet and any management status information with the network segment. An example could be keeping track of packets received on a repeater's ports. An ideal system would place the combined data packet and status field in system memory for future examination by hub management software. The main function of the RIC2A's hub management support logic is to provide this function.

To accomplish this task, the RIC2A utilizes a dedicated hub management interface. This is similar to the Inter-RIC bus since it allows the data packet to be recovered from the receiving RIC2A. Unlike the Inter-RIC bus, the intended recipient is not another RIC2A, but typical National Semiconductor's DP83932 ("SONIC™"). This dedicated bus allows a management status field to be appended at the end of each packet without affecting the operation of the entire repeater system.

In addition to the counters provided on the RIC DP83950B, the RIC2A implements thirteen more (8 bit wide) counters. These counters will count events specified in the Event Count and Interrupt Mask Register2 (ECIMR2). These include items such as Frame Check Sequences, Frame Alignment Errors, Partitions, and Out of Window Collisions. This register also includes "Reset On Read" and "Freeze When Full" control bits.

It should be noted that Counter Decrement (CDEC) will not be used with the ECMR2. Also, real time or event logging interrupts (RTI or ELI) will not be generated for this register.

#### Processor Interface

The RIC2A's processor interface uses an octal bi-directional data bus in order to interface to a system processor. The RIC2A has on-chip registers to indicate the status of the hub management functions, chip configuration, and port status. These registers are accessed by placing the respective address at the Register Address (RA4 - RA0) input pins.

Display update cycles and processor accesses occur utilizing the same data bus. An on-chip arbiter in the processor/display block schedules and controls the accesses and ensures the correct information is written into the display latches. During the display update cycles the RIC2A behaves as a master of its data bus. This is the default state of the data bus. Consequently, a TRI-STATE buffer must be placed between the RIC2A and the system processor's data bus. This ensures that bus contention problems are avoided during simultaneous display update cycles and processor accesses of other devices on the system bus. When the processor accesses a RIC2A register, the RIC2A enables the data buffer and selects the operation, either input or output, of the data pins.

For faster register accesses, the RIC2A provides the added feature of disabling display update cycles. Setting the Disable LED Update bit, DLU in the Lower Event Information register (Page 1H, Address 1FH) stops the RIC LED update cycles. This disables the shared mode of the data bus, leaving the RIC2A in slave access mode. In this mode, the maximum read/write cycle time is reduced to approximately 400 ns.

### 4.3 Description Of Repeater Operations

In order to implement a multi-chip repeater system that behaves as a single logical repeater, special considerations must be taken into account with respect to the data path used for packet repetition. For example, we must consider where in the data path specific operations such as Manchester decoding and elasticity buffering are performed. Additionally, the system state machines, which utilize available network activity signals, must accommodate various packet repetition and collision scenarios according to the IEEE repeater specification.

The RIC2A contains two types of interacting state machines. They are:

1. Port State Machines (PSMs) Every network attachment has its own PSM.
2. Main State Machine (MSM) This state machine controls the shared functional blocks as shown in the block diagram Figure 1.

## 4.0 Functional Description (Continued)

### Repeater Port and Main State Machines

The Port and Main State Machines are described with terminology used in the IEEE Repeater specification. For a detailed explanation of terms, please refer to that specification. References made to repeater states or terms described in the IEEE specification will be shown in italics. Figure 3 shows the Inter-RIC Bus State Diagram and Figure 4 shows the IEEE State Diagram.

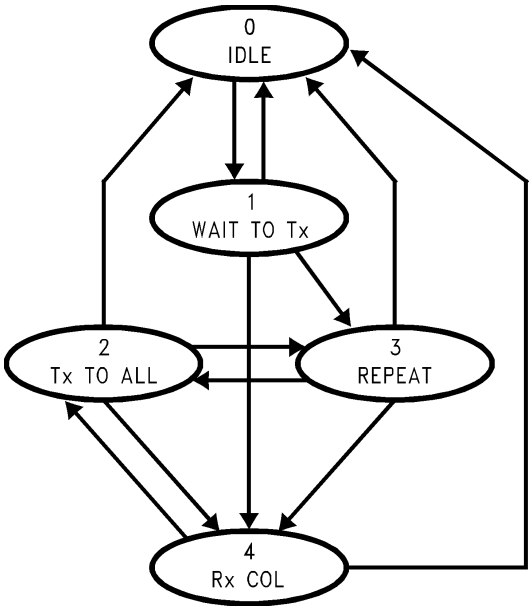



**Figure 3. Inter-RIC bus State Diagram**

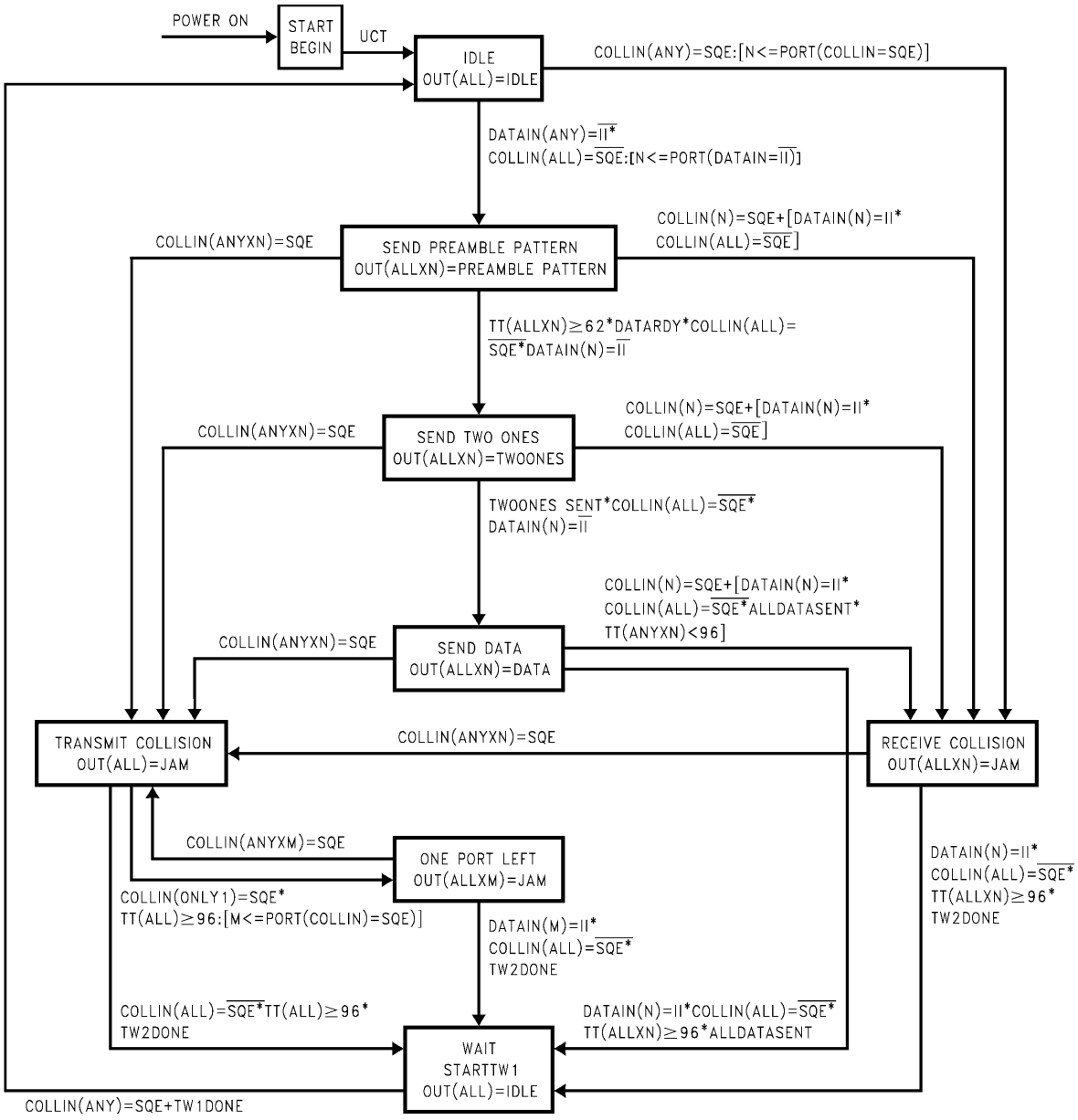



**Figure 4. IEEE Repeater Main State Diagram**

## 4.0 Functional Description (Continued)

### The Port State Machine (PSM)

The two primary functions of the PSM are to:

1. Control the transmission of repeated data, pseudo random data, and jam signals over the attached segments.
2. Determine if a port will be the source of data or collision information to be repeated over the network. This repeater port is known as *PORT N*. An arbitration process is required to enable the repeater to transition from an *IDLE* state to the *SEND PREAMBLE PATTERN* or *RECEIVE COLLISION* states. (See Figure 4) This process is used to locate the port that will be *PORT N* for that particular packet.The data received from *PORT N* is directed to the PLL decoder and transmitted over the Inter-RIC bus. If a collision occurs, the repeater enters the *TRANSMIT COLLISION* state. Then a subsequent arbitration operation is performed in order to determine which port is *PORT M*. *PORT M* is differentiated from the repeater's other ports if the repeater enters the *ONE PORT LEFT* state. In this state *PORT M* does not transmit to its segment. At that time, all other ports are still required to transmit to their segments.

### The Main State Machine (MSM)

The MSM controls the operation of the shared functional blocks in each RIC2A as shown in the block diagram, Figure 1, and performs the majority of the data and collision operations as defined by the IEEE specification.

| Function | Action |
|---|---|
| Preamble Regeneration | Restore the length of the preamble pattern to the defined size. |
| Fragment Extension | Extend received data or collision fragments to meet the minimum fragment length of 96 bits. |
| Elasticity Buffer Control | A portion of the received packet may require storage in an Elasticity Buffer to accommodate preamble regeneration. |
| Jam / Preamble Pattern Generation | In cases of receive or transmit collisions, a RIC2A is required to transmit a jam pattern (1010...).<br>Note: This pattern is the same as that used for preamble regeneration. |
| Transmit Collision Enforcement | The *TRANSMIT COLLISION* state requires a repeater to remain in this state for at least 96 bit times. |
| Data Encoding Control | NRZ formatted data in the elasticity buffer must be encoded into Manchester formatted data prior to re-transmission. |
| Tw1 Enforcement | Enforce the Transmit Recovery Time specification. |
| Tw2 Enforcement | Enforce Carrier Recovery Time specification on all ports with active collisions. |

The interaction of the main and port state machines is visible, in part, through the Inter-RIC bus.

### Inter-RIC Bus (IRB) Operation

### Overview

The IRB consists of eight signals. These signals implement a protocol that may be used to connect multiple RIC2As together. In this configuration, the logical function of a single repeater is maintained. The resulting multi- RIC2A repeater system is compliant with the IEEE 802.3 Repeater Specification and may even encompass several hundred network segments. Figure 5 shows an example of a multiport RIC2A system.

## 4.0 Functional Description (Continued)

The IRB connects multiple RIC2As to realize the following operations:

*Port N* Identification (which port the repeater receives data from)

*Port M* Identification (which port last experienced a collision)

Data Transfer

*RECEIVE COLLISION* identification

*TRANSMIT COLLISION* identification

*DISABLE OUTPUT* (jabber protection)

The following table briefly describes the operation of the Inter-RIC bus signals, the conditions required for a RIC2A to assert a signal, and which RIC2As (in a multi-RIC2A system) would monitor the signal.

| Inter-RIC Bus Signal | Function | Conditions Required for a RIC2A to Drive this Signal | RIC2A Receiving the Signal |
|---|---|---|---|
| $\overline{\text{ACKI}}$ | Input signal to The PSM arbitration chain. This chain is employed to identify *PORT N* and *PORT M*.<br>Note: A RIC2A which contains *PORT N* or *PORT M* may be identified by its $\overline{\text{ACKO}}$ signal being low when its $\overline{\text{ACKI}}$ input is high. | Not applicable | This is dependent upon the method used to cascade RIC2As, described in a following section. |
| $\overline{\text{ACKO}}$ | Output signal from the PSM arbitration chain. | If this RIC2A contains port N, then the device will assert this signal. | This is dependent upon the method used to cascade RIC2As, described in a following section. |
| ACTN | This signal denotes there is activity on *PORT N* or *PORT M.* | A RIC2A must contain *PORT N* or *PORT M.*<br>Note: Although this signal normally has only one source asserting the signal active it is used in a wired-or configuration. | The signal is monitored by all RIC2As in the repeater system. |
| ANYXN | This signal denotes that a repeater port that is not *PORT N* or *PORT M* is experiencing a collision. | Any RIC2A which satisfies the above condition.<br>Note: This bus line is used in a wired-or configuration. | The signal is monitored by all RIC2As in the repeater system. |
| COLN | Denotes *PORT N* or *PORT M* is experiencing a collision. | A RIC2A must contain *PORT N* or *PORT M.* | The signal is monitored by all other RIC2As in the repeater system. |
| IRE | This signal acts as an activity framing signal for the IRC and IRD signals. | A RIC2A must contain *PORT N.* | The signal is monitored by all other RIC2As in the repeater system. |
| IRD | Decoded serial data, in NRZ format, received from the network segment attached to *PORT N.* | A RIC2A must contain *PORT N.* | The signal is monitored by all other RIC2As in the repeater system. |
| IRC | Clock signal associated with IRD and IRE. | A RIC2A must contain *PORT N.* | The signal is monitored by all other RIC2As in the repeater system. |

## 4.0 Functional Description (Continued)

**Methods of RIC2A Cascading**

In order to build multi-RIC2A repeaters, *PORT N* and *PORT M* identification must be performed across all the RIC2As in the system. Inside each RIC2A, the PSMs are arranged in a logical arbitration chain where port 1 is the highest and port 13 the lowest. The top of the chain, the input to port 1 is accessible to the user via the RIC2A's $\overline{\text{ACKI}}$ input pin. The output from the bottom of the chain becomes the $\overline{\text{ACKO}}$ output pin. In a single RIC2A system *PORT N* is defined as the highest port in the arbitration chain with receive or collision activity. *Port N* identification is performed when the repeater is in the *IDLE* state. *PORT M* is defined as the highest port in the chain with a collision when the repeater leaves the *TRANSMIT COLLISION* state. In order for the arbitration chain to function, all that needs to be done is to tie the $\overline{\text{ACKI}}$ signal to a logic high state. In multi-RIC2A systems there are two methods to propagate the arbitration chain between RIC2As:

The first and most straight forward is to extend the arbitration chain by daisy chaining the $\overline{\text{ACKI}}$ $\overline{\text{ACKO}}$ signals between RIC2As. In this approach one RIC2A is placed at the top of the chain (its $\overline{\text{ACKI}}$ input is tied high), then the $\overline{\text{ACKO}}$ signal from this RIC2A is sent to the $\overline{\text{ACKI}}$ input of the next RIC2A and so on. This arrangement is simple to implement but it places some topological restrictions upon the repeater system. In particular, if the repeater is constructed using a backplane with removable printed circuit boards. (These boards contain the RIC2As and their associated components.) If one of the boards is removed then the $\overline{\text{ACKI}}$ $\overline{\text{ACKO}}$ chain will be broken and the repeater will not operate correctly.

The second method of *PORT N or M* identification avoids this problem. This second technique relies on an external parallel arbiter which monitors all of the RIC2As' $\overline{\text{ACKO}}$ signals and responds to the RIC2A with the highest priority. In this scheme each RIC2A is assigned with a priority level. One method of doing this is to assign a priority number which reflects the position of a RIC2A board on the repeater backplane, i.e., its slot number. When a RIC2A experiences receive activity and the repeater system is in the *IDLE* state, the RIC2A board will assert $\overline{\text{ACKO}}$. External arbitration logic drives the identification number onto an arbitration bus and the RIC2A containing *PORT N* will be identified. An identical procedure is used in the *TRANSMIT COLLISION* state to identify *PORT M.* Parallel arbitration is not subject to the problems caused by missing boards, i.e., empty slots in the backplane. The logic associated with asserting this arbitration vector in the various packet repetition scenarios could be implemented in PAL® or GAL® type devices.

Both of the above methods employ the same signals: $\overline{\text{ACKI}}$, $\overline{\text{ACKO}}$ and $\overline{\text{ACTN}}$ to perform *PORT N* or *M* arbitration.

The Inter-RIC bus allows multi-RIC2A operations to be performed in exactly the same manner as if there is only a single RIC2A in the system. The simplest way to describe the operation of Inter-RIC bus is to see how it is used in a number of common packet repetition scenarios.

### 4.4 Examples Of Packet Repetition Scenarios

The operation of RIC2A is described by the following examples of packet repetition scenarios.

**Data Repetition Overview**

When a packet is received at one port, the RIC2A checks the source, and destination addresses of the packet. The port configuration causes either a pseudo random bit sequence, or the received packet to be transmitted to different ports.

If there is a destination address mismatch (secure mode), then the RIC2A will generate a random pattern from the first bit of the data field to that port. The data remains intact on the Inter-RIC bus so other cascaded repeaters could compare the destination address with their local CAMs.

On a valid source address mismatch (secure mode), RIC2A shall switch to random pattern both on the local transmitting ports and the Inter-RIC bus.

**Collision Scenarios Overview**

The RIC2A will adhere to all collision scenarios. When a collision occurs, RIC2A will switch to a jam pattern to comply with IEEE repeater specifications.

**FIFO Condition Overview**

Elasticity buffer error (ELBER) or FIFO overflow burst is another condition that could take place anytime during the packet transmission. The sequence of events for FIFO burst is the same as those for collision.

**Data Repetition Process**

The first task to be performed is *PORT N* identification. This is an arbitration process performed by the Port State Machines in the system. In situations where two or more ports simultaneously receive packets, the Inter-RIC bus operates by choosing one of the active ports, and forcing the others to transmit data (real data or pseudo random data). This is done in accordance with the IEEE specification's allowed exit paths from the *IDLE* state, i.e., to the *SEND PREAMBLE PATTERN* or *RECEIVE COLLISION* states.

The packet begins with a preamble pattern derived from the RIC2A's on chip jam/preamble generator. The data received at *PORT N* is directed through the receive multiplexor to the PLL decoder. Once phase lock has been achieved, the decoded data (in NRZ format) with its associated clock and enable signals, is asserted onto the IRD, IRC, and IRE of the Inter-RIC bus. This serial data stream is received from the bus by all RIC2As in the repeater and directed to their elasticity buffers. Logic circuits monitor the data stream and look for the Start of Frame Delimiter (SFD). When it has been detected, data is loaded into the elasticity buffer for later transmission. This will occur when sufficient preamble has been transmitted and certain internal state machine operations have been fulfilled.

## 4.0 Functional Description (Continued)

Figure 5 shows two RIC2As A and B, daisy chained together with RIC2A-A positioned at the top of the chain. If a packet is received at port B1 of RIC2A-B, and then repeated to the other ports in the system (non-secure mode). Figure 6 shows the functional timing diagram for the packet repetition signals. In this example only two ports in the system are shown. In non-secure mode, the other ports also repeat the packet. It also indicates the operation of the RIC2As' state machines in so far as can be seen by observing the Inter-RIC bus. For reference, the repeater's state transitions are shown in terms of the states defined by the IEEE specification. The location of *PORT N* is also shown. The following section describes the repeater and Inter-RIC bus transitions shown in Figure 6.

The repeater activity is stimulated by the data signal received by port B1. The RIC2As in the system are alerted to forthcoming repeater operation by the falling edges on the $\overline{\text{ACKI}}$ and $\overline{\text{ACKO}}$ daisy chain and the $\overline{\text{ACTN}}$ bus signal. Following a defined start up delay the repeater moves to the *SEND PREAMBLE* state. The RIC2A system utilizes the start up delay to perform port arbitration. When packet transmission begins, the RIC2A system enters the REPEAT state. The expected, for normal packet repetition, sequence of repeater states, *SEND PREAMBLE, SEND SFD* and *SEND DATA* are followed, but are not visible at the Inter-RIC bus. They are then merged into a single REPEAT state. Similarly, the *WAIT* and *IDLE* states appear as a combined Inter-RIC bus IDLE state.

Once a REPEAT operation has begun (e.g. the repeater leaves the *IDLE* state), it is required to transmit at least 96 bits of data or jam/preamble onto its network segments. If the duration of the received signal from *PORT N* is shorter than 96 bits, the repeater transitions to the *RECEIVE COLLISION* state (described later). This behavior is known as fragment extension.

After the packet data has been repeated, including the emptying of the RIC2As' elasticity buffers, the RIC2A performs the *Tw1* transmit recovery operation. This is performed during the *WAIT* state shown in the repeater state diagram.

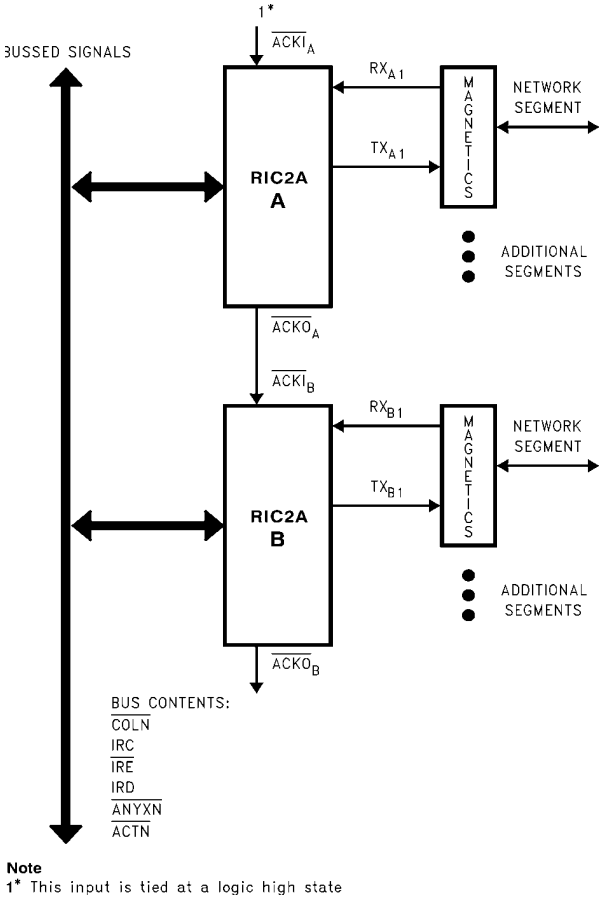
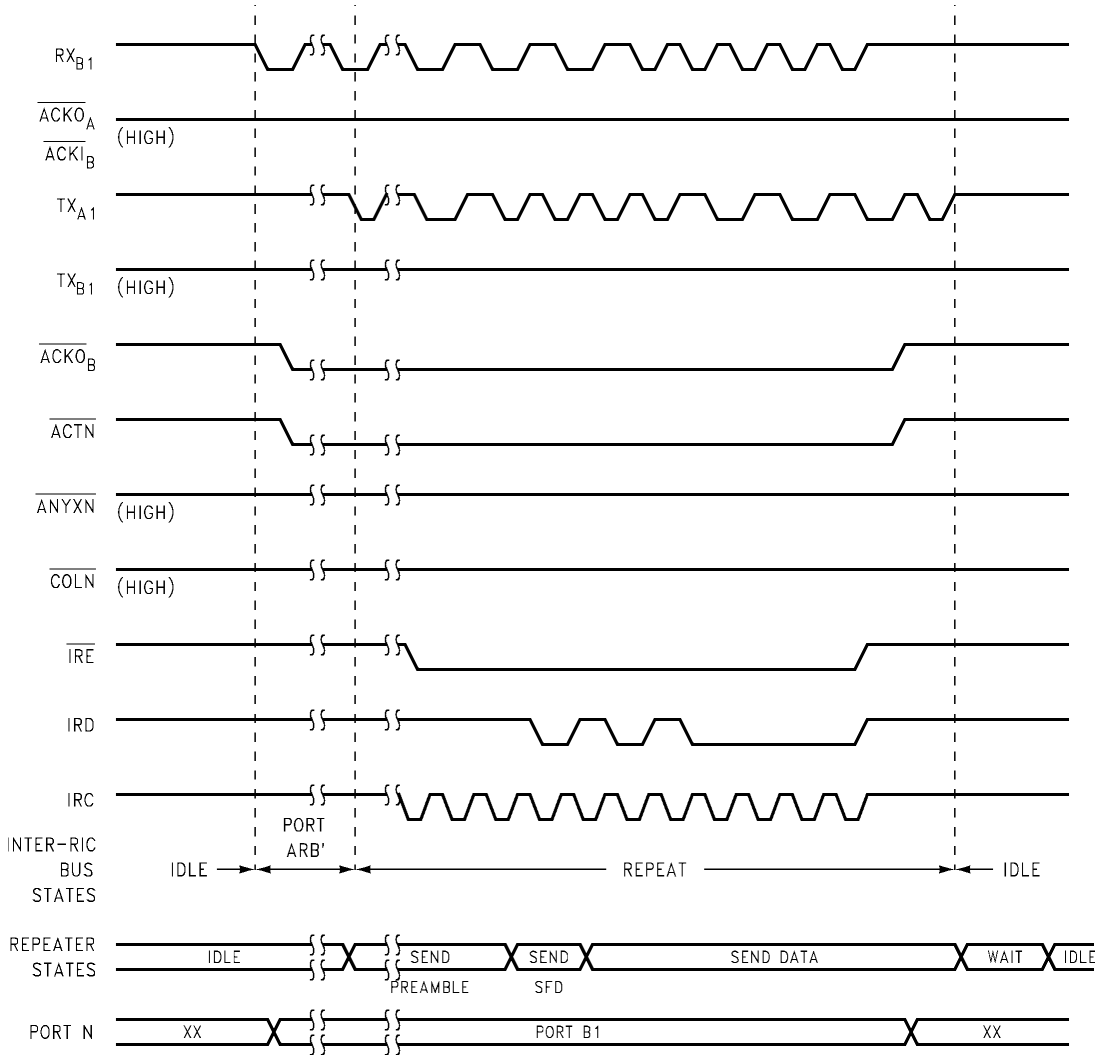
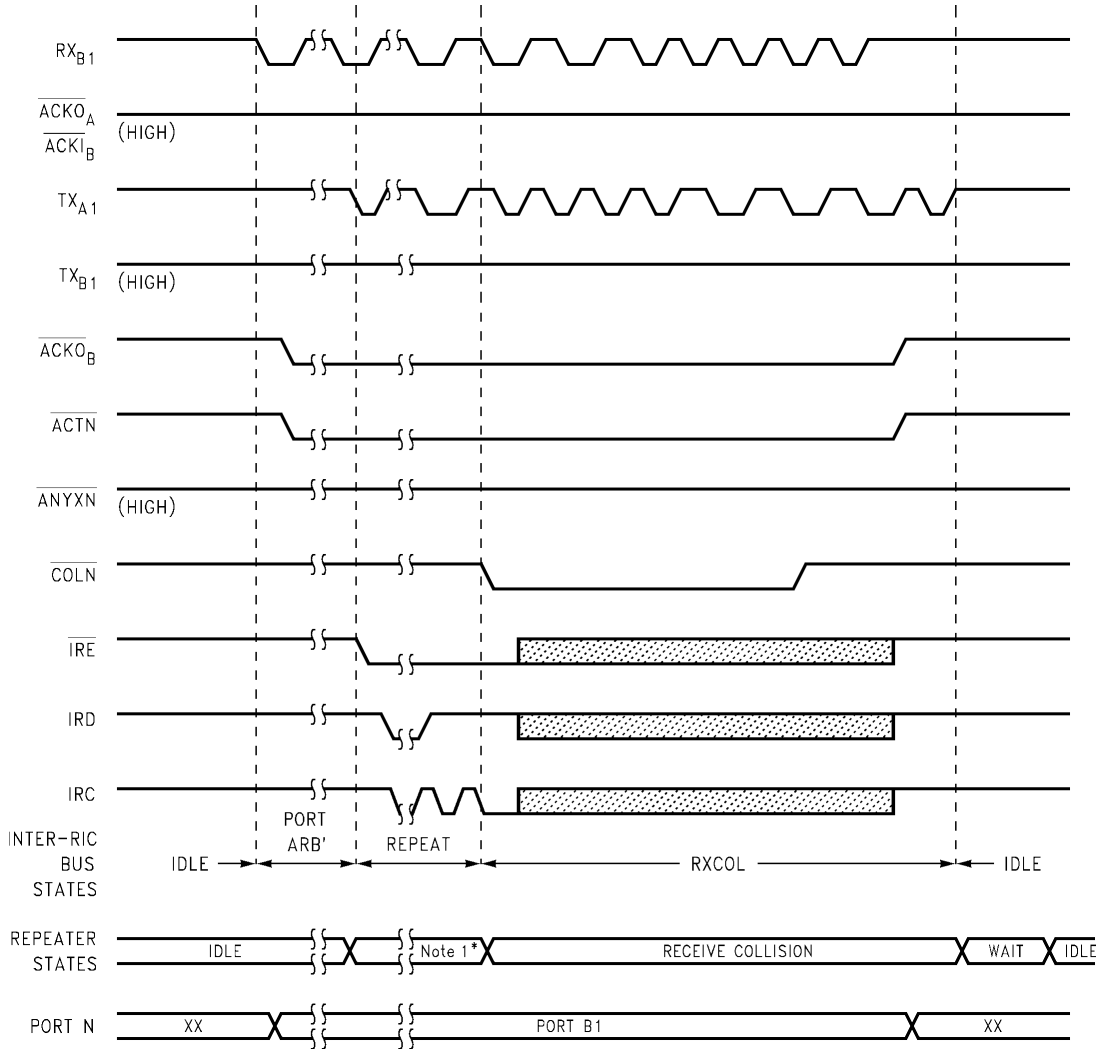



**Figure 5. RIC2A System Topology**




Note: **1*** The activity shown on RX $_{A1}$ represents the transmitted signal on TX $_{A1}$ after being looped back by the attached transceiver.

**Figure 6. Data Repetition**




Note: 1 SEND PREAMBLE, SEND SFD, SEND DATA

AUI port shown.

**Figure 7. Receive Collision**

## 4.0 Functional Description (Continued)

### Receive Collisions (AUI Port only)

A receive collision is a collision which occurs on the network segment attached to the AUI port. The collision is "received" in a similar manner as a data packet is received, and then repeated to the other network segments. Not surprisingly, the receive collision propagation follows a similar sequence of operations as data repetition.

An arbitration process is performed to find *PORT N* and a preamble/jam pattern is transmitted by the repeater's other ports. When the AUI port as *PORT N* detects a collision on its segment the $\overline{\text{COLN}}$ Inter-RIC bus signal is asserted. This forces all the RIC2As in the system to transmit a preamble/jam pattern to their segments. This is important since they may be already transmitting data from their elasticity buffers. The repeater moves to the *RECEIVE COLLISION* state and begins to transmit the jam pattern. The repeater remains in this state until both the following conditions have been fulfilled:

1. at least 96 bits have been transmitted onto the network,
2. the activity has ended.

Under close examination, the repeater specification reveals that the actual end of activity has its own permutations of conditions:

1. collision and receive data signals may end simultaneously,
2. receive data may appear to end before collision signals,
3. receive data may continue for some time after the end of the collision signal.

Network segments using coaxial media may experience spurious gaps in segment activity when the collision signal goes inactive. This arises from the inter-action between the receive and collision signal squelch circuits, implemented in coaxial transceivers, and the properties of the coaxial cable itself. The repeater specification avoids propagation of these activity gaps by extending collision activity by the *Tw2* wait time. Jam pattern transmission must be sustained throughout this period. After this, the repeater will move to the *WAIT* state unless there is a data signal being received by the AUI port as *PORT N.*

The functional timing diagram, Figure 7, shows the operation of a repeater system during a receive collision. The system configuration is the same as earlier described and is shown in Figure 6.

The RIC2As perform the same *PORT N* arbitration and data repetition operations described previously. The system is notified of the receive collision on the AUI port by the $\overline{\text{COLN}}$ bus signal going active. This signal informs the main state machines to send out the jam pattern rather than valid data stored in the elasticity buffers. Once a collision has occurred, the IRC, IRD and $\overline{\text{IRE}}$ bus signals may become undefined. When the collision has ended and the *Tw2* operation performed, the repeater moves to the *WAIT* state.

### Transmit Collisions

A transmit collision is a collision that is detected upon a segment to which the repeater system is transmitting. The port state machine monitoring the colliding segment asserts the $\overline{\text{ANYXN}}$ bus signal. The assertion of $\overline{\text{ANYXN}}$ causes *PORT M* arbitration to begin. The repeater moves to the *TRANSMIT COLLISION* state when the port which had been *PORT N* starts to transmit a Manchester encoded 1 on to its network segment. While in the *TRANSMIT COLLISION* state, all ports of the repeater must transmit the 1010... jam pattern, and *PORT M* arbitration is performed. Each RIC2A is obligated, by the IEEE specification, to ensure all of its ports transmit for at least 96 bits once the *TRANSMIT COLLISION* state has been entered. This transmit activity is enforced by the $\overline{\text{ANYXN}}$ bus signal. While $\overline{\text{ANYXN}}$ is active, all RIC2A ports will transmit jam. To ensure this situation lasts for at least 96 bits, the MSM inside the RIC2As assert the $\overline{\text{ANYXN}}$ signal throughout this period. After this period has elapsed, $\overline{\text{ANYXN}}$ will only be asserted if there are multiple ports with active collisions on their network segments.

There are two possible ways for a repeater to leave the *TRANSMIT COLLISION* state. The most straight forward is when network activity, i.e., collisions and their *Tw2* extensions, end before the 96 bit enforced period expires. Under these conditions the repeater system may move directly to the *WAIT* state when 96 bits have been transmitted to all ports. If the MSM enforced period ends and there is still one port experiencing a collision, the *ONE PORT LEFT* state is entered. This may be seen on the Inter-RIC bus when $\overline{\text{ANYXN}}$ is de-asserted and *PORT M* stops transmitting to its network segment. In this circumstance the Inter-RIC bus transitions to the RECEIVE COLLISION state. The repeater will remain in this state while *PORT M's* collision, *Tw2* collision extension and any receive signals are present. When these conditions are not true, packet repetition finishes and the repeater enters the *WAIT* state.

Figure 8 shows a multi-RIC2A system operating under transmit collision conditions. There are many different scenarios which may occur during a transmit collision, this figure illustrates one of these. The diagram begins with packet reception by port A1. Port B1 experiences a collision, since it is not *PORT N* it asserts $\overline{\text{ANYXN}}$. This alerts the main state machines in the system to switch from data to jam pattern transmission.

Port A1 is also monitoring the $\overline{\text{ANYXN}}$ bus line. Its assertion forces A1 to relinquish its *PORT N* status, start transmitting, stop asserting $\overline{\text{ACTN}}$ and release its hold on the PSM arbitration signals ($\overline{\text{ACKO}}$ A and $\overline{\text{ACKI}}$ B). The first bit it transmits will be a Manchester encoded "1" in the jam pattern. Since port B1 is the only port with a collision, it attains *PORT M* status and stops asserting $\overline{\text{ANYXN}}$. It does however assert $\overline{\text{ACTN}}$, and exert its presence upon the PSM arbitration chain (forces $\overline{\text{ACKO}}$ B low). The MSMs ensure that $\overline{\text{ANYXN}}$ stays active and thus force all of the ports, including *PORT M,* to transmit to their segments.

After some time port A1 experiences a collision. This arises from the presence of the packet being received from port A1's segment and the jam signal the repeater is now transmitting onto this segment. Simultaneous receive and transmit activity on one segment results in a collision. Port A1 fulfills the same criteria as B1, i.e., it has an active collision

on its segment, but in addition it is higher in the arbitration chain. This priority yields no benefits for port A1 since the $\overline{\text{ANYXN}}$ signal is still active. There are now two sources driving $\overline{\text{ANYXN}}$, the MSMs and the collision on port B1.

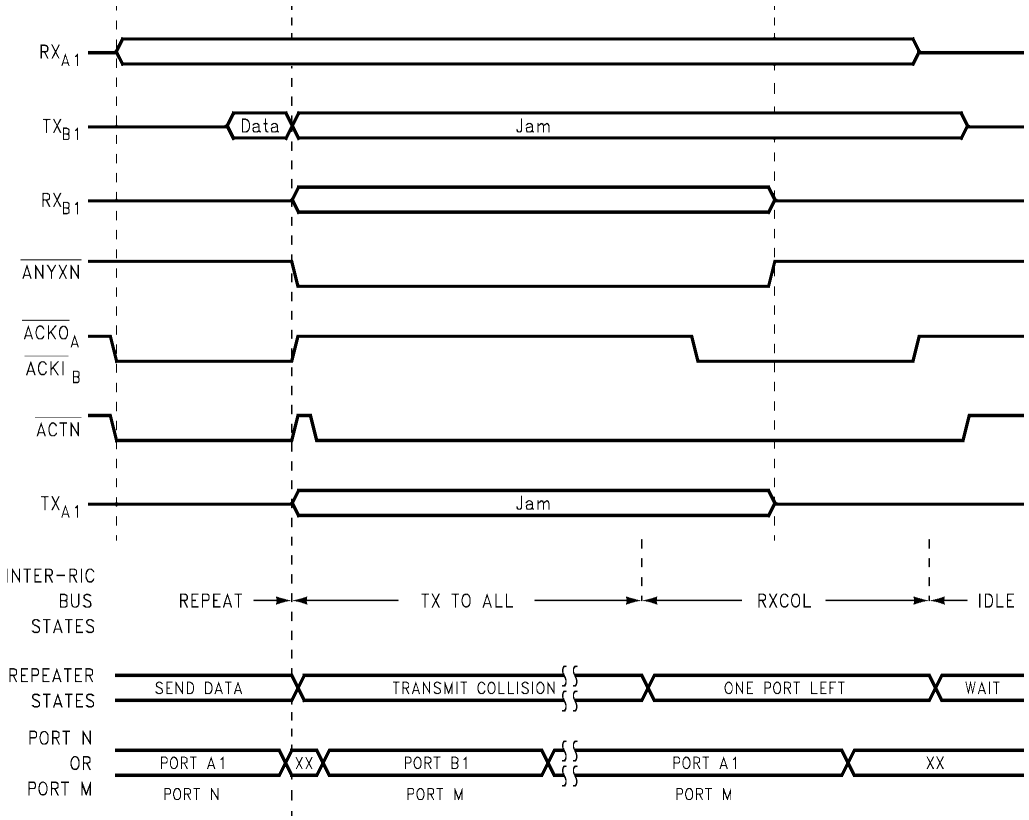
Eventually the collision on port B1 ends and the $\overline{\text{ANYXN}}$ extension by the MSMs expires. There is only one collision on the network (this may be deduced since $\overline{\text{ANYXN}}$ is inactive) so the repeater will move to the *ONE PORT LEFT* state. The RIC2A system treats this state in a similar manner to a receive collision with *PORT M* fulfilling the role of the receiving port. The difference from a true receive collision is that the switch from packet data to the jam pattern has already been made (controlled by $\overline{\text{ANYXN}}$). Thus the state of $\overline{\text{COLN}}$ has no effect upon repeater operations. In common with the operation of the *RECEIVE COLLISION* state, the repeater remains in this condition until the collision and receive activity on *PORT M* subside. The packet repetition operation completes when the *Tw1* recovery time in the *WAIT* state has been performed.

Note: In transmit collision conditions $\overline{\text{COLN}}$ will only go active if the RIC2A which contained *PORT N* at the start of packet repetition contains *PORT M* during the *TRANSMIT COLLISION* and *ONE PORT LEFT* states.

### Jabber Protection

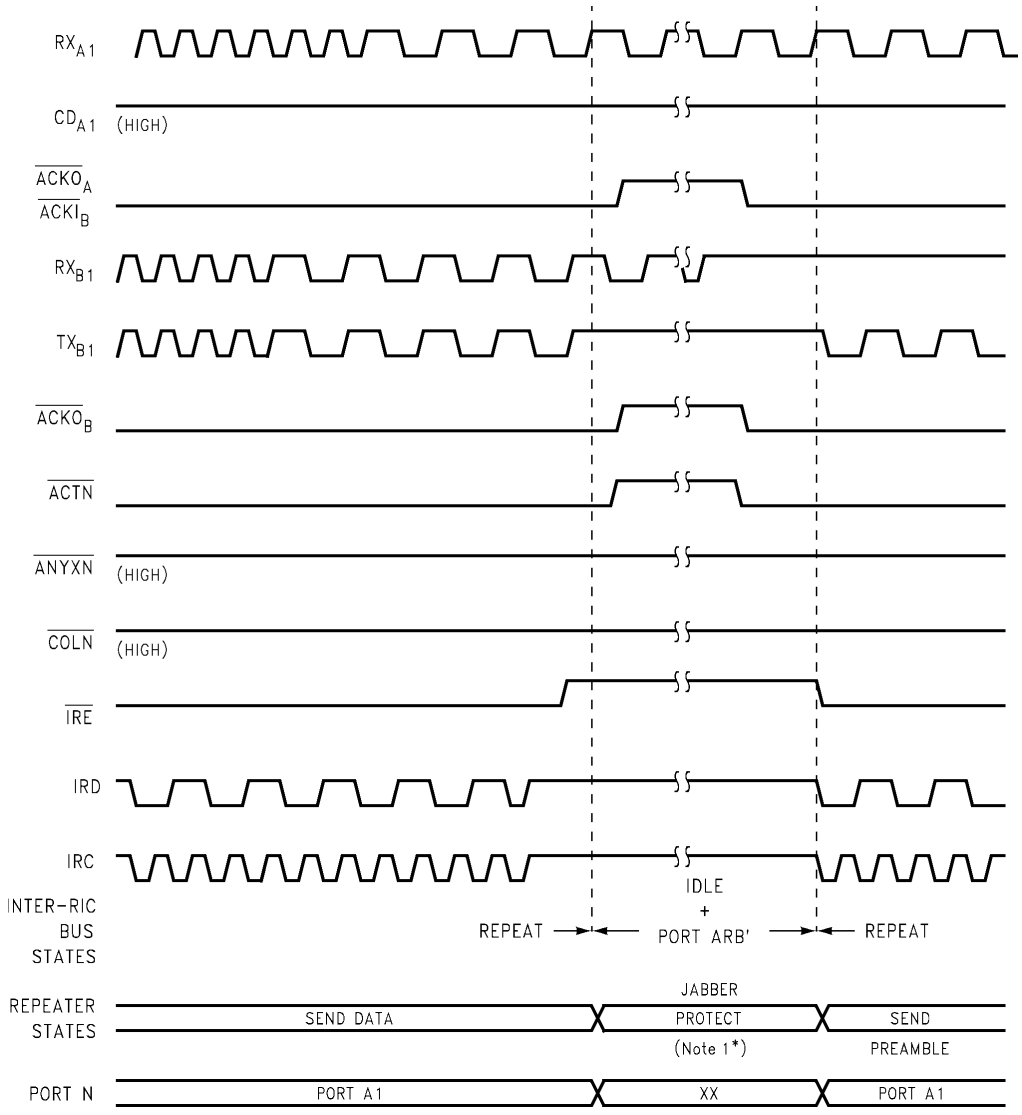
A repeater is required to disable transmit activity if the length of its current transmission reaches the jabber protect limit. This is defined by the specification's *Tw3* time. The repeater disables output for a time period defined by the *Tw4* specification, after this period normal operation may resume.

Figure 9 shows the effect of a jabber length packet upon a RIC2A based repeater system. The **JABBER PROTECT** state is entered from the *SEND DATA* state. While the *Tw4* period is observed the Inter-RIC bus displays the IDLE state. This is misleading since new packet activity or continuous activity (as shown in the diagram) does not result in packet repetition. This may only occur when the *Tw4* requirement has been satisfied.




Note: The Inter-RIC bus is configured to use active low signals. AUI port shown

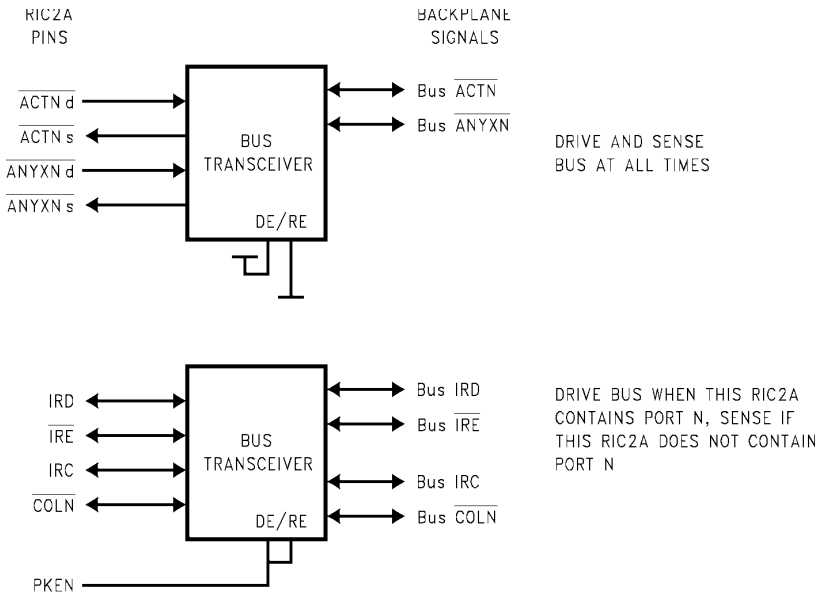
**Figure 8. Transmit Collision**




Note: **1*** The IEEE Specification does not have a jabber protect state defined in its main state diagram, this behavior is defined in an additional MAU Jabber Lockup Protection state diagram.

Note: The Inter-RIC bus is configured to use active low signals. AUI port shown

**Figure 9. Jabber Protect**




Note: DE = Bus Drive Enable active high, /RE = Bus Receive Enable active low

Note: The Inter-RIC bus is configured to use active low signals.

**Figure 10. External Bus Transceiver Connection Diagram**

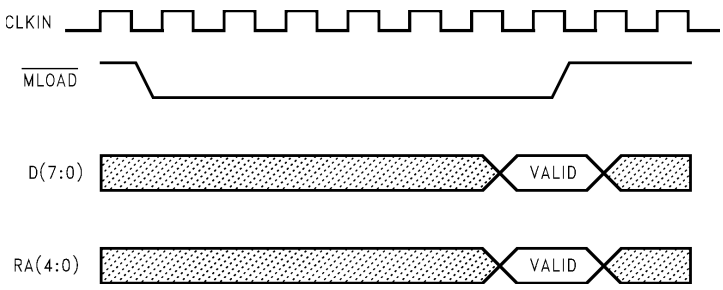



**Figure 11. Mode Load Operation**

### 4.5 Description Of Hardware Connection For Inter-ric Bus

When considering the hardware interface, the Inter-RIC bus may be viewed as consisting of three groups of signals:

1. Port Arbitration chain, namely: $\overline{\text{ACKI}}$ and $\overline{\text{ACKO}}$.
2. Simultaneous drive and sense signals, i.e., $\overline{\text{ACTN}}$ and $\overline{\text{ANYXN}}$. (Potentially these signals may be driven by multiple devices).
3. Drive or sense signals, i.e., $\overline{\text{IRE}}$, IRD, IRC and $\overline{\text{COLN}}$. (Only one device asserts these signals at any instance in time).

The first set of signals are either used as point to point links, or with external arbitration logic. In both cases the load on these signals will not be large, so the on-chip drivers are adequate. This may not be true for signal classes (2) and (3).

The Inter-RIC bus has been designed to connect RIC2As together directly, or via external bus transceivers. The latter is advantageous in large repeaters. In the second application the backplane is often heavily loaded and is beyond the drive capabilities of the on-chip bus drivers. The need for simultaneous sense and drive capabilities on the $\overline{\text{ACTN}}$ and $\overline{\text{ANYXN}}$ signals, and the desire to allow operation with external bus transceivers, makes it necessary for these bus signals to each have a pair of pins on the RIC2A. One drives the bus, the other senses the bus signal. When external bus transceivers are used, they must be open collector / open drain to allow wire-ORing of the signals. Additionally, the drive and sense enables of the bus transceiver should be tied in the active state.

The uni-directional nature of information transfer on the $\overline{\text{IRE}}$, IRD, IRC and $\overline{\text{COLN}}$ signals, means a RIC2A is either driving these signals or receiving them from the bus, but not both at the same time. Thus a single bi-directional input / output pin is adequate for each of these signals. If an external bus transceiver is used with these signals the Packet Enable "PKEN" RIC2A output pin performs the function of a drive enable and sense disable.

Figure 10 shows the RIC2A connected to the Inter-RIC bus via external bus transceivers, such as National's DS3893A bus transceivers.

## 4.0 Functional Description (Continued)

Some bus transceivers are of the inverting type. To allow the Inter-RIC bus to utilize these transceivers, the RIC2A may be configured to invert the active states of the $\overline{\text{ACTN}}$, $\overline{\text{ANYXN}}$, $\overline{\text{COLN}}$ and $\overline{\text{IRE}}$ signals from active low to active high. Thus they become active low once more when passed through an inverting bus driver. This is particularly important for the $\overline{\text{ACTN}}$ and $\overline{\text{ANYXN}}$ bus lines, since these signals must be used in a wired-or configuration. Incorrect signal polarity would make the bus unusable.

### 4.6 Processor and Display Interface

The processor interface pins, which include the data bus, address bus and control signals, actually perform three operations which are multiplexed on these pins. These operations are:

1. The Mode Load Operation, which performs a power up initialization cycle upon the RIC2A.
2. Display Update Cycles, which are refresh operations for updating the display LEDs.
3. Processor Access Cycles, which allows µP's to communicate with the RIC2A's registers.

These three operations are described below.

#### Mode Load Operation

The Mode Load Operation is a hardware initialization procedure performed at power on. It loads vital device configuration information into on chip configuration registers. In addition to its configuration function, the $\overline{\text{MLOAD}}$ pin is the RIC2A's reset input. When $\overline{\text{MLOAD}}$ is low, all of the RIC2A's repeater timers, state machines, segment partition logic and hub management logic are reset.

The Mode Load Operation may be accomplished by attaching the appropriate set of pull up and pull down resistors to the data and register address pins to assert logic high or low signals onto these pins, and then providing a rising edge on the $\overline{\text{MLOAD}}$ pin as is shown in Figure 11. Proper execution of this function not only requires both falling and rising edges of $\overline{\text{MLOAD}}$, but also an active CLKIN throughout. The mapping of chip functions to the configuration inputs is shown in Table 1.

In a complex repeater system, the Mode Load Operation may be performed using a processor write cycle. This would require the $\overline{\text{MLOAD}}$ pin to be connected to the CPU's write strobe via some decoding logic, and included in the processor's memory map.

To support the security options, pin D0 of the data bus during MLOAD is assigned to configure RIC2A. A pull up (non-security mode) or a pull down (security mode) on this pin defines the desired security level. By using this bit, the user could also take advantage of the learning mode, as described below.

#### Learning of Port Source Address(es)

Learning mode could be invoked in two ways according to bit D0 of MLoad configuration. Only the port CAMs are capable of learning the addresses:

1. When D0=0, upon power up and by default, LME, SME, ESA and EDA bits in the Port Security Configuration Register (PSCR) are set globally. This means that each port will learn the address of the node connected to it by the reception of the first good packet. The second address is learned only if it is different from the first one. Only the address of a valid length packet without FAE (Frame Alignment Error) and/or CRC errors can be learned. As soon as the address is learned by any of the two CAM locations, RIC2A will set the corresponding ADV (Address Valid) bit in Port CAM Pointer Register.

   To start the address comparison, the SAC (Start Comparison) bit must be set (SAC=1) by the user. RIC2A will only use this CAM location for comparison when the ADV bit is set (ADV=1), whether LME is 1 or 0. These four bits in PSCR could be disabled later on a per port basis, which allow all the packets regardless of their address to pass through the repeater.
2. When D0=1 for MLOAD, security could still be done, but this time it means that the user should set the LME, SME, ESA and/or EDA bits in the Port Security Configuration Register. The rest of the operation is the same as when D0 is equal to zero.

It is important to note that RIC2A will learn the address of the packet if LME is set regardless of the D0 setting of MLoad, i.e. secure or non-secure mode.

It is also very important to note that for proper address learning, LME and SAC should not be set together.

When the repeater is in non-secure mode, then the comparison will not take place between the incoming address and the learned address.

When the repeater is in secure mode, and the LME bit is set, then the processor read/write access will be ignored for the port CAM entries. That is read/ write cycles are completed, however unknown values are read during the learning process. Data will not be written into the CAM entries until the end of the learning process.

It may be desired not to randomize the outgoing data and transmit the data intact when there is a valid source address mismatch. The Generate Random Pattern bit, GRP in the Global Security Register, will provide the option.

If GRP is set (GRP=1) and there is a source address mismatch, then RIC2A will not generate random pattern; the packet will be transmitted out and the Hub Manager will be informed about the source address mismatch.

For this option to work properly, GRP=ESA=1 and EDA=0. If EDA is also set to 1, then the packet will be randomized on ports with valid DA mismatches, and this functionality will not work.

## 4.0 Functional Description (Continued)

**Table 1. Pin Definitions for Options in the Mode Load Operation**

| Pin Name | Program-ming Function | Effect when Bit is 0 | Effect when Bit is 1 | Function |
|---|---|---|---|---|
| D0 | $\overline{\text{SCRTY}}$ | Security Mode | Non-Security Mode | This bit configures RIC2A security feature options. When D0 =0 LME, SME, ESA, EDA bits in the Port Security Configuration Register (PSCR) are set globally. When D0=1 security can still be done, but now the user needs to set the above bits in the PSCR register. |
| D1 | TW2 | 5 bits | 3 bits | This allows the user to select one of two values for the repeater specification TW2 time. The lower limit (3 bits) meets the IEEE specification. The upper limit (5 bits) is not specification compliant but may provide users with higher network throughput by avoiding spurious network activity gaps when using coaxial (10BASE2, 10BASE5) network segments. |
| D2 | $\overline{\text{CCLIM}}$ | 63 | 31 | The partition specification requires a port to be partitioned after a certain number of consecutive collisions. The RIC2A has two values available to allow users to customize the partitioning algorithm to their environment. Please refer to the Partition State Machine, in data sheet section 7.3. |
| D3 | $\overline{\text{LPPART}}$ | Selected | Not Selected | The RIC2A may be configured to partition a port if the segment transceiver does not loopback data to the port when the port is transmitting to it, as described in the Partition State Machine. |
| D4 | $\overline{\text{OWCE}}$ | Selected | Not Selected | This configuration bit allows the on-chip partition algorithm to include out of window collisions into the collisions it monitors, as described in the Partition State Machine. |
| D5 | $\overline{\text{TXONLY}}$ | Selected | Not Selected | This configuration bit allows the on-chip partition algorithm to restrict segment reconnection, as described in the Partition State Machine. |
| D6 | $\overline{\text{DPART}}$ | Selected | Not Selected | The Partition state machines for all ports may be disabled by writing a logic zero to this bit during the mode load operation. |
| D7 | MIN/MAX | Minimum Mode | Maximum Mode | The operation of the display update block is controlled by the value of this configuration bit, as described in the Display Update Cycles section. |
| RA0 | TP | X | X | All ports (2 to 13) use the internal 10BASE-T transceivers. (Internally configured) |
| RA1 | TP | X | X | |
| RA2 | BINV | Active High Signals | Active Low Sig-nals | This selection determines whether the Inter-RIC signals: IRE, ACTN, ANYXN, COLN and Management bus signal MCRS are active high or low. |
| RA3 | EXPLL | External PLL | Internal PLL | If desired, the RIC2A may be used with an external decoder, this configuration bit performs the selection. |
| RA4 | resv | Not Permitted | Required | To ensure correct device operation, this bit must be written with a logic one during the mode load operation. |

## 4.0 Functional Description (Continued)

### 4.7 Description Of Hardware Connection For Processor And Display Interface

#### Display Update Cycles

The RIC2A possesses control logic and interface pins which may be used to provide status information concerning activity on the attached network segments and the current status of repeater functions. These status cycles are completely autonomous and require only simple support circuitry to produce the data in a form suitable for a light emitting diode "LED" display. The display may be used in one of two modes:

1. Minimum mode - General Repeater Status LEDs
2. Maximum mode - Individual Port Status LEDs

Minimum mode, intended for simple LED displays, makes available four status indicators. The first LED denotes whether the RIC2A has been forced to activate its jabber protect functions. The remaining 3 LEDs indicate if any of the RIC2A's network segments are: (1) experiencing a collision, (2) receiving data, (3) currently partitioned. When minimum display mode is selected the only external components required are a 74LS374 type latch, the LEDs and their current limiting resistors.

Maximum mode differs from minimum mode by providing display information specific to individual network segments. This information denotes the collision activity, packet reception and partition status of each segment. In the case of 10BASE-T segments the link integrity status and polarity of the received data are also made available. The wide variety of information available in maximum mode may be used in its entirety or in part. This allows the system designer to choose the appropriate complexity of status display commensurate with the specification of the end equipment.

The signals provided and their timing relationships have been designed to interface directly with 74LS259 type addressable latches. The number of latches used being depend upon the complexity of the display. Since the latches are octal, a pair is needed to display each type of segment specific data (13 ports means 13 latch bits). The accompanying Table 2 and Table 3 show the function of the interface pins in minimum and maximum modes. Table 4 shows the location of each port's status information when maximum mode is selected. This may be compared with the connection diagram Figure 12.

Immediately following the Mode Load Operation (when the $\overline{\text{MLOAD}}$ pin transitions to a high logic state), the display logic performs an LED test operation. This operation lasts one second. While it is in effect, all of the utilized LEDs will blink on. Thus, an installation engineer is able to test the operation of the display by forcing the RIC2A into a reset cycle ($\overline{\text{MLOAD}}$ forced low). The rising edge on the $\overline{\text{MLOAD}}$ pin starts the LED test cycle. **During the LED test cycle the RIC2A does not perform packet repetition operations.**

The status display possesses a capability to lengthen the time an LED is active. At the end of the repetition of a packet, the display is frozen showing the current activity. This freezing lasts for 30 milliseconds or until a subsequent packet is repeated. Thus at low levels of packet activity, the display stretches activity information to make it discernible to the human eye. At high traffic rates the relative brightness of the LEDs indicates those segments with high or low activity.

It should be mentioned that when the Real Time Interrupt (RTI) occurs, the display update cycle will stop and after RTI is serviced, the display update cycle will resume activity.

**Table 2. Status Display Pin Functions in Minimum Mode**

| Signal Pin Name | Function in MINIMUM MODE |
|---|---|
| D0 | No operation |
| D1 | Provides status information indicating if there is a collision occurring on one of the segments attached to this RIC2A. |
| D2 | Provides status information indicating if one of this RIC2A's ports is receiving a data or collision packet from a segment attached to this RIC2A. |
| D3 | Provides status information indicating that the RIC2A has experienced a jabber protect condition. |
| D4 | Provides Status information indicating if one of the RIC2A's segments is partitioned. |
| D(7:5) | No operation |
| $\overline{\text{STR0}}$ | This signal is the latch enable for the 374 type latch. |
| $\overline{\text{STR1}}$ | This signal is held at a logic one. |

## 4.0 Functional Description (Continued)

**Table 3. Status Display Pin Functions in Maximum Mode**

| Signal Pin Name | Function in MAXIMUM MODE |
|---|---|
| D0 | Provides status information concerning the Link Integrity status of 10BASE-T segments. This signal should be connected to the data inputs of the chosen pair of 74LS259 latches. |
| D1 | Provides status information indicating if there is a collision occurring on one of the segments attached to this RIC2A. This signal should be connected to the data inputs of the chosen pair of 74LS259 latches. |
| D2 | Provides status information indicating if one of this RIC2A's ports is receiving a data or a collision packet from its segment. This signal should be connected to the data inputs of the chosen pair of 74LS259 latches. |
| D3 | Provides Status information indicating that the RIC2A has experienced a jabber protect condition. Additionally it denotes which of its ports are partitioned. This signal should be connected to the data inputs of the chosen pair of 74LS259 latches. |
| D4 | Provides status information indicating if one of this RIC2A's ports is receiving data of inverse polarity. This status output is only valid if the port is configured to use its internal 10BASE-T transceiver. The signal should be connected to the data inputs of the chosen pair of 74LS259 latches. |
| D(7:5) | These signals provide the repeater port address corresponding to the data available on D(4:0). |
| $\overline{\text{STR0}}$ | This signal is the latch enable for the lower byte latches, that is the 74LS259s which display information concerning ports 1 to 7. |
| $\overline{\text{STR1}}$ | This signal is the latch enable for the upper byte latches, that is the 74LS259s which display information concerning ports 8 to 13. |

**Table 4. Maximum Mode LED Definitions**

| **74LS259 Latch Inputs =** $\overline{\text{STR0}}$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 259 Output | Q0 | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 |
| 259 Addr S2-0 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| RIC2A Port Number | | 1 (AUI) | 2 | 3 | 4 | 5 | 6 | 7 |
| RIC2A DO 259 #1 | | | LINK | LINK | LINK | LINK | LINK | LINK |
| RIC2A D1 259 # 2 | ACOL | COL | COL | COL | COL | COL | COL | COL |
| RIC2A D2 259 # 3 | AREC | REC | REC | REC | REC | REC | REC | REC |
| RIC2A D3 259 # 4 | JAB | PART | PART | PART | PART | PART | PART | PART |
| RIC2A D4 259 # 5 | | | BDPOL | BDPOL | BDPOL | BDPOL | BDPOL | BDPOL |

| **74LS259 (or Equiv.) Latch Inputs =** $\overline{\text{STR0}}$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 259 Outputs | Q0 | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 |
| 259 Addr S2-0 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| RIC2A Port Number | 8 | 9 | 10 | 11 | 12 | 13 | | |
| RIC2A DO 259 # 6 | LINK | LINK | LINK | LINK | LINK | LINK | | |
| RIC2A D1 259 # 7 | COL | COL | COL | COL | COL | COL | | |
| RIC2A D2 259 # 8 | REC | REC | REC | REC | REC | REC | | |
| RIC2A D3 259 # 9 | PART | PART | PART | PART | PART | PART | | |
| RIC2A D4 259 # 10 | BDPOL | BDPOL | BDPOL | BDPOL | BDPOL | BDPOL | | |

Note: ACOL= Any Port Collision, AREC= Any Port Reception, JAB= Any Port Jabbering, LINK=Port Link, COL= Port Collision, REC=Port Reception, PART=Port Partitioned, BDPOL=Bad (inverse) Polarity of received data

This shows the LED Output Functions for the LED Drivers when 74LS259s are used. The top table refers to the bank of 4 74LS259s latched with $\overline{\text{STR0}}$, and the lower table refers to the bank of 4 74LS259s latched with $\overline{\text{STR0}}$. For example the RIC2A's D0 data signal goes to 259 #1 and #5. These two 74LS259s then drive the LINK LEDs.

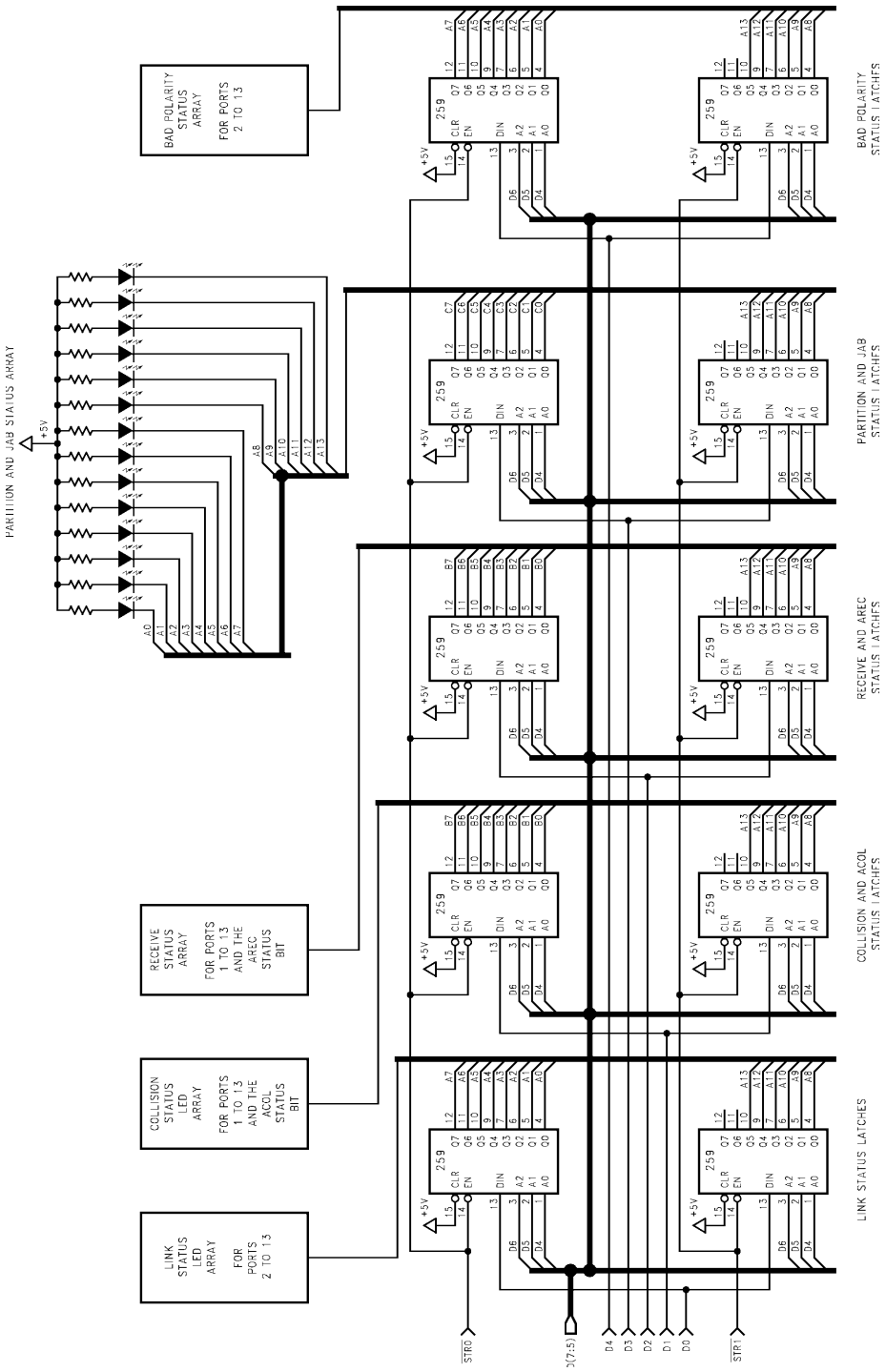



**Figure 12. Maximum Mode LED Display (All Available Status Bits Used)**

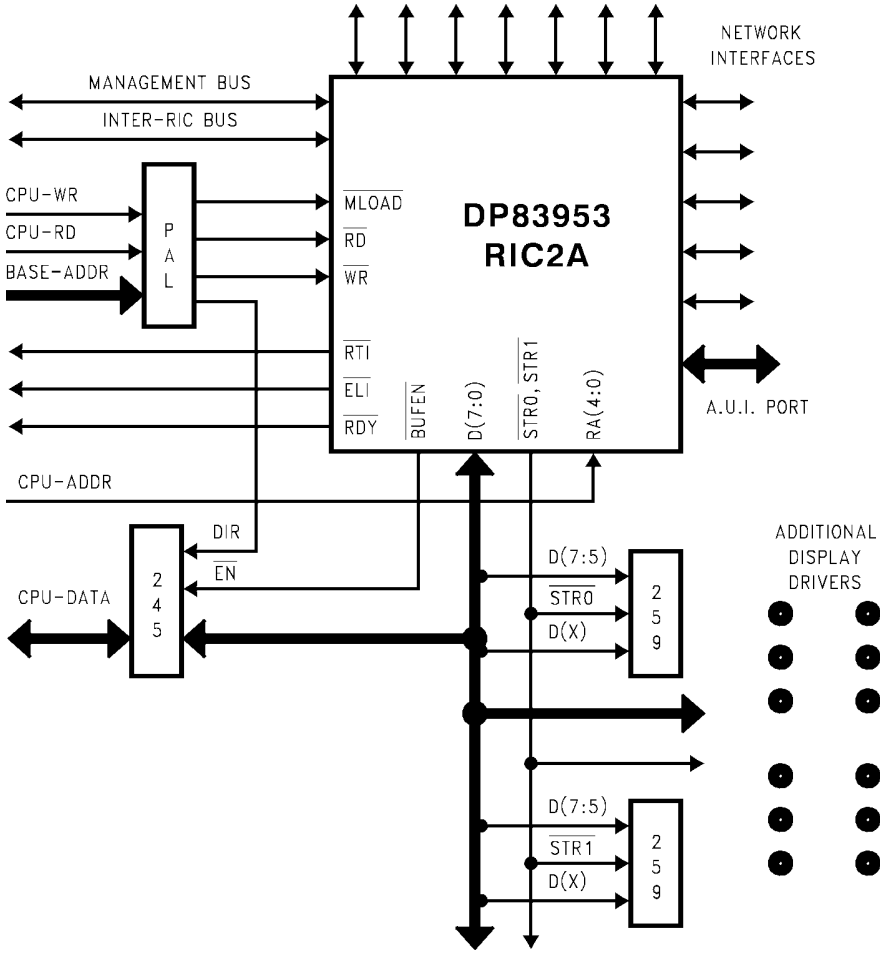



**Figure 13. Processor Connection Diagram**

## 4.0 Functional Description (Continued)

### Processor Access Cycles

Access to the RIC2A's on-chip registers is made via its processor interface, which utilizes conventional non-multiplexed address (five bit) and data (eight bit) busses. Also, the data bus provides data and address information to external chip display latches during the display update cycles. While performing these cycles, the RIC2A behaves as a master of its data bus. Consequently a TRI-STATE bidirectional bus transceiver, e.g. 74LS245, must be placed between the RIC2A and any processor.

The RIC2A provides a scheme to facilitate faster register access. The Lower Event Information Register (Page 1H, Address 1FH) has a Disable LED Update bit (DLU). The setting of this bit causes the RIC2A to stop LED updates. This scheme “unshares” the data bus, holding the RIC2A in slave access mode. This mode reduces the maximum read/write cycle time to approximately 400 ns.

The processor requests a register access by asserting the read “$\overline{RD}$” or write “$\overline{WR}$” input strobes. The RIC2A responds by completing any current display update cycle and asserts the TRI-STATE buffer enable signal “$\overline{BUFEN}$”. If the processor cycle is a write cycle then the RIC2A's data buffers are disabled to prevent contention problems. In order to interface to the RIC2A in a processor controlled system, a PAL device may be used to perform the following operations:

1. To locate the RIC2A in the processor's memory map (address decode),
2. To generate the RIC2A's read and write strobes,
3. To control the direction signal for the 74LS245.

An example of the processor and display interfaces is shown in Figure 13.

### Interrupt Handling

The DP83953 RIC2A offers an alternative method for a faster access to determine the source of the Event Logging Interrupt (ELI) register than the DP83950 RIC.

For an event logging interrupt due to flag found, the **DP83950 RIC** requires the following scheme:

1. Read the Page Select Register (Address 10H) to locate the source of Event Logging Interrupt.
2. Read all the Port Event Recording Registers (Page 1H, Address 11H to 1DH) to find the port and the event responsible for Event Logging Interrupt.

The **DP83953 RIC2A** allows the following alternate scheme for a faster access:

1. Read Page Select Register (Address 10H) to locate the source of Event Logging Interrupt.
2. Read the Event Information Registers (Page 1H, addresses 1EH and 1FH) to locate the port responsible for interrupt.
3. Read the Event Recording register of that port to find which specific event caused the Event Logging Interrupt.

# 5.0 HUB MANAGEMENT SUPPORT

The RIC2A provides information regarding the status of its ports and the packets being repeated. This data is available in three forms:

1. Counted Events - Network events accumulated into the RIC2A's 16 bit Event Counter Registers.
2. Recorded Events - Network events that set bits in the Event Record Registers.
3. Hub Management Status Packets- This is the information sent over the Management Bus in a serial format to be decoded by an Ethernet Controller board.

The processor interface provides access to all counted and recorded event information. This data is port specific and may be used to generate interrupts via the Event Logging Interrupt "$\overline{ELI}$" pin. Since the information is specific to each port, each repeater port has its own event record register and event counter. The counters and event record registers have user definable masks which enable them to be configured to count and record a variety of events. The counters and record registers are designed to be used together so that detailed information (i.e., a count value) can be held on-chip for a specific network condition. More general information, i.e. the occurrence of certain types of events, may be retained in on-chip latches. Thus, the user can configure the counters to increment upon a rapidly occurring event (most likely to be used to count collisions), and the record registers may log the occurrence of less frequent error conditions such as jabber protect packets.

## 5.1 Event Counting Function

The counters may increment upon the occurrence of one of the categories of event as described below.

Potential sources for Counter increment:

**Jabber Protection (JAB):** The port counter increments if the length of a received packet from its associated port causes the repeater state machine to enter the jabber protect state.

**Elasticity Buffer Error (ELBER):** The port counter increments if an Elasticity Buffer underflow or overflow condition occurs during packet reception. This flag is held inactive if a collision occurs during packet reception or if a phase lock error, described below, has already occurred during packet reception.

**Phase Lock Error (PLER):** A phase lock error is caused if the phase lock loop decoder loses lock during packet reception. Phase lock onto the received data stream may or may not be recovered later in the packet, so data errors may have occurred. This flag is held inactive if a collision occurs.

**Non SFD Packet (NSFD):** If a packet is received and the start of frame delimiter (SFD) is not detected the port counter will increment. NSFD counting is inhibited if the packet experiences a collision.

**Out of Window Collision (OWC):** The OWC flag for a port goes active when a port experiences a collision outside of the network slot time.

**Transmit Collision (TXCOL):** The TXCOL flag is enabled when the repeater experiences a transmit collision.

**Receive Collision (RXCOL):** The RXCOL flag for a port goes active when the port is the receive source of network activity and suffers a collision, provided no other network segments experience collisions. At that point, the receive collision flag for the receiving port will be set.

**Partition (PART):** The port counter increments when a port becomes partitioned.

**Bad Link (BDLNK):** The port counter increments when a 10BASE-T port has entered the link lost state.

**Short Event reception (SE):** The port counter increments if the received packet is less than 74 bits long and no collision occurs during reception.

**Packet Reception (REC):** When a packet is received the port counter increments.

In order to utilize the counters the user must choose, from the above list, the desired statistics for counting. This counter mask information must be written to the appropriate registers (i.e. Upper and Lower Event Count Mask Registers). For the exact bit patterns of these registers please read Section 7.0 of the data sheet.

If the counters are configured to count network collisions and the appropriate masks have been set, then whenever a collision occurs on a segment, this information will be latched by the hub management support logic. At the end of repetition of the packet the collision status, respective to each port, is loaded into that port's counter. This operation is completely autonomous and requires no processor intervention.

Each counter is 16 bits long and may be directly read by the processor. Additionally each counter has a number of decodes to indicate the current value of the count. There are three decodes:

low count (a value of 00FF Hex and under),

high count (a value of C000 Hex and above),

full count (a value of FFFF Hex).

The decodes from each counter are logically "ORed" together and may be used as interrupts for the $\overline{ELI}$ interrupt pin. Additionally, the status of these bits may be observed by reading the Page Select Register. In order to enable these threshold interrupts, the appropriate interrupt mask bit must be written to the Management and Interrupt Configuration Register. See Section 7.0 for register details.

In addition to their event masking functions, the Upper Event Counting Mask Register (UECMR) possesses two bits that control operation of the counters. The Reset On Read "ROR" bit resets the counters after performing a processor read cycle. If this ROR bit is not set, which is used to zero the counters, then the counters must be either written with zeros by the processor or allowed to rollover to all zeros. The Freeze When Full "FWF" bit prevents counter rollover by inhibiting count up cycles (these cycles happen when chosen events occur), thereby freezing that particular counter at FFFF Hex.

## 5.0 HUB MANAGEMENT SUPPORT (Continued)

The port event counters may also be controlled by the Counter Decrement ($\overline{\text{CDEC}}$) pin. As the name suggests, a logic low state on this pin will decrement all the counters by a single value. The pulses on $\overline{\text{CDEC}}$ are internally synchronized and scheduled to avoid any conflict with the "up counting" activity. If an “up count” and a “down count” occur simultaneously, then the “down count” is delayed until the “up count” has completed. This combination of up and down counting capability enables the RIC2A's on-chip counters to provide a simple rolling average, or be used as extensions of larger external counters.

Note: If the FWF option is enabled then the count down operation is disabled from those registers which have reached FFFF Hex and consequently have been frozen. Thus, the FWF and $\overline{\text{CDEC}}$ bits will be set to provide rate indication. A frozen counter indicates that a rate has gone out of bounds, due to incrementing too fast or too slowly. If the low count and high count decodes are employed as either interrupts or poll cycles, the direction of the rate excursion may be determined.

### New Hub Management Counters

The are 13 more 8 bit counters on the RIC2A than provided on the DP83950 RIC. These counters will count events specified in the Event Count and Interrupt Mask Register 2 (ECIMR2), such as Frame Check Sequence, Frame Alignment Error, Partition, Out of Window Collision. Also, this register includes "Reset On Read" and "Freeze When Full" control bits.

It should be noted that Counter Decrement (CDEC) will not be used with the ECMR2. Also, no real time or event logging interrupt will be generated for this register.

### Reading the Event Counters

The RIC2A's external data bus is eight bits wide. Since the event counters are 16 bits long, two processor read cycles are required to read the counter value. In order to ensure correct counter values and simultaneously allow event counts and processor accesses, values are stored in a temporary holding register. A read cycle to either the lower or upper byte of a counter causes both bytes to be latched into the temporary holding register. Thus, when the other byte of the counter is obtained, the temporary holding register is accessed (not the actual counter register). This ensures that the upper and lower bytes contain the value sampled at the same instant in time.

There are no restrictions concerning whether the upper or lower byte is read first. However, to ensure the "same instance value" is obtained, the reads of the upper then lower byte (or vice versa) should be performed as consecutive reads of the counter array. Other “non counter” registers may be read in between these read cycles and write cycles may be performed. If another counter is read, or the same byte of the original counter is read again, then the holding register is updated from the counter array, and the unread byte is lost.

If the reset on read option is employed, then the counter is reset after the transfer to the holding register is performed. Processor read and write cycles are scheduled to avoid conflict with count up or count down operations. In the case of a processor read, the count value is stable as it is loaded into the holding register. In the case of a processor write, the newly written value is stable enough to be incremented or decremented by any subsequent count operation. During the period of time when the $\overline{\text{MLOAD}}$ pin is low, (power on reset) all counters are reset to zero and all count masks are forced into the disabled state. Section 7.0 of the data sheet details the address location of the port event counters.

### 5.2 Event Record Function

As stated previously, each repeater port has its own 8 bit Event Recording status register. Each bit may be dedicated to log the occurrence of a particular event (see Section 7.0 for detailed description). The Event Recording Mask Register controls the logging of these events. Additionally, the particular mask bit must be set to record an event. Similar to the scheme employed for the event counters, the recorded events are latched during the repetition of a packet then automatically loaded into the recording registers at the end of packet transmission. When one of the unmasked events occurs, that particular port register bit is set. The register bits for all of the ports are logically "ORed" together to produce a Flag Found "FF" signal. The Page Select Register contains the Flag Found indicator. Additionally, if the appropriate mask bit is enabled in the Management and Interrupt Configuration Register then an interrupt may be generated

A processor read cycle to an Event Record Register resets any bits set in that register. Read operations are scheduled to guarantee that data does not change during the cycle. Any internal bit setting event that immediately follows a processor read will be successful. Events that may be recorded are described below:

**Jabber Protection (JAB):** This flag goes active if the length of a received packet causes the repeater state machine to enter into the Jabber Protect state.

**Elasticity Buffer Error (ELBER):** This goes active if a buffer underflow or overflow condition occurs during packet reception. The flag is held inactive if a collision occurs during packet reception or if a phase lock error has already occurred during packet reception.

**Phase Lock Error (PLER):** A phase lock error is caused if the phase lock loop decoder loses lock during packet reception. A phase lock onto the received data stream may not be recoverable later in the packet and data errors may have occurred. This flag is held inactive if a collision occurs.

**Non SFD Packet (NSFD):** If a packet is received and the start of frame delimiter (SFD) is not detected, this flag will go active. The flag is held inactive if a collision occurs during packet reception.

**Out of Window Collision (OWC):** The out of window collision flag goes active when a port experiences a collision outside of the network slot time.

**Partition (PART):** This flag goes active when a port becomes partitioned.

**Bad Link (BDLNK):** This flag goes active when a 10Base-T port has entered the link lost state.

## 5.0 HUB MANAGEMENT SUPPORT (Continued)

**Short Event reception (SE):** This flag goes active if the received packet is less than 74 bits long and no collision occurs during reception.

5.3 Management Interface Operation

The Hub Management interface provides a mechanism to combine repeater status information with packet information to form a hub management packet. A serial bus interface consisting of carrier sense, receive clock and receive data is designed to connect one or more RIC2A's over a backplane bus to another device, such as the DP83932 "SONIC™" controller. The SONIC combined with the RIC2As form a powerful entity for network statistics gathering.

The management interface consists of four pins:

| | |
|---|---|
| MRXC | Management Receive Clock - 10MHz NRZ Clock output. |
| $\overline{\text{MCRS}}$ | Management Carrier Sense - Input/ Output indicating of valid data stream. |
| MRXD | Management Receive Data - NRZ Data output synchronous to MRXC |
| $\overline{\text{PCOMP}}$ | Packet Compress - Input to truncate the packet's data field. |

The first three signals will emulate the interface between an Ethernet controller and a phase locked loop decoder (such as the DP83932 SONIC and the DP83910 SNI). These signals are driven by the RIC2A that receives the packet. The MRXC and MRXD are comprised of a NRZ serial data stream compatible with the DP83932. The $\overline{\text{PCOMP}}$ signal is driven by logic circuitry on the processor board. The actual data stream transferred as MRXD is derived from the transferred data (IRD) on the Inter-RIC bus. The difference between these two data streams are:

1. At the end of packet repetition a hub management status field is appended to the data stream. This status field, consisting of 7 bytes, is shown in Table 6, and Figure 14. The information is obtained from several packet status registers, which are described below. In accordance with the IEEE Specification the least significant bit of a byte is transmitted first.
2. While the data of the repeated packet is being transferred over the management bus, the received clock signals on the MRXC pin may be inhibited. This is controlled by the Packet Compress pin $\overline{\text{PCOMP}}$. If $\overline{\text{PCOMP}}$ is asserted during repetition of the packet then MRXC signals are inhibited when the number of bytes (after SFD) transferred over the management bus equals the number displayed in the Packet Compress Decode Register. This register provides a means to delay the effect of the $\overline{\text{PCOMP}}$ signal, which may be generated early in the packet's repetition. The packet compression feature reduces the amount of memory required to buffer packets as they are received and are waiting to be processed by hub management software. In this application, an address decoder, which forms part of the packet compress logic, would monitor the address fields as they are received over the management bus. If the destination address is not the address of the management node inside the hub, then packet compression could be employed. In this manner only the portion of the packet meaningful for hub management interrogation (i.e., the address fields,) is transferred to the SONIC and buffered into memory.

   If the repeated packet ends before $\overline{\text{PCOMP}}$ is asserted or before the required number of bytes have been transferred, then the hub management status field is directly appended to the received data at a byte boundary. If the repeated packet is significantly longer than the value in the Decode Register requires, and $\overline{\text{PCOMP}}$ is asserted, the status fields will be delayed until the end of packet repetition. During this delay period MRXC clocks are inhibited, but the MCRS signal remains asserted.

Note: If $\overline{\text{PCOMP}}$ is asserted late in the packet, i.e., after the number of bytes defined by the packet compression register, then packet compression will not occur.

The Management Interface may be fine tuned to meet the timing considerations of the SONIC and the access time of its associated packet memory. This refinement may be performed in two ways:

1. The default mode of operation of the Management interface is to only transfer packets over the bus which have a start of frame delimiter. Thus "packets" that are only preamble/jam and do not convey any source or destination address information are inhibited. This filtering may be disabled by writing a logic zero to the Management Interface Configuration or "MIFCON" bit in the Management and Interrupt Configuration Register. See Section 7.0 for details.
2. The Management bus has been designed to accommodate situations of maximum network utilization, for example, when collision generated fragments occur. (These collision fragments may violate the IEEE 802.3 IFG specification.) The IFG required by the SONIC is a function of the time taken to release space in the receive FIFO and to perform end of packet processing (write status information into memory). These functions are primarily memory operations and consequently depend upon bus latency and the memory access time of the system. In order to allow the system designer some discretion in choosing the speed of this memory, the RIC2A may be configured to protect the SONIC from a potential FIFO overflow condition. This is performed by utilizing the InterFrame Gap (IFG) Threshold Select Register.

   The value (plus one) held in this register defines the minimum allowable InterFrame Gap on the management bus, measured in network bit times. If the gap is smaller than this number, $\overline{\text{MCRS}}$ is asserted but MXRC clocks are inhibited. Consequently, no data is transferred. So, the system designer may decide whether or not to gather statistics or to monitor a subset on all packets, even though they only occur with very small IFGs.

The status field, shown in Table 6, contains information of six different types. They are contained in seven Packet Status Registers "PSRs":

1. The RIC2A and port address fields [PSR(0) and (1)] can uniquely identify the repeater port receiving the packet out of a potential maximum of 832 ports sharing the same management bus (64 RIC2As each with 13 ports).

## 5.0 HUB MANAGEMENT SUPPORT (Continued)

Thus all of the other status fields can be correctly attributed to the relevant port.

2. The status flags that the RIC2A produces for the event counters or recording latches are supplied with each packet [PSR(2)]. Additionally, the clean receive CLN status is supplied to allow the user to determine the reliability of the address fields in the packet. The CLN status bit [PSR(1)] is set if no collisions are experienced during the repetition of the address fields.
3. The RIC2A has an on-chip timer to indicate when, relative to the start of packet repetition, a collision, if any, occurred [PSR(3)]. There is also a timer that indicates how many bit times of IFG were seen on the network between repetition of this packet and the preceding one. This is provided by [PSR(6)].
4. If packet compression is employed, the receive byte count contained in the SONIC's packet descriptor will indicate the number of bytes transferred over the management bus rather than the number of bytes in the packet. For this reason the RIC2A that receives the packet counts the number of received bytes and transfers this over the management bus [PSR(4),(5)].
5. Appending a status field to a data packet will obviously result in a CRC error being flagged by the SONIC. For this reason, the RIC2A monitors the repeated data stream to check for CRC and FAE errors. In the case of FAE errors, the RIC2A provides additional dummy data bits so that the status fields are always byte aligned. For packets of non-valid length the CRC and FAE error bits are not set. Refer to Table 5 for a complete description of actions relating packet length to the setting of the Jab and CRC bits, and learn functions.
6. As a final check upon the effectiveness of the management interface, the RIC2A transfers a bus specific status bit to the SONIC. This flag Packet Compress Done PCOMPD [PSR(0)], may be monitored by hub management software to check if the packet compression operation is enabled.

Figure 15 shows an example of a packet being transmitted over the management bus. The first section of the diagram (moving from left to right) shows a short preamble and SFD pattern. The second region contains the packet's address and the start of the data fields. At this time, logic on the processor/SONIC card would determine if packet compression should be used on this packet. If the $\overline{\text{PCOMP}}$ signal is asserted, then packet transfer will stop when the number of bytes transmitted equals the value defined in the decode register. Hence, the MRXC signal is idle for the remainder of the packet's data and CRC fields. The final region shows the transfer of the RIC2A's seven bytes of packet status.

The following pages describe the Hub Management registers that constitute the management status field.

Note that Packet Status Register 5 (PSR5) can be configured to remain identical in the RIC2A as in the RIC, or PSR5 can be modified to include the RUNT and SAM (source address mismatch) information. PSR5 register bit allocation is determined by the value of bit D2, MPS (Modify Packet Status), in the Global Security Register. When the MPS bit is set, PSR5 register is modified.

**Table 5. Relation of Packet Length to Jab Bit, CRCER bit and Learn**

| Packet Length | Jab Bit (D2) of PSR2 Register | CRCER Bit (D7) of PSR1 Register | Learn |
|---|---|---|---|
| Length< Min. size packet without CRC error | no | no | no |
| Length<Min size packet with CRC error | no | no | no |
| Valid length packet w/o CRC error | no | no | yes |
| Valid length packet with CRC error | no | yes | no |
| Max size<length<Jab size without CRC error | no | no | no |
| Max size< length<Jab size with CRC error | no | no | no |
| Jab size packet without CRC error | yes | no | no |
| Jab size packet with CRC error | yes | no | no |

## 5.0 HUB MANAGEMENT SUPPORT (Continued)

**Table 6. Hub Management Status Field**

| Packet status Register PSR | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| PSR(0) | | A5 | A4 | A3 | A2 | A1 | A0 | PCOMPD | resv |
| PSR(1) | | CRCER | FAE | COL | CLN | PA3 | PA2 | PA1 | PA0 |
| PSR(2) | | SE | OWC | NSFD | PLER | ELBER | JAB | CBT9 | CBT8 |
| PSR(3) Collision Bit Timer | | CBT7 | CBT6 | CBT5 | CBT4 | CBT3 | CBT2 | CBT1 | CBT0 |
| PSR(4) Lower Repeat Byte Count | | RBY7 | RBY6 | RBY5 | RBY4 | RBY3 | RBY2 | RBY1 | RBY0 |
| PSR(5) Upper Repeat Byte Count | Note 2 MPS=0 | RBY15 | RBY14 | RBY13 | RBY12 | RBY11 | RBY10 | RBY9 | RBY8 |
| | MPS=1 | resv | resv | resv | SAM | RUNT | RBY10 | RBY9 | RBY8 |
| PSR(6) Inter Frame Gap Bit Timer | | IBT7 | IBT6 | IBT5 | IBT4 | IBT3 | IBT2 | IBT1 | IBT0 |

Note 1: These registers may only be reliably accessed via the management interface. Due to the nature of these registers they may not be accessed (read or write cycles) via the processor interface.

Note 2: When MPS (Modify Packet Status) bit in the Global Security Register is:
MPS=0, Do not modify Packet Status Register 5. The RIC2A PSR5 is the same as the RIC PSR5.
MPS=1, The PSR5 register is modified in the RIC2A.

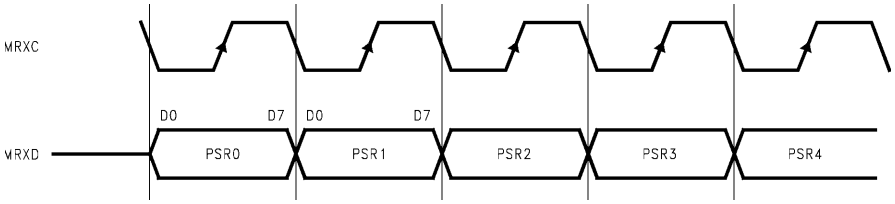



**Figure 14. Management Bus Packet Status Register TIming**

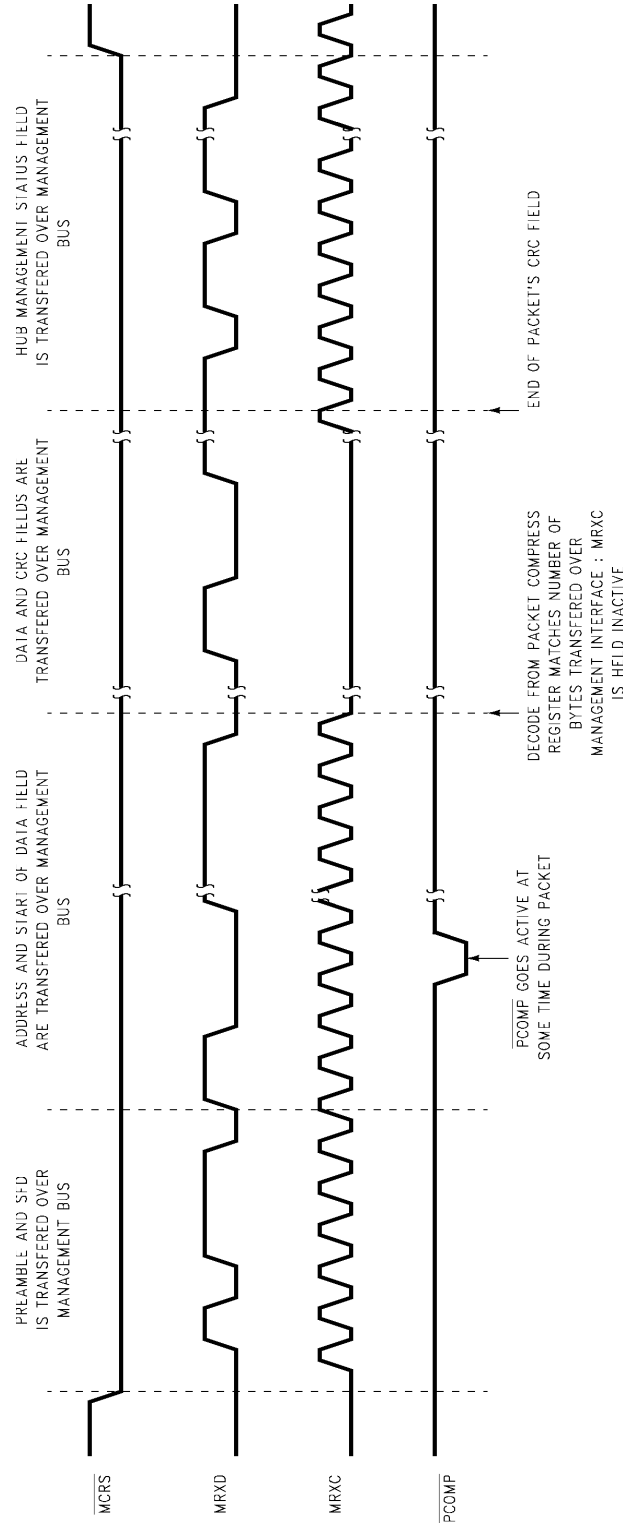



**Figure 15. Operation of the Management Bus**

### Packet Status Register 0

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| A5 | A4 | A3 | A2 | A1 | A0 | PCOMPD | resv |

| Bit | Symbol | Description |
|---|---|---|
| D0 | resv | **Reserved for future use**: This bit is currently undefined.<br>Management software should not examine the state of this bit. |
| D1 | PCOMPD | **Packet Compression Done:** If packet compression is utilized, this bit informs the user that compression was performed, i.e., the packet was long enough to require compression. |
| D(7:2) | A(5:0) | **RIC2A Address (5:0**): This address is defined by the user and is supplied when writing to the RIC2A Address Register. It is used by hub management software to distinguish between RIC2As in a multi-RIC2A system. |

### Packet Status Register 1

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| CRCER | FAE | COL | CLN | PA3 | PA2 | PA1 | PA0 |

| Bit | Symbol | Description |
|---|---|---|
| D(3:0) | PA(3:0) | **Port Address:** This field defines the port which is receiving the packet. |
| D4 | CLN | **Clean Receive:** This bit is asserted provided no collision activity occurs during repetition of the source and destination address fields, and the packet is of sufficient size to contain these fields. |
| D5 | COL | **Collision:** If a receive or transmit collision occurs during packet repetition the collision bit is asserted. |
| D6 | FAE | **Frame Alignment Error:** This bit is asserted if a Frame Alignment Error occurred in the repeated packet. |
| D7 | CRER | **CRC Error:** This bit is asserted if a CRC Error occurred in the repeated packet.<br>This status flag should not be tested if the COL bit is asserted since the error may be simply due to the collision. |

**Packet Status Register 2**

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| SE | OWC | NSFD | PLER | ELBER | JAB | CBT9 | CBT8 |

| Bit | Symbol | Description |
|---|---|---|
| D(1:0) | CT(9:8) | **Collision Timer Bits 9 and 8:** These two bits are the upper bits of the collision bit timer. |
| D2 | JAB | **Jabber Event:** This bit indicates that the receive packet was long enough to force the repeater into the jabber protect state. |
| D3 | ELBER | **Elasticity Buffer Error** During the packet an Elasticity Buffer underflow or overflow condition occurred. |
| D4 | CRER | **Carrier Error Event:** The packet suffered sufficient jitter/noise corruption to cause the PLL decoder to lose lock. |
| D5 | NSFD | **Non SFD:** The repeated packet did not contain a Start of Frame Delimiter. When this bit is set the Repeat Byte Counter counts the length of the entire packet. When this bit is not set the byte counter only counts post SFD bytes.<br>Note: The operation of this bit is not inhibited by the occurrence of a collision during packet repetition (see description of the Repeat Byte Counter below). |
| D6 | OWC | **Out of Window Collision:** The packet suffered an out of window collision. |
| D7 | SE | **Short Event:** The received activity was so small it met the criteria to be classed as a short event. |

### Modified Packet Status Register 5 (MPS=1 in GSR register)

RIC2A provides an option for a new Packet Status Register 5 (PSR5) field. On the seven bytes of management status field, PSR5 has been modified to indicate the source address mismatch information (SAM bit) for security purposes.

By using this option, the maximum received byte count changes to 2048 ($2^{11}$). As soon as the counter reaches this number, it will freeze, instead of rolling over and starting again on the reception of the next packet.

A RUNT bit has also been added to this register indicating whether the last packet received by a port was RUNT. (A packet is RUNT when its length is greater than or equal to Short Event and less than or equal to 64 bytes from SFD.)

The other registers comprise the remainder of the collision timer register [PSR(3)], the Repeat Byte Count registers [PSR(4) and PSR(5)], and the Inter Frame Gap Counter "IFG" register [PSR(6)].

**Modified Packet Status Register 5 (MPS=1 in GSR register)**

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| resv | resv | resv | SAM | RUNT | RBY10 | RBY9 | RBY8 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | NA | RBY8 | Eighth bit of receive byte count |
| D1 | NA | RBY9 | Ninth bit of receive byte count |
| D2 | NA | RBY10 | Tenth bit of receive byte count |
| D3 | NA | RUNT | **RUNT:** A packet whose length is less or equal to 64 bytes from SFD and greater than or equal to SE length.<br>0: Last packet received was not a runt<br>1: Last packet received was a runt |
| D4 | NA | SAM | **Source Address Mismatch:**<br>0: Source address match occurred for the last packet<br>1: Source address mismatch occurred for the last packet |
| D[7:5] | NA | resv | **Reserved for Future Use: reads as a logic 0** |

## 5.0 HUB MANAGEMENT SUPPORT (Continued)

### Collision Bit Timer

The Collision Timer counts, in bit times, the time between the start of repetition of the packet and the detection of the packet's first collision. When a collision occurs, the Collision counter increments as the packet repeats and freezes. The value in the counter is only valid when the collision bit "COL" in [PSR(1)] is set.

### Repeat Byte Counter

The Repeat Byte Counter is a 16 bit counter that can perform two functions. In cases where the transmitted packet possesses an SFD, the byte counter counts the number of received bytes after the SFD field. Alternatively, if no SFD is repeated, the counter reflects the length of the packet (counted in bytes) starting at the beginning of the preamble field. When performing the latter function, the counter is shortened to 7 bits when MPS =0 in the GSR register. Thus, the maximum count value is 127 bytes. The counter is shortened to 11 bits when MPS =1 in the GSR register. In this configuration, the maximum received byte count changes to 2048 bytes. The mode of counting is indicated by the "NSFD" bit in [PSR(2)]. In order to check if the received packet was genuinely a Non-SFD packet, the status of the COL bit should be checked. During collisions SFD fields may be lost or created, Management software should be robust to this kind of behavior.

### Inter Frame Gap (IFG) Bit Timer

The IFG counter counts, in bit times, the period in between repeater transmissions. The IFG counter increments whenever the RIC2A is not transmitting a packet. If the IFG is long, i.e., greater than 255 bit times, the counter holds this value. Thus a count value equal to 255 should be interpreted as 255 or more bit times.

### 5.4 Description of Hardware Connection for Management Interface

The RIC2A has been designed so that it may be connected to the management bus directly or to external bus transceivers. External bus transceivers are advantageous in larger repeaters because system backplanes are often heavily loaded beyond the drive capabilities of the on-chip bus drivers.

The unidirectional nature of information transfer on the $\overline{\text{MCRS}}$, MRXD and MRXC signals, means a single open drain output pin is adequate for each of these signals. The Management Enable (MEN) RIC2A output pin performs the function of a drive enable for an external bus transceiver if one is required.

In common with the Inter-RIC bus signals ($\overline{\text{ACTN}}$, $\overline{\text{ANYXN}}$, $\overline{\text{COLN}}$ and $\overline{\text{IRE}}$) the $\overline{\text{MCRS}}$ active level asserted by the $\overline{\text{MCRS}}$ output is determined by the state of the BINV Mode Load configuration bit.

## 6.0 Port Block Functions

The RIC2A has 13 port logic blocks (one for each network connection). In addition to the packet repetition operations already described, the port block performs two other functions:

1. the physical connection to the network segment (transceiver function).
2. it provides a means to protect the network from malfunctioning segments (segment partition).

Each port has its own status register. This register allows the user to determine the current status of the port and configure a number of port specific functions.

### 6.1 Transceiver Functions

The RIC2A may connect to network segments in two ways:

1. over AUI cable to transceiver boxes,
2. to twisted pair cable via a simple interface.

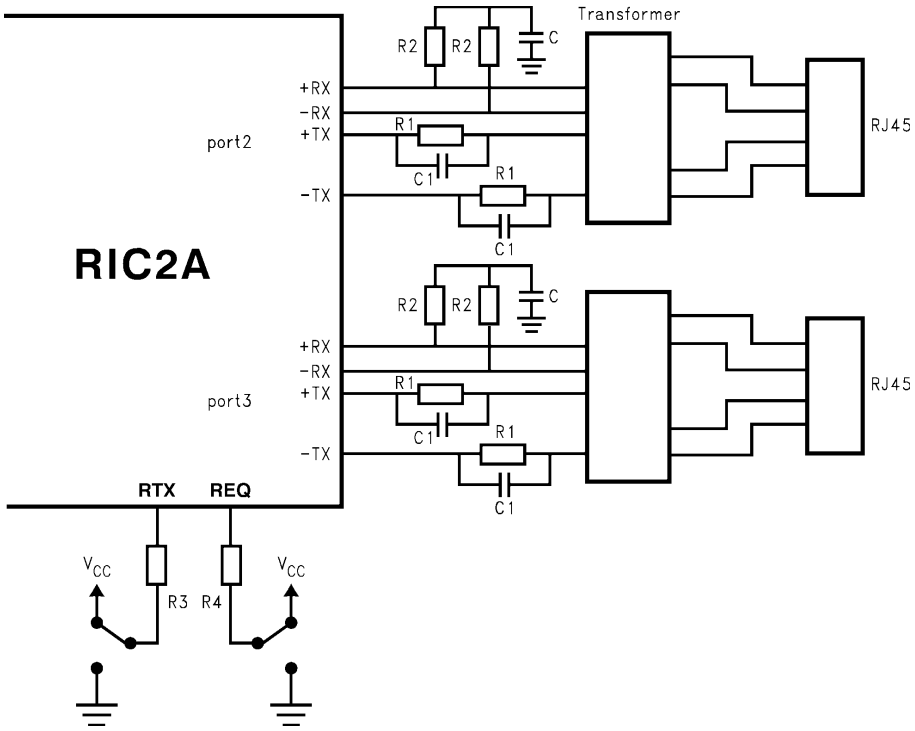
The first method is only supported by RIC2A port 1 (the AUI port). Twisted pair cable connection, (2), is only supported on ports 2 to 13.

### 10BASE-T Transceiver Operation

The RIC2A contains virtually all of the digital and analog circuitry required for connection to 10BASE-T network segments. The RIC2A design even integrates transmit filters. The Figure 16 shows the connection for a RIC2A port to a 10BASE-T segment

The operation of the 10BASE-T transceiver's logical functions may be modified by software control. The default operation allows the transceivers to transmit and expect reception of link pulses. If the designer writes a logical one to the $\overline{\text{GDLNK}}$ bit of a port's status register, the default mode may be modified. The port's transceiver will operate normally but will not transmit link pulses or monitor their reception. Therefore, a link fail state and the associated modification of transceiver operation will not occur.

The on-chip 10BASE-T transceivers automatically detects and correct the polarity of the received data stream. This polarity detection scheme relies upon the polarity of the received link pulses and the end of packet waveform. Polarity detection and correction may be disabled by software control.




Pre-emphasis resistor network/filters and per port buffer/driver are all integrated in the RIC2A.

Where C = 0.01 µF, C1 = 1.0 nF, R2 = 49.9Ω, and R1 =10Ω. All values are typical and ± 1%.

**Figure 16. Port Connection to a 10BASE-T Segment**

## 6.0 Port Block Functions (Continued)

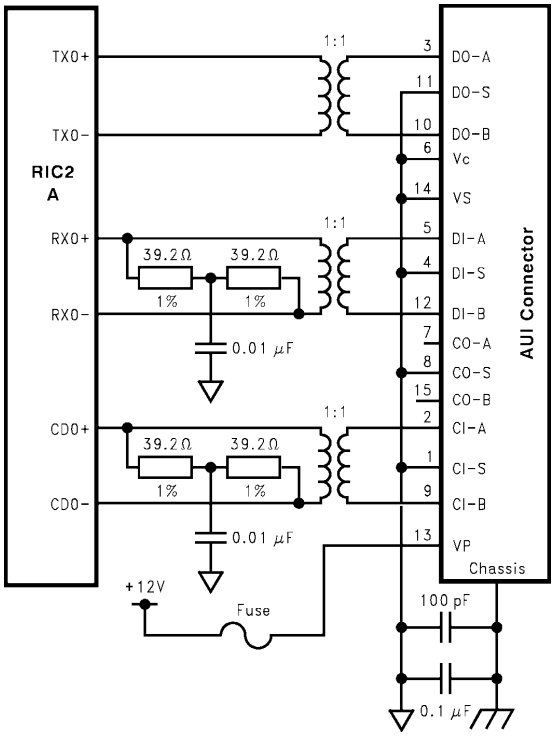



**Figure 17. AUI Port Connection**

### 6.1.1 Magnetics Specifications

This section describes the required magnetics to be used with RIC2A Twisted pair ports. The new transformer is the only external magnetics required. In this configuration, a transformer requires a 1:2 turn ratio on the transmit path and a 1:1 turn ratio on the receive path. In addition, the system designer must determine if chokes are necessary. They may not be when careful layout techniques are employed.

### 6.1.2 IEEE Conformance

The RIC2A was tested for IEEE conformance on different platforms. This testing brought out a number of points that designers should be aware of when developing a RIC2A based system. Specifically, designers may want to make provisions to optimize their systems for conformance to Receive Noise Immunity and Transmitter Differential Output Voltage, as these have shown to be the most difficult items in the specification to meet.

**Receiver Differential Noise Immunity**

A repeater using the RIC2A may have difficulty meeting all parameters of Receiver Differential Noise Immunity (IEEE 802.3 section 14.3.1.3.2) without strict layout and design considerations aimed at reducing the reception of 20, 25 and 30 MHz signals. Even though a system shy of these specification sections shall function flawlessly in commercial environments, National Semiconductor recommends a 3 pole low-pass Butterworth receive filter with a cut off frequency of 15 MHz for those concerned about full IEEE compliance. Figure 21 is an example of such a filter. Systems implementing this filter have been shown to comply with the noise immunity specification.

Filters like this are often packaged within magnetics modules. These modules are currently available from Halo, Belfuse, Pulse, and Valor magnetics suppliers. This is a point for reference only. National Semiconductor does not qualify, recommend or claim conformance with any such device.

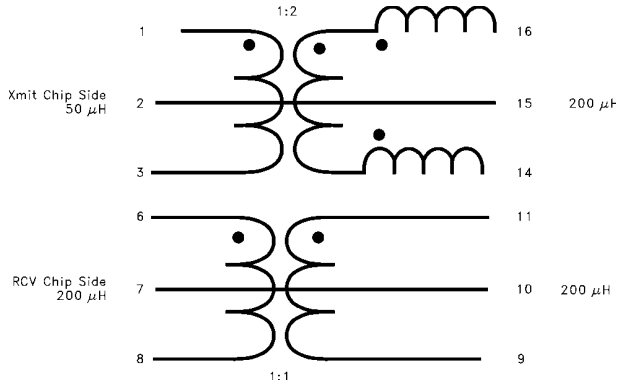
**Peak Differential Output Voltage ($V_{OD}$)**

Without any resistive loading on the $R_{EQ}$ and $R_{TX}$ pins, a RIC2A based repeater will pass all conformance tests with the possible exception of Peak Differential Output Voltage ($V_{OD}$ into a resistive load). With only one twisted pair port loaded and transmitting a packet, and with the system running at 5.25V in a $0^{o}$C chamber, the output may marginally violate the Peak Differential Output Voltage ($V_{OD}$ into a resistive load) at the upper limit. With all twelve twisted pair ports loaded and transmitting, and with the system running at 4.75V in an $80^{o}$C chamber, the output may marginally violate Peak Differential Output Voltage ($V_{OD}$ into a resistive load) at the lower limit.

Please note that this parameter is also related to layout considerations, so these results may not be observed with every design. Also, the violation of this specification under these conditions will not affect a normal network. The RIC2A has undergone endurance testing in many platforms and has not shown any loss of data attributed to out of spec $V_{OD}$.

A discussion of $V_{OD}$ on the RIC2A would not be complete without the inclusion of comments about the RTX and REQ pins. The RTX and REQ pins can be used to tune the internal transmit filter and wave shaping circuitry. The RTX input can be used to adjust the differential voltage ($V_{OD}$) of all output drivers. By placing a resistor between RTX and $V_{DD}$, the peak-to-peak Voltage will be increased. Conversely,

## 6.0 Port Block Functions (Continued)




Note 1: 1:2 turn ratio on transmit and 1:1 turn ratio on receive sides.

Note 2: Transmit common mode chokes may not be required.

Note 3: Please consult individual vendors for pin outline. Typical 16 pin module shown.

**Electrical Specification:**

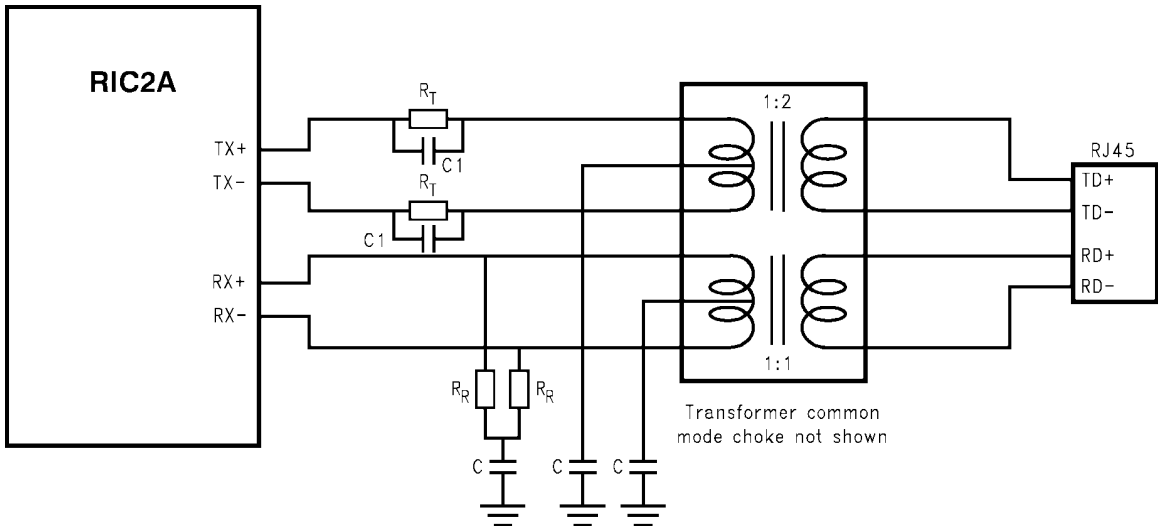
| | | |
|---|---|---|
| OCL | (1-3) | min. 50 µH @ 1 MHz, 0.1Vrms |
| | (6-8), (9-11), (14-16) | min. 200 µH @ 1 MHz, 0.1Vrms |
| Cww | (1-3) to (14-16) | 12 pF max @ 1 MHz, 0.1Vrms |
| | (6-8) to (9-11) | 12 pF max @ 1 MHz, 0.1Vrms |
| LL | (1,3 & short 14,16) | 0.3 µH max @ 1MHz, 0.1Vrms |
| | (6,8 & short 9,11) | 0.3 µH max @ 1MHz, 0.1Vrms |
| DCR | (1-3) | 0.35 Ω max |
| | (6-8) = (11-9) | 0.5 Ω max |
| | (14-16) | 1.0 Ω max |

| | | |
|---|---|---|
| HIPOT | (1,3 to 14,16) | |
| | (6,8 to 9,11) | 2000 Vrms for 1 minute |

**Glossary:**

| | |
|---|---|
| OCL: | Open circuit inductance |
| Cww: | Interwinding capacitance |
| LL: | Leakage inductance |
| DCR: | DC resistance |

**Figure 18. Magnetics Requirements**




Where C = 0.01 µF, C1 = 1.0 nF, $R_R$ = 49.9Ω, and $R_T$ = 10Ω. All values are typical and ± 1%.

**Figure 19. Twisted Pair Interface to the Transformer and RJ45**

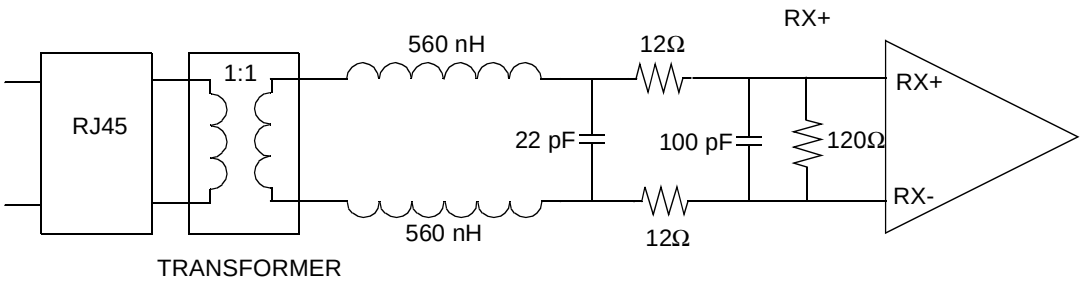



**Figure 20. Sample Twisted Pair Receive Filter**

connecting the resistor to GND will decrease $V_{OD}$. The degree of change is related to the resistor value.

The REQ input can be used to adjust the shape of the waveform for all outputs. By placing a resistor between REQ and $V_{DD}$, the amplitude of the pre-emphasis waveform will be increased. Conversely, connecting the resistor to GND will decrease the amplitude. As with RTX, the degree of change here is related to the resistor value.

**Early Link Pulses**

IEEE 802.3 specification, section 14.3.1.2.3 and 14.3.1.2.1 can be interpreted as requiring a period of silence between repeated packets ranging from 8 to 24 mS. The RIC2A may, however, send an early link pulse as soon as 200nS after successfully transmitting a packet. This may be considered an IEEE compliance issue, but National Semiconductor views early link pulses as having no impact on system performance. Again, the RIC2A has undergone extensive endurance testing sessions and has not shown any loss of data.

### 6.2 Segment Partition

The RIC2A's ports have dedicated partition state machines to perform all functions defined by the IEEE algorithm. Refer to the "Partitioning State Diagram for Port X", Figure 9-6 in the IEEE 802.3 Repeater Specifications. Several device configuration options are available to customize this algorithm for various applications during power up (the Mode Load cycle).

The RIC2A provides five different options:

1. Operation of the 13 partition state machines may be disabled via the disable partition $\overline{\text{DPART}}$ configuration bit (pin D6).
2. The value of consecutive collision counts required to partition a segment (the CCLimit specification) may be set at either 31 or 63 consecutive collisions.
3. The use of the TW5 specification in the partition algorithm differentiates between collisions that occur early in a packet (before TW5 has elapsed) and those that occur late in the packet (after TW5 has elapsed). These late or "out of window" collisions can be regarded in the same manner as early collisions if the Out of Window Collision Enable $\overline{\text{OWCE}}$ option is selected. This configuration bit is applied to the D4 pin during the Mode Load operation.
4. The operation of the ports' state machines reconnecting of a segment may also be modified by the user. The Transmit Only (/TXONLY) configuration bit allows the designer to prevent segment reconnection unless the reconnecting packet is sourced by the repeater. For this case, the repeater transmits on to the segment rather than the segment transmitting when the repeater is idle. The normal reconnection mode does not differentiate between such packets. The /TXONLY configuration bit is input on pin D5 during the Mode Load cycle.
5. The RIC2A may be configured to use an additional criterion for segment partition. This is referred to as loopback partition. If this operation is selected, the partition state machine monitors the receive and collision inputs from a network segment to discover if they are active when the port is transmitting. This determines if the network transceiver is looping back the data pattern from the cable. A port may be partitioned if no data or collision signals are seen by the partition logic in the following window: 61 to 96 network bit times after the start of transmission. See datasheet Section 7.0 for details. A segment partitioned by this operation may be reconnected in the normal manner.

### 6.3 Port Status Register Functions

All RIC2A ports have their own port status registers. Additionally, these registers provide pertinent status information concerning the port and the network segment such as the following operations:

1. Port Disable
2. Link Disable
3. Partition Reconnection
4. Selection between normal and reduced squelch levels

Note that the link disable and port disable options are mutually exclusive functions. For example, disabling link does not affect receiving and transmitting from/to that port and disabling a port does not disable link.

When a port is disabled, packet transmission and reception between the port's segment and the rest of the network is prevented.

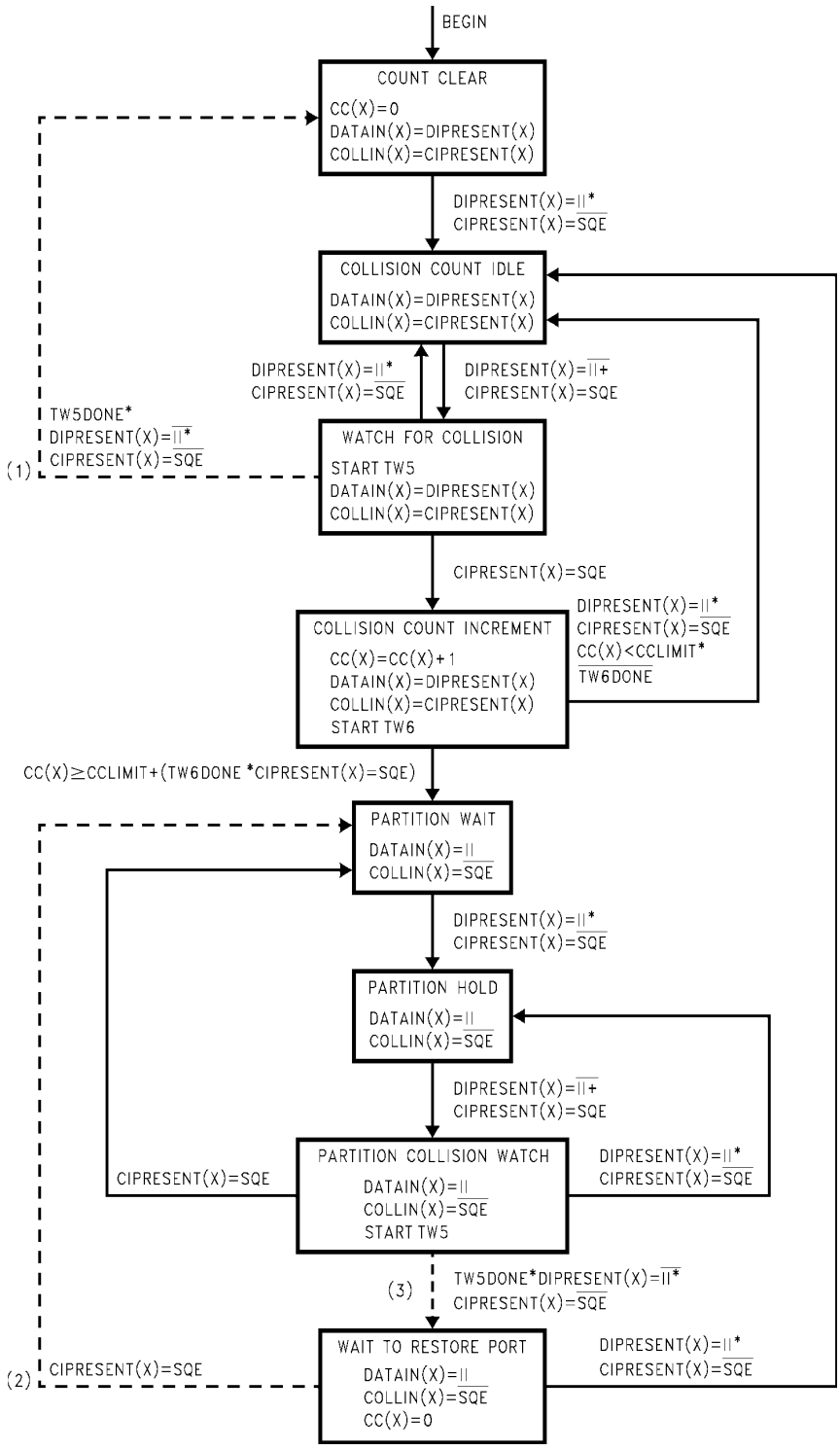



**Figure 21. IEEE Segment Partition Algorithm**

## 6.0 Port Block Functions (Continued)

### 6.4 Local Ports and Expected Activity

The RIC2A incorporates security options into the repeater. The configuration of the security features can be performed globally or on a per port basis. Upon packet reception by the RIC2A, depending on port configuration, the repeater will either: transmit the actual data intact to the port, or transmit pseudo random data to the port during the data field of the packet.

RIC2A security features can be globally enabled/disabled during the MLOAD process, or via the RIC2A Configuration Register bit GSE (global security enable). When GSE is set, the device will, for all ports, set the port SME (Security Mode), ESA (Source Address Security), EDA (Destination Address Security), LME (Learn Mode Enable) bits in the Port Security Configuration Register (PSCR).

Learning Mode for all the port CAMs can be globally enabled during the MLOAD process, or via the GLME (Global Learn Mode Enable) bit in the Global Security Register (GSR). When GLME is set, the device will, for all ports, set the LME (Learn Mode Enable) bit in the PSCR register.

In a multi RIC2A repeater environment, each RIC2A will get the packet intact over the InterRIC bus (except those with mismatched source addresses when configured in the security mode). Each RIC2A will transmit either the real, intact data or pseudo random data depending on the port configuration.

#### Configuration of the Source and Destination Address Security Modes

1. Security Mode Disabled

   RIC2A performs the same repeater operations as the RIC. The received data is transmitted to all ports, and on the InterRIC bus.

2. Security Mode Enabled

   a. When a port's ESA =0 and,

      i. The port's EDA=0, then the repeater will repeat the data on the port, and the Inter-RIC bus.

      ii. The port's EDA=1, then the repeater will repeat the data on destination address match. On a destination address mismatch, the repeater will transmit random data on that port. In both cases, the repeater will transmit data on the Inter-RIC bus.

   b. When a port's ESA=1 and,

      i. The port's EDA=0, then on a valid source address match, the repeater will repeat the data on that port, and on the Inter-RIC bus. If source address mismatch occurs, then the repeater will transmit random data to the port, and on the Inter-RIC bus.

      ii. The port's EDA=1, then on a valid source and destination address match, the repeater will repeat the data on the port. If source address matches, but the destination address does not match, then the repeater will transmit random data to that port. In both of these cases, the repeater will repeat the data on the Inter-RIC bus. When source address mismatch occurs, then the repeater will transmit random data to the port and on the Inter-RIC bus.

Table 7 describes the types of transmitted data, either actual or pseudo random data, of each port and over the Inter-RIC bus. It is assumed that the repeater is powered up in security mode (GLME=0).

For example, suppose the repeater is in security mode (SME=1) and configured to perform address comparison only on destination addresses (ESA=0 and EDA=1). If a packet is received whose destination address does not match with that stored address in a designated CAM, then all the transmitting ports switch to random packet, while the data is transmitted intact over the Inter-RIC bus. The other cascaded repeaters will compare the packet's destination address with their own internal CAMs for proper decision making.

**Table 7. Local Ports and Inter-RIC Bus Data Field Contents**

| SME | ESA | EDA | Source Address of Packet | Destination Address of Packet | Transmitting ports | Inter-RIC Bus |
|---|---|---|---|---|---|---|
| 0 | X | X | X | X | repeat | repeat |
| | | 0 | X | X | repeat | repeat |
| | | | match | match | repeat | repeat |
| | 0 | | match | mismatch | random | repeat |
| | | 1 | mismatch | match | repeat | repeat |
| 1 | | | mismatch | mismatch | random | repeat |
| | | | match | match | repeat | repeat |
| | | | match | mismatch | repeat | repeat |
| | 1 | 0 | mismatch | X | random | random |
| | | | match | match | repeat | repeat |
| | | 1 | match | mismatch | random | repeat |
| | | | mismatch | X | random | random |

Note: SME: Security Mode bit in the Port Security Configuration Register (PSCR).
ESA: Source Address Security bit in the PSCR register.
EDA: Destination Address Security bit in the PSCR register.

# 7.0 RIC2A Registers

## RIC2A Register Address Map

The RIC2A's registers may be accessed by applying the required address to the five Register Address (RA(4:0)) input pins. Pin RA4 makes the selection between the upper and lower halves of the register array. The lower half of the register map consists of 16 registers:

- one RIC2A Real Time Status and Configuration register,
- 13 Port Real Time Status registers,
- one RIC2A Configuration Register
- one Real Time Interrupt Status Register.

These registers may be directly accessed at any time via the RA(4:0) pins, (RA4 = 0).

The upper half of the register map, (RA4 = 1), is organized as 15 pages of registers. These pages include registers for port security configuration (global and on a per port basis), event count registers, port CAM and shared CAM locations, CAM location mask registers, etc. See Memory Map and Register Description sections for details.

Register access within these pages is performed using the RA(4:0) pins, (RA4 = 1). Page switching is performed by writing to the Page Selection bits (PSEL3,2,1, and 0). These bits are found in the Page Select Register, located at address 10 hex on each page of the upper half of the register array. At power on these bits default to 0 Hex, i.e., page zero.

**Table 8. RIC2A Register Address Map**

| | Name | | | |
|---|---|---|---|---|
| **Address** | **PAGE (0)** | **PAGE (1)** | **PAGE (2)** | **PAGE (3)** |
| 00H | RIC2A Status and Configuration Register | | | |
| 01H | Port 1 Status Register | | | |
| 02H | Port 2 Status Register | | | |
| 03H | Port 3 Status Register | | | |
| 04H | Port 4 Status Register | | | |
| 05H | Port 5 Status Register | | | |
| 06H | Port 6 Status Register | | | |
| 07H | Port 7 Status Register | | | |
| 08H | Port 8 Status Register | | | |
| 09H | Port 9 Status Register | | | |
| 0AH | Port 10 Status Register | | | |
| 0BH | Port 11 Status Register | | | |
| 0CH | Port 12 Status Register | | | |
| 0DH | Port 13 Status Register | | | |
| 0EH | RIC2A Configuration Register | | | |
| 0FH | Real Time Interrupt Register | | | |
| 10H | Page Select Register | | | |
| 11H | Device Type Register | Port 1 Event Record Register (ERR) | res | res |
| 12H | Lower Event Count Mask Register (ECMR) | Port 2 ERR | Port 1 Lower Event Count Register (ECR) | Port 8 Lower ECR |
| 13H | Upper ECMR | Port 3 ERR | Port 1 Upper ECR | Port 8 Upper ECR |
| 14H | Event Record Mask Register | Port 4 ERR | Port 2 Lower ECR | Port 9 Lower ECR |
| 15H | **ECIMR - 2** | Port 5 ERR | Port 2 Upper ECR | Port 9 Upper ECR |
| 16H | Management/Interrupt Configuration Register | Port 6 ERR | Port 3 Lower ECR | Port 10 Lower ECR |
| 17H | RIC2A Address Register | Port 7 ERR | Port 3 Upper ECR | Port 10 Upper ECR |
| 18H | Packet Compress Decode Register | Port 8 ERR | Port 4 Lower ECR | Port 11 Lower ECR |

Note: Registers written in **bold** are not present in the RIC.

## 7.0 RIC2A Registers (Continued)

**Table 8. RIC2A Register Address Map (Continued)**

| | | | | |
|---|---|---|---|---|
| 19H | res | Port 9 ERR | Port 4 Upper ECR | Port 11 Upper ECR |
| 1AH | res | Port 10 ERR | Port 5 Lower ECR | Port 12 Lower ECR |
| 1BH | res | Port 11 ERR | Port 5 Upper ECR | Port 12 Upper ECR |
| 1CH | res | Port 12 ERR | Port 6 Lower ECR | Port 13 Lower ECR |
| 1DH | **GSR** | Port 13 ERR | Port 6 Upper ECR | Port 13 Upper ECR |
| 1EH | res | **Upper EIR** | Port 7 Lower ECR | res |
| 1FH | IFG Threshold Select | **Lower EIR** | Port 7 Upper ECR | res |

**Name**

| **Address** | **PAGE (4)** | **PAGE (5)** | **PAGE (6)** | **PAGE (8)** |
|---|---|---|---|---|
| 11H | Port 1 ECR-2 | Port 1 CAM 1 | Port 5 PCPR | Port 9 PSCR |
| 12H | Port 2 ECR-2 | Port 1 CAM 2 | Port 5 CAM 1 | Port 9 PCPR |
| 13H | Port 3 ECR-2 | Port 2 PSCR | Port 5 CAM 2 | Port 9 CAM 1 |
| 14H | Port 4 ECR-2 | Port 2 PCPR | Port 6 PSCR | Port 9 CAM 2 |
| 15H | Port 5 ECR-2 | Port 2 CAM 1 | Port 6 PCPR | Port 10 PSCR |
| 16H | Port 6 ECR-2 | Port 2 CAM 2 | Port 6 CAM 1 | Port 10 PCPR |
| 17H | Port 7 ECR-2 | Port 3 PSCR | Port 6 CAM 2 | Port 10 CAM 1 |
| 18H | Port 8 ECR-2 | Port 3 PCPR | Port 7 PSCR | Port 10 CAM 2 |
| 19H | Port 9 ECR-2 | Port 3 CAM 1 | Port 7 PCPR | Port 11 PSCR |
| 1AH | Port 10 ECR-2 | Port 3 CAM 2 | Port 7 CAM 1 | Port 11 PCPR |
| 1BH | Port 11 ECR-2 | Port 4 PSCR | Port 7 CAM 2 | Port 11 CAM 1 |
| 1CH | Port 12 ECR-2 | Port 4 PCPR | Port 8 PSCR | Port 11 CAM 2 |
| 1DH | Port 13 ECR-2 | Port 4 CAM 1 | Port 8 PCPR | Port 12 PSCR |
| 1EH | Port 1 PSCR | Port 4 CAM 2 | Port 8 CAM 1 | Port 12 PCPR |
| 1FH | Port 1 PCPR | Port 5 PSCR | Port 8 CAM 2 | Port 12 CAM 1 |

**Name**

| **Address** | **PAGE (9)** | **PAGE (10)** | **PAGE (11)** | **PAGE (12)** |
|---|---|---|---|---|
| 11H | Port 12 CAM 2 | SCAM Lo 3 | SCAM Lo 8 | SCAM Lo 13 |
| 12H | Port 13 PSCR | CLMR Lo Loc 3 | CLMR Lo Loc 8 | CLMR Lo Loc 13 |
| 13H | Port 13 PCPR | CLMR Hi Loc 3 | CLMR Hi Loc 8 | CLMR Hi Loc 13 |
| 14H | Port 13 CAM 1 | SCAM Lo 4 | SCAM Lo 9 | SCAM Lo 14 |
| 15H | Port 13 CAM 2 | CLMR Lo Loc 4 | CLMR Lo Loc 9 | CLMR Lo Loc 14 |
| 16H | SCVR 1 | CLMR Hi Loc 4 | CLMR Hi Loc 9 | CLMR Hi Loc 14 |
| 17H | SCVR 2 | SCAM Lo 5 | SCAM Lo 10 | SCAM Lo 15 |
| 18H | SCVR 3 | CLMR Lo Loc 5 | CLMR Lo Loc 10 | CLMR Lo Loc 15 |
| 19H | SCVR 4 | CLMR Hi Loc 5 | CLMR Hi Loc 10 | CLMR Hi Loc 15 |
| 1AH | SCAM Lo 1 | SCAM Lo 6 | SCAM Lo 11 | SCAM Lo 16 |
| 1BH | CLMR Lo Loc 1 | CLMR Lo Loc 6 | CLMR Lo Loc 11 | CLMR Lo Loc 16 |
| 1CH | CLMR Hi Loc 1 | CLMR Hi Loc 6 | CLMR Hi Loc 11 | CLMR Hi Loc 16 |
| 1DH | SCAM Lo 2 | SCAM Lo 7 | SCAM Lo 12 | SCAM Lo 17 |
| 1EH | CLMR Lo Loc 2 | CLMR Lo Loc 7 | CLMR Lo Loc 12 | CLMR Lo Loc 17 |
| 1FH | CLMR Hi Loc 2 | CLMR Hi Loc 7 | CLMR Hi Loc 12 | CLMR Hi Loc 17 |

Note: Registers written in **bold** are not present in the RIC.

## 7.0 RIC2A Registers (Continued)

**Table 8. RIC2A Register Address Map (Continued)**

| | Name | | | |
|---|---|---|---|---|
| **Address** | **PAGE (13)** | **PAGE (14)** | **PAGE (15)** | |
| 11H | SCAM Lo 18 | SCAM Lo 23 | SCAM Lo 28 | |
| 12H | CLMR Lo Loc 18 | CLMR Lo Loc 23 | CLMR Lo Loc 28 | |
| 13H | CLMR Hi Loc 18 | CLMR Hi Loc 23 | CLMR Hi Loc 28 | |
| 14H | SCAM Lo 19 | SCAM Lo 24 | SCAM Lo 29 | |
| 15H | CLMR Lo Loc 19 | CLMR Lo Loc 24 | CLMR Lo Loc 29 | |
| 16H | CLMR Hi Loc 19 | CLMR Hi Loc 24 | CLMR Hi Loc 20 | |
| 17H | SCAM Lo 20 | SCAM Lo 25 | SCAM Lo 30 | |
| 18H | CLMR Lo Loc 20 | CLMR Lo Loc 25 | CLMR Lo Loc 30 | |
| 19H | CLMR Hi Loc 20 | CLMR Hi Loc 25 | CLMR Hi Loc 30 | |
| 1AH | SCAM Lo 21 | SCAM Lo 26 | SCAM Lo 31 | |
| 1BH | CLMR Lo Loc 21 | CLMR Lo Loc 26 | CLMR Lo Loc 31 | |
| 1CH | CLMR Hi Loc 21 | CLMR Hi Loc 26 | CLMR Hi Loc 31 | |
| 1DH | SCAM Lo 22 | SCAM Lo 27 | SCAM Lo 32 | |
| 1EH | CLMR Lo Loc 22 | CLMR Lo Loc 27 | CLMR Lo Loc 32 | |
| 1FH | CLMR Hi Loc 22 | CLMR Hi Loc 27 | CLMR Hi Loc 32 | |

Note: Registers written in **bold** are not present in the RIC.

**Register Array Bit Map Addresses 00H to 10H**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 00 | BINV | X | X | $\overline{\text{APART}}$ | $\overline{\text{JAB}}$ | $\overline{\text{AREC}}$ | /ACOL | resv |
| 01 to 0D | DISPT | SQL | PTYPE1 | PTYPE0 | $\overline{\text{PART}}$ | $\overline{\text{REC}}$ | $\overline{\text{COL}}$ | $\overline{\text{GDLNK}}$ |
| 0E | MINMAX | $\overline{\text{DPART}}$ | /TXONLY | $\overline{\text{OWCE}}$ | /LPPART | $\overline{\text{CCLIM}}$ | TW2 | GSE |
| 0F | IVCTR3 | IVCTR2 | IVCTR1 | IVCTR0 | ISRC3 | ISRC2 | ISRC1 | ISRC0 |
| 10 | FC | HC | LC | FF | PSEL3 | PSEL2 | PSEL1 | PSEL0 |

**Register Array Bit Map Addresses 11H to 1FH Page (0)**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 11 | 1 | 0 | 0 | 1 | 0 | 0 | X | X |
| 12 | BDLNKC | PARTC | RECC | SEC | NSFDC | PLERC | ELBERC | JABC |
| 13 | resv | resv | OWCC | RXCOLC | TXCOLC | resv | FWF | ROR |
| 14 | BDLNKE | PARTE | OWCE | SEE | NSFDE | PLERE | ELBERE | JABE |
| 15 | res | ISAM | FWF-2 | ROR-2 | OWCC-2 | PARTC-2 | FAEC | FCSC |
| 16 | /IFC | /IHC | /ILC | /IFF | $\overline{\text{IREC}}$ | /ICOL | /IPART | MIFCON |
| 17 | A5 | A4 | A3 | A2 | A1 | A0 | resv | resv |
| 18 | PCD7 | PCD6 | PCD5 | PCD4 | PCD3 | PCD2 | PCD1 | PCD0 |
| 1D | res | GLME | res | DSM | res | MPS | GRP | SAC |
| 1F | IFGT7 | IFGT6 | IFGT5 | IFGT4 | IFGT3 | IFGT2 | IFGT1 | IFGT0 |

## 7.0 RIC2A Registers (Continued)

**Register Array Bit Map Addresses 11H to 1FH Page (1)**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 11 to 1D | BDLNK | PART | OWC | SE | NSFD | PLER | ELBER | JAB |
| 1E | ER8 | ER7 | ER6 | ER5 | ER4 | ER3 | ER2 | ER1 |
| 1F | DLU | res | res | ER13 | ER12 | ER11 | ER10 | ER9 |

**Register Array Bit Map Addresses 11H to 1FH Pages (2) and (3)**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 11 | -- | -- | -- | -- | -- | -- | -- | -- |
| even locations | EC7 | EC6 | EC5 | EC4 | EC3 | EC2 | EC1 | EC0 |
| odd locations | EC15 | EC14 | EC13 | EC12 | EC11 | EC10 | EC9 | EC8 |

**Register Array Bit Map Addresses 11H to 1FH Page (4)**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 11 to 1D | EC7 | EC6 | EC5 | EC4 | EC3 | EC2 | EC1 | EC0 |
| 1E | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 1F | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |

**Register Array Bit Map Addresses 11H to 1FH Page (5)**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 11, 12 | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |
| 13 | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 14 | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 15, 16 | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |
| 17 | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 18 | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 19, 1A | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |
| 1B | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 1C | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 1D, 1E | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |
| 1F | res | EDA | ESA | SAM | MCE | BCE | SME | LME |

## 7.0 RIC2A Registers (Continued)

**Register Array Bit Map Addresses 11H to 1FH Page (6)**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 11 | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 12, 13 | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |
| 14 | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 15 | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 16, 17 | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |
| 18 | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 19 | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 1A, 1B | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |
| 1C | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 1D | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 1E, 1F | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |

**Register Array Bit Map Addresses 11H to 1FH Page (8)**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 11 | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 12 | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 13, 14 | PCAMx _D7 | PCAMx _D6 | PCAMx _D5 | PCAMx _D4 | PCAMx _D3 | PCAMx _D2 | PCAMx _D1 | PCAMx _D0 |
| 15 | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 16 | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 17, 18 | PCAMx _D7 | PCAMx _D6 | PCAMx _D5 | PCAMx _D4 | PCAMx _D3 | PCAMx _D2 | PCAMx _D1 | PCAMx _D0 |
| 19 | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 1A | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 1B, 1C | PCAMx _D7 | PCAMx _D6 | PCAMx _D5 | PCAMx _D4 | PCAMx _D3 | PCAMx _D2 | PCAMx _D1 | PCAMx _D0 |
| 1D | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 1E | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 1F | PCAMx _D7 | PCAMx _D6 | PCAMx _D5 | PCAMx _D4 | PCAMx _D3 | PCAMx _D2 | PCAMx _D1 | PCAMx _D0 |

Note: For Port CAM register bits (PCAMx_D[7:0]) and Shared CAM register bits (SCAM**x**_D[7:0]) **x** represents the port number.

## 7.0 RIC2A Registers (Continued)

**Register Array Bit Map Addresses 11H to 1FH Page (9)**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 11 | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |
| 12 | res | EDA | ESA | SAM | MCE | BCE | SME | LME |
| 13 | ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |
| 14, 15 | PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |
| 16 | ADV8 | ADV7 | ADV6 | ADV5 | ADV4 | ADV3 | ADV2 | ADV1 |
| 17 | ADV16 | ADV15 | ADV14 | ADV13 | ADV12 | ADV11 | ADV10 | ADV9 |
| 18 | ADV24 | ADV23 | ADV22 | ADV21 | ADV20 | ADV19 | ADV18 | ADV17 |
| 19 | ADV32 | ADV31 | ADV30 | ADV29 | ADV28 | ADV27 | ADV26 | ADV25 |
| 1A | SCAMx_D7 | SCAMx_D6 | SCAMx_D5 | SCAMx_D4 | SCAMx_D3 | SCAMx_D2 | SCAMx_D1 | SCAMx_D0 |
| 1B | P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |
| 1C | PTR2 | PTR1 | PTR0 | P13 | P12 | P11 | P10 | P9 |
| 1D | SCAMx_D7 | SCAMx_D6 | SCAMx_D5 | SCAMx_D4 | SCAMx_D3 | SCAMx_D2 | SCAMx_D1 | SCAMx_D0 |
| 1E | P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |
| 1F | PTR2 | PTR1 | PTR0 | P13 | P12 | P11 | P10 | P9 |

**Register Array Bit Map Addresses 11H to 1FH Pages (AH, BH, CH, DH, EH, FH)**

| Address (Hex) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 11 | SCAMx_D7 | SCAMx_D6 | SCAMx_D5 | SCAMx_D4 | SCAMx_D3 | SCAMx_D2 | SCAMx_D1 | SCAMx_D0 |
| 12 | P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |
| 13 | PTR2 | PTR1 | PTR0 | P13 | P12 | P11 | P10 | P9 |
| 14 | SCAMx_D7 | SCAMx_D6 | SCAMx_D5 | SCAMx_D4 | SCAMx_D3 | SCAMx_D2 | SCAMx_D1 | SCAMx_D0 |
| 15 | P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |
| 16 | PTR2 | PTR1 | PTR0 | P13 | P12 | P11 | P10 | P9 |
| 17 | SCAMx_D7 | SCAMx_D6 | SCAMx_D5 | SCAMx_D4 | SCAMx_D3 | SCAMx_D2 | SCAMx_D1 | SCAMx_D0 |
| 18 | P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |
| 19 | PTR2 | PTR1 | PTR0 | P13 | P12 | P11 | P10 | P9 |
| 1A | SCAMx_D7 | SCAMx_D6 | SCAMx_D5 | SCAMx_D4 | SCAMx_D3 | SCAMx_D2 | SCAMx_D1 | SCAMx_D0 |
| 1B | P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |
| 1C | PTR2 | PTR1 | PTR0 | P13 | P12 | P11 | P10 | P9 |
| 1D | SCAMx_D7 | SCAMx_D6 | SCAMx_D5 | SCAMx_D4 | SCAMx_D3 | SCAMx_D2 | SCAMx_D1 | SCAMx_D0 |
| 1E | P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |
| 1F | PTR2 | PTR1 | PTR0 | P13 | P12 | P11 | P10 | P9 |

Note: For Port CAM register bits (PCAMx_D[7:0]) and Shared CAM register bits (SCAM**x**_D[7:0]) **x** represents the port number.

## 7.0 RIC2A Registers (Continued)

### RIC2A Status and Configuration Register (Address 00H)

The lower portion of this register contains real time information concerning the operation of the RIC2A. The D7 bit represent the chosen configuration of the transceiver interface employed.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| BINV | resv | resv | $\overline{\text{APART}}$ | $\overline{\text{JAB}}$ | $\overline{\text{AREC}}$ | $\overline{\text{ACOL}}$ | resv |

| Bit | R/W | Symbol Access | Description |
|---|---|---|---|
| D0 | R | resv | **Reserved for future use**<br>Reads as a logic 1. |
| D1 | R | $\overline{\text{ACOL}}$ | **Any Collisions**<br>0: A collision is occurring at one or more of the RIC2A's ports.<br>1: No collisions. |
| D2 | R | $\overline{\text{AREC}}$ | **Any Receive**<br>0: one of the RIC2A's ports is the current packet or collision receiver.<br>1: No packet or collision reception within this RIC2A. |
| D3 | R | $\overline{\text{JAB}}$ | **Jabber Protect**<br>0: The RIC2A has been forced into jabber protect state by one of its ports or by another port on the Inter-RIC bus, (Multi-RIC2A operations).<br>1: No jabber protect conditions exist. |
| D4 | R | $\overline{\text{APART}}$ | **Any Partition**<br>0: One or more ports are partitioned.<br>1: No ports are partitioned. |
| D5 | R | resv | **Reserved** |
| D6 | R | resv | **Reserved**<br>Reads as a logic 1. |
| D7 | R | BINV | **Bus Invert**<br>This register bit informs whether the Inter-RIC signals: IRE, ACTN, ANYXN, COLN and Management bus signal MCRS are:<br>0: active high.<br>1: active low. |

## 7.0 RIC2A Registers (Continued)

### Port Real Time Status Registers (Address 01H to 0DH)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| DISPT | SQL | PTYPE1 | PTYPE0 | $\overline{\text{PART}}$ | $\overline{\text{REC}}$ | $\overline{\text{COL}}$ | $\overline{\text{GDLNK}}$ |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | $\overline{\text{GDLNK}}$ | **Good link**<br>0: Link pulses are being received by the port.<br>1: Link pulses are not being received by the port logic.<br>Note: Writing a 1 to this bit will cause the 10Base-T transceiver not to transmit or monitor the reception of link pulses. If port 1 (AUI port) is read, then this bit is undefined. |
| D1 | R | $\overline{\text{COL}}$ | **Collision**<br>0: A collision is happening or has occurred during the current packet.<br>1: No collisions have occurred as yet during this packet. |
| D2 | R | $\overline{\text{REC}}$ | **Receive**<br>0: This port is now or has been the receive source of packet or collision information for the current packet.<br>1: This port has not been the receive source during the current packet. |
| D3 | R/W | $\overline{\text{PART}}$ | **Partition**<br>0: This port is partitioned.<br>1: This port is not partitioned.<br>Writing a logic one to this bit forces segment reconnection and partition state machine reset. Writing a zero to this bit has no effect. |
| D(5,4) | R | PTYPE0<br>PTYPE1 | **Partition Type 0**<br>**Partition Type 1**<br>The partition type bits provide information specifying why the port is partitioned.<br><br>**PTYPE1** / **PTYPE0** / **Information**<br>0 / 0 / Consecutive collision limit reached<br>0 / 1 / Excessive length of collision limit reached<br>1 / 0 / Failure to see data loopback from transceiver in monitored window<br>1 / 1 / Processor forced reconnection |
| D6 | R/W | SQL | **Squelch Levels**<br>0: Port operates with normal IEEE receive squelch level.<br>1: Port operates with reduced receive squelch level.<br>Note 1: In addition to hysteresis that DP83950 RIC provides on normal receive squelch, DP83953 RIC2A provides a hysteresis when operating in the reduced squelch level mode. |
| D7 | R/W | DISPT | **Disable Port**<br>0: Port operates as defined by repeater operations.<br>1: All port activity is prevented. |

## 7.0 RIC2A Registers (Continued)

### RIC2A Configuration Register (Address 0EH)

This register displays the state of a number of RIC2A configuration bits loaded during the Mode Load operation.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| MINMAX | $\overline{\text{DPART}}$ | $\overline{\text{TXONLY}}$ | $\overline{\text{OWCE}}$ | $\overline{\text{LPPART}}$ | $\overline{\text{CCLIM}}$ | TW2 | GSE |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R | GSE | **Global Security Enable**<br>0: RIC2A operates in security mode with Learning Mode enabled by default for all ports.<br>1: RIC2A operates in non-security mode. |
| D1 | R | TW2 | **Carrier Recovery Time**<br>0: TW2 set at 5 bits.<br>1: TW2 set at 3 bits. |
| D2 | R | $\overline{\text{CCLIM}}$ | **Consecutive collision limit**<br>0: Consecutive collision limit set at 63 collisions.<br>1: Consecutive collision limit set at 31 collisions. |
| D3 | R | $\overline{\text{LPPART}}$ | **Loopback Partition**<br>0: Partitioning upon lack of loopback from transceivers is enabled.<br>1: Partitioning upon lack of loopback from transceivers is disabled. |
| D4 | R | $\overline{\text{OWCE}}$ | **Out of Window Collision Enable**<br>0: Out of window collisions are treated as in window collisions by the segment partition state machines.<br>1: Out of window collisions are treated as out of window collisions by the segment partition state machines. |
| D5 | R | $\overline{\text{TXONLY}}$ | **Only Reconnect upon Segment Transmission**<br>0: A segment will only be reconnected to the network if a packet transmitted by the RIC2A onto that segment fulfills the requirements of the segment reconnection algorithm.<br>1: A segment will be reconnected to the network by any packet on the network that fulfills the requirements of the segment reconnection algorithm. |
| D6 | R | $\overline{\text{DPART}}$ | **Disable Partition**<br>0: Partitioning of ports by on-chip algorithms is prevented.<br>1: Partitioning of ports by on-chip algorithms is enabled. |
| D7 | R | MINMAX | **Minimum / Maximum Display Mode**<br>0: LED display set in minimum display mode.<br>1: LED display set in maximum display mode. |

## 7.0 RIC2A Registers (Continued)

### Real Time Interrupt Register (Address 0FH)

The Real Time Interrupt register (RTI) contains information which may change on a packet by packet basis. Any remaining interrupts which have not been serviced before the following packet is transmitted are cleared. Since multiple interrupt sources may be displayed by the RTI a priority scheme is implemented. A read cycle to the RTI gives the interrupt source and an address vector indicating the RIC2A port that generated the interrupt.

The order of priority for the display of interrupt information is as follows (in secure mode only):

1. Source Address Mismatch (feature of the RIC2A that is not present in the RIC.)
2. The receive source of network activity (Port N),
3. The first RIC2A port showing collision,
4. A port partitioned or reconnected.

During the repetition of a single packet it is possible that multiple ports may be partitioned or alternatively reconnected. The ports have equal priority in displaying partition/reconnection information. This data is derived from the ports by the RTI register as it polls consecutively around the ports.

Reading the RTI clears the particular interrupt for all cases. If no interrupt sources are active, the RTI returns a no valid interrupt status.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| IVCTR3 | IVCTR2 | IVCTR1 | IVCTR0 | ISRC3 | ISRC2 | ISRC1 | ISRC0 |

| Bit | R/W | Symbol Access | Description |
|---|---|---|---|
| D(3:0) | R | ISCR(3:0) | **Interrupt Source**<br>These four bits indicate the reason why the interrupt was generated. |
| D(7:4) | R | IVCTR(3:0) | **Interrupt Vector**<br>This field defines the port address responsible for generating the interrupt. |

The following table shows the mapping of interrupt sources onto the D3 to D0 pins. Essentially each of the three interrupt sources has a dedicated bit in this field. If a read to the RTI produces a low logic level on one of these bits then the interrupt source may be directly decoded. Associated with the source of the interrupt is the port where the event is occurring. If no unmasked events (receive, collision, etc.) have occurred when the RTI is read, then an all ones pattern is driven by the RIC2A onto the data pins.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Comments |
|---|---|---|---|---|---|---|---|---|
| **PA3** | **PA2** | **PA1** | **PA0** | **1** | **1** | **1** | **0** | **Source Address Mismatch<br>PA(3:0) = port address for the mismatch** |
| PA3 | PA2 | PA1 | PA0 | 1 | 1 | 0 | 1 | first collision<br>PA(3:0) = collision port address |
| PA3 | PA2 | PA1 | PA0 | 1 | 0 | 1 | 1 | receive<br>PA(3:0) = receive port address |
| PA3 | PA2 | PA1 | PA0 | 0 | 1 | 1 | 1 | partition reconnection<br>PA(3:0) = partition port address |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | no valid interrupt |

## 7.0 RIC2A Registers (Continued)

### Page Select Register ((All pages) Address 10H)

The Page Select register performs two functions:

1. It enables switches to be made between register pages,
2. It provides status information regarding the Event Logging Interrupts.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| FC | HC | LC | FF | PSEL3 | PSEL2 | PSEL1 | PSEL0 |

| bit | R/W | Symbol | Description |
|---|---|---|---|
| D(3:0) | R/W | PSEL(3:0) | **Page Select Bits:** When read these bits indicate the currently selected Upper Register Array Page. Write cycles to these locations facilitates page swapping. The page select bits are latched on the rising edge of the read strobe. |
| D4 | R | FF | **Flag Found:** This indicates one of the unmasked event recording latches has been set. |
| D5 | R | LC | **Low Count:** This indicates one of the port event counters has a value less than 00FF Hex. |
| D6 | R | HC | **High Count:** This indicates one of the port event counters has a value greater than C000 Hex. |
| D7 | R | FC | **Full Counter:** This indicates one of the port event counters has a value equal to FFFF Hex. |

### Device Type Register (Page 0H Address 11H)

This register may be used to distinguish different revisions of RIC. It will return the value 91 H for the DP83953 RIC2A device. It will return the value $8X_H$ for the DP83952 RICII device, or the value $0X_H$ for the DP83950 RIC device. Write operations to this register have no effect upon the contents.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | X | X |

### Lower Event Count Mask Register (Page 0H Address 12H)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| BDLNKC | PARTC | RECC | SEC | NSFDC | PLERC | ELBERC | JABC |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | JABC | **Jabber Count Enable:** Enables recording of Jabber Protect events. |
| D1 | R/W | ELBERC | **Elasticity Buffer Error Count Enable:** Enables recording of Elasticity Buffer Error events. |
| D2 | R/W | PLERC | **Phase Lock Error Count Enable:** Enables recording of Carrier Error events. |
| D3 | R/W | NSFDC | **Non SFD Count Enable:** Enables recording of Non SFD packet events. |
| D4 | R/W | SEC | **Short Event Count Enable:** Enables recording of Short events. |
| D5 | R/W | RECC | **Receive Count Enable:** Enables recording of Packet Receive (port N status) events that do not suffer collisions. |
| D6 | R/W | PARTC | **Partition Count Enable:** Enables recording of Partition events. |
| D7 | R/W | BDLNKC | **Bad Link Count Enable:** Enables recording of Bad Link events. |

Note: 1 = enable, 0 = disable

## 7.0 RIC2A Registers (Continued)

### Upper Event Count Mask Register (Page 0H Address 13H)

The bits in this register effect the Upper and Lower Port Event Count Registers (ECR) on Page (3) addresses 12H to 1FH, and Page (4) addresses 12H to 1DH.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| resv | resv | OWCC | RXCOLC | TXCOLC | resv | FWF | ROR |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | ROR | **Reset on Read:** This bit selects the action a read operation has upon a port's event counter:<br>0: No effect upon register contents.<br>1: The counter register is reset. |
| D1 | R/W | FWF | **Freeze When Full:** This bit controls the freezing of the Event Count registers when the counter is full (FFFF Hex). |
| D2 | R | resv | **Reserved for future use:** This bit should be written with a low logic level. |
| D3 | R/W | TXCOLC | **Transmit Collision Count Enable:** Enables recording of transmit collision events only. |
| D4 | R/W | RXCOLC | **Receive Collision Count Enable:** Enables recording of receive collision events only. |
| D5 | R/W | OWCC | **Out of Window Collision Count Enable:** Enables recording of out of window collision events only. |
| D(7:6) | R | resv | **Reserved for future use:** These bits should be written with a low logic level. |

Note: To count all collisions, both the TXCOLC and RXCOLC bits must be set. The OWCC bit should not be set because the port counter will be incremented twice when an out of collision window collision occurs. The OWCC bit alone should be set if only out of window collisions are to be counted.

### Event Record Mask Register (Page 0H Address 14H)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| BDLNKE | PARTE | OWCE | SEE | NSFDE | PLERE | ELBERE | JABE |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | JABE | **Jabber Enable:** Enables recording of Jabber Protect events. |
| D1 | R/W | ELBERE | **Elasticity Buffer Error Enable:** Enables recording of Elasticity Buffer Error events. |
| D2 | R/W | PLERE | **Phase Lock Error Enable:** Enables recording of Carrier Error events. |
| D3 | R/W | NSFDE | **Non SFD Enable:** Enables recording of Non SFD packet events. |
| D4 | R/W | SEE | **Short Event Enable:** Enables recording of Short events. |
| D5 | R/W | OWCE | **Out of Window Collision Count Enable:** Enables recording of Out Of Window Collision events only. |
| D6 | R/W | PARTE | **Partition Enable:** Enables recording of Partition events. |
| D7 | R/W | BDLNKE | **Bad Link Enable:** Enables recording of Bad Link events. |

Note: Writing a 1 enables the event to be recorded.

## 7.0 RIC2A Registers (Continued)

### Event Count and Interrupt Mask Register 2 (ECIMR-2) (Page 0H Address 15H)

The bits in this register effect the Port Event Count Register 2, PECR-2 on Page 4, Addresses 11H to 1DH.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| res | ISAM | FWF-2 | ROR-2 | OWCC-2 | PARTC-2 | FAEC | FCSC |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | FCSC | **Frame Check Sequence Count Enable:** This bit enables counting the packets with frame check sequence error.<br>0: Disable the frame check sequence count.<br>1: Enable the frame check sequence count. |
| D1 | R/W | FAEC | **Frame Alignment Error Count Enable:** This bit enables counting the packets with frame alignment error.<br>0: Disable the frame alignment error count.<br>1: Enable the frame alignment error count. |
| D2 | R/W | PARTC-2 | **Partition Count Enable:** This bit enables recording of partition events.<br>0: Disable the partition count.<br>1: Enable the partition count. |
| D3 | R/W | OWCC-2 | **Out of Window Collision Count Enable:** This bit enables counting of out of window collision events.<br>0: Disable the out of window collision count.<br>1: Enable the out of window collision count. |
| D4 | R/W | ROR-2 | **Reset On Read:** This bit enables the counter register to reset upon reading the port event's counter.<br>0: No effect upon reading the register contents.<br>1: The counter register is reset by reading the contents of the register. |
| D5 | R/W | FWF-2 | **Freeze When Full:** This bit controls the freezing of the Event Count registers when the counter is full (FF Hex).<br>0: No effect on the event count register.<br>1: Freeze the event count register when the counter is full. |
| D6 | R/W | $\overline{\text{ISAM}}$ | **Interrupt on the Source Address Mismatch Mask**<br>0: Interrupts will be generated on a source address mismatch mask. ($\overline{\text{RTI}}$ pin becomes active.)<br>1: No interrupts are generated. |
| D7 | R | resv | **Reserved for Future Use**<br>reads as a logic 0 |

## 7.0 RIC2A Registers (Continued)

### Interrupt and Management Configuration Register (Page 0H Address 16H)

This register powers up with all bits set to one and must be initialized by a processor write cycle before any events will generate interrupts.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{IFC}}$ | $\overline{\text{IHC}}$ | $\overline{\text{ILC}}$ | $\overline{\text{IFF}}$ | $\overline{\text{IREC}}$ | $\overline{\text{ICOL}}$ | $\overline{\text{IPART}}$ | MIFCON |

| bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | MIFCON | **Management Interface Configuration**<br>0: All Packets repeated are transmitted over the Management bus.<br>1: Packets repeated by the RIC2A that do not have Start of Frame Delimiters are not transmitted over the Management bus. |
| D1 | R/W | $\overline{\text{IPART}}$ | **Interrupt on Partition**<br>0: Interrupts will be generated ($\overline{\text{RTI}}$ pin goes active) if a port becomes Partitioned.<br>1: No interrupts are generated by this condition. |
| D2 | R/W | $\overline{\text{ICOL}}$ | **Interrupt on Collision**<br>0: Interrupts will be generated ($\overline{\text{RTI}}$ pin goes active) if this RIC2A has a port that experiences a collision (single RIC2A applications,) or contains a port that experiences a receive collision or is the first port to suffer a transmit collision in a packet in Multi-RIC2A applications.<br>1: No interrupts are generated by this condition. |
| D3 | R/W | $\overline{\text{IREC}}$ | **Interrupt on Receive**<br>0: Interrupts will be generated ($\overline{\text{RTI}}$ pin goes active) if this RIC2A contains the receive port for packet or collision activity.<br>1: No interrupts are generated by this condition. |
| D4 | R/W | $\overline{\text{IFF}}$ | **Interrupt on Flag Found**<br>0: Interrupts will be generated ($\overline{\text{ELI}}$ pin goes active) if one or more than one of the flags in the flag array is true.<br>1: No interrupts are generated by this condition. |
| D5 | R/W | $\overline{\text{ILC}}$ | **Interrupt on Low Count**<br>0: Interrupt generated ($\overline{\text{ELI}}$ pin goes active) when one or more of the Event Counters holds a value less than 256 counts.<br>1: no effect |
| D6 | R/W | $\overline{\text{IHC}}$ | **Interrupt on High Count**<br>0: Interrupt generated ($\overline{\text{ELI}}$ pin goes active) when one or more of the Event Counters holds a value in excess of 49152 counts.<br>1: No effect |
| D7 | R/W | $\overline{\text{IFC}}$ | **Interrupt on Full Counter**<br>0: Interrupt generated ($\overline{\text{ELI}}$ pin goes active) when one or more of the Event Counters is full.<br>1: No effect |

### RIC2A Address Register (Page 0H Address 17H)

This register may be used to differentiate between RIC2As in a multi-RIC2A repeater system. The contents of this register form part of the information available through the management bus.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| A5 | A4 | A3 | A2 | A1 | A0 | resv | resv |

## 7.0 RIC2A Registers (Continued)

### Packet Compress Decode Register (Page 0H Address 18H)

This register is used to determine the number of bytes in the data field of a packet which are transferred over the management bus when the packet compress option is employed. The register bits perform the function of a direct binary decode. Thus up to 255 bytes of data may be transferred over the management bus if packet compression is selected.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| PCD7 | PCD6 | PCD5 | PCD4 | PCD3 | PCD2 | PCD1 | PCD0 |

### Global Security Register (GSR) (Page 0H Address 1DH)

This register provides various security configuration options. For example, enable learning mode for all the ports; starting address comparison; use the modified packet status register 5 for the management bus; generate random pattern on source address mismatch; disable port on source address mismatch.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| resv | GLME | resv | DSM | resv | MPS | GRP | SAC |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | SAC | **Start Address Comparison**<br>0: Do not begin comparison<br>1: Begin comparison |
| D1 | R/W | GRP | **Generate Random Pattern:** This bit controls generating the random pattern on a valid source address mismatch. In any event, the Hub Manager will be informed on the SA mismatch.<br>0: Generate the random pattern<br>1: Do not generate the random pattern |
| D2 | R/W | MPS | **Modify Packet Status Register 5:** This bit enables modifying the packet status register 5, PSR5 on the 7 management bytes, over the management bus.<br>0: Do not modify the PSR5<br>1: Modify the PSR5 |
| D3 | R | resv | **Reserved for Future Use**<br>For proper operation, this bit must be 0. |
| D4 | R/W | DSM | **Disable the Port on a Source Address Mismatch**<br>0: Do not disable the port on a valid source address mismatch<br>1: Disable the port on a valid source address mismatch |
| D5 | R | resv | **Reserved for Future Use** |
| D6 | R/W | GLME | **Global Learn Mode Enable**<br>0: Do not enable the learn mode for all ports<br>1: Enable the learn mode for all ports<br><br>Note: The GLME is not a status bit. Reading this bit indicates what value was last written to it. |
| D7 | R | resv | **Reserved for Future Use**<br>reads as a logic 0 |

## 7.0 RIC2A Registers (Continued)

### Inter Frame Gap Threshold Select Register (Page 0H Address 1FH)

This register is used to configure the hub management interface to provide a certain minimum inter frame gap between packets transmitted over the management bus. The value written to this register, plus one, is the magnitude in bit times of the minimum IFG allowed on the management bus.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| IFGT7 | IFGT6 | IFGT5 | IFGT4 | IFGT3 | IFGT2 | IFGT1 | IFGT0 |

### Port Event Record Registers (Page 1H Addresses 11H to 1DH)

These registers hold the recorded events for the specified RIC2A port. The flags are cleared when the register is read.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| RCON | PART | OWC | SE | NSFD | PLER | ELBER | JAB |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R | JAB | **Jabber:** A Jabber Protect event has occurred. |
| D1 | R | ELBER | **Elasticity Buffer Error:** A Elasticity Buffer Error has occurred. |
| D2 | R | PLER | **Phase Lock Error:** A Phase Lock Error event has occurred. |
| D3 | R | NSFD | **Non SFD:** A Non SFD packet event has occurred. |
| D4 | R | SE | **Short Event:** A short event has occurred. |
| D5 | R | OWC | **Out of Window Collision:** An out of window collision event has occurred. |
| D6 | R | PART | **Partition:** A partition event has occurred. |
| D7 | R | BDLNK | **Bad Link:** A link failure event has occurred. |

### Upper Event Information Register (Upper EIR) (Page 1H Address 1EH)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| ER8 | ER7 | ER6 | ER5 | ER4 | ER3 | ER2 | ER1 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R | ER1 | 0: Flag found not generated by event on port 1<br>1: Flag found generated by event on port 1 |
| D1 | R | ER2 | 0: Flag found not generated by event on port 2<br>1: Flag found generated by event on port 2 |
| D2 | R | ER3 | 0: Flag found not generated by event on port 3<br>1: Flag found generated by event on port 3 |
| D3 | R | ER4 | 0: Flag found not generated by event on port 4<br>1: Flag found generated by event on port 4 |
| D4 | R | ER5 | 0: Flag found not generated by event on port 5<br>1: Flag found generated by event on port 5 |
| D5 | R | ER6 | 0: Flag found not generated by event on port 6<br>1: Flag found generated by event on port 6 |
| D6 | R | ER7 | 0: Flag found not generated by event on port 7<br>1: Flag found generated by event on port 7 |
| D7 | R | ER8 | 0: Flag found not generated by event on port 8<br>1: Flag found generated by event on port 8 |

## 7.0 RIC2A Registers (Continued)

### Lower Event Information Register (Lower EIR) (Page 1H Address 1FH)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| DLU | resv | resv | ER13 | ER12 | ER11 | ER10 | ER9 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R | ER9 | 0: Flag found not generated by event on port 9<br>1: Flag found generated by event on port 9 |
| D1 | R | ER10 | 0: Flag found not generated by event on port 10<br>1: Flag found generated by event on port 10 |
| D2 | R | ER11 | 0: Flag found not generated by event on port 11<br>1: Flag found generated by event on port 11 |
| D3 | R | ER12 | 0: Flag found not generated by event on port 12<br>1: Flag found generated by event on port 12 |
| D4 | R | ER13 | 0: Flag found not generated by event on port 13<br>1: Flag found generated by event on port 13 |
| D5 | R | resv | **Reserved for Future Use**<br>reads as a logic 0 |
| D6 | R | resv | **Reserved for Future Use**<br>reads as a logic 0 |
| D7 | R/W | DLU | **Disable the LED Updates:** This bit disables the LED display updates for a faster processor register access<br>0: Re-enable the LED update cycle.(Note)<br>1: Disable the LED update cycles<br>Note: The LED update cycle will be re-enabled only when the network and the RIC2A internal state machines are idle |

### Port Event Count Register (Pages 2H and 3H)

The Event Count Register (ECR) shows the instantaneous value of the specified port's 16 bit counter. The counter increments when an enabled event occurs. The counter may be cleared when it is read, and prevented from rolling over when the maximum count is reached, by setting the appropriate control bits in the Upper Event Count mask register. Since the RIC2A's processor port is octal and the counters are 16 bits long, a temporary holding register is employed for register reads. When one of the counters is read, either high or low byte first, all 16 bits of the counter are transferred to a holding register. Provided the next read cycle to the counter array accesses the same counter's other byte, then the read cycle accesses the holding register. This avoids the problem of events occurring in between the two processor reads and indicating a false count value. In order to enter a new value to the holding register a different counter must be accessed, or the same counter byte must be re-read.

**Lower Byte**

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| EC7 | EC6 | EC5 | EC4 | EC3 | EC2 | EC1 | EC0 |

**Upper Byte**

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| EC15 | EC14 | EC13 | EC12 | EC11 | EC10 | EC9 | EC8 |

## 7.0 RIC2A Registers (Continued)

### Port Event Count Register 2 (PECR-2) (Page 4H Addresses 11H to 1DH)

The Port Event Count Register 2 (PECR-2) shows the instantaneous value of the specified port's 8 - bit counter. The counter increments when an enabled event occurs. The counter may be cleared when it is read, and prevented from rolling over when the maximum count is reached, by setting the appropriate control bits in the ECIMR - 2 register.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| EC7 | EC6 | EC5 | EC4 | EC3 | EC2 | EC1 | EC0 |

### Port Security Configuration Register (PSCR) (Pages 4H, 5H, 6H, 8H, 9H)

This register sets up the various security modes for the RIC2A. It provides port specific information such as enabling/disabling the security mode, passing broadcast packets, etc. In addition, comparison on destination address, source address, or both can be selected. The system can also qualify learning mode on a per port basis.

Note: Bit D0 is **only** for the port CAMs, and **not** for the shared CAMs

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| res | EDA | ESA | SAM | MCE | BCE | SME | LME |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | LME | **Learning Mode**<br>0: Disable Learn Mode for port CAMs.<br>1: Enable Learn Mode for port CAMs. |
| D1 | R/W | SME | **Security Mode**<br>0: Disable Security Mode.<br>1: Enable Security Mode. |
| D2 | R/W | BCE | **Accept Broadcast:** Enables the repeater to pass/repeat a packet with an all 1's destination address.<br>0: Replace the broadcast packets with random packets.<br>1: Pass broadcast packets.<br>Note: SA mismatch is still valid for broadcast packets. |
| D3 | R/W | MCE | **Accept Multicast:** Enables the repeater to pass/repeat a packet with the LSB of '1' in the most significant byte of the destination address.<br>0: Replace the multicast packets with random packets.<br>1: Pass multicast packets.<br>Note: SA mismatch is still valid for multicast packets. |
| D4 | R | SAM | **Source Address Match/Mismatch**<br>0: Source address match occurred for the packet.<br>1: Source address mismatch occurred for the packet. |
| D5 | R/W | ESA | **Source Address Security**<br>0: Do not employ source address to implement security.<br>1: Employ source address to implement security. |
| D6 | R/W | EDA | **Destination Address Security**<br>0: Do not employ destination address to implement security.<br>1: Employ destination address to implement security. |
| D7 | R | resv | **Reserved for Future Use**<br>reads as a logic 0 |

## 7.0 RIC2A Registers (Continued)

### Port CAM Pointer Register (PCPR) (Pages 4H, 5H, 6H, 8H, 9H)

This register indicates which bytes of the six ethernet address bytes has been stored in the CAM locations. When a byte has been loaded into the CAM location, the pointer increments. Thus, this register indicates which byte will get written on the subsequent CAM location access. After the complete address is stored in any of the two CAMs, the user must set the "address valid", ADV, bit so the address is not over-written mistakenly. When in learning mode, this register could be read to see if an address has been learned.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| ADV | PTR2 | PTR1 | PTR0 | ADV | PTR2 | PTR1 | PTR0 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R | PTR0 | D0 of the pointer for the port CAM location 1 |
| D1 | R | PTR1 | D1 of the pointer for the port CAM location 1 |
| D2 | R | PTR2 | D2 of the pointer for the port CAM location 1 |
| D3 | R/W | ADV | **ADdress Valid**<br>0: Address is not valid in port CAM location 1<br>1: Address is valid in port CAM location 1 |
| D4 | R | PTR0 | D0 of the pointer for the port CAM location 2 |
| D5 | R | PTR1 | D1 of the pointer for the port CAM location 2 |
| D6 | R | PTR2 | D2 of the pointer for the port CAM location 2 |
| D7 | R/W | ADV | **ADdress Valid**<br>0: Address is not valid in port CAM location 2<br>1: Address is valid in port CAM location 2 |

### Port CAM Register (Pages 5H, 6H, 8H, 9H)

This register accesses the 48 bits of the port CAM address. Six write/read cycles are required to load/read the entire 48 bit address.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| PCAMx_D7 | PCAMx_D6 | PCAMx_D5 | PCAMx_D4 | PCAMx_D3 | PCAMx_D2 | PCAMx_D1 | PCAMx_D0 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D(7:0) | R/W | PCAMx _D(7:0) | This register accesses the **P**ort **CAM** for the particular port. Note that **x** represents the port number.<br><br>1st access: bits [7: 0] of the address,<br>2nd access: bits [15: 8] of the address,<br>3rd access: bits [23:16]of the address,<br>4th access: bits [31: 24] of the address,<br>5th access: bits [39: 32] of the address,<br>6th access: bits [47: 40] of the address. |

## 7.0 RIC2A Registers (Continued)

### Shared CAM Validation Register 1 (SCVR 1) (Page 9H Address 16H)

This register indicates the validity of an Ethernet address stored in any one of the shared CAMs. When a “1” is written in a specific SCVR, upon starting the network security, the CAM contents will be used for address comparison.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| ADV8 | ADV7 | ADV6 | ADV5 | ADV4 | ADV3 | ADV2 | ADV1 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | ADV1 | **ADdress Valid 1**<br>0: Address is not valid in CAM 1<br>1: Address is valid in CAM 1 |
| D1 | R/W | ADV2 | **ADdress Valid 2**<br>0: Address is not valid in CAM 2<br>1: Address is valid in CAM 2 |
| D2 | R/W | ADV3 | **ADdress Valid 3**<br>0: Address is not valid in CAM 3<br>1: Address is valid in CAM 3 |
| D3 | R/W | ADV4 | **ADdress Valid 4**<br>0: Address is not valid in CAM 4<br>1: Address is valid in CAM 4 |
| D4 | R/W | ADV5 | **ADdress Valid 5**<br>0: Address is not valid in CAM 5<br>1: Address is valid in CAM 5 |
| D5 | R/W | ADV6 | **ADdress Valid 6**<br>0: Address is not valid in CAM 6<br>1: Address is valid in CAM 6 |
| D6 | R/W | ADV7 | **ADdress Valid 7**<br>0: Address is not valid in CAM 7<br>1: Address is valid in CAM 7 |
| D7 | R/W | ADV8 | **ADdress Valid 8**<br>0: Address is not valid in CAM 8<br>1: Address is valid in CAM 8 |

Note: Before writing to and changing any bits in this register, read the register first and then only change the desired bits. By doing this, a previous entry will not be invalidated mistakenly.

## 7.0 RIC2A Registers (Continued)

### Shared CAM Validation Register 2 (SCVR 2) (Page 9H, Address 17H)

This register indicates the validity of an Ethernet address stored in any one of the shared CAMs. When a “1” is written in a specific SCVR, upon starting the network security, the CAM contents will be used for address comparison.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| ADV16 | ADV15 | ADV14 | ADV13 | ADV12 | ADV11 | ADV10 | ADV9 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | ADV9 | **ADdress Valid 9**<br>0: Address is not valid in CAM 9<br>1: Address is valid in CAM 9 |
| D1 | R/W | ADV10 | **ADdress Valid 10**<br>0: Address is not valid in CAM 10<br>1: Address is valid in CAM 10 |
| D2 | R/W | ADV11 | **ADdress Valid 11**<br>0: Address is not valid in CAM 11<br>1: Address is valid in CAM 11 |
| D3 | R/W | ADV12 | **ADdress Valid 12**<br>0: Address is not valid in CAM 12<br>1: Address is valid in CAM 12 |
| D4 | R/W | ADV13 | **ADdress Valid 13**<br>0: Address is not valid in CAM 13<br>1: Address is valid in CAM 13 |
| D5 | R/W | ADV14 | **ADdress Valid 14**<br>0: Address is not valid in CAM 14<br>1: Address is valid in CAM 14 |
| D6 | R/W | ADV15 | **ADdress Valid 15**<br>0: Address is not valid in CAM 15<br>1: Address is valid in CAM 15 |
| D7 | R/W | ADV16 | **ADdress Valid 16**<br>0: Address is not valid in CAM 16<br>1: Address is valid in CAM 16 |

Note: Before writing to and changing any bits in this register, read the register first and then only change the desired bits. By doing this, a previous entry will not be invalidated mistakenly.

### Shared CAM Validation Register 3 (SCVR 3) (Page 9H, Address 18H)

This register indicates the validity of an Ethernet address stored in any one of the shared CAMs. When a “1” is written in a specific SCVR, upon starting the network security, the CAM contents will be used for address comparison.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| ADV24 | ADV23 | ADV22 | ADV21 | ADV20 | ADV19 | ADV18 | ADV17 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | ADV17 | **ADdress Valid 17**<br>0: Address is not valid in CAM 17<br>1: Address is valid in CAM 17 |
| D1 | R/W | ADV18 | **ADdress Valid 18**<br>0: Address is not valid in CAM 18<br>1: Address is valid in CAM 18 |
| D2 | R/W | ADV19 | **ADdress Valid 19**<br>0: Address is not valid in CAM 19<br>1: Address is valid in CAM 19 |
| D3 | R/W | ADV20 | **ADdress Valid 20**<br>0: Address is not valid in CAM 20<br>1: Address is valid in CAM 20 |
| D4 | R/W | ADV21 | **ADdress Valid 21**<br>0: Address is not valid in CAM 21<br>1: Address is valid in CAM 21 |
| D5 | R/W | ADV22 | **ADdress Valid 22**<br>0: Address is not valid in CAM 22<br>1: Address is valid in CAM 22 |
| D6 | R/W | ADV23 | **ADdress Valid 23**<br>0: Address is not valid in CAM 23<br>1: Address is valid in CAM 23 |
| D7 | R/W | ADV24 | **ADdress Valid 24**<br>0: Address is not valid in CAM 24<br>1: Address is valid in CAM 24 |

Note: Before writing to and changing any bits in this register, read the register first and then only change the desired bits. By doing this, a previous entry will not be invalidated mistakenly.

## 7.0 RIC2A Registers (Continued)

### Shared CAM Validation Register 4 (SCVR 4) (Page 9H Address 19H)

This register indicates the validity of an Ethernet address stored in any one of the shared CAMs. When a “1” is written in a specific SCVR, upon starting the network security, the CAM contents will be used for address comparison.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| ADV32 | ADV31 | ADV30 | ADV29 | ADV28 | ADV27 | ADV26 | ADV25 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | ADV25 | **ADdress Valid 25**<br>0: Address is not valid in CAM 25<br>1: Address is valid in CAM 25 |
| D1 | R/W | ADV26 | **ADdress Valid 26**<br>0: Address is not valid in CAM 26<br>1: Address is valid in CAM 26 |
| D2 | R/W | ADV27 | **ADdress Valid 27**<br>0: Address is not valid in CAM 27<br>1: Address is valid in CAM 27 |
| D3 | R/W | ADV28 | **ADdress Valid 28**<br>0: Address is not valid in CAM 28<br>1: Address is valid in CAM 28 |
| D4 | R/W | ADV29 | **ADdress Valid 29**<br>0: Address is not valid in CAM 29<br>1: Address is valid in CAM 29 |
| D5 | R/W | ADV30 | **ADdress Valid 30**<br>0: Address is not valid in CAM 30<br>1: Address is valid in CAM 30 |
| D6 | R/W | ADV31 | **ADdress Valid 31**<br>0: Address is not valid in CAM 31<br>1: Address is valid in CAM 31 |
| D7 | R/W | ADV32 | **ADdress Valid 32**<br>0: Address is not valid in CAM 32<br>1: Address is valid in CAM 32 |

Note: Before writing to and changing any bits in this register, read the register first and then only change the desired bits. By doing this, a previous entry will not be invalidated mistakenly.

## 7.0 RIC2A Registers (Continued)

### Shared CAM Register (Pages 9H, AH, BH, CH, DH, EH, FH)

This register accesses the 48 bits of the shared CAM address. Six write/read cycles are required to load/read the entire 48 bit address.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| SCAMx_D7 | SCAMx_D6 | SCAMx_D5 | SCAMx_D4 | SCAMx_D3 | SCAMx_D2 | SCAMx_D1 | SCAMx_D0 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D(7:0) | R/W | SCAMx_D(7:0) | This register accesses the **S**hared **CAM** location for the particular port. Note that **x** represents the port number.<br><br>1st access: bits [7: 0] of the address,<br>2nd access: bits [15: 8] of the address,<br>3rd access: bits [23:16]of the address,<br>4th access: bits [31: 24] of the address,<br>5th access: bits [39: 32] of the address,<br>6th access: bits [47: 40] of the address. |

## 7.0 RIC2A Registers (Continued)

### CAM Location Mask Register (CLMR) (Pages 9H, AH, BH, CH, DH, EH, FH)

Each shared CAM has a CLMR, therefore there are 32 CLMRs. Any of the 32 CAMs can be shared among the ports. For example, multiple ports can share a single ethernet address, or multiple addresses can be associated with a single port. Assigning CAMs to ports, or vice-versa, is done through these registers.

**CLMR Lo Byte Location**

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | P1 | 0: CAM entry does not belong to port 1<br>1: CAM entry belongs to port 1 |
| D1 | R/W | P2 | 0: CAM entry does not belong to port 2<br>1: CAM entry belongs to port 2 |
| D2 | R/W | P3 | 0: CAM entry does not belong to port 3<br>1: CAM entry belongs to port 3 |
| D3 | R/W | P4 | 0: CAM entry does not belong to port 4<br>1: CAM entry belongs to port 4 |
| D4 | R/W | P5 | 0: CAM entry does not belong to port 5<br>1: CAM entry belongs to port 5 |
| D5 | R/W | P6 | 0: CAM entry does not belong to port 6<br>1: CAM entry belongs to port 6 |
| D6 | R/W | P7 | 0: CAM entry does not belong to port 7<br>1: CAM entry belongs to port 7 |
| D7 | R/W | P8 | 0: CAM entry does not belong to port 8<br>1: CAM entry belongs to port 8 |

**CLMR Hi Byte Location**

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| PTR2 | PTR1 | PTR0 | P13 | P12 | P11 | P10 | P9 |

| Bit | R/W | Symbol | Description |
|---|---|---|---|
| D0 | R/W | P9 | 0: CAM entry does not belong to port 9<br>1: CAM entry belongs to port 9 |
| D1 | R/W | P10 | 0: CAM entry does not belong to port 10<br>1: CAM entry belongs to port 10 |
| D2 | R/W | P11 | 0: CAM entry does not belong to port 11<br>1: CAM entry belongs to port 11 |
| D3 | R/W | P12 | 0: CAM entry does not belong to port 12<br>1: CAM entry belongs to port 12 |
| D4 | R/W | P13 | 0: CAM entry does not belong to port 13<br>1: CAM entry belongs to port 13 |
| D5 | R | PTR0 | D0 of the pointer into the CAM location |
| D6 | R | PTR1 | D1 of the pointer into the CAM location |
| D7 | R | PTR2 | D2 of the pointer into the CAM location |

## 8.0 Board Layout Recommendations

There are numerous methods to layout PCB boards to achieve successful proper operation. Two options for the RIC2A layout are presented here. These NSC recommendations have not been empirically proven in the laboratory.

### Power and Ground Planes

Standard analog design techniques should be utilized when laying out the power supply traces on the board. If a digital power supply is used, NSC recommends a one pole RC filter designed with a cut-off frequency of 1 kHz to improve the signal jitter performance. See the figure below

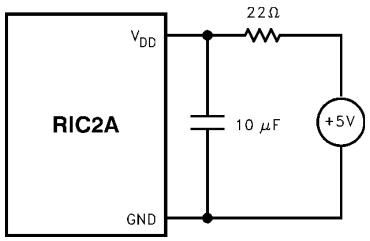



**Figure 22. Recommended RC Filter**

This methodology facilitates locking of the PLL when capturing the incoming signal. This filtering scheme should be implemented on each of the analog power and ground pins ($V_{DD}$A, GNDA, $V_{DD}$PLL, GNDPLL, $V_{DD}$WS, GNDWS). Additionally, NSC recommends ferrite chokes to isolate the various ground signals.

### Power Plane

The power plane should be divided into distinct islands to reduce and isolate noise effects. The signal traces which traverse across multiple islands should be minimized and impedance matched to reduce standing wave reflections.

The power plane for the device may be divided into three regions as shown in the figure: Digital $V_{DD}$, Analog $V_{DD}$ and PLL $V_{DD}$. Or, it may be divided in only two regions: by combining the Analog and PLL $V_{DD}$ regions into one, and leaving the Digital $V_{DD}$ as a separate region. NSC recommends ferrite beads to isolate the Digital $V_{DD}$ and Analog $V_{DD}$ regions

### Ground Plane

Option 1: The ground plane is one single uniform plane.

Option 2: The ground plane for the RIC2A is divided into islands to minimize the effects of noise. The signal traces which traverse across multiple islands should be minimized and impedance matched to reduce standing wave reflections.

The ground plane for the device may be divided into three regions as shown in the figure: Digital GND, Analog GND and PLL GND. Or, it can be divided in only two regions: by combining the Analog and PLL GND regions into one, and leave the Digital GND as a separate region. NSC recommends a ferrite bead between the Digital GND (Board GND) and Analog GND regions for isolation.

The ground pin on the external 40 MHz oscillator should be connected to the RIC2A's digital ground region. (The output of the oscillator will be connected to RIC2A's CLKIN signal, pin 100.)

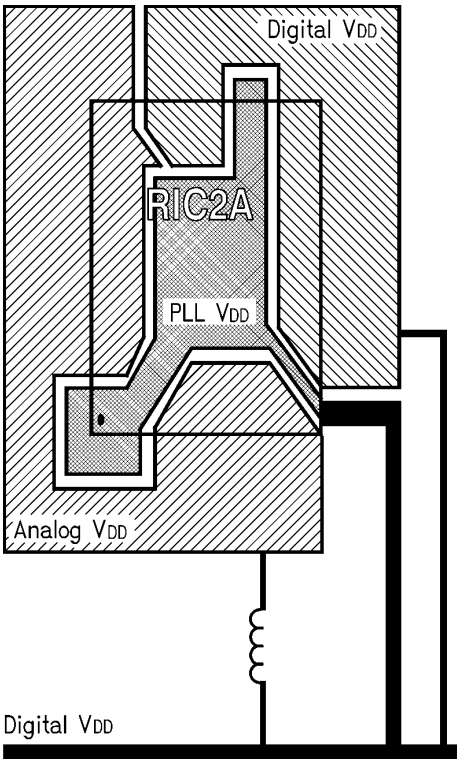



**Figure 23. The RIC2A Power Plane is Divided into Islands to Reduce and Isolate Noise Effects**

All the port ground pins on the RIC2A should be connected to the digital ground region. If only two regions are created, the GNDPLL, GNDWS, and GNDA pins on the RIC2A should be connected to the analog ground region.

### Device Layout

Due to high device power dissipation, additional layout considerations should be applied to ease that process. Placing an additional metal layer right below the device placement (on the component layer) will sink additional current into the ground plane and will aid in cooling the device. See Figure 25. The metal traces should be placed between the last corner pins on both sides. Make the traces as thick as possible. Multiple vias to ground should be placed on these metal traces (and as many as layout will allow).

### The RIC2A Airflow Fan

For a RIC2A design, a fan is recommended to increase airflow and keep junction temperature down.

#### Decoupling Capacitors

National strongly recommends decoupling capacitors between the power and ground pins. See Figure 26 below for specific placement and value.

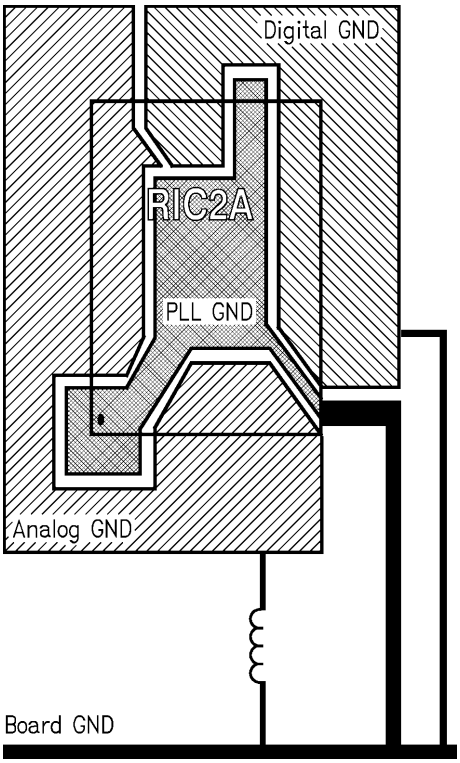



**Figure 24. The RIC2A Ground Plane is Divided into 3 Regions to Minimize Noise Effects**

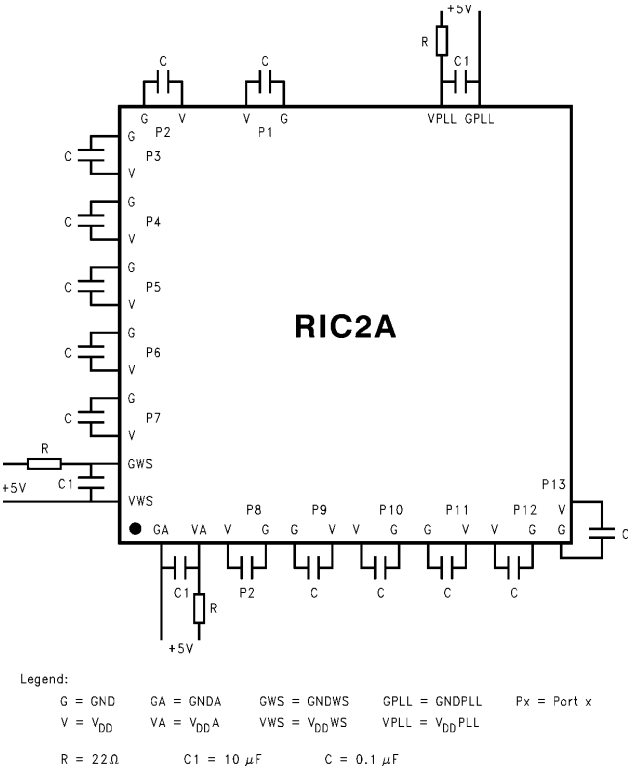



**Figure 26. Configuration for Decoupling Capacitors across Power & Ground Pins**

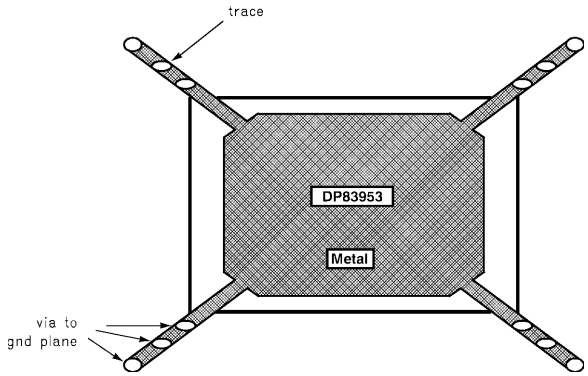



**Figure 25. The RIC2A Metal Layer Configuration used to Sink Additional Current**

## 9.0 DC and AC Specification

### Absolute Maximum Ratings

| | |
|---|---|
| Supply Voltage ($V_{DD}$) | 0.5V to 7.0V |
| DC Input Voltage ($V_{in}$) | -0.5V to $V_{DD}$ +0.5V |
| DC Output Voltage ($V_{out}$) | -0.5V to $V_{DD}$ +0.5V |
| Storage Temperature Range ($T_{STG}$) | -65 °C to 150 °C |
| Power Dissipation for chip ($P_D$) | 3.15 Ω |
| Lead Temp. ($T_L$) (Soldering, 10 sec) | 260 °C |
| ESD Rating ($R_{zap}$ = 1.5K, $C_{zap}$ = 120pF) | 2000V |

### Recommended Operating Conditions

| | |
|---|---|
| Supply Voltage ($V_{DD}$) | 5V ± 5% |
| Ambient Temperature | 0 to 70°C |

*Note: Absolute maximum ratings are those values beyond which the safety of the device can be guaranteed. These ratings are not meant to imply that the device should be operated at these limits.*

### DC Specifications Measured at $T_A$ = 0 °C to 70 °C, $V_{DD}$ = 5V ±5%, unless otherwise specified

**PROCESSOR, LED, TWISTED PAIR PORTS, INTER-RIC and MANAGEMENT INTERFACES**

| Symbol | Description | Conditions | Min | Max | Units |
|---|---|---|---|---|---|
| $V_{OH}$ | Minimum high level output voltage | $I_{OH}$ = -8 mA | 3.5 | | V |
| $V_{OL}$ | Minimum low level output voltage | $I_{OL}$ = 8 mA | | 0.4 | V |
| $V_{IH}$ | Minimum high level input voltage | | 2.0 | | V |
| $V_{IL}$ | Maximum low level input voltage | | | 0.8 | V |
| $I_{IN}$ | Input current | $V_{IN}$ = $V_{DD}$ or GND | -1.0 | 1.0 | µA |
| $I_{OZ}$ | Maximum TRI-STATE output leakage current | $V_{OUT}$ = $V_{DD}$ or GND | -10 | 10 | µA |
| $I_{CC}$ | Average supply current | $V_{IN}$ = $V_{DD}$ or GND<br>$V_{DD}$ = 5.25 | | 870 | mA |
| **AUI (PORT 1)** | | | | | |
| $V_{OD}$ | Differential output voltage (TX±) | 78Ω termination & 270Ω pulldowns | ±550 | ±1200 | mV |
| $V_{OB}$ | Differential output voltage imbalance (TX) | 78Ω termination & 270Ω pulldowns | Typical: 40 mV | | |
| $V_U$ | Undershoot voltage (TX±) | 78Ω termination & 270Ω pulldowns | Typical: 80 mV | | |
| $V_{DS}$ | Differential squelch threshold (RX±, CD±) | | -175 | -300 | mV |
| $V_{CM}$ | Differential input common mode voltage (RX±, CD±) (Note 1) | | 0 | 5.5 | V |
| **TWISTED PAIR (PORTS 2-13)** | | | | | |
| $V_{RON}$ | Minimum receive squelch threshold | Normal Mode<br>Reduced Mode | ±300 (Note 2) | ±585<br>±340 | mV<br>mV |

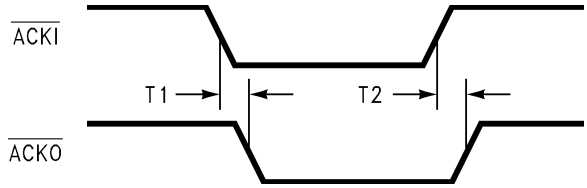
Note 1: This parameter is guaranteed by design and is not tested.

Note 2: The operation in reduced mode is not guaranteed below 300 mV.

## 9.0 DC and AC Specification (Continued)

## AC Specifications

### Port Arbitration Timing

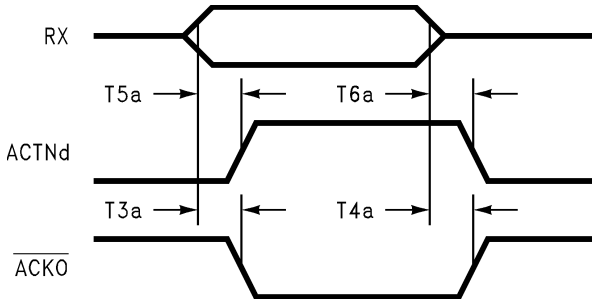



| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T1 | ackilackol | $\overline{ACKI}$ low to $\overline{ACKO}$ low | | 24 | ns |
| T2 | ackihackoh | $\overline{ACKI}$ high to $\overline{ACKO}$ high | | 21 | ns |

Note 1: Timing valid with no receive or collision activities.
Note 2: In these diagrams the Inter-RIC and Management Busses are shown using active high signals. Active low signals may also be used. See section 5.5 Mode Load Operation?

### Receive Timing AUI Port

Receive activity propagation start up and end delays for AUI port



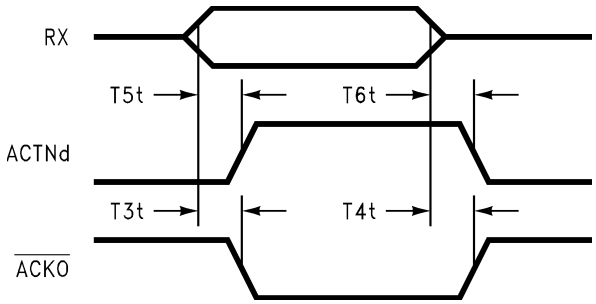

| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T3a | rxaackol | RX active to $\overline{ACKO}$ low | | 66 | ns |
| T4a | rxiackoh | RX inactive to $\overline{ACKO}$ high | | 325 | ns |
| T5a | rxaactna | RX active to ACTNd active | | 105 | ns |
| T6a | rxiactni | RX inactive to ACTNd inactive | | 325 | ns |

Note: ACKI assumed high

## 9.0 DC and AC Specification (Continued)

### Receive Timing-10Base-T Ports

Receive activity propagation start up and end delays for 10BASE-T ports

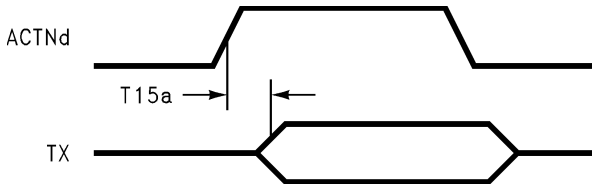



| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T3t | rxaackol | RX active to $\overline{\text{ACKO}}$ low | | 240 | ns |
| T4t | rxiackoh | RX inactive to $\overline{\text{ACKO}}$ high | | 255 | ns |
| T5t | rxaactna | RX active to ACTNd active | | 270 | ns |
| T6t | rxiactni | RX inactive to ACTNd inactive | | 265 | ns |

Note: $\overline{\text{ACKI}}$ assumed high

### Transmit Timing-AUI Ports

Transmit activity propagation start up and end delays for AUI port



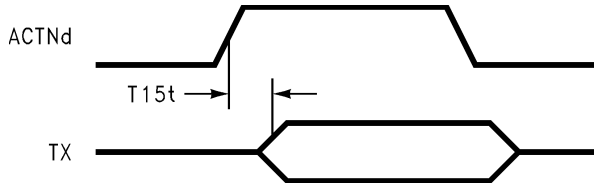

| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T15a | actnatxa | ACTNd active to TX active | | 585 | ns |

Note: $\overline{\text{ACKI}}$ assumed high

## 9.0 DC and AC Specification (Continued)

### Transmit Timing-10Base-T Ports

Receive activity propagation start up and end delays for 10BASE-T ports



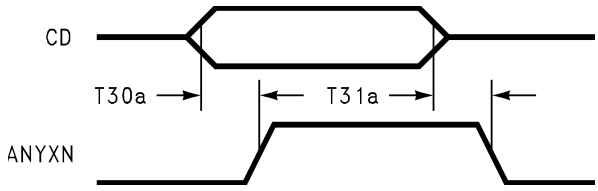

| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T15t | actnatxa | ACTNd active to TX active | | 790 | ns |

Note: $\overline{\text{ACKI}}$ assumed high

### COLLISION TIMING - AUI PORT

Collision activity propagation start up and end delays for AUI port

### Transmit Collision Timing

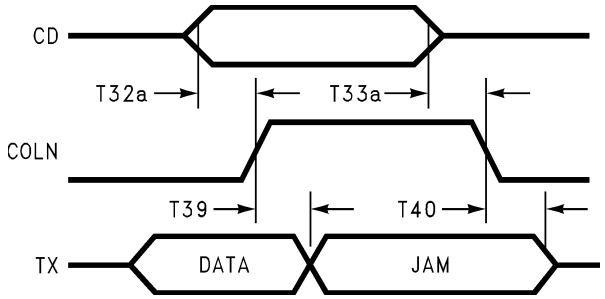



| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T30a | cdaanyxna | CD active to ANYXN active | | 65 | ns |
| T31a | cdianyxni | CD inactive to ANYXN inactive (Note 1, 2) | | 400 | ns |

Note 1: TX collision extension has already been performed and no other port is driving ANYXN
Note 2: Includes TW2

## 9.0 DC and AC Specification (Continued)

### Receive Collision Timing



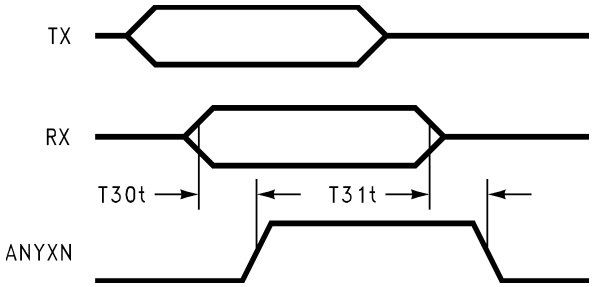

| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T32a | cdacolna | CD active to COLN active (Note 1) | | 55 | ns |
| T33a | cdicolni | CD inactive to COLN inactive | | 215 | ns |
| T39 | colnajs | COLN active to start of jam | | 400 | ns |
| T40 | colnije | COLN inactive to end of jam (Note 2) | | 800 | ns |

Note 1: PKEN assumed high

Note 2: Assuming reception ended before COLN goes inactive. TW2 is included in this parameter. Assuming ACTNd to ACTNs delay is 0.

### Collision Timing-10BASE-T Ports

Collision activity propagation start up and end delays for 10BASE-T ports

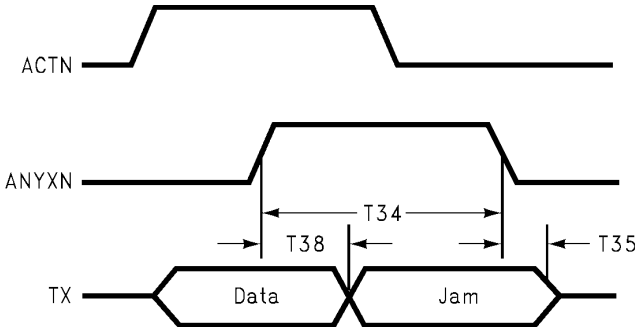



| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T30t | colaanya | Collision active to ANYXN active | | 800 | ns |
| T31t | colianyi | Collision inactive to ANYXN inactive (Note 1) | | 400 | ns |

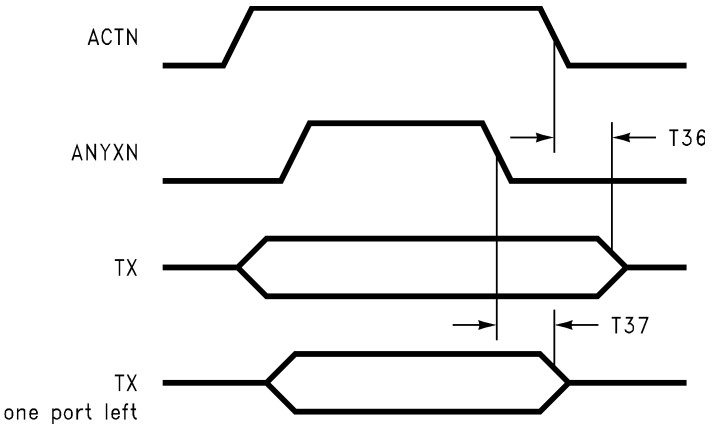
Note: TX collision extension has already been performed and no other port is asserting ANYXN.

## 9.0 DC and AC Specification (Continued)

**Collision Timing-AUI Port**



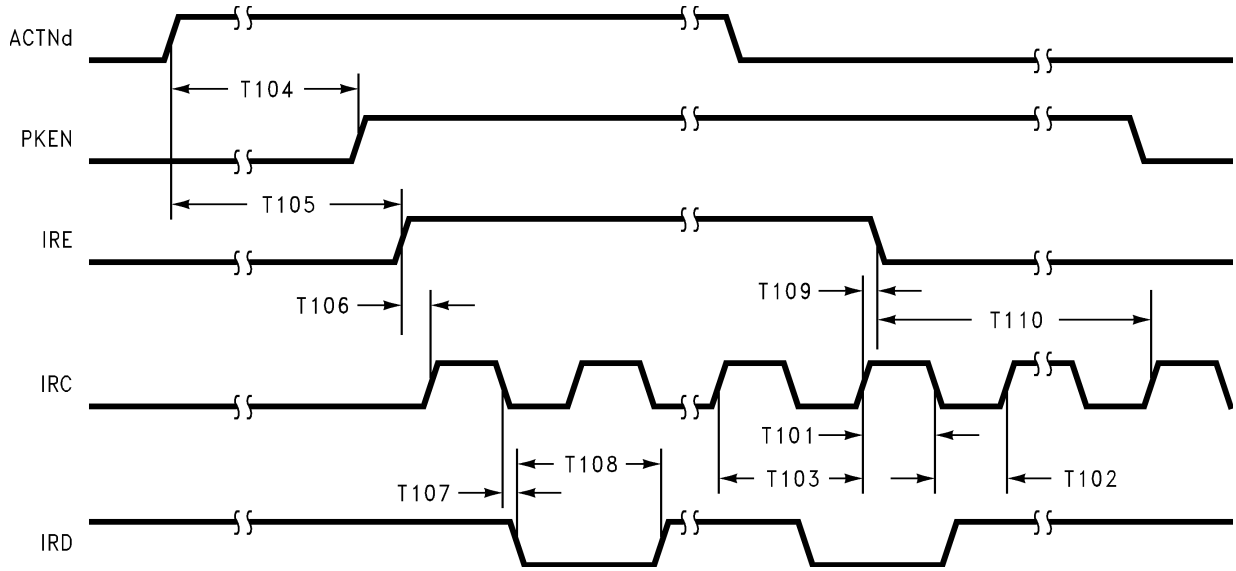

| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T34 | anyamin | ANYXN active time | 96 | | bits |
| T35 | anyitxai | ANYXN inactive to TX to all inactive | 120 | 170 | ns |
| T38 | anyasj | ANYXN active to start of jam | | 400 | ns |




| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T36 | actnitxi | ACTN inactive to TX inactive | | 405 | bits |
| T37 | anyitxoi | ANYXN inactive to TX "one port left" inactive | 120 | 170 | ns |

## 9.0 DC and AC Specification (Continued)

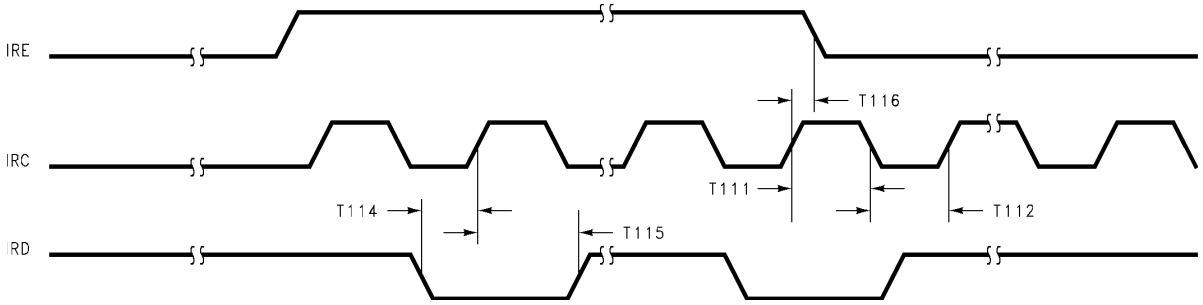
### Inter RIC Bus Output Timing




| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T101 | ircoh | IRC output high time | 45 | 55 | ns |
| T102 | ircol | IRC output low time | 45 | 55 | ns |
| T103 | ircoc | IRC output cycle time | 90 | 110 | ns |
| T104 | actndapkena | ACTNd active to PKEN active | 555 | | ns |
| T105 | actndairea | ACTNd active to IRE active | 560 | | ns |
| T106 | ireairca | IRE output active to IRC active | | 1.8 | µs |
| T107 | irdov | IRD output valid from IRC | | 10 | ns |
| T108 | irdos | IRD output stable valid time | 90 | | ns |
| T109 | ircohirei | IRC output high to IRE inactive | 30 | 70 | ns |
| T110 | ircclks | number of IRCs after IRE inactive | 5 | | clks |

## 9.0 DC and AC Specification (Continued)

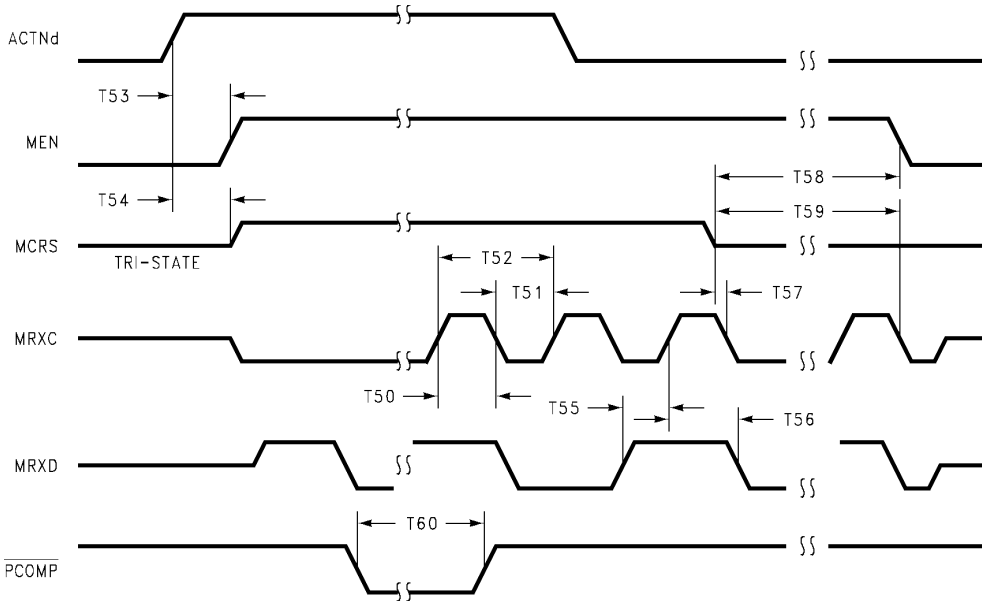
**Inter RIC Bus Input Timing**




| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T111 | ircih | IRC input high time | 20 | | ns |
| T122 | ircil | IRC input low time | 20 | | ns |
| T114 | irdisirc | IRD input setup to IRC | 5 | | ns |
| T115 | irdihirc | IRD input hold from IRC | 10 | | ns |
| T116 | ircihirei | IRC input high to IRE inactive | 10 | 90 | ns |

## 9.0 DC and AC Specification (Continued)

### Management Bus Timing

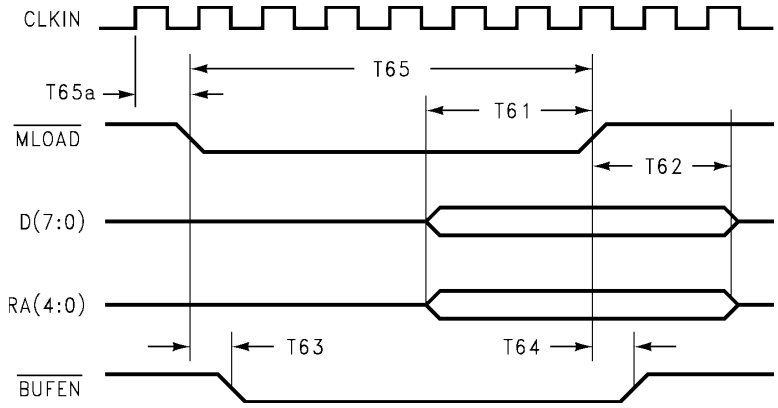



| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T50 | mrxch | MRXC high time | 45 | 55 | ns |
| T51 | mrxcl | MRXC low time | 45 | 55 | ns |
| T52 | mrxcd | MRXC cycle time | 90 | 110 | ns |
| T53 | actndamena | ACTNd active to MEN active | | 715 | ns |
| T54 | actndamcrsa | ACTNd active to MCRS active | | 720 | ns |
| T55 | mrxds | MRXD setup | 40 | | ns |
| T56 | mrxdh | MRXD hold | 45 | | ns |
| T57 | mrxclmcrsi | MRXC low to MCRS inactive | -5 | 6 | ns |
| T58 | mcrsimenl | MCRS inactive to MEN low | | 510 | ns |
| T59 | mrxcclks | min. number of MRXCs after MCRS inactive | 5 | 5 | Clks |
| T60 | pcompw | PCOMP pulse width | 20 | | ns |

Note: The preamble on this bus consists of the following string; 01011

## 9.0 DC and AC Specification (Continued)

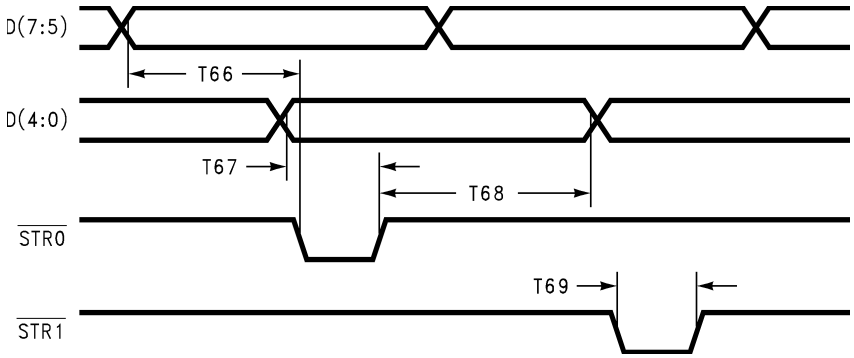
### MLOAD TIMING




| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T61 | mldats | data setup | 10 | | ns |
| T62 | mldath | data hold | 10 | | ns |
| T63 | mlabufa | MLOAD active to BUFEN active | | 35 | ns |
| T64 | mlibufi | MLOAD inactive to BUFEN inactive | | 35 | ns |
| T65 | mlw | MLOAD width | 800 | | ns |
| T65a | clkinm | CLKIN setup to MLOAD | 10 | | ns |

Note: Both edges of MLOAD have to be valid for proper setup timing

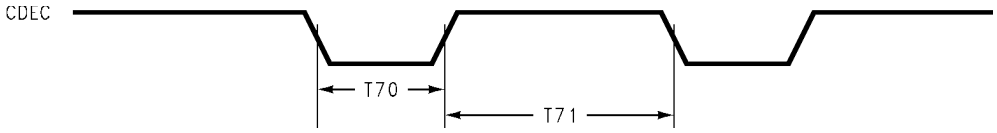
### STROBE TIMING




| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T66 | stradrs | Strobe address setup | 80 | 115 | ns |
| T67 | strdats | Strobe data setup | 22 | 28 | ns |
| T68 | strdath | Strobe data hold | 172 | 178 | ns |
| T69 | strw | Strobe width | 30 | 65 | ns |

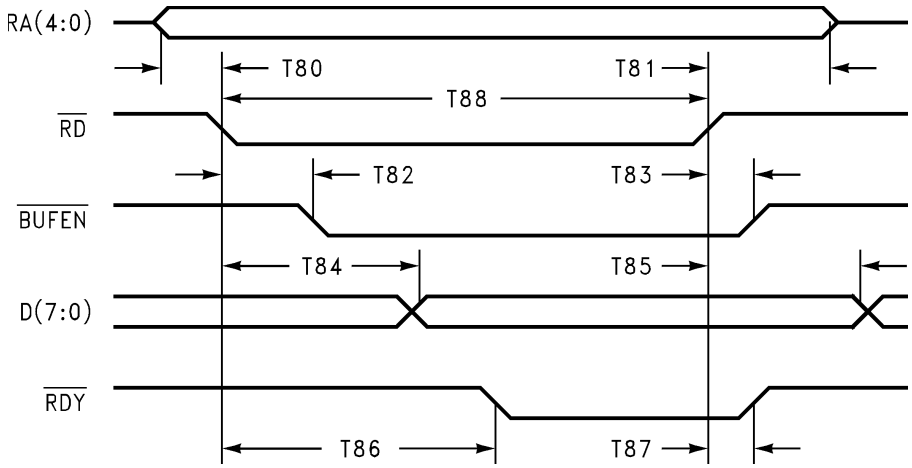
## 9.0 DC and AC Specification (Continued)

### CDEC TIMING




| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T70 | cdecpw | CDEC pulse width | 20 | 100 | ns |
| T71 | cdeccdec | CDEC to CDEC width | 200 | | ns |

### REGISTER READ TIMING

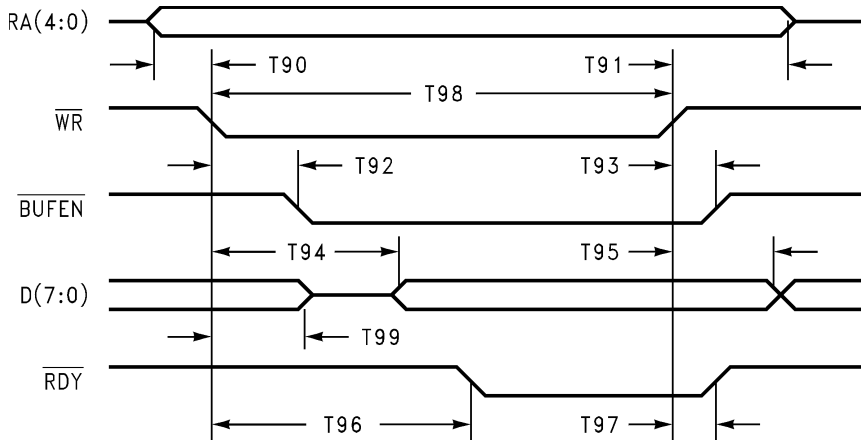



| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T80 | rdadrs | Read address setup | 0 | | ns |
| T81 | rdadrh | Read address hold | 0 | | ns |
| T82 | rdabufa | Read active to BUFEN active | 95 | 345 | ns |
| T83 | rdibufi | Read inactive to BUFEN inactive | | 35 | ns |
| T84 | rdadatv | Read active to Data valid | 245 | | ns |
| T85 | rddath | Read Data hold | 75 | | ns |
| T86 | rdardya | Read active to RDY active | 340 | 585 | ns |
| T87 | rdirdyi | Read inactive to RDY inactive | | 30 | ns |
| T88 | rdw | Read width | 600 | | ns |

Note: Minimum high time between read/write cycle is 100 ns.

## 9.0 DC and AC Specification (Continued)

### REGISTER WRITE TIMING




| Number | Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|---|
| T90 | | Write address setup | 0 | | ns |
| T91 | wradrh | Write address hold | 0 | | ns |
| T92 | wrabufa | Write active to BUFEN active | 95 | 355 | ns |
| T93 | wribufi | Write inactive to BUFEN inactive | | 35 | ns |
| T94 | wradatv | Write active to Data valid | | 275 | ns |
| T95 | wrdath | Write Data hold | 0 | | ns |
| T96 | wrardya | Write active to RDY active | 340 | 585 | ns |
| T97 | wrirdyi | Write inactive to RDY inactive | | 30 | ns |
| T98 | wrw | Write width | 600 | | ns |
| T99 | wradt | Write active to data TRI-STATE | | 350 | ns |

Note 1: Assuming zero propagation delay on external buffer.
Note 2: Minimum high time between read/write cycle is 100 ns.

## 9.0 DC and AC Specification (Continued)

### AC Timing Test Conditions

All specifications are valid only if the mandatory isolation is employed and all differential signals are taken to be at AUI side of the transformer.

| | |
|---|---|
| Input Pulse Levels (TTL/CMOS) | GND to 3.0V |
| Input Rise and Fall Times (TTL/CMOS) | 5ns |
| Input and Output Reference Levels (TTL/CMOS) | 1.5V |
| Input Pulse Levels (Diff.) | $2V_{p-p}$ |
| Input and Output Reference Levels (Diff.) | 50% Point of the Differential |
| TRI-STATE Reference Levels | Float (ΔV)±0.5V |

Output Load (See Figure Below)

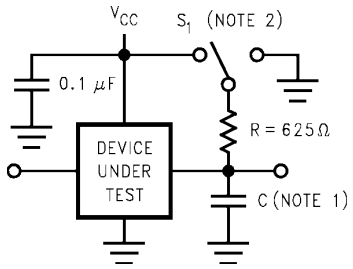



**Figure 27. Output Loading for the Device Under Test**

Note 1: 100 pF, includes scope and jig capacitance.
Note 2: S1 = Open for timing tests for push pull outputs.

S1 = $V_{DD}$ for $V_{OL}$ test.

S1 = GND for $V_{OH}$ test.

S1 = $V_{DD}$ for High Impedance to active low and active low to High Impedance measurements.

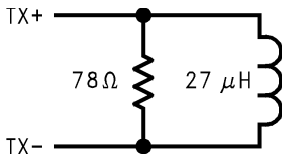
S1 = GND for High Impedance to active high and active high to High Impedance measurements.

### Capacitance $T_A$ = 25°C, f = 1 MHz

| Symbol | Parameter | Typ | Units |
|---|---|---|---|
| $C_{IN}$ | Input Capacitance | 7 | pf |
| $C_{OUT}$ | Output Capacitance | 7 | pf |

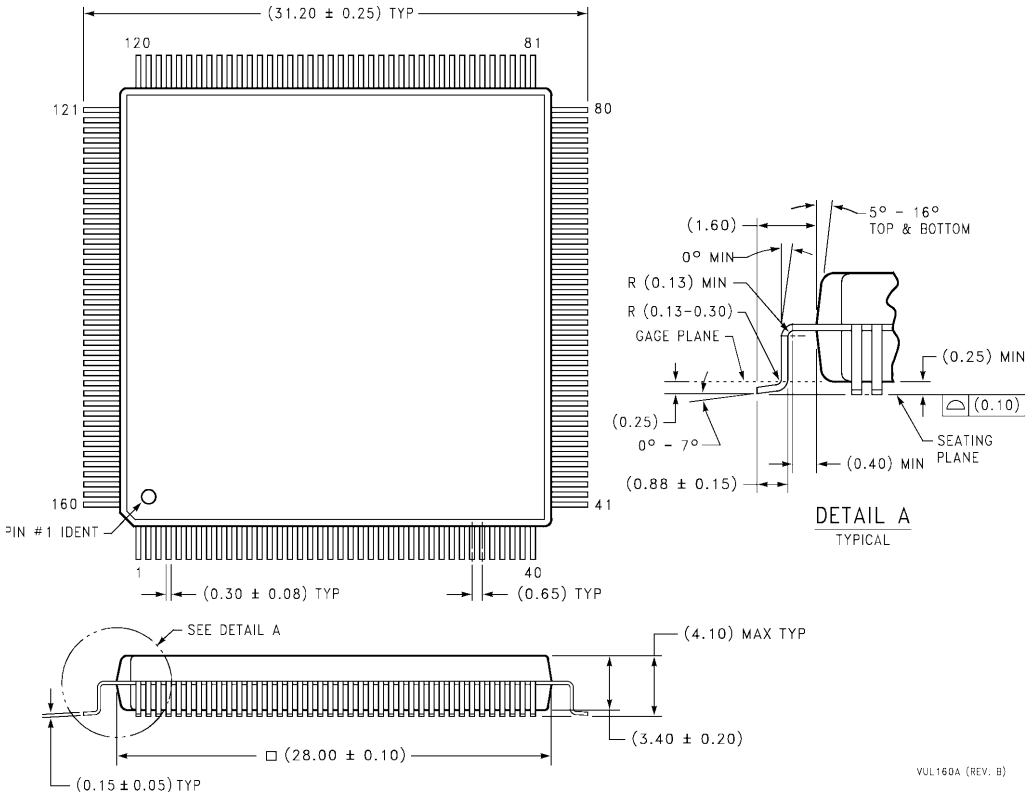
Derating Factor

Output timing are measured with a purely capacitive load for 50pF. The following correction factor can be used for other loads: $C_L \geq$50pF+0.3ns/pF




Note: In the above diagram, the TX+ and TX- signals are taken from the AUI side of the isolation (transformer).

## 10.0 Physical Dimensions inches (millimeters) unless otherwise noted




**Molded Plastic Quad Package, JEDEC**
**Order Number DP83953VUL**
**NS Package Number VUL160A**

### LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

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

**National Semiconductor Corporation**
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com

**National Semiconductor Europe**
Fax: (+49) 0-180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32

**National Semiconductor Asia Pacific Customer Response Group**
Tel: 65-254-4466
Fax: 65-250-4466
Email: sea.support@nsc.com

**National Semiconductor Japan Ltd.**
Tel: 81-3-5620-6175
Fax: 81-3-5620-6179

www.national.com