National Semiconductor

ADVANCE INFORMATION
May 1996

QR1001 QuickRing Enhanced Data Stream Controller

QR1001 QuickRing™ Enhanced Data Stream Controller

General Description

QuickRing technology is a point-to-point data transfer architecture designed to facilitate high speed data streams between devices, boards and systems. The QuickRing architecture can be applied both inside the chassis as well as outside the chassis to increase data throughput. Each QR1001 QuickRing Controller in a ring is capable of streaming up to 280 MSamples per second on 6-bit links. This device is intended for use in applications that handle high-bandwidth data streams associated with graphics, compressed and uncompressed video, disk arrays, high-speed localized networks, multiprocessor systems, and peripherals over cable. The QR1001 QuickRing Controller can be used to augment the performance of traditional backplane buses in personal computers, workstations, and high-end systems. The QR1001 is useful for routing high-bandwidth streams in systems that are larger or topologically more complex than bus-based systems.

The QR1001 Enhanced QuickRing Controller is an upgrade of the QR0001 features and performance. The devices are pin compatible and the QR1001 can be configured to operate in QR0001 mode. The QR1001 and QR0001 can interoperate in the same ring.

Features

- Low latency multicast and broadcast (non-reservation)
- User-controllable packet size
- 160-pin PQFP package
- 16 node single ring capability
- Support for multi-ring topologies
- Supports separate ring and client clock rates

Ring Interface

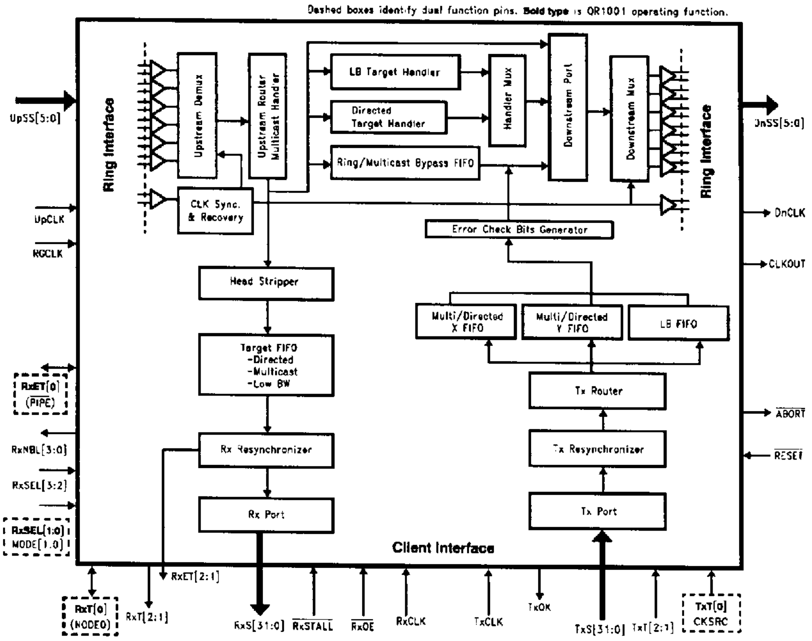
- Data capture up to 280 MSamples/s on 6-bit wide link
- 40 MHz maximum ring clock frequency
- Low Voltage Differential Signals (LVDS) used on ring
- Reservation-based protocol for directed transfers (single target) to eliminate FIFO overflow
- Reservation protocol turned off during multicast
- Error detection detects 1- and 2-bit errors

Client Interface

- Type field increased to 3 bits for multicast and packet-size control
- Up to 160 MBytes/s data rate at both Tx and Rx ports
- 32-bit transmit and receive data ports
- Readable internal diagnostic register

Block Diagram QR1001

Dashed boxes identify dual function pins. **Bold type** is QR1001 operating function.



TL/F/12048-1

TRI-STATE® is a registered trademark of National Semiconductor Corporation.
QuickRing™ is a trademark of Apple Computer, Incorporated.

Table of Contents

1.0 Signal Description

Pin Name	I/O	No.	Description
$\overline{\text{RESET}}$	I	1	**RESET:** Asserting this signal clears all registers and FIFOs. When this input is released, the operation mode is determined and the initialization sequence begins.
$\overline{\text{ABORT}}$	O	1	**ABORT:** When asserted, it indicates that a failure was detected on the ring. $\overline{\text{ABORT}}$ is negated by asserting Reset or reading the diagnostic register, nibble Ah.
$\overline{\text{PIPE}}$	I	1	**PIPE:** When $\overline{\text{PIPE}}$ is negated (non-pipelined timing), at the Client ports, both the symbol and type fields correspond to each other during the same clock cycle. When $\overline{\text{PIPE}}$ is asserted (pipelined timing), the timing of the type field leads by one clock at the receive port and trails by one clock at the transmit port. (The type and symbol fields are pipelined.)
QR1001 **RxET[0]**	**O**		**Extended Receive Early Type:** After release of reset, this pin becomes the third bit in the early type field.
NODE0	I	1	**Node0:** When asserted, the controller is configured as having Node ID 0. Node 0 is responsible for governing the initialization process in the ring.
QR1001 **RxT[0]**	**O**		**Extended Receive Type Field:** In QR1001 operation mode, the function of this pin is the same as NODE0 until the release of reset. This pin becomes the third receive port type bit. This gives 8 symbol functions instead of 4. They are used to identify the multicast (broadcast) heads and explicit tail symbols.
RGCLK	I	1	**Ring Clock:** This clock input is the time-base for the ring interface and chip core. When the CKSRC pin is asserted, a ring clock input should be present. When CKSRC is negated, RGCLK should be tied to ground.
CKSRC	I	1	**Clock Source:** Designates the source of the ring clock. When asserted, RCGLK is the clock source and 4 to 5 clock delays are added to every symbol passing through the Up Port (an elasticity buffer is activated). Multiple nodes can have CKSRC asserted and be ring clock sources. (In 5 node rings and larger, there should be a clock source at every node.) When this pin is negated, both ring and core clocks are derived from the differential UpCLK.
QR1001 **TxT[0]**	**I**		**Extended Transmit Type Field:** In QR1001 operation mode, the function of this pin is the same as CKSRC until the release of reset. After release of reset, this pin becomes the third transmit port type bit. This identifies the multicast (broadcast) head and the explicit tail symbol used to fix stream and packet length on the client and ring ports.
CLKOUT	O	1	**Clock Out:** If CKSRC is asserted, then CLKOUT is derived from RGCLK. If CKSRC is negated, then CLKOUT is derived from UpCLK.
UpCLK	I	2	**Upstream Clock:** This LVDS input clock comes from the upstream node and drives the ring interface and chip core when CKSRC is negated.
UpSS[5:0]	I	12	**Upstream Sub-Symbol:** These 6 LVDS inputs for the Ring interface receive the divided 42-bit symbol from the downstream port of the previous node.
DnCLK	O	2	**Downstream Clock:** This LVDS output clock signal is derived from the clock that drives the Ring interface. The transitions on the DnSS are source synchronous with transitions on the DnCLK signal.
DnSS[5:0]	O	12	**Downstream Sub-Symbol:** These 6 LVDS outputs for the Ring interface drive the divided 42-bit symbol for the upstream port of the next node.
TxCLK	I	1	**Transmit Clock:** On the Client interface, all transmit port signals are synchronous to the rising edge of this clock
TxT[2:1]	I	2	**Transmit Type:** On the Client interface, this field defines (as head, data, frame, tails or null) the contents of the symbol on TxS: In the previous clock cycle when $\overline{\text{PIPE}}$ is asserted, pipelined timing. In the current clock cycle when $\overline{\text{PIPE}}$ is negated, non-pipelined (bridge) timing.
TxS[31:0]	I	32	**Transmit Symbol:** On the Client interface, these signals form the data bus of the transmit port.

1.0 Signal Description (Continued)

Pin Name	I/O	No.	Description
TxOK	O	1	**Transmit Okay:** On the Client interface, this is the transmit port status signal. It tells the client whether or not another non-null symbol can be accepted. Loading of non-null symbols must cease within 20 symbols of the negation of TxOK. Transmission may not resume until TxOK is reasserted.
RxCLK	I	1	**Receive Clock:** On the Client interface, all receive port signals are synchronous to the rising edge of this clock.
RxT[2:1]	O	2	**Receive Type:** On the Client interface, this field defines (as head, data, frame, tails or null) the contents of the symbol on RxS: In the next clock cycle for when $\overline{\text{PIPE}}$ is asserted, pipelined timing. In the current clock cycle when $\overline{\text{PIPE}}$ is negated, non-pipelined (bridge) timing.
RxS[31:0]	O	32	**Receive Symbol:** On the Client interface, these signals form the data bus of the receive port.
RxSTALL	I	1	**Receive Stall:** On the Client interface, when $\overline{\text{RxSTALL}}$ is asserted: When $\overline{\text{PIPE}}$ is asserted, pipelined timing: non-null RxS shall remain for the next clock cycle. When $\overline{\text{PIPE}}$ is negated, non-pipelined timing: RxT will indicate a null for the next clock cycle and RxS shall remain.
$\overline{\text{RxOE}}$	I	1	**Receive Output Enable:** On the Client interface, when asserted this signal enables outputs RxS[31:0]. When negated, the RxS are TRI-STATE®.
RxET[2:1]	O	2	**Receive Early Type:** On the Client interface, this field identifies in advance whether the information entering the Rx Port block is a head, data, frame, tail or null.
RxNBL [3:0]	O	4	**Receive Nibble:** On the Client interface, it contains one of the 16 selectable fields of two readable intemal registers (Diagnostics register, symbol on RxS output).
RxSEL [3:0]	I	4	**Receive Select:** On the Client interface, selects one of the 16 fields appearing on the RxNBL, Codes from 0 to 7 select 4 bit fields at the current output driver of RxS, codes of 8 or above select internal diagnostics status bits.
QR1001 **MODE[0]** (RxSEL[0]) **MODE[1]** (RxSEL[1])	 I I		**Operation Mode Selection (Either QR0001 or 0R1001):** The operation mode selection occurs on the release of reset. If MODE0 and MODE1 are sampled low on the release of reset, the part will operate in QR0001 mode. If MODE0 is high and MODE1 is low, the part will operate in QR1001 Enhanced mode. The two other conditions are reserved for future use. **RxSEL3 RxSEL2 MODE1 MODE0 Operation Mode** X X 0 0 QR0001 X X 0 1 **QR1001 Enhanced QuickRing** X X 1 1 Reserved X X 1 0 Reserved
V_{CC}	N/A	13	**Power Pin**
GND	N/A	29	**Ground Pins**

Note 1: Signal Name: The overline indicates that the signal is active low.

Note: The following sections assume a 50 MHz ring clock. The QR1001-33VUL and the QR1001-40VUL have maximum ring clocks of 33 MHz and 40 MHz respectively.

2.0 Basic Structure

The QuickRing Controller has two interfaces: the Ring Interface and the Client Interface. Each interface has two ports (see *Figure 2-1*). All ports on the QR1001 are unidirectional so that incoming and outgoing data can be queued simultaneously.

The two **Ring Interface** ports are:

(1) upstream port for arriving traffic,

(2) downstream port for departing traffic.

The Ring Interface forms the link to other nodes on the point-to-point QuickRing architecture. QuickRing technology connects multiple nodes by attaching the upstream port of each node to the downstream port of another node. The ring ports, upstream and downstream, are 6 bits wide plus a clock. The ring interface is implemented using LVDS drivers and receivers. The Ring Interface signals are not accessible from the board except through the controller. The on board logic connects to the QR1001 controller via the Client interface.

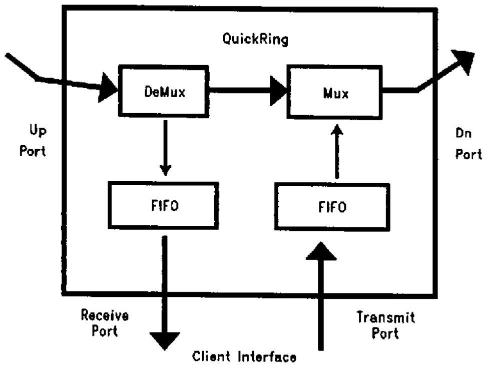
The two **Client Interface** ports are:

(1) the transmit port for locally generated symbol streams,

(2) the receive port for locally-absorbed symbol streams.

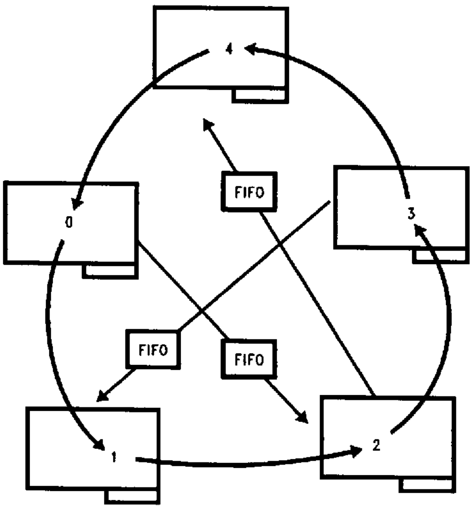
The transmit and receive ports have a 32-bit data path which use TTL compatible I/Os. The Transmit (Tx) and Receive (Rx) ports each have a separate clock plus control signals for information flow. Also, some QR1001 internal status bits can be read through the receive interface. All on board circuitry interfaces to the Client transmit and receive ports, never to the Ring ports.

QuickRing technology transmits data streams between nodes on the ring. The goal of QuickRing technology is to pipeline data streams and not just to facilitate memory access. Imagine connecting two cards together via a FIFO chip. One card can load data into its side of the FIFO, and the other card can extract data from the other side of the FIFO. QuickRing is logically equivalent to placing a large FIFO between pairs of Quick-Ring nodes (see *Figure 2-2*).



TL/F/12048-5

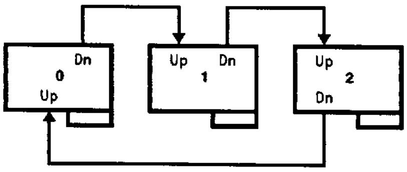
FIGURE 2-1. The QuickRing Controller Has Four Ports



TL/F/12048-3

FIGURE 2-2. Logical Data Flow In Ring (QuickRing Virtual FIFOs)

Figure 2-3 shows that data physically moves in a ring from card to card, data traverses the ring until it arrives at the final destination. Physical data flow is unidirectional, and propagates downstream between nearest neighbors.



TL/F/12048-7

FIGURE 2-3. Physical Data Flow In QuickRing

3.0 Client Interface

3.1 TYPE AND SYMBOL FIELDS AT THE CLIENT PORTS

The QuickRing client can multiplex multiple independent data streams to and from the transmit (Tx) and receive (Rx) ports of the controller. The symbol consists of the main data fields, TxS[31:0] and RxS[31:0]. The type fields, TxT[2:0] and RxT[2:0], identify the nature of the symbol field information as: directed head, multicast head, data, data-tail, frame, frame-tail or NoSymbol (see Table 3-1). The multicast head, data-tail and frame-tail are not identified in the QR0001. A third type bit has been added to the QR1001 to identify these new symbol types. The addition of multicast allows data to be sent from one source to multiple target nodes. Tail symbols mark the end of data streams in fixed length packet or multicast mode.

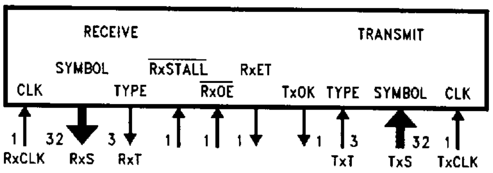
The transmit port can be thought of as the input to a bank of fast, deep FIFOs, connected to other nodes on the ring. The receive port can be treated as the output of the bank of FIFOs connected to other nodes on the ring. *Figure 3-2* illustrates the controller's client interface.

TABLE 3-1. QR1001 Client Type Field Definitions (TxT[2:0] and RxT[2:0])

Type	Name	Description
0	Directed Head	First symbol of a stream or packet, specifying the path to a single target. This stream is reservation based.
1	Multi-cast Head	First symbol of a packet destined for one or more targets. This stream does not use the reservation based ring protocol.
2	Data	32-bit payload symbol is a Data.
3	Data-Tail	Data symbol which is the last symbol in a fixed length and multicast stream or packet.
4	Frame	Specially marked 32-bit payload symbol is a Frame.
5	Frame-Tail	Frame symbol which is the last symbol in a fixed length and multicast stream or packet.
6	Reserved	Future use
7	NoSymbol	No associated symbol

QR0001 Client Type Field Definitions (TxT[2:1] and RxT[2:1])

Type	Name	Description
0	Head	Associated symbol is a head symbol
1	Data	Associated payload symbol is a data
2	Frame	Associated payload symbol is a frame
3	Null	No symbol associated with this Type



TL/F/12048-8

Figure 3.2: Client Ports of a QuickRing Controller

3.2 CLIENT TRANSMIT PORT

Figure 3-3 shows the block diagram of the transmit port. The transmit block of QR1001 is formed by: Tx Port, Tx Resynchronizer, Tx Router, and 3 independent FIFOs. All of these blocks form the transmit pipeline.

1) The Tx Port is the first stage into the transmit pipeline. The Transmit port is a 4 deep pipeline.

2) The Tx Resynchronizer handles the frequency difference between the Tx Port and the ring logic of the controller. The Resynchronizer is a 32-deep asynchronous FIFO.

3) The Tx Router directs the streams to the appropriate channel efficiently (described later).

4) FIFOs X and Y are meant for handling one independent high bandwidth stream each (either directed or multicast), and the LB (Low Bandwidth) FIFO is meant for low bandwidth transmissions. The FIFOs contain the payload (data or frame) part of the client stream. (The Head information is held in a separate holding latch internally).

   The **sole** purpose of providing two normal (high bandwidth) FIFOs (X and Y) is so the client may switch from transmitting one stream to another without slowing down or wasting available ring bandwidth during the context switch.

   On release of $\overline{\text{RESET}}$, any payload symbols at the transmit port are ignored until the first head symbol is presented at the input of the Tx Port. QR1001 always checks for consecutive heads and ignores all redundant heads in variable length mode. The type and symbol fields are latched internally according to the timing specified by the state of $\overline{\text{PIPE}}$.

   When the client starts a transmission, it writes a head followed by a stream of payloads. QR1001 receives these symbols through the transmit port and directs them to either the X, Y or LB FIFO. Any head symbol with the CONN (see Section 3.6) field equal to 1 is always routed to the LB FIFO, as is every payload symbol following such a head. Any other head with the CONN field equal to 0, and all payloads following such a head, are routed to either the X or Y FIFO.

   QR1001 can handle **one independent data stream through each of the X and Y FIFOs, a total of two streams at once.** Even if the FIFO is not full, the FIFO will store data associated only with a single head. Multiple data streams with various heads will not be held in a single FIFO. (Except the LB FIFO which will queue more than one head and associated payloads.) The subsequent data streams, with different heads, will be held in the Tx pipeline, until either FIFO X or Y empties. Then the data (with the different head) is allowed to further proceed in the pipeline.

3.0 Client Interface (Continued)

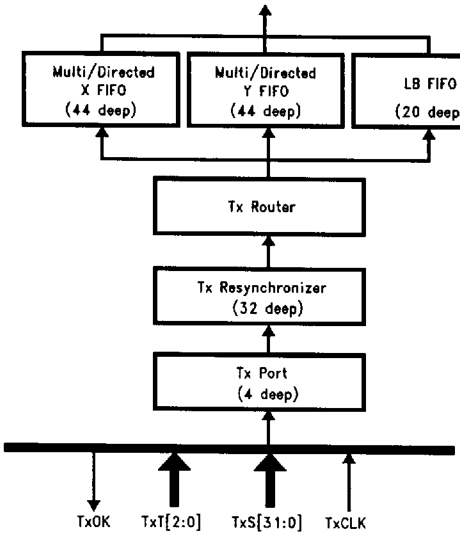
5) The LB FIFO is different. Several streams with different heads can flow through the LB FIFO at one time. When more than one payload is loaded following a single head, each payload is sent onto the ring with its own head and associated reservation protocol. The LB FIFO is intended to allow small content streams better access to the ring and a greater Target FIFO availability.

For all transmissions, low bandwidth or normal, QR1001 will keep TxOK asserted as long as there is space for 20 or more symbols in the transmit pipeline. As soon as the transmit pipeline has space for **only** 20 more symbols, TxOK negates. The initial negation of TxOK indicates to the client interface that it must stop transmitting, non-null symbols soon. TxOK is the only handshake mechanism at the transmit port. If TxOK asserts again, the count is voided and the client can write to the TxPort as many symbols as it wants. If TxOK negates again, the client must stop writing non-null symbols within 20 valid transactions.

The client may pause transmission at any time by presenting the null type code to the transmit port.

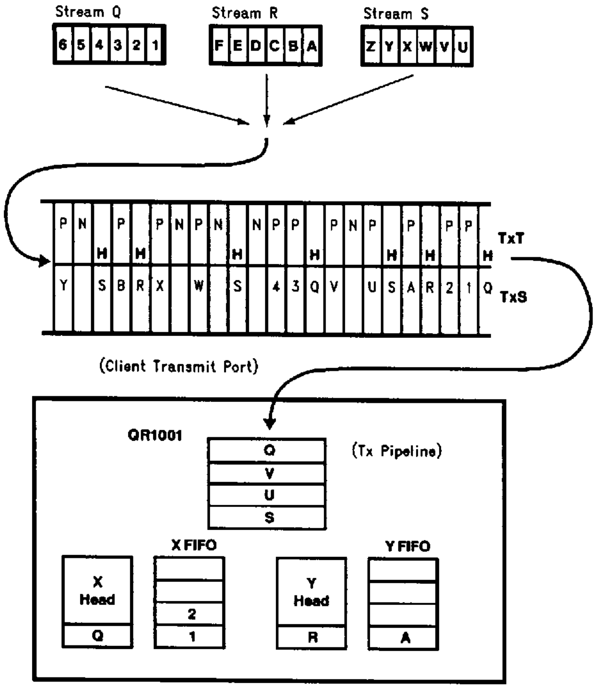
When the client interface wishes to begin transmission of a data stream, the client first writes a head (H) to the transmit port. From then on, every payload symbol (P) (type = data, frame, data-tail or frame-tail) sent to the transmit port is assumed to belong to the stream identified by the head. The data stream that the client writes at the transmit port is unbounded. However, if a new head, differing in at least one bit, is written to the transmit port, the data stream that follows is associated with the new head. If at any time the client is not prepared to transmit either a payload or a new head, a NoSymbol (N) may be introduced into the transmit data stream. NoSymbols (Nulls) do not propagate into the QR1001 QuickRing controller. Logically distinct data streams can be multiplexed together and loaded into the QR1001 transmit port. The client is free to switch between source streams at its convenience, as long as it introduces a new head when the switch occurs.

Figure 3-4 shows how three independent streams (high bandwidth) may be multiplexed from the Client Transmit Port into the QuickRing controller. Stream Q goes first, sending 2 payloads. It is followed by 1 payload from stream R, then by 2 payloads from stream S. Two more symbols from stream Q are sent, etc.



TL/F/12048-2

FIGURE 3-3. QR1001 TxPort



TL/F/12048-9

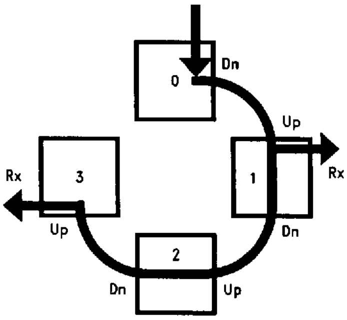
FIGURE 3-4. Logically Distinct Streams of Data Can Be Multiplexed into the Tx Port

3.3 CLIENT TRANSMIT PORT—NEW FEATURES

The client transmit port accommodates the multicast and fixed ring packet size features. This is an upgrade to the QR0001 functionality. *Figure 3-3* shows the X FIFO and Y FIFO used for both the Multicast and Directed streams. The LB FIFO is unchanged from the QR0001. It never handles Fixed size packets, either multicast or directed.

3.3.1 Multicast

The Multicast is used to send non-reservation based streams to a subset of all nodes in the system. If all nodes are in the subset, then the multicast is a broadcast. Non-reservation means the voucher and ticket FIFO reservation protocol is turned off. This reduces ring latency but makes transfers unreliable because target FIFO space is not guaranteed by reservation.



TL/F/12048–10

FIGURE 3-5. Multicast Data Flow Source Node 0, Target Nodes 1 and 3

Multicast uses the stream header to address the delivery pattern. There is a group field and a multicast field. The multicast field has 16 bits, one for each node in the local ring. If the bit corresponding to the node address is set and there is Target FIFO space available, then that node will absorb the multicast packet, clear the bit and pass it downstream. If there is no FIFO space, the bit is still cleared and packet forwarded only. If the bit is not set, the node only passes the multicast packet downstream. The stream is eliminated from the ring when all bits are cleared in the Multicast field. The Multicast field must never have a bit set that corresponds to a node that is not present in the ring. For example, if the ring Max ID is 8, no bit greater than 8 should be set in the Multicast field.

The group field is available to designate multicast targets in rings beyond bridge hops (see Section 3.13 for discussion of bridge hops). Intelligent bridges must identify the multicast field patterns in ring hops. These group field maps must be set up during configuration, prior to transmitting the multicast. The group field is passed through the target RxPort exactly as it was entered at the source TxPort. The bridge logic external to the QR1001 must have the intelligence to look up the group in a table and set the appropriate multicast bits for the next ring. If another bridge hop is in the route, the group field must be correct for the next intelligent bridge.

Multicast packets are always fixed in length and never Low Bandwidth on the ring. The length is set by one of 3 occurrences at the TxPort.

1. The loading of a tail, either frame or data.
2. The loading of another head of any type.
3. The loading of the twenty-first payload symbol of any type. This occurrence also attaches the previous head to the next multicast packet.

3.3.2 Fixed Packet Mode

Fixed packet mode allows the client to control the contiguous packet size of transmitted data onto the ring. This packet benefits the client design because the client can guarantee delivery of contiguous streams at the Target RxPort. The maximum fixed stream size is limited to the maximum ring packet size of 20 payload symbols.

A fixed packet at the RxPort may be longer than the original transmitted packet if access symbols are inserted on the ring, which may result in nulls at the RxPort. If the RxPort is stalled, the packet will be compressed in the Rx pipeline. This will eliminate nulls. Any shorter size contiguous packets can be used, such as 16 symbols for 64 byte payloads.

Fixed packet mode is entered by setting the appropriate value in the access field of the stream header. Table 3-2 shows this access field value in the second row for the fixed packet mode. This field is reserved in the Client interface header format in the QR0001 mode. The field can be one of 2 values as represented in the Table 3-2.

Data streams at the transmit client port may send fixed packets of different lengths. There are three ways to transmit a fixed size packet onto the ring.

1) Transmitting a tail symbol defines the last payload symbol of a fixed packet.
2) Transmitting a new head symbol; the previous payload becomes the last symbol, tail, of the fixed packet and will appear as a tail at the receiving node.
3) Transmitting the twenty-first payload symbol. The twentieth symbol becomes the last symbol, tail, of the fixed packet (max packet length) and will appear as such at the receiving node.

Any of the above conditions will initiate the voucher and ticket reservation on the ring. The packetized data is launched at the downstream ring port when a ticket has been received from the target node. The packet then travels to the destination and appears at the receive port as one continuous stream (no interleaved streams, but possibly inserted nulls due to deleted access symbols which were inserted on the ring) ending with the tail symbol.

3.0 Client Interface (Continued)

TABLE 3-2. Directed Head Access Field Description

ACC[1,0]	Name	Description
0	Variable Directed	The stream marked by this head shall follow a specified path to a single target using the voucher-ticket protocol within each ring and may modify packetization. Streams may arrive at target RxPort non-contiguously.
1	Fixed Directed	The stream marked by this head shall follow a specified path to a single target using the voucher-ticket protocol within each ring. The only modification of the packet may be the insertion of vouchers and/or tickets (null symbols at the Rx port). This mode of operation requires a Tail symbol, either data or frame, be used as the final payload symbol. Streams less than 20 payload symbols will be contiguous throughout the ring.
2	Reserved	Do not use
3	Reserved	Do not use

3.3.3 Tail Symbols

Tail symbols are an added feature that gives flexibility to the device. The tail symbol is necessary for the fixed length directed stream and multicast operation. Variable directed streams ignore tail symbols. If a tail symbol is transmitted during a variable directed stream operation, the received type will be converted to a normal data or frame.

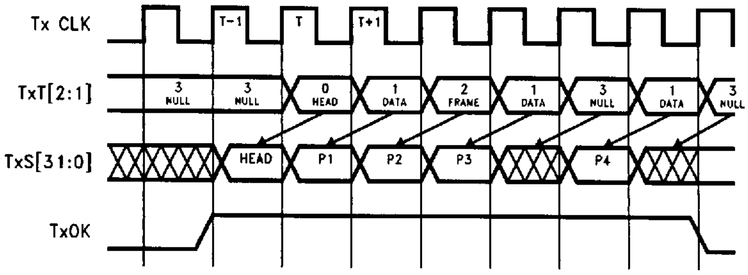
TABLE 3-3. Tail Symbol Format

TxT[2:0] RxT[2:0]	TxS[31:0] RxS[31:0]								
2:0	31:30	29:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
3	Data-Tail Payload								
5	Frame-Tail Payload								

3.3.4 Tail Type Conversion

There are several situations in which a transmitted type will be converted before arriving at the receive port. QR1001 will modify the transmitted type field under conditions given in Table 3-4.

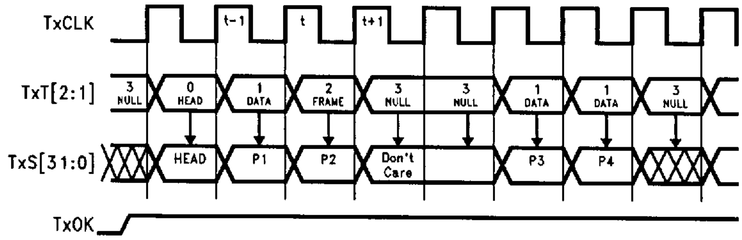
Transmit Port Timing when $\overline{\text{PIPE}}$ Is Asserted



TL/F/12048-11

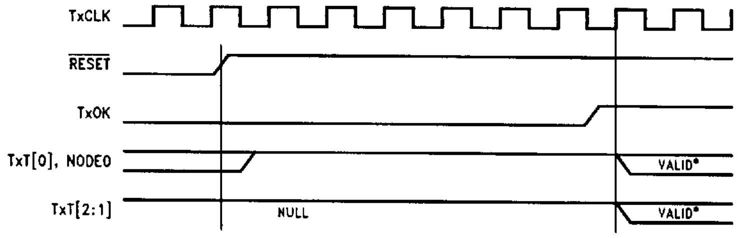
FIGURE 3-6. When $\overline{\text{PIPE}}$ Is asserted, the Type Field lags the Symbol Field by one clock cycle at the transmit port.

Transmit Port Timing when $\overline{\text{PIPE}}$ Is Negated



TL/F/12048-12

FIGURE 3-7. When $\overline{\text{PIPE}}$ Is negated, the Type Field and the Symbol Field are loaded during the same clock cycle.



TL/F/12048-13

*Type is valid when initialization complete (Node ID and Max ID received)

FIGURE 3-8. TxPort Timing Following Initialization

TABLE 3-4. Tail Type Conversion

Packet Length	Tx Type	Rx Type
Fixed & Multicast	Data followed by head	Data-Tail followed by head
	Frame followed by head	Frame-Tail followed by head
Variable	Data-Tail	Data
	Frame-Tail	Frame
Fixed & Multicast	20th Symbol (data)	Data-Tail
	20th Symbol (frame)	Frame-Tail

3.4 TRANSMIT PORT TIMING RELATIONSHIPS

When $\overline{\text{PIPE}}$ is asserted (low voltage level) the type field, TxT, accompanying the symbol field, TxS, is loaded into the controller one clock cycle after the symbol that it identifies. See *Figure 3-6.* When a symbol is presented on TxS at time t, then the corresponding code is presented on TxT at time t+1. The purpose of delivering the TxT field one clock cycle after the symbol is so a simple, synchronous state machine has one full clock cycle to compute the TxT code without using external latches on the symbol field.

When $\overline{\text{PIPE}}$ is negated (high voltage level), the type field, TxT, accompanying the symbol field, TxS, is loaded into the controller during the same clock cycle as the symbol it identifies. This non-pipelined mode supports back-to-back client port connection during bridging. Refer to *Figure 3-7.*

The TxOK function and timing remains unchanged regardless of the level of the $\overline{\text{PIPE}}$ signal, giving a 20 symbol warning that transmission of non-null symbols may need to cease.

The timing sequence for the valid transmission of TxT[2:0] is (see *Figure 3-8*):

1) Release reset. TxT[2:0] should be driven high (null) after reset is released.

2) Wait for TxOK to assert.

3) TxT[2:0] is valid on the next rising edge of TxCLK.

When TxOK asserts, TxT[2:0] may be driven with a non-null type. Even though the node ID and max node ID may not have been received at the Rx Port, symbols can be written to the Tx Port when TxOK asserts. They will not be sent to the ring by the controller until the initialization is completed and the source node has the node ID to stamp in the source field. It is recommended that start of transmission be delayed until max node ID for the ring is checked by the sourcing client. This will prevent an address greater than the maximum node address being introduced into the ring. This would cause an abort condition that must be cleared by another reset and intialization.

When operating in either QR1001 or QR0001 mode, it is recommended that TxT[2:1] be held in a high state (null type) until type fields are valid.

3.5 CLIENT RECEIVE PORT

Figure 3-9 shows the QR1001 receive block and part of the forwarding path.

1) The Upstream Router/Multicast Handler, LB Target Handler, Directed Target Handler, and the Ring FIFO are part of the forwarding path.

   The LB Target Handler processes LB vouchers targeted to this node into tickets. These tickets are forwarded to the source node through the downstream port.

   The Directed Target Handler processes vouchers targeted to this node into tickets. These tickets are forwarded to the source node through the downstream port.

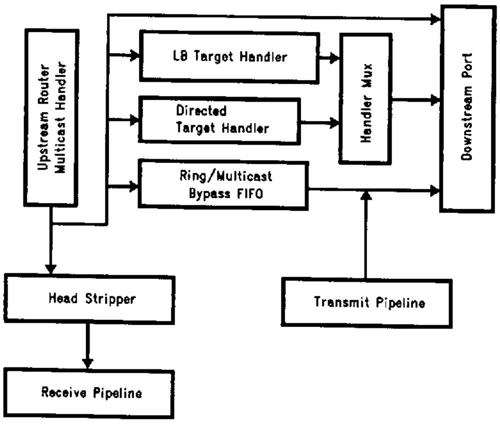
   The Ring FIFO stores incoming data from the upstream port that is intended to be forwarded to other nodes on the ring.

2) The Head Stripper is the first of the receive blocks. It removes all heads except those identifying the beginning of a stream.

3) The Target FIFO reserves space for 3 normal packets and 6 LB packets.

4) The Rx Resynchronizer handles the frequency difference between the client interface and the ring logic of the controller.

5) The Rx Port is the last stage between a data stream and the client interface.



TL/F/12048-04

FIGURE 3-9. QR1001 Target and Multicast Handlers and Bypass FIFO

On release of Reset, the first two non-null symbols that appear at the RxS[31:0] are the node ID of the controller, and the largest maximum ID number on the ring. RxT will indicate a non-null type for these two IDs. (See Section 9.1.)

RxET alerts the client interface, up to 20 symbols prior to the output stage, to the type of data entering the receive pipeline. The RxT, when $\overline{\text{PIPE}}$ is asserted, leads by one clock the symbol that it identifies; thus, giving the client a one clock cycle advance notice of the symbol about to appear at the RxS[31:0]. The client may choose to stall the symbol at RxS[31:0] if desired.

At the Rx Port, a contiguous, variable length data stream is unbounded, and data belonging to the same head is marked by a single initial head appearance. At the Rx Port, there may be no evidence that the stream is packetized on the ring. A new head will appear only when there is a change in data stream context. A long data stream transmitted in multiple packets from one node to another, will appear at the Rx Port as a single head followed by a long data stream, unless broken by a different, unique stream.

In cases where one target node is the subject of multiple transmissions from several nodes, multiple streams, marked by head symbols, will appear multiplexed at the Rx Port. The same will occur if one source node is sending different streams to the same target. A stream is treated as a different stream if the 32-bit head symbol differs by at least one bit.

3.5.1 Upstream Router

Multicast mode does not use the voucher-ticket protocol on the ring. For this reason, multicast ring-packets will arrive at a target without a FIFO reservation.

At the Upstream Router/Multicast handler, the multicast head is checked for the Multicast Field bit to be set that corresponds to that node's ring address. For example, Bit 1 corresponds to Node 1. If the node bit in the multicast field is set then the bit is cleared and the packet is routed to the Head Stripper. The Head Stripper is the beginning of the receive path. If there are still any multicast bits set, then the packet is routed to the Ring/Multicast Bypass FIFO and forwarded onto the next node. If there are no more bits set, then the packet is eliminated from the ring. If the node address bit is not set, the multicast packet is routed to the Bypass FIFO and immediately forwarded. This allows for the minimum delay through each node and the almost simultaneous multicast to each destination. If the packet should return to the source node, and the bit is set for that node, the packet is routed to the head stripper. The packet is not forwarded a second time from the source node.

3.5.2 Receive Pipeline

Since tickets and vouchers are not used during the multicast function, there are different functions in the target handler depending on the type of packet. In multicast, if a ticket is available, the ticket counter is decremented and the packet is copied into the Target FIFO. When the mulitcast packet in the Target FIFO is unloaded through the RxPort by the client, then the ticket counter is incremented and another multicast packet can be copied. If all tickets are outstanding, the packet is not copied locally. This will result in a multicast target not receiving the multicast packet data.

3.5.3 Head Stripper

Shifting Hop Fields

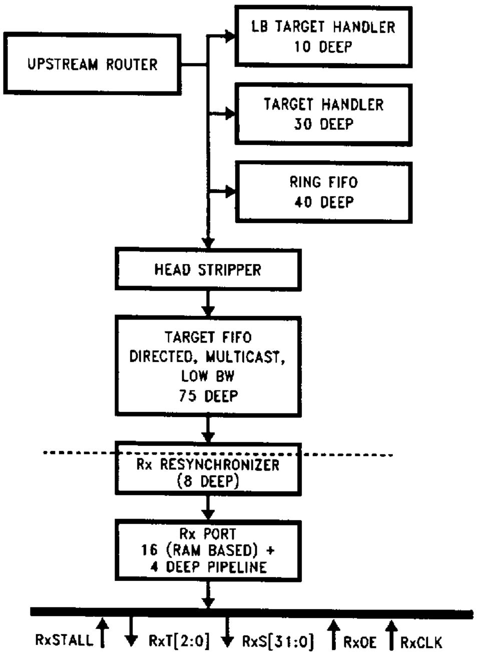
The Head Stripper has the function of shifing hop fields in the QR1001 and QR0001. There is a difference in the QR1001 mode because of the addition of Multicast heads and the Hop 5 Field being changed to Hop Count Field (HCNT). The Head Stripper only shifts the 4 remaining Hop fields in Directed heads. The HCNT is never shifted. Instead, the HCNT is decremented as it passes the Head Stripper. No shifting occurs for the Multicast heads.

When the device is operating in QR0001 mode, all 5 fields are treated as hop fields (no hop count field) and all fields are shifted as in the QR0001 device. See Section 3.13 for more details on hop field shifting.

Fixed Packets

The Fixed length packets will have a head at the beginning of each packet that emerges from the RxPort. The head stripper that usually eliminates redundant heads used in the ring packetization is deactivated when Fixed length packets are traversing the receive pipeline. It is only variable, directed, redundant heads that are stripped. Fixed packets will always appear at the RxPort with a head at the beginning and a tail symbol at the end.

3.0 Client Interface (Continued)



UPSTREAM ROUTER
LB TARGET HANDLER 10 DEEP
TARGET HANDLER 30 DEEP
RING FIFO 40 DEEP
HEAD STRIPPER
TARGET FIFO DIRECTED, MULTICAST, LOW BW 75 DEEP
Rx RESYNCHRONIZER (8 DEEP)
Rx PORT 16 (RAM BASED) + 4 DEEP PIPELINE
RxSTALL
RxT[2:0]
RxS[31:0]
RxOE
RxCLK

TL/F/12048-14

FIGURE 3-10. Rx Port Client Interface

Multicast Packets

Multicast packets are fixed in length. Just as with the Fixed packets, they will always appear at the RxPort beginning with a head symbol and ending with a tail symbol. Redundant heads are not stripped at the receive port when operating in multicast or fixed packet mode.

3.6 Receive Port Timing Relationships

When $\overline{\text{PIPE}}$ is asserted (low level), the type field, RxT, at time t indicates the type of symbol presented at the output, RxS, at time t+1. See *Figure 3-11*. This is true as long as $\overline{\text{RxSTALL}}$ is negated.

If $\overline{\text{RxSTALL}}$ is asserted when $\overline{\text{PIPE}}$ is asserted (pipeline timing mode) **the RxS[31:0] output will stall at the first non-null symbol encountered in the pipeline after $\overline{\text{RxSTALL}}$ is asserted.** The symbol on RxS, when $\overline{\text{RxSTALL}}$ is asserted, will persist through the next clock cycle unless it corresponds to a null symbol. The $\overline{\text{RxSTALL}}$ input signal is only capable of holding a non-null symbol at the RxS output.

The client may need to examine some symbols within the symbol stream in order to determine their disposition. It is highly desirable to do so without employing added data path buffering external to the controller. QuickRing allows the client to examine the contents of the symbol at the RxS output through $\overline{\text{RxSTALL}}$, or a combination of RxSEL and RxNBL even with the RxS output drivers disabled.

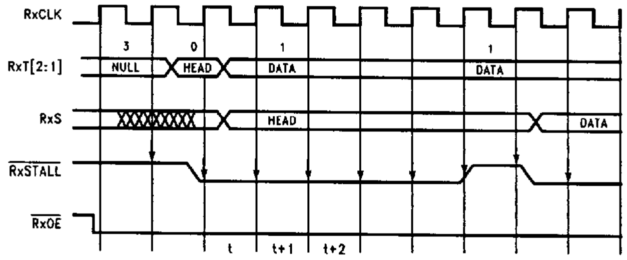
To further aid in the receive stream management, the symbol type field just entering the Rx Port Block of the receive pipeline is visible on RxET. Thereby the client can preview the symbol type in the receive pipeline before it appears on the RxT outputs. Thus it is possible to detect the presence of a head, data, frame, or tail in the pipeline even if up to 19 more symbols are stored ahead of it.

When $\overline{\text{PIPE}}$ is negated (high level), then the value of the type field RxT/TxT at time t corresponds to the value of the symbol field at the same time t. (See *Figure 3-7*.) When two QuickRing controllers are connected to form a bridge, the TxOK is connected to $\overline{\text{RxSTALL}}$ of the other controller. (Care should be taken in board layout to make sure timing between TxOK and $\overline{\text{RxSTALL}}$ to satisfy the setup/hold times.)

If $\overline{\text{RxSTALL}}$ is asserted when $\overline{\text{PIPE}}$ is negated (high level), at the next positive edge of clock:

1) RxT is forced to Null and

2) The RxS[31:0] persists.



TL/F/12048-15

FIGURE 3-11. $\overline{\text{RxSTALL}}$ Sampling Edge (Pipelined Mode)

3.0 Client Interface (Continued)

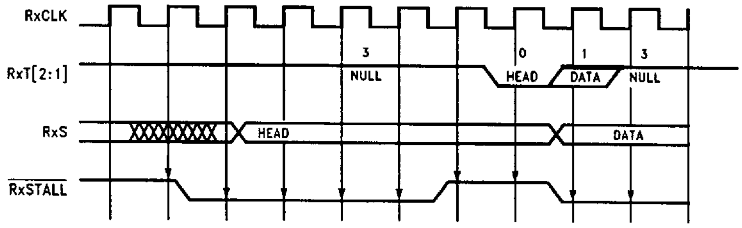
To summarize the Client Port Timing:

At the Rx Port, a non-null symbol remains valid at the RxS output in the presence of $\overline{\text{RxSTALL}}$.

At the Tx Port, when TxOK negates it indicates that the FIFO is nearly full and the client must stop transmission within 20 non-null symbols.

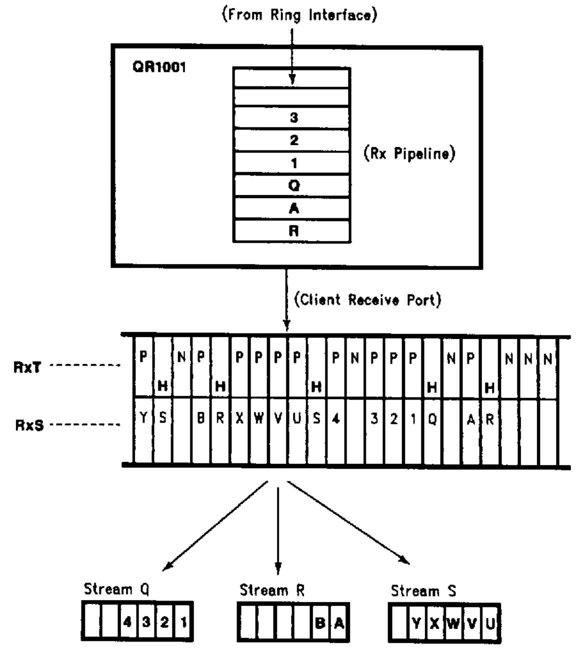
The $\overline{\text{PIPE}}$ input determines how the type fields, RxT and TxT, identify a symbol as it appears at RxS and TxS respectively.

At the receive port, many different arriving streams may be multiplexed together. Every switch to a new stream context is marked by a new head symbol. A single, variable length stream loaded into the Tx client port may be multiplexed with another data stream on the ring. This may cause the de-muxed variable length stream at the Rx port to appear different from the original stream loaded into the controller at the source node. The QuickRing protocol does not preserve the order of multiplexed streams, but it does preserve the first-in-first-out ordering of each individual stream.



TL/F/12048-16

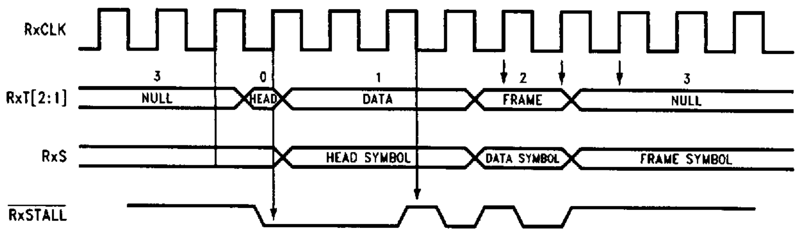
FIGURE 3-12. $\overline{\text{RxSTALL}}$ Sampling Edge (Non-Pipelined Mode)



TL/F/12048-17

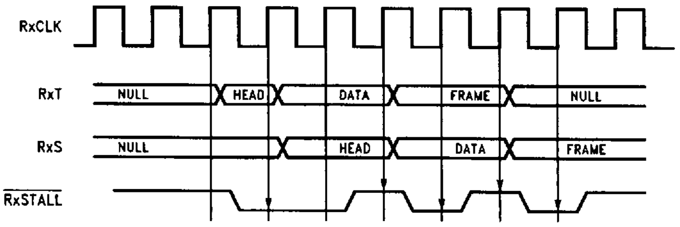
FIGURE 3-13. The Individual Ordering of Each Stream Is Preserved

3.0 Client Interface (Continued)



TL/F/12048-18

FIGURE 3-14. Holding Head Symbols for 2 Clocks and Holding Data and Frame Symbols for 1 Clock Each



TL/F/12048-19

FIGURE 3-15. When RxSTALL is asserted, it Stalls RxS and only Indirectly Stalls RxT. RxT always Identifies a FUTURE state of RxS, NEVER the present state, when PIPE is asserted.

TABLE 3-5. External Behavior of the RxSTALL Function

Input		Present State	Next State	Output
RxSTALL	RxT[2:0] = 7	Volatile	Volatile	RxS[31:0]
F	F	F	"F"	"Sy"
F	F	T	"F"	"Sy"
F	T	F	T	Sx
F	T	T	T	Sx
T	F	F	F	Sx
T	F	T	"F"	"Sy"
T	T	F	F	Sx
T	T	T	F	Sx

"Sy" emphasizes new symbols always start out as non-volatile.
Sx is the value of RxS[ ] during the present state.
Sy is the value of the symbol that follows Sx.

3.0 Client Interface (Continued)

Figure 3-4 relates to *Figure 3-13*. It shows that even though stream Q was loaded first, in *Figure 3-4*, stream R, arrives first, in *Figure 3-13*. Notice that at the output, the order within each individual stream is preserved.

3.6.1 Client Receive Port Interface Options (PIPE asserted)

One of the possiblilities when interfacing to the Client Receive Port, is to hold RxSTALL negated during normal operation. When the Client interface detects a non-null type it may assert RxSTALL during the next clock cycle to stall that particular symbol and the next non-null type (if available). See *Figure 3-14*.

3.6.1.1. RxSTALL Behavior

The timing diagram shown in *Figure 3-11* illustrates the timing of the RxSTALL input signal for QR1001.

1) At the rising edge of clock cycle *t* RxT is valid.

2) At the rising edge of *t+1*, RxSTALL must be asserted to stall the RxS. (RxS head symbol is present.)

3) At the rising edge of *t+2* RxS remains stalled.

RxSTALL is always sampled on the rising edge of the receive clock, regardless of operating mode.

External Behavior for RxSTALL

The following definitions and Table 3-5 are used to explain the function provided by the RxSTALL signal.

RxSTALL is false when at a high voltage and true when at a low voltage.

RxT[ ] = 7 means No-new-symbol is available for presentation at RxS[ ] on the next clock cycle.

VOLATILE is a state variable that is true when assertion of RxSTALL cannot inhibit the overwriting of the symbol at RxS[ ] by a new symbol. This is the bit that the system designer uses to recall if RxSTALL was ever left unasserted after the current symbol arrived.

RxS[ ] is the Rx Port data path output.

Sx is the value of RxS[ ] during the present state.

Sy is the value of the symbol that follows Sx.

Pipelined Mode

In Table 3-5.1, the T'[ ] variable represents the type of the next arriving symbol in the receiver pipeline. All of the type field columns—T'[ ], and RxT[ ] in the present and next states—are shown to take on either a null code or a non-null code of Ty or Tz. Although these are two- or three-bit fields, for the purpose of determining the present or next state, these columns represent binary values distinguishing between a null or non-null type code. The non-null codes are shown as Ty or Tz to illustrate their correlation to the value of RxS[ ] and their relative order in the pipeline (Tz follows Ty). RxS[ ] in the present state always has a value of Sx.

The *nullRxS* variable is not explicitly provided (but is derivable) outside the device; it should be interpreted to mean that the value of RxS[ ] in the current clock cycle is not eligible to be frozen at RxS[ ] during the next clock cycle by the assertion of RxSTALL during the current cycle. The output variable *newRxS* (also not explicitly provided but derivable) indicates whether the RxS[ ] output is actually updated with a new non-null symbol during the current clock cycle. Note that whenever *newRxS* is true, RxS[ ] changes from the old Sx to a new Sy. Note also that, although RxS[ ] may represent a null symbol (nullRxS is TRUE) it always retains the value of the last non-null symbol to be issued.

TABLE 3-5.1. QR1001 Pipelined Mode (PIPE = 0)

Row #	Input		Present State			Next State				Possible Next Row
	RxSTALL	T'[ ]	RxT[ ]	nullRxS	RxS[ ]	RxT[ ]	nullRxS	newRxS	RxS[ ]	
1	F	Null	Null	—	Sx	Null	T	F	Sx	1, 2, 5, 6
2	F	Ty	Null	—	Sx	Ty	T	F	Sx	3, 4, 9, 10
3	F	Null	Ty	—	Sx	Null	F	T	Sy	1, 2, 7, 8
4	F	Tz	Ty	—	Sx	Tz	F	T	Sy	3, 4, 11, 12
5	T	Null	Null	T	Sx	Null	T	F	Sx	1, 2, 5, 6
6	T	Ty	Null	T	Sx	Ty	T	F	Sx	3, 4, 9, 10
7	T	Null	Null	F	Sx	Null	F	F	Sx	1, 2, 7, 8
8	T	Ty	Null	F	Sx	Ty	F	F	Sx	3, 4, 11, 12
9	T	Null	Ty	T	Sx	Null	F	T	Sy	1, 2, 7, 8
10	T	Tz	Ty	T	Sx	Tz	F	T	Sy	3, 4, 11, 12
11	T	Null	Ty	F	Sx	Ty	F	F	Sx	3, 4, 11, 12
12	T	Tz	Ty	F	Sx	Ty	F	F	Sx	3, 4, 11, 12

3.0 Client Interface (Continued)

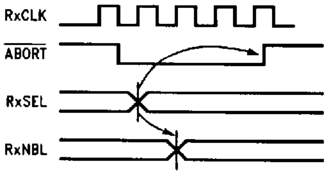
Understanding RxSTALL Operations (PIPE asserted)

Correctly predicting the behavior of the Rx port can be done by learning the following principles:

1) RxT always identifies a FUTURE state of RxS, NEVER the present state.
2) A symbol will only stall (remain NON-VOLATILE) if RxSTALL is asserted in the first clock cycle during which the symbol appears at RxS.
3) Even a NON-VOLATILE symbol can be overwritten if RxSTALL is not asserted.
4) Once RxSTALL is released a stalled symbol becomes VOLATILE, and cannot be stalled again.
5) RxS always holds a (non-null) symbol; RxSTALL controls whether or not the symbol is VOLATILE or NON-VOLATILE.
6) The null type code does not represent a symbol, but the ABSENCE of a symbol.
7) Even a VOLATILE symbol sits stably at RxS until another symbol arrives to overwrite it.

3.6.2 Rx Port Timing Relationships

Several receive port signals on the QR1001 have new relationships to the rising edge of the receive clock, even in QR0001 mode. RxSEL signals are now synchronous to the rising edge of RxCLK; in the original QR0001, RxSEL was independent of the receive clock. When RxSEL is changed, RxNBL is updated in the subsequent clock cycle. (*Figure 3-16*).



TL/F/12048-20

FIGURE 3-16. RxSEL Timing Relationships

As described in the previous section, RxSTALL is now latched with the rising edge of RxCLK.

TABLE 3-6. QR1001 Non-Pipelined Mode/Bridge Mode (PIPE = 1)

Input		Next State	
RxSTALL	T'[ ]	RxT[ ]	RxS[ ]
F	Null	Null	Null
F	Ty	Ty	Sy
T	X	Null	Sx

3.7 CLIENT INTERFACE FIELD DEFINITIONS

Table 3-7 shows the symbol field definitions for the Tx and Rx ports. Refer to Section 3.15 for details.

TABLE 3-7. Tx and Rx Port Symbol Field Definitions

Field	Decriptions
Type[2:0]	At the client ports, distinguishes heads, data, frame, tails and null.
CONN[1:0]	The connection code provides two types of transmission, normal and low-bandwidth. Low-bandwidth streams are transmitted with higher priority.
TRGT[3:0]	The target field contains the node ID of the target of the associated payload.
SRCE[3:0]	The source field contains the node ID of the source of the associated payload.
HOP1[3:0] HOP2[3:0] HOP3[3:0] HOP4[3:0]	In a multiple-ring topology they supplement source and target ID fields to route streams as they hop from ring to ring. Can provide unique stream identification (SID) for those who's source-to-target routes are identical.
HOP5[3:0]	*QR0001 Mode only*. Same function as above.
HCNT[3:0]	*QR1001 mode only*. Can be used to identify the number of bridge hops or location of stream ID that has been shifted by ring hops. The hop count field is decremented at each receive port.

At the client ports ACCess field should be [00] (variable packets) or [01] (fixed packets).

Table 3-8 shows the values of the connection field (CONN[1:0]). If a LB connection is requested, QuickRing takes the stream of payloads and parcels the data or frame symbols presented at the Tx Port and transmits them in 2 symbol ring packets, 1 head and 1 payload.

TABLE 3-8. Connection Field Definitions

CONN [1:01	Name	Description
0	Normal	Queue and accumulate symbols from the same stream at will, to maximize system efficiency and minimize system load.
1	Low Bandwidth (LB)	Do not concatenate with other data symbols from the same stream. Results in two symbol packets, head and payload.
2	N/A	Reserved
3	N/A	Reserved

3.8 CLIENT TYPE FIELDS

The TxT[2:0] and RxT[2:0] fields are the type fields at the transmit and receive ports, respectively They are encoded as shown in Table 3-9. Each 32-bit symbol written or read from the client ports is associated with one type field.

Note: When operating in QR0001 mode, only 2 type bits are used as in the QR0001 device.

3.0 Client Interface (Continued)

TABLE 3-9. Client Type Field Definitions (TxT/RxT)

Type	Name	Description
0	Directed Head	First symbol of a stream or packet, specifying the path to a single target. This stream is reservation based.
1	Multi-cast Head	First symbol of a packet destined for one or more targets. This stream does not use the reservation based ring protocol.
2	Data	32-bit payload symbol is a Data.
3	Data-Tail	Data symbol which is the last symbol in a fixed length and multicast stream or packet.
4	Frame	Specially marked 32-bit payload symbol is a Frame.
5	Frame Tail	Frame symbol which is the last symbol in a fixed length and multicast stream or packet.
6	Reserved	Future use
7	NoSymbol	No associated symbol

3.9 TRANSMIT PORT HEAD FIELDS

A head symbol must be loaded into the transmit port to begin transmission, or when the context of the loaded symbols is switched to another stream. Redundant heads are acceptable to the controller transmit port. If multiple heads are loaded without intervening data or frame symbols, then all but the last head are ignored. The transmit head information—access, connection, target and hop field—must be provided at the client transmit port. The controller adds its own ID to the head's source field internally, based on the local node ID value that was set during initialization. Table 3-10 shows the format of a head symbol that the local client must load into the Tx Port to establish a connection.

TABLE 3-10. TxPort Head Format

TxT[2:0]	TxS[31:0]								
2:0	31:30	29:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
Type	Acc	Conn	Srce	Trgt	HOP				
0	0 Vary	Conn 0/1*	XXXX	Trgt	Hop1	Hop2	Hop3	Hop4	HCNT
0	1 Fixed	0	XXXX	Trgt	Hop1	Hop2	Hop3	Hop4	HCNT
0	2 & 3	Reserved							
1	XX	0	XXXX	Group Field		Multicast Field			

*0: non-low-bandwdth packet, and 1: low-bandwidth packet

3.10 RECEIVE PORT HEAD FIELDS

The receive port head symbol format contains the same fields as are found in heads at the transmit port, but are shifted from their original positions when they exit the receive port. The purpose is to support routing of streams in multiple-ring topologies. Head symbols only appear at the receive port when there is a stream context change in variable mode or a new packet in fixed and multicast mode. Redundant head symbols are deleted when operating in variable packet mode. In fixed and multicast packet mode, head symbols appear at the beginning of each packet. Head symbols at the receive port hold valid information in the access, connection, source, target, and hop fields. Table 3-11 shows the format of the head field at the receive port of the controller in the same ring as the original source. If a ring hop has occured, another shift of fields would have occured with respect to the original source.

TABLE 3-11. Receive Port Head Fields

RxT[2:0]	RxS[31:0]								
2:0	31:30	29:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
Type	Acc	Conn	Trgt	HOP					
0	0/1	Conn	Trgt	Hop1	Hop2	Hop3	Hop4	Srce	HCNT

3.11 PAYLOAD SYMBOLS AT THE Rx AND Tx PORTS

Payload symbols at the transmit or receive ports follow the head symbol that identifies them. A payload consists of a sequence of data and/or frame symbols that are distinguished by a 2 or 4 (1 or 2 in QR0001 mode) in the accompanying type field, refer to Table 3-12. The final payload in a fixed size packet is designated by the tail symbol types 3 and 5 (data-tail and frame-tail). When $\overline{\text{PIPE}}$ is asserted, the type field at the receive port leads the symbol field that it identifies by one clock cycle, and at the transmit port it lags the symbol field by one clock. However, if the controller is in non-pipelined timing ($\overline{\text{PIPE}}$ is negated) the type field corresponds to the symbol at the same clock.

TABLE 3-12. Payload Symbols at Tx/Rx Ports

Type[2:0]	Tx/RxS[31:0]
2	DATA. User Defined Information
3	DATA-TAIL. User Defined Information (fixed/muiticast)
4	FRAME. User Defined Information
5	FRAME-TAIL. User Defined Information (fixed/muiticast)

3.12 NULL SYMBOLS AT THE Rx AND Tx PORTS

The lack of a head symbol or payload symbol is indicated, at the Rx port by NoSymbol (null) symbols with type field of 7 and all RxS bits "don't cares". At the Tx port, if a head or payload is not ready to be transmitted, a null symbol code should be presented at the TxT. The value of the type fields at the client ports is as indicated in Table 3-13.

TABLE 3-13. Null Symbol Format

Type[2:0]	Tx/Rx S[31:0]
7	NULL. Don't Cares

3.0 Client Interface (Continued)

3.13 THE HOP FIELDS AND THE UNIQUENESS OF SYMBOL STREAMS

The identity of a symbol stream is fixed by the combination of the connection, source, target, and hop fields in the stream's head symbol. If the heads of symbol streams differ in any of these fields, then they represent different symbol streams.

The symbols of a unique stream will always arrive in order. Multiple streams targeted at the same node may arrive interleaved. The interleaving will always be indicated by an appropriate head symbol, identifying the switch in stream context.

The Hop Fields were created to route packets through bridges in multiple ring topologies. In a single ring topology the hop fields may be used to identify different streams. The hop fields can be used to distinguish different data streams that have the same source and destination. If only one of the fields is used for stream ID, there can be 16 different streams having the same source and target. However, in multiple ring topologies, every time a bridge is crossed, one hop field is used. Therefore, it is lost for identifying unique data streams from a single source node. Table 3-14 shows the hop field locations at the transmit port and Table 3-15 shows the hop fields at the receive port.

3.14 HOP COUNT FIELD

The QR1001—operating in QR1001 mode ONLY—employs the Hop Count field (HCNT) which replaces Hop Field 5. This enables such functions as a bridging node to also be addressed as a leaf node (an intelligent node which also functions as a bridge). For example, the source node will identify the number of bridge hops in the header. Each Head Stripper decrements the HCNT field. If the HCNT is Fh when received at a leaf node, then external logic would decide there are no more bridges to the destination and will not pass the stream onto the next bridged ring. If the value of HCNT is 0h or greater (not Fh) in the HCNT then the stream would be bridged to the next ring by external logic. The HCNT decrementing is integrated into the QR1001 part. The HCNT field is not shifted when the other hop fields are shifted.

Client Port Field Formats (QR1001 mode)

Tx Port Head Field

Type	Acc	Conn	Srce	Trgt	HOP				
TxT[2:0]	TxS[31:30]	TxS[29:28]	TxS[27:24]	TxS[23:20]	TxS[19:16]	TxS[15:12]	TxS[11:8]	TxS[7:4]	TxS[3:0]
0/1	0/1	Conn	XXXX	Trgt	HOP1	HOP2	HOP3	HOP4	HCNT

Rx Port Head Field (same ring as Srce node)

Type	Acc	Conn	Trgt	HOP					
RxT[2:0]	RxS[31:30]	RxS[29:28]	RxS[27:24]	RxS[23:20]	RxS[19:16]	RxS[15:12]	RxS[11:8]	RxS[7:4]	RxS[3:0]
0/1	0/1	Conn	Trgt	HOP1	HOP2	HOP3	HOP4	Srce	HCNT

Tx and Rx Ports Payload Symbols

T[2:0]	Tx/Rx S[31:0]
2	DATA. User Defined Information
4	FRAME. User Defined Information
3/5	DATA-TAIL/FRAME-TAIL. Last symbol in fixed/multicast stream

Tx and Rx Ports Null Symbols

T[2:0]	Tx/Rx S[31:0]
7	Null. Don't Cares

Node ID Format

RxT[2:0]	RxS[31:28]	RxS[27:0]
4	Node ID	1 1 1 1 1 1 1

Max ID Format

RxT[2:0]	RxS[31:28]	RxS[27:0]
4	Max ID	1 1 1 1 1 1 1

3.0 Client Interface (Continued)

If the particular node is a bridge only to another ring, the $\overline{\text{PIPE}}$ signal should be negated. The HOP fields rotate the same regardless of the state of the $\overline{\text{PIPE}}$ input. The HCNT field (only in QR1001 mode) does NOT rotate position.

During QR0001 mode operation, the QR1001 device maintains the 5 HOP fields. All 5 HOP fields rotate as in the QR0001 device. This allows interoperability of QR0001 and QR1001 (using QR0001 mode). It should be noted that when both devices are in the same ring and QR1001 is not operating in the QR0001 mode, the QR0001 will ignore all QR1001 added features.

The actual rotation of the Source, Target, and HOP fields occurs in the client receive pipeline (see Tables 3-14 through 3-17). The HOP fields rotate at each target Receive pipeline as follows (from Receive Client perspective for the QR1001):

1) Source bits [27:24] shift to bit field [7:4].
2) All other HOP fields (including Target field) move up one HOP field to the next more significant 4-bit position, HOP 4 moves from [7:4] → [11:8].

Given this information the system interface should be able to determine how the Source,Target and HOP fields rotate in a Ring of Rings architecture including 4 HOPs. The HCNT field can be used to identify the number of ring hops remaining in the path to the target node.

When the Client Rx Port receives a data stream, the head contains the path taken by the stream to reach this particular target. Looking at the Head and determining where the source is and how many HOPs the stream encountered requires knowledge of the ring topology. There are options on how to determine the return address. For example, it can be assumed that during the initialization process each node will build, or be provided, a table of addresses of all the NODEs in the system. When a Head is received, it can be compared to the address table to determine the source of the data stream. Other options are available and will be provided in future Application notes. It should be noted that all unused HOP fields can be used as Stream ID.

TABLE 3-14. HOP Fields at Tx Port (OR1001 Mode)

27:24	23:20	19:16	15:12	11:8	7:4	3:0
Srce	Trgt	Hop 1	Hop 2	Hop 3	Hop 4	HCNT

TABLE 3-15. HOP Fields at Rx Port (QR1001 Mode)

27:24	23:20	19:16	15:12	11:8	7:4	3:0
Trgt	Hop 1	Hop 2	Hop 3	Hop 4	Srce	HCNT

TABLE 3-16. HOP Fields at Tx Port (QR0001 Mode)

27:24	23:20	19:16	15:12	11:8	7:4	3:0
Srce	Trgt	Hop 1	Hop 2	Hop 3	Hop 4	Hop 5

TABLE 3-17. HOP Fields at Rx Port (QR0001 Mode)

27:24	23:20	19:16	15:12	11:8	7:4	3:0
Trgt	Hop 1	Hop 2	Hop 3	Hop 4	Hop 5	Srce

3.15 SUMMARY OF CLIENT PORT FIELD FORMATS

A complete listing of symbol formats for the QR1001 is shown in the table titled **Client Port Field Formats (QR1001 mode)** on the previous page.

3.16 READABLE REGISTERS

The client can read the two internal registers, Diagnostics Register and Receive Symbol Register at any time, through inputs RxSEL[3:0] and outputs RxNBL[3:0]. The RxS register can also be read while $\overline{\text{RxSTALL}}$ is asserted. RxSEL[3:0] selects a 4-bit field within the 32-bit internal registers and shows that field on the RxNBL[3:0] outputs.

TABLE 3-18. Diagnostics Register (Read Only)

31:24	23:20	19:12	11:8	7:4	3:0
Reserved	FIFO Flags	Syndrome Word: S[6:0]	Error Status	Max ID	Node ID

Node ID: Address of the Node

Max ID: Largest ID on the ring.

Error Status: Individual bits are set depending on the origin of the error.

Bit 8 is set due to an EDC detection.

Bit 9 is set due to an Abort symbol received at the Up port.

Bit 10 is set due to an error in the packets sequence, i.e, two consecutive heads. (Detected on the upstream port of the ring.)

Bit 11 is set due to an invalid address detected (on the ring).

Syndrome Word: Points to the bit in error detected through EDC. All zeros if no error(s).

FIFO Flags: Contains status flags for the Target, X, Y, and LB FIFOs. Section 5.1 has details on this register.

Reserved: Reserved for future expansion

TABLE 3-19. RxS Register (Read Only)

31:0
The most recent 32 bits the RxS received.

Table 3-20 gives the decode for reading the various register bits.

Reading Registers

Reading the error status register will clear ABORT as well as allow the syndrome words to be overwritten. To guarantee all information related to the current error is accessed, the following sequence is suggested for reading error registers.

1. Set RxSEL = C; read RxNBL syndrome word
2. Set RxSEL = B; read RxNBL syndrome word
3. Set RxSEL = A; read RxNBL error status

3.0 Client Interface (Continued)

TABLE 3-20. Register Access Decode

RxSEL[3:0]	RxNBL[3:0]	Description
0	RxS[3:0]	
1	RxS[7:4]	
2	RxS[11:8]	
3	RxS[15:12]	
4	RxS[19:16]	
5	RxS[23:20]	
6	RxS[27:24]	
7	RxS[31:28]	
8	Diagnostics[3:0]	Node ID
9	Diagnostics[7:4]	Max ID
10	Diagnostics[11:8]	Error Status
11	Diagnostics[15:12]	Syndrome Word
12	Diagnostics [19:16]	Syndrome Word
13	Diagnostics [23:20]	FIFO Flags
14	Diagnostics [27:24]	Reserved
15	Diagnostics [31:28]	Reserved

Reset Condition for Registers

The RxSEL nibbles 0 to 9 are not cleared by the RESET. The values in those nibbles will remain the same as before RESET was asserted. The Node ID and Max ID may change during a RESET if the ring configuration changes. All other nibbles are cleared by the RESET.

3.17 Error Detection

The error detection code (EDC) field, CB[6:0], provides redundant parity checking to verify symbol integrity. The QR1001 implements a modified Hamming code algorithm that provides Double Error Detection (DED).

The QR1001 provides the syndrome word that can be read from the Diagnostics register to show bit(s) detected in error. To reduce latency effects, no error correction is performed.

The syndrome word, S[6:0], consist of the Ex-OR of the incoming check bits that were sent within the packet, CB[6:0], and new generated check bits for the packet (new generated, NGCB[6:0]).

A correct EDC field is transmitted with each symbol emitted from the downstream port. Every symbol, including nulls, that is received at the upstream port is passed through an EDC checking circuit. Any inconsistency causes the $\overline{\text{ABORT}}$ signal to assert, and an abort symbol will be transmitted at the downstream port. EDC fields are propagated through the chip core as required to support the above described functionality. The EDC field is not visible at the client ports.

Table 3-21 shows the matrix of data bits. An "X" indicates the bits that are "Exclusive-ORed" to generate each particular check bit. Check bit 6 is generated by "Exclusive ORing" all data and all check bits. CB[6:0] form the syndrome word.

Given a Single or Double bit error, the code has the following properties:

1) If the syndrome word: **S[6:0] is zero (0),** then there is No Error.
2) If any of the syndrome bits: **S[5:0], is not zero (0), and S6 is zero,** then there is a Double Error. The particular bits in error can not be determined.
3) If any of the syndrome bits: **S[5:0] are not zero (0) and S6 is one,** then there is a Single Error in either the data or the check bits.
4) If a single bit in S[5:0] is one and S6 is one, then the corresponding CB is in error and the data is correct.

   If more than 1 bit in S[5:0] is set to one and S6 is one, then the syndrome bits point to the data bit in error. See columns of Table 3-21. S[5:0] pinpoints to the position number in the mapping diagram at which bit is in error.

   If all remaining bits, S[5:0] are zero and S6 is one, then CB6 is in error.

The syndrome bits associated with an EDC error may be observed by reading the diagnostic register. However, the actual data packet causing the error cannot be identified at the client receive port.

TABLE 3-21. Error Detection Matrix

	35-Bit Data Word																																		
CHECK BITS	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
CB0	X		X		X		X		X	X		X		X		X		X		X		X		X	X		X		X		X	X		X	X
CB1			X	X			X	X		X	X			X	X			X	X			X	X		X	X			X	X		X	X		X
CB2			X	X	X	X				X	X	X	X					X	X	X	X				X	X	X	X				X	X	X	
CB3	X	X								X	X	X	X	X	X	X	X								X	X	X	X	X	X	X				
CB4										X	X	X	X	X	X	X	X	X	X	X	X	X	X	X											
CB5	X	X	X	X	X	X	X	X	X																										
CB6	Exclusive "OR" All Data Bits (1–35) and All Check Bits (0–5)																																		

C6 Aids in DED (Double Error Detection)

4.0 Ring Interface

This section gives information on the protocol implemented to control data flow on the ring. It is useful to know the protocol fundamentals to better understand the flow at the client ports. Details are purposely excluded to simplify this section. Details are available upon request.

4.1 TYPE AND SYMBOL FIELD AT THE RING PORTS

On the ring path (upstream and downstream ports), the type and symbol fields organize data transmissions. Data on the ring flows in bounded streams called packets. Before data flows in the ring, packets are formed by each controller internally. Packets have one head and one or more payload symbols. Inside a controller, packets can be found that may originate from any other node on the system. Each packet has its own head symbol which contains the same information as the client port heads. The type field marks each symbol as a head, payload, tail, or access.

4.2 DATA AND FRAMES

In QuickRing technology, there are two types of payload symbols, data symbols and frame symbols, but their distinction is only of interest to the clients. The QuickRing controller does not discriminate between them, except to preserve their identity.

The Frame symbol can be used to identify a special kind of data of interest to the clients. For example, it can be used to designate the beginning, end, or address of a stream. It may also distinguish between data streams at the client interface. QR1001 maintains this symbol identity but treats it only as another payload symbol.

4.3 SYMBOL FLUX ON RING

At the client (transmit and receive) ports, the length of a payload stream that is uninterrupted by a head is unbounded. On the ring (upstream and downstream ports), payloads are bounded; there is an upper bound that gives the concept of a packet. There are two types of packets: normal and low bandwidth (LB). There is a ring protocol defining the symbol sequence. For normal packets, the maximum number of payload symbols associated with one head is 20 symbols. Variable length packets can have payloads anywhere between 1 and 20. The multicast and fixed length packets are controlled by the client as to how long they are on the ring. **The largest packet ever on the ring is 21 symbols in all.** However, packets may be less than 21 symbols. The LB packet always consists of a Head and one payload on the ring even though it is unbounded on the client ports.

4.4 DATA ON THE RING (HEAD, PAYLOAD, TAIL)

QuickRing technology transports streams of payload symbols from source nodes to target nodes through the ring interconnect. QuickRing technology internally assembles packets from the data that the client writes into the transmit port. In variable length mode, this data is eventually transmitted in packets of 1 to 20 payload symbols. A head symbol precedes the packet and the last payload symbol of a packet is specially marked as the ring tail of that packet (this is different from data-tail and frame-tail on the client). The head holds the source and the destination node IDs, plus other information that uniquely identifies the stream to which the payload symbols belong.

Payload symbols consist of 32 bits of user defined information plus 3 type bits. The type bits on the ring are different from the type bits on the client but this manipulation is invisible to the client and will not be discussed in this document.

4.5 ACCESS SYMBOLS ON THE RING (VOUCHER, TICKET, ABORT, NULL)

Access symbols are used on the ring to control the traffic flow. They are never visible to the client ports and never hop to other rings. Vouchers and tickets are used for the reservation protocol which prevents FIFO overflow. The abort symbol is generated by a node that detects a ring error condition and informs all other nodes in the ring that an error condition existed. The null symbol is used when no traffic is present. It serves the purpose of an idle symbol.

Before a source node can launch a packet onto the ring, permission to transmit must first be granted by the target node. To get permission to transmit, the source node sends a voucher to the target node. To grant permission to transmit, the target node checks for space available in the Target FIFO and sends a ticket back to the source node. This is done only in response to a voucher. When the source node sends a voucher, the target node may (1) absorb the voucher and return a ticket or (2) if no voucher queue space is available, return it. If the source receives the ticket, then it may send one packet to the target that returned the ticket. If the source received its own returned voucher, then it will retransmit the voucher after 100 clocks to try again for a ticket. The number of retries for the voucher is unlimited until the target returns a ticket. Under normal circumstances, the target will return a ticket in response to a voucher, even if it saves accumulated vouchers in a queue (room for 30 vouchers) and issues tickets with significant delays. The return of a voucher to its source should occur only if resources for queuing vouchers in the target node are exhausted.

A voucher and ticket pair will always make a complete loop on the ring. This delay before launching a packet adds to the latency inherent in the QuickRing protocol. The latency then is directly proportional to the number of nodes in the ring. Since the voucher and ticket symbols have the highest priority in bypassing nodes, the latency is minimized and kept almost constant as long as the vouchers are not stuck in a deep queue at the target handler.

4.6 RING ERRORS

The $\overline{\text{ABORT}}$ signal is intended to show the ring is operating in an error free condition. If an error occurs on the ring, it will be logged by the first node to detect it and then all nodes will be notified (abort symbol circulated) that an error was detected. The $\overline{\text{ABORT}}$ function is not intended to be an error correction or recovery mechanism by itself. It may give useful information to an error recovery and correction protocol existing at a system level.

4.0 Ring Interface (Continued)

The ABORT signal will be asserted if an EDC error, illegal node address (greater than Max ID) or illegal sequence of symbols (sequential heads for example) is detected. All ring symbols, including nulls, are checked for EDC. The detection of an error condition launches an abort symbol at the detecting nodes downstream port. In addition, the erred symbol which triggers the ABORT assertion may go to the Bypass FIFO to continue downstream. If the erred symbol is launched before the abort symbol, it will trigger more ABORT assertions if it arrives at an upstream port before the abort symbol. The initial occurrence of an abort is captured in the diagnostic register. The ABORT pin is also asserted, flagging an abort condition to the client. No subsequent abort condition will be logged until the diagnostic register is read or reset occurs.

The ABORT signal is cleared by asserting RESET which means all nodes in the ring must perform the reset and initialization. QR1001 does NOT require reset and initialization procedures to clear the ABORT signal. The ABORT signal and associated error status bits are cleared by reading the error status bits ("read to clear") of the diagnostic register (RxSEL = A). ABORT signal is cleared a minimum of 3 clock cycles after reading the diagnostic register. The syndrome word bits are also reset, and overwritten in the following clock cycle.

When a ring abort condition is flagged, an abort symbol is sent downstream from the node detecting the abort. If the abort was due to an EDC failure, this abort symbol can overwrite any symbol that is currently at the downstream port. All transmitted symbols, including heads, tickets and vouchers, may be overwritten. The possibility exists of corrupting transmitted data, and/or inhibiting transmission by corrupting access symbols or heads. The possibility also exists that the error was a transient event on a harmless ring symbol and no further problems occur. This is when the "read to clear" function becomes useful. If the abort was due to an illegal sequence of symbols or an illegal address, the abort symbol will replace the erred symbol and cause no further corrupted symbols. All these situations may be relieved by using a global reset.

A global reset must always be used in the case of an abort caused by an illegal node address. This is because data queued to the transmit FIFO waits for the receipt of a ticket before being sent out on the ring. If the header includes an invalid target address, a ticket will never be received and data remains in the transmit FIFO. The only way to clear this transmit FIFO and avoid a potential deadlock situation is through a global reset. When an abort occurs, polling the error status of the diagnostic register will indicate if an invalid address has been issued.

5.0 Clock Signals

There are four clock domains in QuickRing, one for each port. The transmit and receive ports are clocked by TxCLK and RxCLK, respectively. The QR1001 core logic is clocked either from RGCLK (CKSRC asserted) or from UpCLK (CKSRC negated). The upstream port is always clocked by UpCLK. Each controller derives the DnCLK from the clock that drives the downstream port.

TABLE 5-1. Clock Signal

QR0001 Interface	CKSRC	Clock Source
Downstream Port, CLKOUT	H	RGCLK
	L	UpCLK

The client ports are asynchronous from each other. The Upstream and Downstream ports are synchronous with each other when CKSRC is negated, and frequency locked, not phase locked, when CKSRC is asserted.

For all clocks, the minimum period is 20 ns, for a maximum frequency of 50 MHz.

5.1 FIFO Flags

QR1001 provides the capability to monitor status of the transmit and receive port FIFOs. A description of the corresponding register bits is shown in Table 5-2 and refer to Table 3-20 also.

TABLE 5-2. Diagnostic Register FIFO Flag Bits (RxSEL = D)

23	22	21	20
Target FIFO Available = 1	Tx X FIFO Empty = 1	Tx Y FIFO Empty = 1	LB FIFO Empty = 1

The target FIFO available flag indicates that at least one ticket is available at the time this register is polled. Each of the Tx FIFO empty flags is high (= 1) when there is no data in the FIFO at the time the register is polled. The status of each Tx FIFO (X, Y, and Low bandwidth) may be monitored. These flags do NOT indicate the presence of data that is currently being transmitted at the client ports but has not yet reached the Tx FIFOs.

6.0 ABORT Signal

The ABORT signal is an output of the QR1001. Any one of the following events can cause the ABORT signal to be asserted:

- EDC error (checked on all symbols, including ring access symbols; vouchers, tickets, nulls, aborts)
- Illegal symbol sequence detected on the QuickRing (for example, sequential heads)
- Node ID detected greater than maximum node ID in the ring
- Received an Abort symbol at the upstream port

The ABORT signal can be cleared in 2 ways. One is asserting RESET which leads to a global reset and initialization of the ring. The other is to read the Diagnostic register that contains the error status (RxSEL = A). This last method only clears the error status and syndrome word registers and does not lead to or require a global reset. If there is a packet blocking one of the transmit FIFOs because a voucher, ticket or head was corrupted, reading the diagnostic register will not clear that blockage. The blockage can only be cleared by a global reset.

7.0 Bridges

QuickRing technology has the ability to link rings together and pass traffic from one ring to another. This is referred to as bridging rings. The header has HOP fields which can be used to address nodes in rings other than the source node's local ring.

In a single ring, the maximum number of nodes is sixteen. Within a ring, the $\overline{\text{PIPE}}$ signal will most probably be asserted for ease of interfacing to the QR1001. In a multiple QuickRing topology; the individual rings may be connected together through bridges. The system may implement a very basic bridge or an intelligent bridge.

For a basic bridge implementation, two QR1001 controllers can be directly connected through their client ports, providing a bridge between two rings. The $\overline{\text{PIPE}}$ signal should be negated in bridge operation so that the Rx and Tx port timing will be compatible. Also, the TxOK signal is connected to the $\overline{\text{RxSTALL}}$ signal. QR0001 timing characteristics will support bridging. However, an external flip-flop may be used to achieve additional setup and hold time margins. For more sophisticated bridges, external logic can implement an added layer of protocol.

The HOP fields are used, with respect to bridges, when implementing multi-ring topologies. The HOP fields may be used as desired in a single ring topology (i.e: to distinguish various data streams). Refer to Client Interface Field Definitions sections for more information.

8.0 Little/Big Endian Issues

This QR1001 datasheet uses strict little endian labeling conventions to indicate bit positions. The device itself is neither big endian nor little endian. No assumption is made in QR1001 about the relative significance of bytes within any payload symbol. Table 8-1 is for reference only:

Big Endian: MSB(yte) of the information (data/address) is stored at the least significant address location.

Little Endian: LSB(yte) of the information (data/address) is stored at the least significant address location.

TABLE 8-1. Big/Little Endian

Big Endian

MSb(it)			LSb(it)
bit-0			bit 31
[0:7]	[8:15]	[16:23]	[24:31]
MSB(yte)			LSB(yte)
byte 0	byte 1	byte 2	byte 3
address: (n)	address: (n + 1)	address: (n + 2)	address: (n + 3)

Little Endian

MSb(it)			LSb(it)
bit 31			bit 0
[31:24]	[23:16]	[15:8]	[7:0]
MSB(yte)			LSB(yte)
byte 3	byte 2	byte 1	byte 0
address: (n + 3)	address: (n + 2)	address: (n + 1)	address: (n)

9.0 Reset and Initialization

9.1 RESET

The controller must be reset after power up. Reset can be released from each node in any order but only after all nodes in a local ring have asserted reset and are simultaneously in the reset state. External logic should assert $\overline{\text{RESET}}$ to all nodes on the ring during system power up and when a reset is required to correct an abort condition. The release of $\overline{\text{RESET}}$ to the single node in the ring with NODE0 asserted will begin the initialization process. The ring initialization proceeds to completion only after $\overline{\text{RESET}}$ has been released to all nodes.

The first 2 non-null symbols that appear at the receive port are the node ID number and the largest ID on the ring. The type associated with this information will indicate a frame symbol. These values will be present for one clock cycle. The information can later be retrieved by reading the Diagnostics Register through the RxNBL and RxSEL at any time.

9.2 NODE0 SELECTION AND INITIALIZATION

Only one QuickRing controller in a ring can be designated as Node 0 (NODE0 asserted). For all other controllers on the ring, NODE0 must be negated. Once the ring has completed the initialization process, the ring is ready for normal operation. During normal operation, there are no differences between node 0 and all other nodes.

9.3 NODE ID ASSIGNMENT

After Reset is released to each QuickRing controller, node 0 begins the node ID assignment. Each node receives the node ID of its upstream neighbor on RxS[31:28], assigns its own address as node ID + 1, and passes its new node ID to its downstream neighbor. When this sequence returns to node 0, the last node ID is declared the max node ID and circulated to all nodes in the ring.

After initialization, the first two non null symbols that appear at the receive port indicate to the client interface the node ID number of the controller, and the largest ID number on the ring. This information can be used to configure the client interface.

The node ID (Node ID) and largest node ID in the system (Max ID) can be read later from the internal diagnostics register in the controller.

9.4 SEQUENCE FOR NODE0

As soon as $\overline{\text{RESET}}$ is released by node 0, it begins to send a probe symbol downstream. When this symbol returns, all nodes in the ring have come out of reset and are ready for initialization.

1) The initialization begins by the controller with the NODE0 signal asserted assigning itself to be node 0.
2) Node 0 then begins to transmit a stream of identical symbols at the downstream port whose high order four bits are [0,0,0,0]. No lower order bits are used in the initialization procedure.
3) The first downstream node to receive this symbol increments the value to [0,0,0,1] and becomes node 1.
4) The node forwards the symbol containing a value of 1.

9.0 Reset and Initialization (Continued)

5) Each node in turn increments the value and takes on its own unique node ID.
6) When the node 0 receives the symbol at its upstream port, the value of the symbol is the largest ID number in the ring.
7) Node 0 stores this value and forwards it around the ring.
8) All nodes store the Max ID and forward this symbol unmodified.
9) Once this symbol returns to node 0, node 0 begins to transmit an initialization complete symbol.
10) As soon as each node, including node 0, detects the initialization complete symbol, its initiallzation sequence is completed, and it may begin to transmit vouchers, tickets and packets.
11) During this sequence, the controller forwards the node ID and the Max ID to the receive port.

9.5 SEQUENCE FOR ALL OTHER NODES ON THE RING

1) Release of $\overline{\text{RESET}}$ from a node causes that node to begin monitoring its upstream port. The node transmits all symbols it receives at the downstream port.
2) When the upstream port receives the first non-null type, the accompanying symbol field is interpreted as the node ID of the upstream neighbor.
3) The node increments the symbol field value, stores the new value internally, and then forwards the incremental value at its downstream port. The stored value becomes the local node ID. This node ID number is propagated to the node's receive port so the client interface can learn its ID number.
4) Some time later, the Max ID symbol arrives at the upstream port. The value of this symbol is stored internally and represents the numerically greatest node ID residing on the ring. The node forwards this symbol to its downstream port. This value is also forwarded to the receive port, so the client can learn the number of nodes on the ring.
5) The initialization complete symbol will arrive at the upstream port. As soon as this symbol is forwarded to the downstream port, the node may commence normal ring operation. If at any point there is a node ID detected whose value is greater than the greatest ID residing on the ring, an abort symbol is sent and it is considered as a failure. For the client interface, the first 2 non null symbols that appear at the receive port are the node ID number and the largest ID on the ring.

In a 1 node ring, the complete initialization process will take about 42 clock cycles from the release of $\overline{\text{RESET}}$ to the delivery of Max ID at the client receive port. TxOK will assert at about 23 clock cycles. Each additional node in the ring will add about 8 clock cycles to the total initialization time. For example, all nodes in a 3 node ring will have the Max ID delivered in about 58 clock cycles.

10.0 QR1001 Operation Flow

10.1 RING TRAFFIC FLOW PRIORITIES FOR DOWNSTREAM PORT TRANSMISSIONS

Alter the initialization process is complete, all nodes on the ring are ready to transfer packets. Data streams can be queued and de-queued to and from the QR1001 through the Client Interface on all nodes. As traffic builds on the ring, the QR1001 prioritizes the information flow through the node onto the ring. Following is the list, in descending order, of the paths inside the QR1001 that process information to be sent onto the ring through the downstream (DnSS) port. Refer to the QR1001 block diagram.

1) The highest priority is given to tickets and vouchers passing through the node. This QR1001 is simply forwarding the access symbols on the ring that are destined for other nodes. This serves to minimize the latency involved in the FIFO reservation protocol.
2) LB Target Handler: Sends out low bandwidth tickets generated by this node. (Includes voucher rejects when exceeding the LB voucher storage capacity.)
3) Target Handler: Sends out normal tickets generated by this node. (Includes voucher rejects when exceeding the normal bandwidth voucher storage capacity.)
4) Local sourced reservation vouchers launched by this node.

   LB FIFO: Generates a voucher when the LB symbol gets to the head of the FIFO.

   X or Y FIFO: For variable sized packets, it generates a voucher as soon as the first payload symbol is loaded into the FIFO. Also, generates another voucher as soon as the 21st symbol (associated with the same head) is loaded into the FIFO. Further, continues to generate a voucher for each packet (maximum bundle of 20 symbols). For the fixed length packets, it generates the voucher when the tail symbol arrives in the FIFO (refer to Section 3.3.4).
5) Ring FIFO: This QR0001 is forwarding data destined to other nodes.
6) Sends out locally generated packets from this node's X, Y, or LB FIFO. At the beginning of each packet, the traffic flow priorities are checked. The source (X, Y, or LB) FIFO can transmit when the Ring FIFO has at least 28 empty positions.

10.2 INSIDE THE SOURCE NODE (DEVICE TRASMITTING DATA)

At the source node, as soon as QR1001 latches a head and a payload symbol, in the X or Y FIFO, it sends a voucher to the target node. The source node waits until the target node sends a ticket back before transmitting a packet. During this time the client interface can write payloads into the controller.

When QR1001 detects that a single packet will not be enough to transmit all data in FIFO X or Y, another voucher is sent to the target node. Additional vouchers are sent, as soon as the controller deems it necessary, to complete the transmission. This action is intended to hide the latency between the transmission of vouchers and the receipt of tickets. A maximum of 3 vouchers can be outstanding from both the X and Y FIFOs at the source node. Vouchers have higher priority than payloads, and they can be launched interleaved in current outgoing packets.

At the source node, only when a ticket is received is a packet sent. A packet is formed by 1 head and anywhere from 1 to 20 payload symbols. The largest packet is 21 symbols, 1 head symbol and 20 payload symbols. The last payload symbol of a packet is always identified as a tail. This is encoded in the ring type field.

10.0 QR1001 Operation Flow (Continued)

When operating in variable mode the OuickRing controller does not wait for any specific number of data symbols to send a packet. If the client interface is slow in writing data at the transmit port, QuickRing could send packets with less than 21 symbols. The same will occur in transmissions where the total number of payloads is not a multiple of 20. The QuickRing controller is designed to transmit the data in the transmit pipeline as soon as possible.

Fixed or multicast operation guarantees that a packet will not be transmitted on the ring until one of the following is queued at the client port: a tail symbol, the 21st payload symbol, or a new head. This allows the user to define the size of a packet being transmitted on the ring. The voucher for the packet will not be generated until the final symbol of the fixed length packet is queued in the transmit FIFO. This may add some additional latency to the first fixed length packet launched, but subsequent packets will only incur the latency the client adds in getting the tail symbols into the transmit port.

10.3 SUMMARY OF SOURCE NODE ACTIONS

QR0001 and QR1001 directed mode send a voucher from the source node to the target node as soon as it has at least one payload to transmit. (In the X or Y FIFOs.)

QR0001 and QR1001 variable length mode send an additional voucher as soon as it identifies that one packet is not going to be enough to transmit all the data in the transmit pipeline.

QR1001 fixed length mode waits for the tail to arrive in the transmit FIFO before launching the voucher.

No packet, except multicast packet, is sent until a ticket is received. This includes low bandwidth packets.

QR0001 and QR1001 directed mode do not wait for data; therefore, packets could vary in size.

The largest and most efficient packet is one formed by 1 head symbol and 20 payload symbols, 21 symbols in all.

Low Bandwidth packets are 1 head and 1 payload symbol on the ring (unlimited on the client).

The client interface must stop writing data at the transmit port within 20 non-null symbols after TxOK negates. The count is reset if TxOK asserts again.

10.4 INSIDE THE TARGET NODE

At the target node, if the receiving controller has space for one 20 symbol packet in the Target FIFO, it will send a ticket immediately to the source node in response to a voucher. A QR1001 Target FIFO has space for 3 normal packets and 6 low bandwidth packets; therefore, the controller can have only 3 outstanding normal tickets and 6 outstanding low bandwidth tickets. If all tickets have been given, the receiving QR1001 will queue incoming vouchers in one of two special buffers, called Target Handler and LB Target Handler. The Target Handler can store 30 vouchers and the LB Target Handler can store 10 vouchers for low bandwidth transmission.

At the target node, a ticket is made available as soon as a packet has exited the Target FIFO to the Rx Resynchronizer.

If the target node cannot return a ticket or store the voucher to be handled later, it will return the rejected voucher to the source node. (The Source will sink the voucher and the node will then resend the voucher after 100 clock cycles.)

10.5 SUMMARY OF TARGET NODE ACTIONS

The target FIFO can store 3 normal packets and 6 low bandwidth packets. Therefore, only 3 normal tickets and 6 LB tickets can be outstanding at one time.

QR1001 can store 30 normal vouchers and 10 LB vouchers before returning a voucher-reject to the source node.

The Head Stripper will remove redundant heads before entering the Target FIFO, unless they are fixed length or multicast.

Data may arrive at the Rx Port on every tick of the clock unless the client stops the flow through the $\overline{\text{RxSTALL}}$ input.

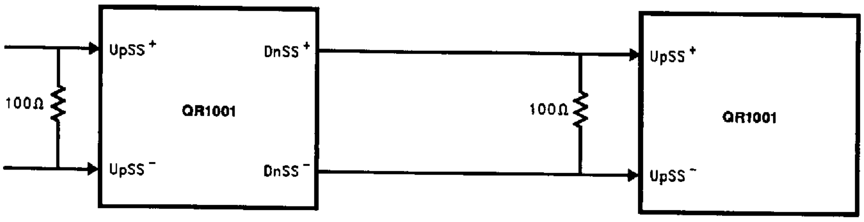
RxET can be used to monitor the kind of data entering the receive pipeline up to 20 symbols before it appears at the receive port. When the Rx pipeline is free flowing in the unblocked pipe ($\overline{\text{RxSTALL}}$ is negated), RxET will indicate the Type:

1. three clock cycles earlier then the symbol (RxS) in the pipelined timing and
2. two clock cycles earlier then the symbol (RxS) in the non-pipelined timing.

11.0 Board Considerations

11.1 UPSTREAM PORT SIGNAL TERMINATION

The ring interface upstream port signals, UpSS[5:0] and UpCLK, need external termination resistors. The termination should be a 100Ω resistor between the differential signal pair. The resistor should be placed as close to the upstream port pins as possible. Minimum parasitic inductance and capacitance is desirable. Surface mount chip resistors with ±1% tolerance are recommended. See *Figure 11-1.*



TL/F/12048-21

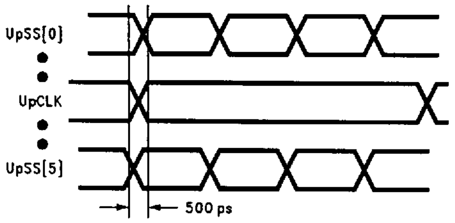
FIGURE 11-1. Termination between the Differential Signal Pair

11.0 Board Considerations (Continued)

11.2 QuickRing PHYSICAL LAYER DETAILS

If the QuickRing signals on one of the 6 lines were switching between 1 and 0 on every bit, they would be switching at 175 MHz. The bit rate is twice the maximum possible frequency of the signals. QuickRing's 175 MHz data signals dictate special care for the physical layer design and layout. The use of LVDS (Low Voltage Differential Signals) enables the very high frequency operation. The LVDS also eases design because the differential signals are forgiving to certain impedance discontinuity in the signal path. If the discontinuities are at the same electrical distance and have the same magnitude, they will incur minimum distortion to the differential signal. Each single ended signal may appear to have reflections, but if the differential pair has the same minor reflections, then the differential signal will be minimally affected. The skew between the pairs and inside the pairs are critical design criteria. The basic guidelines for transporting the ring signals should be observed.

The skew between pairs is critical. The 350 MBaud signals only provide a bit width of about 2.86 ns. The QuickRing UpPort needs 2.36 ns of this bit width (including transitions) to successfully sample the value for each subsymbol. This allows for a total skew budget of 0.5 ns. This budget allows a window into which all signals must fit under all circumstances. The interconnect between DnPort and UpPort should be limited to 500 ps. Care should be taken to provide skew margin. The 500 ps includes the skew from driver outputs. The remainder can be divided between PCB traces, connectors, headers, cables and all other media used in the signal path. As ring clock frequency decreases, the skew specification can be loosened, but it is recommended to keep skew as minimal as possible.



TL/F/12048-22

FIGURE 11-2. Upstream Port Signals Must Occur in the 500 ps Window at the Receiver Input

The skew within a pair needs to be controlled because of the EMI considerations. The simultaneous and opposite transitions on paths within a pair create equal and opposing electromagnetic fields. These fields, the source of EMI, serve to cancel each other thereby reducing EMI. The skew within a pair should be controlled by using the same length traces which have the same impedance discontinuities. They should also be close to each other so that the opposing single ended electromagnetic fields cancel one another out.

The length of node interconnects is not critical to the operation of QR until it degrades signal integrity. Nodes in a ring can have different length interconnects. The maximum length of the interconnect depends on two qualities of the interconnect: transition time degradation and amplitude attenuation. Any extension in transition time, due to the high frequency filter affect of the interconnect, subtracts from the skew budget. The trade-offs between the skew and transition time degradation must be balanced to allow for the correct amount of sample time for the UpPort.

The signal attenuation affects the differential signal amplitude at the receiver input. The receiver requires a differential voltage of at least 100 mV to guarantee correct operation. As long as the differential voltage is guaranteed to be at least 100 mV and all the skew budget specifications are met, the receiver will operate correctly.

12.0 Power and Decoupling Issues

12.1 POWER ISSUES

The QR1001 device internally has five distinct power regions.

1) Logic power pin names
   (V_{CC} 4, 5, 12; GND 18, 19, 28, 29);
2) Client receive port output power pin names
   (V_{CC} 6, 7, 8, 9, 10, 11; GND 20, 21, 22, 23, 24, 25, 26, 27);
3) LVDS power pin names
   (V_{CC} 1, 3; GND 1, 2, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 17);
4) PLL and delay element power pins
   (V_{CC} 2; GND 9);.
5) Bandgap power: Quiet V_{CC} and Quiet GND
   (QV_{CC}: pin 10; QGND: pins 121, 160)

It is recommended that the PC Board have separate Gnd and V_{CC} planes. Also, Power Region 2 should be carefully decoupled from the power plane. The decoupling aids in damping the Receive Port current spikes to the remaining plane.

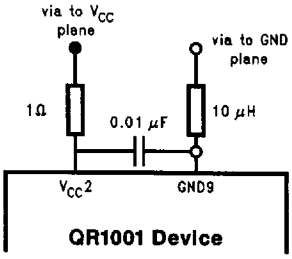
12.2 DECOUPLING ISSUES

The high frequency, mixed signal operation of QuickRing technology requires good power supply noise suppression. Experience with the QR0001 showed extra attention must be given to the power regions 2 and 4 for maximum frequency operation. The QR1001 design has taken this susceptibility into account and noise suppression features have been incorporated. However, their effectiveness has not been verified yet. As a precaution, the lessons learned will be suggested for the QR1001.

It is currently recommended that 0.001 µF capacitors be placed locally on all V_{CC} pins of the device to provide an even filtering. Bulk capacitors, 4.7 µF to 10 µF, should also be placed very close to the device to provide additional low frequency noise filtering. Refer to *Figure 12-2*. In addition, an extra 0.01 µF should be placed in parallel to get high and low frequency filtering for each V_{CC} pin shown in *Figure 12-2*. However, it is recommended each capacitor should have a via directly to the V_{CC} and Ground planes.

For power region 4 (PLL and delay element power pin), a decoupling capacitor should be placed as close to the pins as possible (between $V_{CC}2$ and GND9). A 10 µH inductor for the ground lead can be used in series to block high frequency noise from using this path. A trace from the pin directly to the capacitor is recommended and separate vias to ground plane. Refer to *Figure 12-1*. Another proven implementation has used a 1Ω resistor in series to the V_{CC} pin, $V_{CC}2$, and an additional capacitor, 4.7 µF, in parallel to the capacitor but on the power plane side of the resistors.

12.0 Power and Decoupling Issues (Continued)



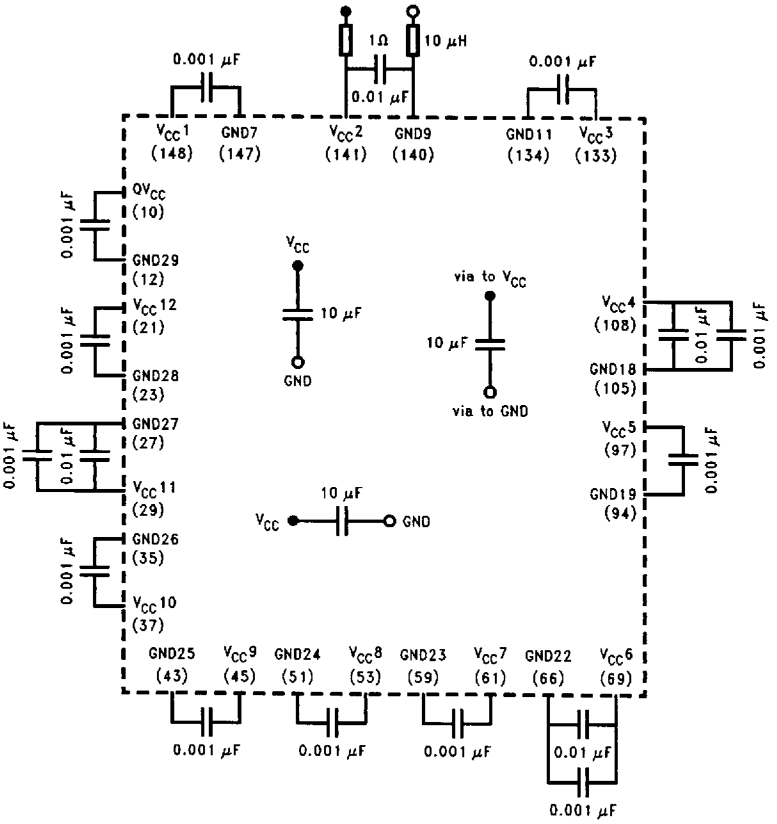
TL/F/12048–23

FIGURE 12-1. PLL Decoupling

Following is a list of PCB recommendations:

1. Use one ground plane and at least one V_{CC} plane.
2. Use a range of decoupling capacitors values. These decoupling caps should be high Q factor chip capacitors (chip caps have little parasitic inductance). The different QuickRing port frequencies require decoupling over a range of frequencies. The decoupling caps should not share vias to the ground plane, as this would defeat the noise suppression across the desired frequency range.
3. Decoupling caps should be placed as close to the device power pins as possible. Extreme care should be taken placing the power region 4 decouple caps close to the power pins. This decoupling cap is recommended as 0.1 μF. The caps at the ends of RxPort (region 2) are recommended to be 0.001 μF and 0.01 μF. These caps should be used as noise barriers between this region and the high frequency ring port region.

   Note: Further testing is being done to find the most efficient size and number of capacitors to use for maximum performance.
4. Care should be taken for ensuring RGCLK, TxCLK, and RxCLK have clean transitions and no reflections or ringing. Transmission line design techniques must be used. As a rule of thumb, if the signal path electrical length (propagation delay time) is greater than 0.125 times the clock transition time, the line should be terminated. For example, if the clock edge is a minimum of 2 ns, the trace electrical length should be less than 250 ps or termination should be used. Assuming a typical 140 ps per inch propagation delay, the clock signal traces should be no more than 2 inches long without termination.



TL/F/12048–24

FIGURE 12-1. QR1001 Power Region Isolation

13.0 DC Electrical Characteristics

Parametrics Disclaimer

The current AC and DC specifications contained in this document are target design specifications. Currently, this information does not represent all actual guaranteed tested timing parameters. Guaranteed specifications will be provided after full device characterization. For more specific information regarding DC and AC parameters, contact National Semiconductor.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

DC Supply Voltage (V_{CC})	−0.5 to +7.0V
DC Input voltage (V_{IN})	−0.5V to V_{CC} +0.5V
Storage Temperature Range (T_{STG})	−55°C to +150°C
Power Dissipation (P_D)	2.2W
ESD Rating	2000V

Recommended Operating Conditions

Supply Voltage, V_{CC}	4.75V to 5.25V
Operating Free Air Temperature	0°C to 70°C

DC TTL Specifications, Client Ports

T_A = 0°C to 70°C, V_{CC} = 5V ±5%, unless otherwise specified

Symbol	Parameter	Conditions	Min	Max	Units
V_{IH}	Minimum High Level Input Voltage	(Note A)	2.0		V
V_{IL}	Maximum Low Level Input voltage	(Note A)		0.8	V
V_{OH}	Minimum High Level Output Voltage	I_{OH} = −400 µA	2.4		V
V_{OL}	Maximum Low Level Output Voltage	I_{OL} = 8 mA		0.4	V
I_I	Input Leakage Current	V_{IN} = V_{CC} (Note A)	−1.0	1.0	µA
I_{IN}	Input High Current	V_{IN} = 2.0V (Note A)		1.0	µA
I_{IL}	Input Low Current	V_{IN} = 0.8 (Note A)	−1.0		µA
I_{DD}	Dynamic Supply Current	V_{CC} = 4.5V, 5V, 5.5V V_{IH} = 2.4V, V_{IL} = 0.4V (Note A)		450	mA
I_{CC}	Static Operating Supply Current	RESET, RxSTALL, RxOE = 3.5V Other CLIENT INPUTS = 0.4V V_{CC} = 4.5V, 5V, 5.5V (Note A)		200	mA
I_{OZ}	Maximum TRI-STATE Output Leakage Current	$\overline{\text{RxOE}}$ = 2V (Note A)	−10	10	µA

Note A: Limit guaranteed by test program.

Note B: Limit based on simulation results.

DC Electrical Characteristics, Ring Ports

DC DIFFERENTIAL GENERATOR SPECIFICATIONS, T_A = 0°C to 70°C, V_{CC} = 5V ±5%, unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OH}	Output Voltage High, V_{OA} and V_{OB}	R_{Load} = 100Ω (Note B)	V_{OL} + 0.3	1.4	1.5	V
V_{OL}	Output Voltage Low, V_{OA} and V_{OB}	R_{Load} = 100Ω (Note B)	0.9	1.0	V_{OH} − 0.3	V
V_{OD}	Differential Output Voltage	R_{Load} = 100Ω (Note B)	300	400	500	mV
ΔV_{OD}	Differential Voltage Change between Complimentary Output States	R_{Load} = 100Ω (Note B)		0	50	mV
ΔV_{OS}	Output Offset Voltage Change between Complimentary Output States	R_{Load} = 100Ω (Note B)		0	50	mV

DC Receiver Specifications T_A = 0°C to 70°C, V_{CC} = 5V ±5%, unless otherwise specified

Symbol	Parameter	Conditons	Min	Typ	Max	Units
V_I	Input Voltage, $V_{IA'}$ and $V_{IB'}$	V_{gpd} = ±900 mV (Note B)	0		3	V
V_{TH}	Differential High Input Threshold	(Note 1, B)			+100	mV
V_{TL}	Differential Low Input Threshold	(Note B)	−100			mV

Note 1: V_{gpd} = ground potential difference voltage between the generator and receiver.

AC Receiver Specifications T_A = 0°C to 70°C, V_{CC} = 5V ±5%, unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{pSKEW}	Receiver Propagation Delay Skew	Any Two Channels on IC (Notes 2, 3, B)	0		250	ps
t_{pwd}	Pulse Width Distortion (t_{PLH}−t_{PHL})	One Channel at Receiver Output (Notes 2, 3, B)	−250		+250	ps
t_{SKEWIN}	Differential Channel to Channel Skew That Can Be Tolerated at Receiver Inputs	Measured at 50% of Transition (Notes 2, 3, B)			500	ps

Note 2: These specifications are not tested but verified by design.

Note 3: A 300 mV differential signal is used to stimulate the receiver input circuitry.

AC Differential Generator Specifications

T_A = 0°C to 70°C, V_{CC} = 5V ±5%, unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_r	V_{OA} and V_{OB} Rise Time. 20% to 80%	Z_{Load} = 100Ω (Note B)	200	400	600	ps
t_f	V_{OA} and V_{OB} Fall Time. 80% to 20%		200	400	600	ps
t_{SKEW}	Generator Propagation Delay	Any Two Channels on IC (Note B)			200	ps
t_{pwd}	Pulse Width Distortion (t_{PLH}−t_{PHL})	One Channel, Difference between Differential Prop Delays (Note B)	−200		+200	ps

14.0 AC Timing Parameters

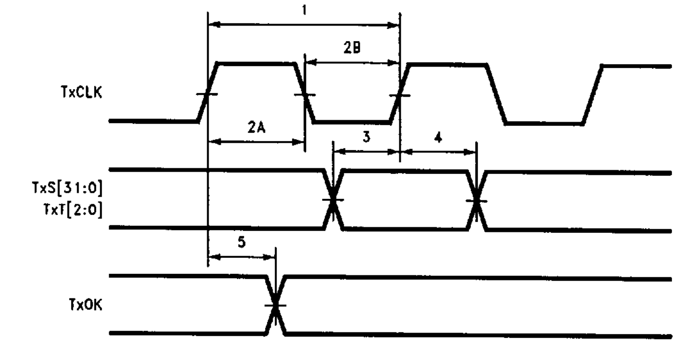
Parametrics Disclaimer

The current AC and DC specifications contained in this document are target design specifications, limited sampled empirical data, and some characterization data. Currently, this information does not represent all actual guaranteed tested timing parameters. Guaranteed specifications will be provided after full device characterization. For more specific information regarding DC and AC parameters, contact National Semiconductor.

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

AC TTL PARAMETERS

Tx Port Timing



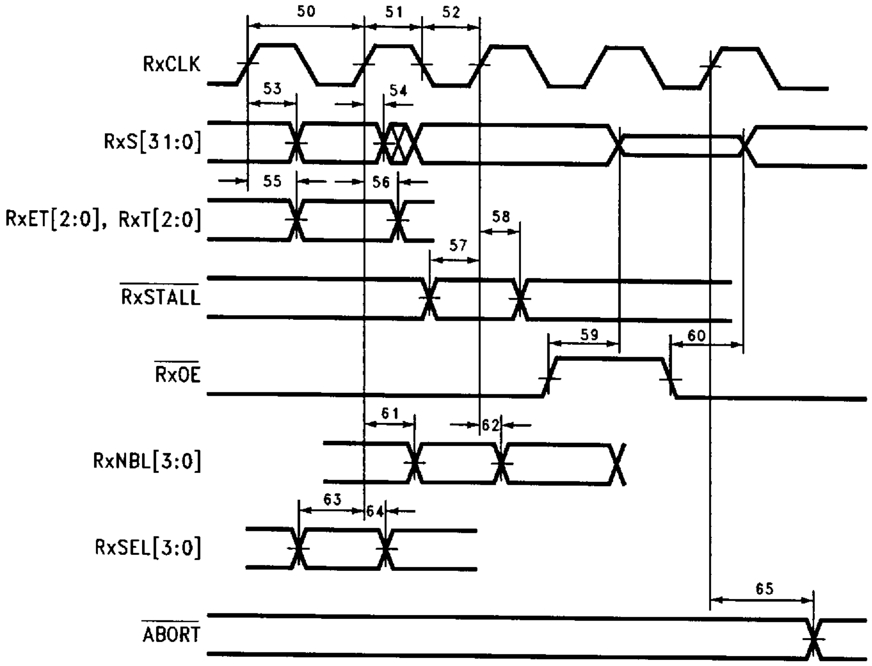
TL/F/12048-25

#	Symbol	Description	Min	Typ	Max	Units
1	t_{TCKP} (Note 1)	Transmit Clock Period (1/f = T: @50 MHz, T = 20 ns)	20			ns
2A	t_{TCKPWH}	Transmit Clock Pulse Width High (f_{max} = 50 MHz)	9			ns
2B	t_{TCKPWL}	Transmit Clock Pulse Width Low (f_{max} = 50 MHz)	9			ns
3	t_{TDS}	Symbol and Type Set Up to clock High	5			ns
4	t_{TDH}	Symbol and Type Hold Time	2			ns
5	t_{TOK}	Clock High to TxOK Valid		8.7	10	ns

Note 1: This parameter is dependent on clock frequency.

14.0 AC Timing Parameters (Continued)

Rx Port Timing



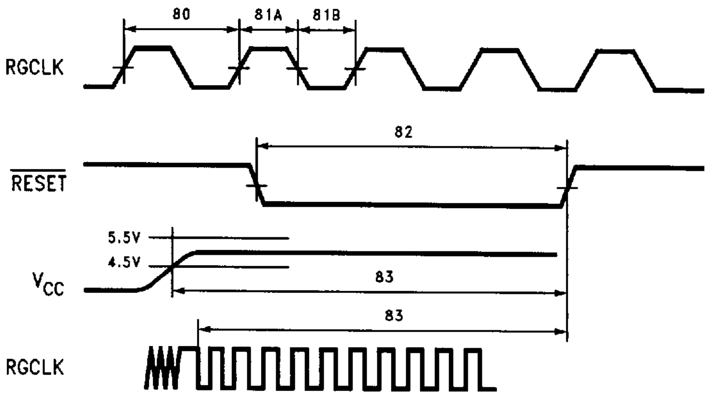
TL/F/12048-26

#	Symbol	Description	Min	Typ	Max	Units
50	t_{RCKP} (Note 1)	Receive Clock Period	20			ns
51	t_{RCKPWH}	Receive Clock Pulse Width High (f_{max} = 50 MHz)	9			ns
52	t_{RCKPWL}	Receive Clock Pulse Width Low (f_{max} = 50 MHz)	9			ns
53	t_{RSACC}	Clock High to Symbol Access Time			12	ns
54	t_{RSVAL}	Symbol Valid after Clock High	3	9.4		ns
55	t_{RTACC}	Clock High to Type Access Time			14	ns
56	t_{RTVAL}	Type Valid after Clock High	3	8.8		ns
57	t_{RSTLS}	$\overline{\text{RxSTALL}}$ Set Up to Clock High	8			ns
58	t_{RSTLH}	$\overline{\text{RxSTALL}}$ Hold from Clock High	0			ns
59	t_{RSHZ}	$\overline{\text{RxOE}}$ Negated to Symbol Tri-Stated		4.5	8	ns
60	t_{RSLZ}	$\overline{\text{RxOE}}$ Asserted to Symbol Low Z		5	9	ns
61	t_{RNBACC}	Clock High to NIBBLE Access Time		8.6	15	ns
62	t_{RNBVAL}	NIBBLE Valid after Clock High	4	3		ns
63	t_{RxSELS}	RxSEL Set Up to Clock High	9			ns
64	t_{RxSELH}	RxSEL Hold from Clock High	0			ns
65	t_{RABT}	Clock High to ABORT Valid		12.5	15	ns

Note 1: This parameter is dependent on clock frequency.

14.0 AC Timing Parameters (Continued)

Reset and Other Miscellaneous Timing 1



TL/F/12048–27

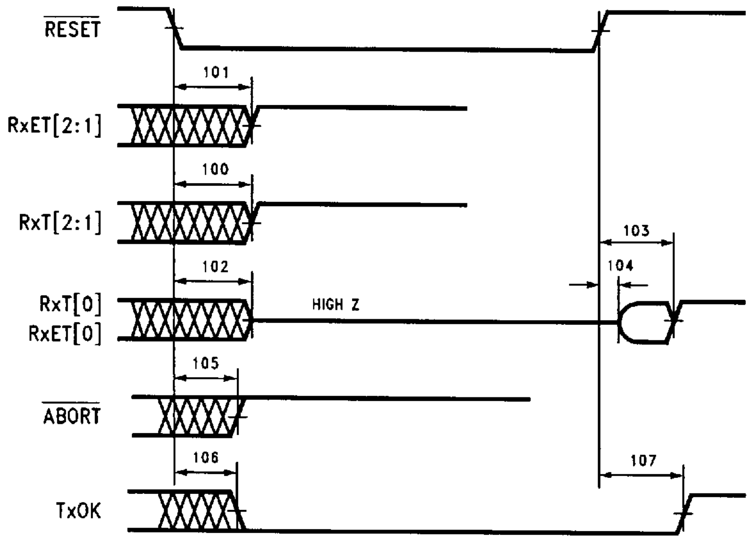
#	Symbol	Description	Min	Typ	Max	Units
80	t_{RGCKP} (Note 1)	RGCLK Clock Period	20		40	ns
81A	$t_{RGCKPWH}$	RGCLK Clock Pulse Width High (f_{max} = 50 MHz)	8			ns
81B	$t_{RGCKPWL}$	RGCLK Clock Pulse Width Low (f_{max} = 50 MHz)	8			ns
82	t_{RSPW} (Note 1)	RESET Pulse Width @50 MHz @33 MHz	 230* 320*	 100 150		 ns ns
83	t_{PLLS}	Phase Lock Loop Set	3**			ms
84	t_{RGCLKS}	RGCLK Set Up Time to V_{CC} = 4.0V	0***			ns

*	At 50 MHz, i.e., $t_{RCKP} = t_{RGCKP} = t_{TCKP} = 20$ ns Otherwise, $t_{RSPW} = t_{RCKP} + 7(t_{RGCKP}) + t_{TCKP} + 50$ ns
**	At one node, otherwise, t_{PLLS} = 1.5 (note number + 1) ms
***	Only applies to clock source node

Note 1: This parameter is dependent on clock frequency.

14.0 AC Timing Parameters (Continued)

Reset and Other Miscellaneous Timing 2



TL/F/12048-28

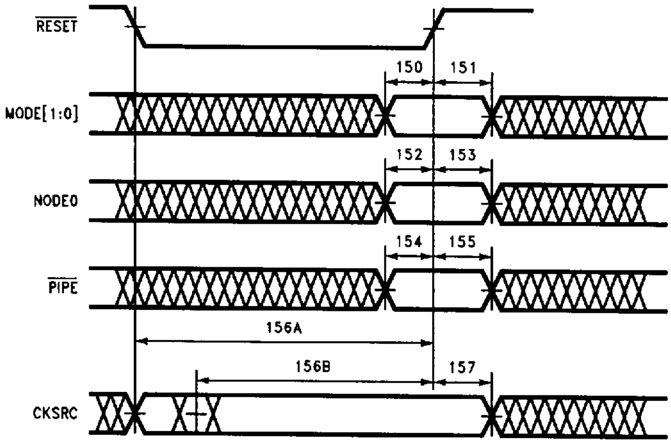
#	Symbol	Description	Min	Typ	Max	Units
100	t_{RSET}	$\overline{RESET}$ Asserted to RxET[2:1] High		25	55*	ns
101	t_{RST}	$\overline{RESET}$ Asserted to RxT[2:1] High		22	55*	ns
102	t_{RST0HZ}	$\overline{RESET}$ Asserted to RxT[0]/RxET[0] High Z		22	55*	ns
103	t_{RST0H}	$\overline{RESET}$ Negated to RxT[0]/RxET[0] High	10	25	55*	ns
104	t_{RST0LZ}	$\overline{RESET}$ Negated to RxT[0]/RxET[0] Low Z	10	22	55*	ns
105	t_{RSABT}	$\overline{RESET}$ Asserted to $\overline{ABORT}$ Valid		10.2	35	ns
106	$t_{RSTXOKN}$	$\overline{RESET}$ Asserted to TxOK Low		14.6	35	ns
107	t_{RSTXOK} (Note 1)	$\overline{RESET}$ Negated to TxOK High @50 MHz @33 MHz (Note A)		150 220	191** 282**	ns ns

*	At $t_{RCKP} = 20$ ns. Otherwise $t_{RSET} = t_{RST} = 2t_{RCKP} + 15$ ns
**	At 50 MHz, i.e., $t_{RCKP} = t_{RGCKP} = t_{TCKP} = 20$ ns Otherwise, $t_{RSTXOKmax} = t_{RCKP} + 7(t_{RGCKP}) + t_{TCKP} + t_{TOK}$

Note 1: This parameter is dependent on clock frequency.

14.0 AC Timing Parameters (Continued)

Reset and Other Miscellaneous Timing 3

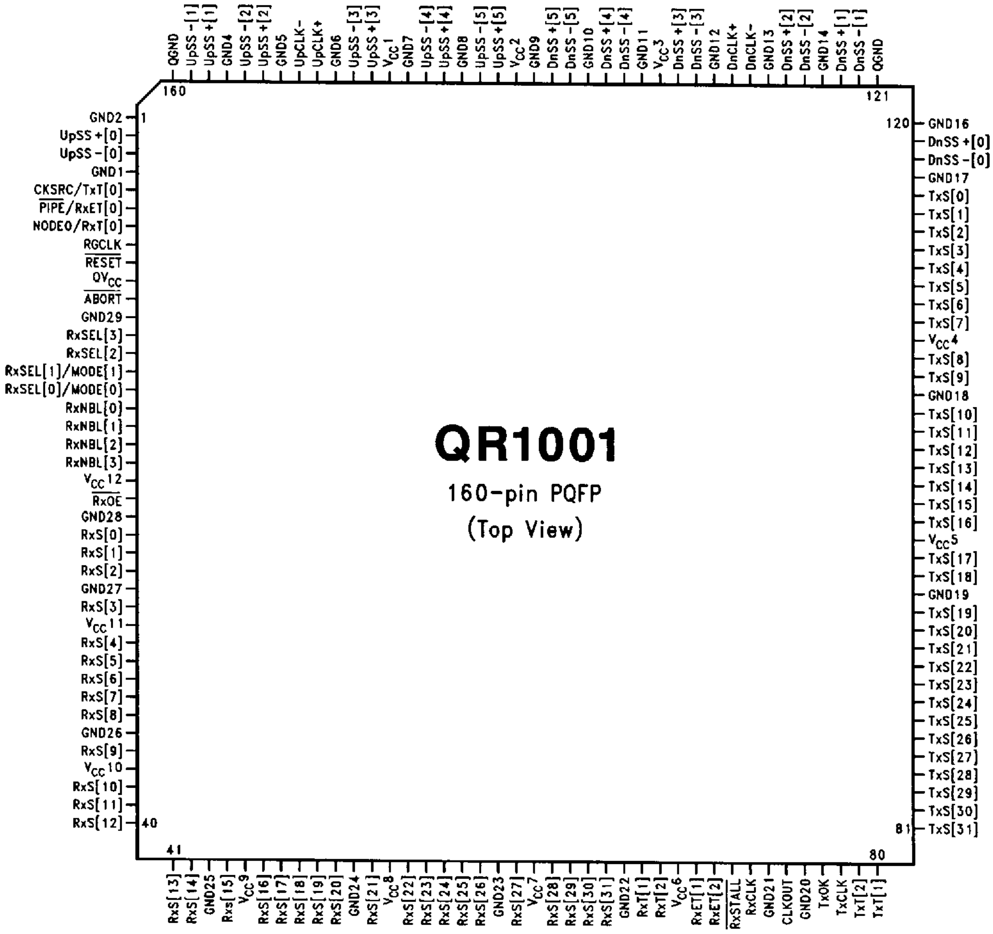


TL/F/12048-29

#	Symbol	Description	Min	Typ	Max	Units
150	t_{MODES}	MODES Set Up Time to RESET High	9			ns
151	t_{MODEH}	MODES Hold Time from RESET High	0			ns
152	t_{NODE0S}	NODE0 Set Up Time to RESET High	9			ns
153	t_{NODE0H}	NODE0 Hold Time from RESET High	0			ns
154	t_{PIPES}	PIPE Set Up Time to RESET High	9			ns
155	t_{PIPEH}	PIPE Hold Time from RESET High	0			ns
156A	$t_{CKSRCSA}$	CKSRC Set Up Time to RESET High (Initial applied power or changed CKSRC state)	same as #83	see #83		ms
156B	$t_{CKSRCSB}$	CKSRC Set Up Time to RESET High	1	0.05		ms
157	t_{CKSRCH}	CKSRC Hold Time from RESET High	0			ns

Note 1: This parameter is depenent on clock frequency.

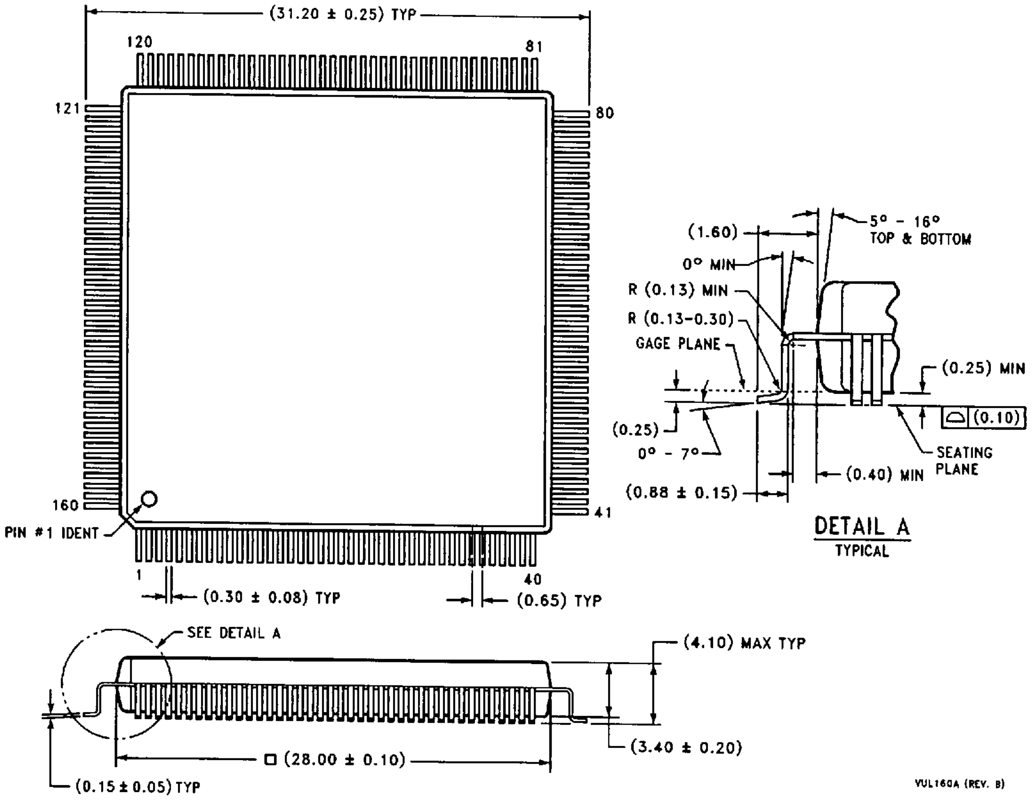
15.0 Connection Diagram



TL/F/12048-6

Order Number:	Description
QR1001-33VUL	Maximum Ring Clock Frequency of 33 MHz (Point-to-Point Raw Bandwidth of 132 MB/s)
QR1001-40VUL	Maximum Ring Clock Frequency of 40 MHz (Point-to-Point Raw Bandwidth of 160 MB/s)

Physical Dimensions inches (millimeters) unless otherwise noted



160-Lead (28mm x 28mm) Molded Plastic Quad Flat Package, JEDEC
Order Number QR1001
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
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

http://www.national.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
Français Tel: +49 (0) 180-532 93 58
Italiano Tel: +49 (0) 180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2308
Fax: 81-043-299-2408

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.