Preliminary Data Sheet

VSC7201C

1 GByte/Sec SCI Compliant Link Controller

Features

- Conforms to IEEE SCI Standards: IEEE Standard 1596-1992
- Sends and Receives SCI Data in 2ns for 1GByte/s Data Rate
- 18 Signal Parallel Link Interface
- High Speed Link Interface Conforms to Low Voltage Differential I/O Standard (IEEE P1596.3)
- 64-Bit Bi-Directional GTL System Interface
- +3.3V and +2V Power Supplies Required
- 4 Entry Send Queue, 8 Entry Receive Queue Buffers
- 269 Tape Ball Grid Array Package (TBGA - 50 mil centers)
- IEEE Std 1149.1 Test Access Port for Diagnostics

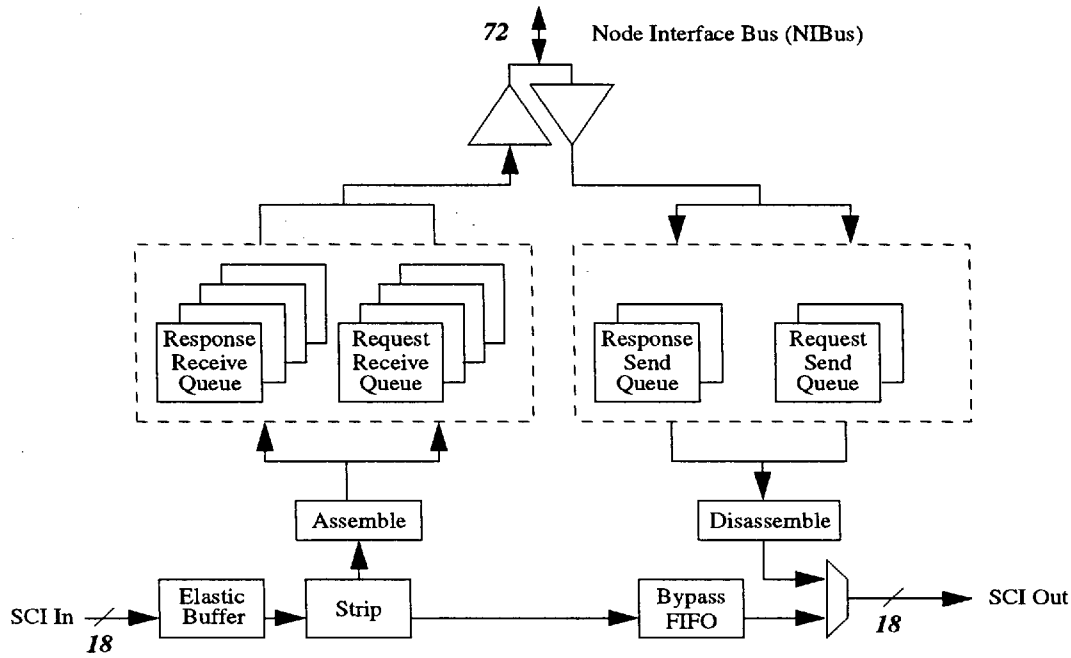
1.0 Introduction

The Scalable Coherent Interface (SCI) provides services similar to those commonly offered in a computer bus architecture. In a multiprocessor environment, however, the scalability of a traditional bus is limited by physics; specifically, problems associated with tapped transmission lines, reflections and capacitive loading. To overcome these problems SCI uses a collection of fast point-to-point unidirectional links instead of a physical bus.

A packet transfer protocol is used by SCI to implement various transactions on the high speed links. This arrangement scales very well from a small number of nodes to a large number of nodes. And SCI has specified protocol and link management to guarantee deadlock free transmission ensuring forward progress on all nodes.

The VSC7201C SCI DataPump implements the link physical transport layer of SCI including guaranteed delivery and forward progress protocol.

VSC7201C - Block Diagram



741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

2.0 Terms and Conventions

For reference consult the following documents;

1) IEEE Std 1149.1-1990 IEEE Standard Test Access Port And Boundary-Scan Architecture.

2) IEEE Std 1212-1991 IEEE Standard Control and Status Register Architecture for Microcomputer Buses.

3) IEEE Std 1596-1992 IEEE Standard for Scalable Coherent Interface (SCI).

2.1 Bit and Byte Ordering

The addressable unit in SCI is the byte. SCI packets are constructed from 2-byte (doublet) symbols.

Bit zero is always the most significant bit of a symbol, byte zero is always the most significant byte of a symbol, and the most significant doublet of the address always comes first. Bytes within a packet progress from byte[0] to byte[N] with increasing time. This is big-endian packet notation.

Registers on the DataPump are defined as 4 bytes (quadlet) in size. Data transfers on the NIbus between the DataPump and node interface logic are 8 bytes (octlet) in size. For address invariance, the mapping of bytes within a packet to bytes within a quadlet or octlet is always the same, with byte[0] being the most significant. On the NIBus this means that DATA[0] maps to bit[0] of byte[0] and DATA[63] maps to bit[7] of byte[7]. Within a register, byte[0] is the most significant and byte[3] is the least significant. This big-endian notation implies that smaller address values are more significant than larger ones.

For the defined packets and registers, the sizes of all fields within the data unit (e.g. doublet, quadlet, octlet) are specified; the bit position of each field is implied by the size of the fields to its right or left with the leftmost bit position being 0, or most significant. This labelling convention is more compact than bit-position labels, and avoids the question of whether 0 should be used to label the most or least-significant bit.

2.2 Numerical Values

Decimal, hexadecimal, and binary numbers are used within this specification. Decimal numbers are represented in their standard 0, 1, 2, ... format. Hexadecimal numbers are represented by a string of one or more hexadecimal (0-9, A-F) digits followed by the subscript 16. Binary numbers are represented by a string of one or more binary (0,1) digits followed by the subscript 2. The character x or X is sometimes used as a character in hexadecimal and binary formats and represents the "don't care" value. Each x or X is one digit in size.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

3.0 Quick Signal Pin Reference

3.1 Node Interface Bus (NIBus) Signals

Signal	Type	Level	# Pins	Description
PDATA[0:63]	IO	GTL	64	Data for packet transfer.
PPARITY[0:7]	IO	GTL	8	Byte parity on PDATA. Good parity is odd. PPARITYO corresponds to PDATA [0:7]
NDPSEL	In	GTL	1	DataPump chip select.
PCMND[0:2]	In	GTL	3	DataPump access command for queue and register reads and writes.
NNIACK	In	GTL	1	From node interface logic indicating successful access command completion.
NNIRDY	In	GTL	1	Transfer flow control from node interface logic. Assert equals ready to transfer. De-assert for wait.
NDPACK	Out	GTL	1	From DataPump indicating successful access command completion.
PRCVREQ	Out	GTL	1	Indicates Receive Request Queue has data.
PRCVRSP	Out	GTL	1	Indicates Receive Response Queue has data.
PSNDREQ	Out	GTL	1	Indicates Send Request Queue has space.
PSNDRSP	Out	GTL	1	Indicates Send Response Queue has space.
NINTRNI	Out	GTL	1	Node interface logic interrupt for asynchronous link events and errors.
NERRNI	Out	GTL	1	Asserted low when a node interface bus error occurs.
PCLKSTB	Out	GTL	1	Indicates clockStrobe packet was output on the SCI link.
NRESET	In	GTL	1	Resets DataPump and SCI link.
NSYNCRQ	In	GTL	1	Request to send Sync packet.
NSCRUB	In	GTL	1	Selects DataPump as SCI link scrubber.
PPC[0,1]	Out	GTL	2	Performance counters outputs.
CLKNI	In	TTL	1	Input Clock for NIBus.
CLKHI, NCLKHI	In	Diff.GTL	2	Differential SCI clock input for multiplication to 500Mhz.
PCKHMPY[0:1]	In	GTL	2	Divider value for CLKHI to produce internal SCI clock. See SCI clock divide table in Section 5.1.17.
PCLK250	In	TTL	1	Works in conjunction with PCKHMPY[0:1] to determine internal clock rate. See SCI clock divide table in Section 5.1.17.
VGREF	In	ANALOG	1	GTL external reference input

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

3.2 SCI Link Interface Signals

Signal	Type	Level	# Pins	Description
PSCISI, NSCISI	In	LVDS	2	SCI differential input strobe.
PSCIFI,NSCIFI	In	LVDS	2	SCI differential input flag.
PSCIDI[0:15], NSCIDI[0:15]	In	LVDS	32	SCI differential input data.
PSCISO, NSCISO	Out	LVDS	2	SCI differential output strobe.
PSCIFO, NSCIFO	Out	LVDS	2	SCI differential output flag.
PSCIDO[0:15], NSCIDO[0:15]	Out	LVDS	32	SCI differential output data.

3.3 Test Access Port and Internal Scan Test Signals

Signal	Type	Level	# Pins	Description
PTDI	In	TTL	1	JTAG Test Access Port (TAP) Test Data In.
PTMS	In	TTL	1	TAP Test Mode Select
PTCK	In	TTL	1	TAP Test Clock.
PTDO	Out	TTL	1	TAP Test Data Out.
NTRST	In	TTL	1	TAP Test Reset.
PCKHSEL	In	TTL	1	CLKHI bypass select mux. When asserted CLKHI input will bypass the PLL and drive the internal SCI clock directly.
PTSTMD	In	TTL	1	Not used. Always assert low.
PSTEP	In	TTL	1	Not used. Always assert low.
PSTOP	In	GTL	1	Not used. Always assert low.
PMAINTMD[0:1]	In	TTL	2	PMAINTMD0 toggles NIBus parity checking. PMAINTMD1 modifies initialization sequence for test purposes.
PDIV10OUT	Out	TTL	1	Internal 1GHz PLL clock divided by 10 test output.
IPNC	In	TTL	1	Factory test scan input. Tie to VCC.
OPNC	Out	TTL	1	Factory test scan output. Do not connect.
TE	In	TTL	1	Factory test enable. Tie to VCC.

Total Signal Pins = 181

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

4.0 SCI Overview

The objective of the Scalable Coherent Interface (SCI) standard is to provide a high performance interconnect system between processors and processor elements for tightly coupled, cache coherent data communication.

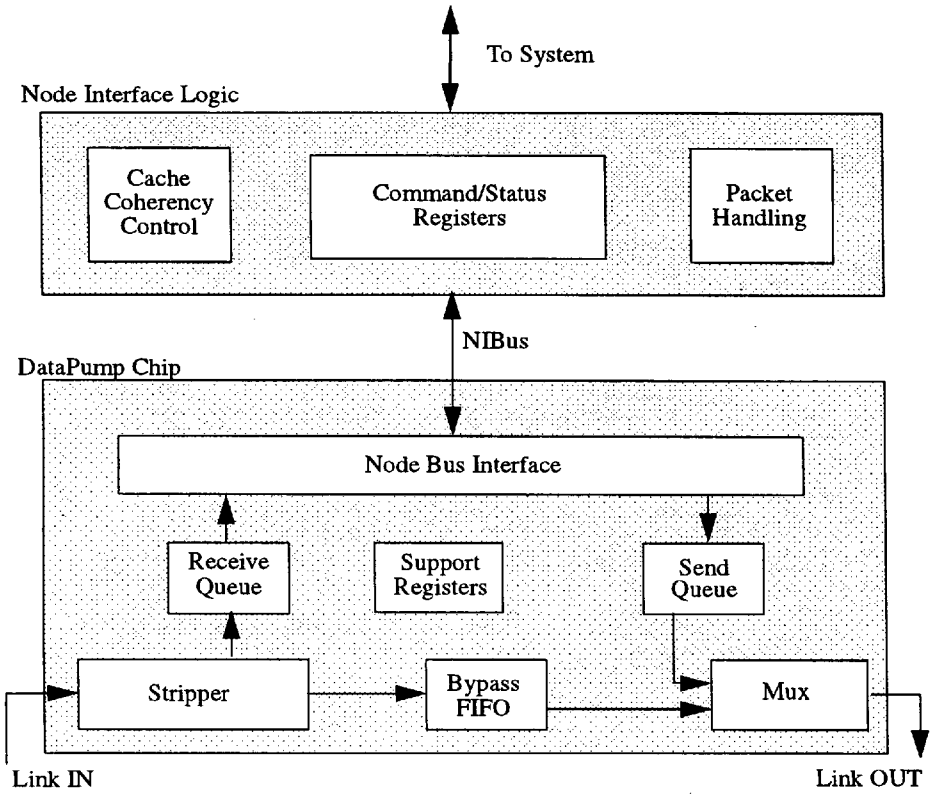
SCI utilizes point-to-point links and passes data packets to avoid the problems of bus design such as shared resource bandwidth bottlenecks and design of multi-drop, high speed backplane transmission lines.

The DataPump chip provides high speed SCI links, sends and receives packets, and manages the data transfer on the SCI physical layer.

4.1 SCI Node Model

A complete SCI node consists of high speed SCI input and output links, queues for receiving and sending packets, a bypass FIFO for storage when the node is sending a packet, and upper level protocol management for transaction handling. Figure 4.1 shows a block diagram of an SCI node.

Figure 4.1 : SCI Node Mode



The SCI function is broken into two distinct blocks. The physical link interface and queues for packets are handled on the DataPump chip, also known as a link controller. The higher level protocol such as packet handling, cache coherence, CSR (control and status register) register support, and interface to the system is handled in node-specific interface logic.

Therefore, the DataPump chip takes care of getting data packets on and off the high speed SCI link and transferring those packets to and from the node interface logic.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

4.2 DataPump Block Description

The basic block diagram of the DataPump is shown in Figure 4.1. Data is received at the stripper block on the link inputs.

The start of a data packet contains header information. The first 16-bit symbol in the header is the nodeId address of the target node to receive the packet. The stripper block checks this targetId to see if the packet is for this node. If so, the packet is stripped off the ringlet. If there is receive queue space available, the packet will be stored for later unloading by the node controller. If there is no queue space available the received packet will be discarded and a message will be sent to the sender to retry the packet again later.

The receive queue block is split into storage for two types of packets, requests and responses. There is room for up to four packets of each type. Within receive request or response queues, slots are assigned on a first-in/first-out basis.

If the packet is not intended for this node it is sent through the bypass FIFO. The bypass FIFO is required to store a packet as it is received if the DataPump is sending data at the same time. The send queue can only begin to transmit a packet if the bypass FIFO is empty. It is only allowed to transmit one packet at a time before it must again check the bypass FIFO. Therefore, the storage size of the bypass FIFO must be large enough to buffer the largest packet size which can be sent.

The send queue is also split into two types, requests and responses, of which there are two queue slots for each type. The queue slots are loaded from the node interface logic and the slot loading order and transmission order is based on packet age.

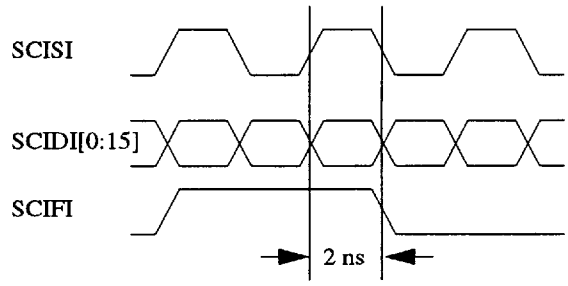
The DataPump also contains bus interface logic for communication with the node interface logic and on-board registers for node control and status register (CSR) support. However, on-chip registers are only provided to control the operation of the DataPump - full CSR support according to IEEE Std 1212-1991 must be provided by the node interface logic.

4.3 Physical Layer Connection

The SCI physical layer connection consists of 18 differential input and output signals. The inputs consist of 16 data signals (PSCIDI[0:15]), a flag bit (PSCIFI) used to delimit packets, and a strobe signal (PSCISI) for latching incoming data. The outputs consist of 16 data (PSCIDO[0:15]), a flag (PSCIFO), and strobe (PSCISO).

Each 16-bit data quantity transferred is called a symbol and is clocked on each rising and falling edge of the strobe signal. The 16 data plus flag signals transition in phase every 2ns (250 MHz) giving an effective data transfer rate of 1Gbyte/sec. Figure 4.2 illustrates these signals using single-ended notation.

Figure 4.2 : SCI Link Signals



The received strobe PSCISI is used for latching data and flag, but is not used for generating the output strobe PSCISO. The DataPump chip generates its own internal clock for clocking data through the high speed data path and driving out PSCIDO[0:15], PSCISO, and PSCIFO.

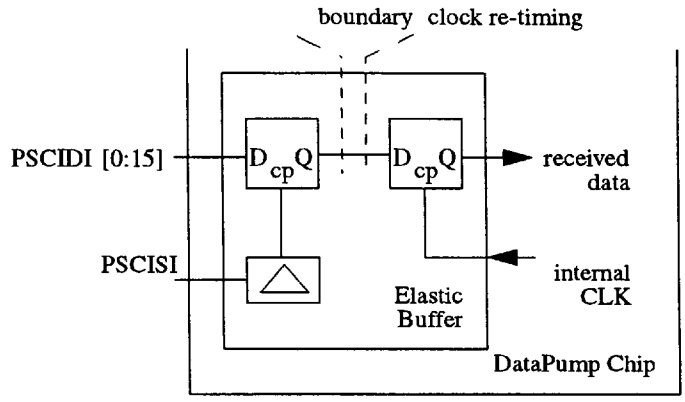
The internally generated clock is not guaranteed to be in phase with nor at precisely the same frequency as the incoming PSCISI. Therefore, all SCI link input signals are received on chip through the elastic buffer which re-times the signals to the internal high speed clock and inserts or deletes symbols with a logic state machine to account for small frequency

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

differences between incoming symbols and the internal clock. The clocking boundaries and elastic buffer data re-timing function are shown in Figure 4.3.

Symbols called idle symbols are transmitted between data packets or if no packets are being sent. The elastic buffer will only delete idle symbols, not symbols within a packet. Every packet must be followed by at least one idle symbol. This guarantees that there will always be enough idle symbols received which can be deleted in order to make up for the worst case clock frequency difference.

Figure 4.3 : Elastic Buffer Data Re-timing



4.4 Packet Formats

SCI defines two groups of packet types; those packets involved in the logical protocol (send and echo packets), and other special link related packets.

The VSC7201C does not support any of the special init packets except for the SYNC packet.

4.4.1 Basic Send and Echo Packets

The packets involved in the logical protocol consist of four types; request-send, request-echo, response-send, and response-echo.

Logical protocol transactions are initiated with a requester and completed by a responder. Each transaction consists of two sub-actions; a request sub-action wherein command and possibly data are passed to the responder and a response sub-action where completion status and possibly data are returned to the requester.

Each sub-action involves two packet transfers. One is a send packet initiated at the output link of a producer node and the second is an echo packet returned by the consumer node and received on the input link of the producer node.

Hence, normal SCI transactions are usually four-way transactions initiated with a send-request packet from a requester. The target of the request (the responder) sends an acknowledgment of receipt of the request packet by returning a request-echo packet. When the responder is ready with the requested data, it sends a response-send packet and the original requester acknowledges with a response-echo packet.

All packets are an integer multiple of four symbols in length. The DataPump supports all SCI packet types except 256-byte block sizes. Therefore, the largest packet size is 96-bytes, or 48 16-bit symbols.

The DataPump also uses full 16-bit nodeIds for targetId and sourceId decoding.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 4.4 : Request Send Packet Format

targetId (<$FFF0_{16}$)
command
sourceID
control
addressOffset[00.15]
addressOffset[16.31]
addressOffset[32.47]
ext (0 or 16 bytes)
data (0, 16, 32, 48 or 64 bytes)
cyclic-redundancy code (CRC)

The basic request-send packet construction is shown in Figure 4.4. Each block in Figure 4.4 comprises a 16-bit symbol value. The targetId is the 16-bit nodeId of the target node. Node ID values above $FFF0_{16}$ are special reserved values.

The command and control fields contain command and flow-control information. Some of the bits used for flow-control are modified by the DataPump chip. See sections 4.6 and 4.7. The sourceId is the 16-bit node ID of the sending node. The 48-bit address offset is interpreted by the responder.

A packet may also contain a 16-byte extended header. The presence of extended header is indicated by a bit (com.eh) in the command field.

The CRC (cyclic redundancy code) symbol at the end of the packet allows for error checking the entire data packet upon reception.

When transmitting a packet from the send queue, the DataPump forms the CRC and appends it to the end of the packet. When receiving a packet into the receive queue, the DataPump forms the CRC as the packet is being received and checks the generated value against this transmitted value to ensure data correctness.

Figure 4.5 : Request-Echo and Response-Echo Packet Format

targetId (<$FFF0_{16}$)
command
sourceId
cyclic-redundancy code (CRC)

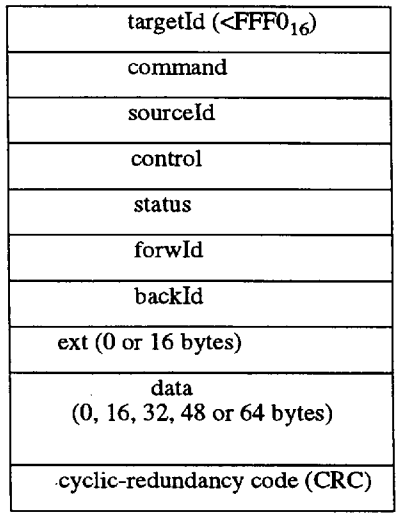
The node interface logic external to the DataPump does not need to check or generate the CRC.

Echo packets are 4 symbols long containing the source and target IDs exchanged, a command symbol which is a modified version of the send command, and the CRC.

The DataPump automatically generates all required echo packets as a result of received packets. The node interface logic external to the DataPump cannot generate or receive echo packets.

The basic echo packet is shown in Figure 4.5.

Figure 4.6 : Response-Send Packet Format



The basic response packet is shown in Figure 4.6. The response packet is similar to request packet except a status symbol and two nodeId pointers called forwId and backId are returned. These are used in the cache coherency scheme.

The DataPump does not, however, handle any upper level cache coherency protocol management. Response packets are simply queued with a length of packet indicator.

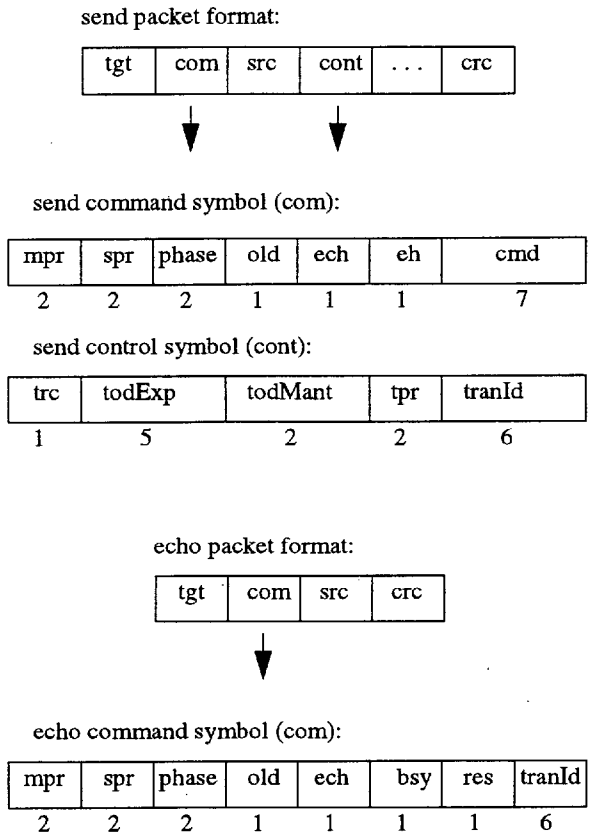
The DataPump also manages data flow control information contained within packets and idle symbols. When no packet is being received, 16-bit idle symbols are received. The DataPump stores and transmits these idles, while checking and managing flow control information contained in the idle symbol. For more detailed information about packet types and fields within packets, consult the IEEE Std 1596-1992.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

4.4.2 Command and Control Symbols

The DataPump uses certain bits in the command and control fields for transaction and flow control. The format for these fields is defined in Figure 4.7. These fields are explained briefly in Table 4.1

Figure 4.7 : .Command and Control Symbols



741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

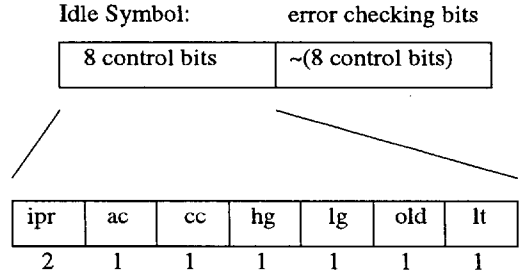
Table 4.1: Command and Control Bit Fields

Bit Field	Description
com.mpr	Maximum ringlet priority field. Passed but not used by DataPump.
com.spr	Send priority. Passed but not used by DataPump.
com.phase	Used by queue control on DataPump. See 4.7 Queue Allocation.
com.old	Packet aging used by scrubber DataPump to remove old packets.
com.ech	Indicates echo packet; com.ech=0 for send, com.ech=1 for echo. Set by DataPump during transmission.
com.eh	Indicates use of extended header. If set to 1 a 16-byte extended header is present.
com.cmd	Specifies transaction command.
com.bsy	Used in echo packet to indicate no queue space available on target node. See 4.7 Queue Allocation.
com.res	In echo packet indicates request or response echo; com.res=0 for request, com.res=1 for response.
com.tranId cont.tranId	Transaction Id set by requester in send-request packet. This tranId value is used in all subaction packets related to the request.
cont.trc	Trace bit. Passed but not used by DataPump.
cont.todExp	Time-of-death; exponent. Passed by but not used by DataPump.
cont.todMant	Time-of-death; exponent. Passed by but not used by DataPump.
cont.tpr	Transmit priority, set by node interface. Passed but not used by DataPump.

4.4.3 Idle Symbols

Idle symbols fill the spaces between packets. They contain bits associated with ringlet flow control which the DataPump uses to manage the SCI link. Figure 4.8 shows the bit fields and Table 4.2 defines these fields. The least significant byte is the complement of the most significant byte and is used for a simple parity check.

Figure 4.8 : Idle Symbol Fields



741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Table 4.2: Idle Symbol Field Descriptions

Bit Field	Description
idle.ipr	Idle-priority used for ringlet priority; not used. Set to low.
idle.ac	Allocation count, toggles when all nodes are enabled to transmit.
idle.cc	Circulation count, indicates when idle has circulated around ringlet.
idle.hg	High go bit; not used. Set to low.
idle.lg	Low go bit; see Section 4.6 Bandwidth Allocation.
idle.old	Packet aging bit - used by scrubber to remove old packets from ringlet
idle.lt	Low type priority class bit; not used. Set to low.

4.4.4 SYNC Packet and its Flag Encoding

The sync packet is used during initialization for synchronizing the link and can also be sent during normal operation by asserting the NSYNCRQ pin. Although the VSC7201C doesn't have autodeskewing on its input pins, future versions may use the received SYNC packet for deskewing SCI inputs. The format for the SYNC packet is shown in Figure 4.9. No other SCI defined special packets (e.g. ABORT, RESET, etc.) are supported or needed by the VSC7201C. For further information on reset and initialization and the use of the SYNC packet, see sections 4.10 DataPump Reset.

Figure 4.9 : SYNC Packet Format

Symbol		Flag
0	1111111111111111	1
1	0000000000000000	0
2	0000000000000000	0
3	0000000000000000	0
4	0000000000000000	0
5	0000000000000000	0
6	0000000000000000	0
7	0000000000000000	0

SYNC Packet

4.4.5 Packet and Idle Alignment to SCI Strobe

For performance reasons the DataPump aligns all packet transmisions to start with the first symbol driven when PSCISO (strobe out) is high. The DataPump also expects to receive packets aligned so that the first symbol is received when PSCISI (strobe in) is high.

Therefore, the minimum number of idle symbols between packets will be 2 and idle deletion and insertion will be done on idle pairs. Idle pairs are also aligned with strobe high followed by strobe low.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

4.5 Transaction Commands

The command in an SCI send packet identifies one of four main types of transactions. Those are response-expected request, move request, event request, and response. The DataPump only decodes the transaction command to determine if it is greater or less than 124 indicating a response versus request packet, or if it is an event00 (clockStrobe) command.

Data payload size within a send packet is only checked for the SCI legal multiple of 8 symbols and for excessive length. When the end of a send packet is received it is checked for length equal to a multiple of 8 symbols and if not is rounded out to 8 symbols, then processed normally.

A send packet not targeted for this VSC7201C is not stripped and is only checked for size less than or equal to 48 symbols. Packets of this length will be bypassed without error as long as they are a multiple of 8 symbols.

Packets which are stripped by the DataPump are checked for length less than or equal to the maximum queue buffer size of 48 symbols. As long as stripped packets are of this size and are a multiple of 8 symbols they will be queued normally.

The transaction types uniquely treated by the DataPump are summarized in Table 4.3. Packet sizes for these commands are only checked as outlined above. For example, a nread256 command implies a data payload of 256-bytes but is not checked for that payload size by the DataPump and could in fact be only 64-bytes.

Table 4.3: DataPump Receive Packet Handling

Received Packet Type	Packet Handling	Comments
response-expected request	strip if targeted node and generate echo; else bypass	DataPump only checks command value is less than 124 and not event00
response	strip if targeted node and generate echo; else bypass	DataPump only checks that command value is greater than or equal to 124
directed move (dmovexx)	strip if targeted node and generate echo; else bypass	Command value not distinguished from response-expected request
broadcast move (rmovexx, smovexx)	not supported	Command value not distinguished from response-expected request
clockStrobe (event00, time-of-day clock)	strip and toss if targeted node; else bypass	See Section 4.5.1 Event00 (clockStrobe) Transaction
event16, event64	not supported	Command value not distinguished from event00
echo packet	strip if targeted node; else bypass	stripped echo packet information is used to determine send packet status; see send packet handling in 4.7 Queue Allocation

4.5.1 Event00 (clockStrobe) Transaction

The event00 transaction is supported by the DataPump as the (time-of-day) clockStrobe signal. If targeted to a DataPump, the Event00 packet is accepted immediately and no echo is generated. If received and not targeted to the DataPump, it bypasses the packet.

The DataPump which transmits an event00 targeted to itself from its send queue is called the clockStrobe master. Datapumps which bypass the event00 are clockStrobe slaves. When the event00 packet is transmitted from the send queue it is immediately removed from the queue. When the event00 packet completes its trip around the ringlet and returns to the clockStrobe master, it is stripped and discarded.

When an event00 packet is transmitted on the DataPump's outputs, the clockStrobe pin PCLKSTB is asserted to the node interface logic. PCLKSTB is deasserted when the event00 is received and stripped. The node interface logic can use this signal to enable a timer to measure time around the ringlet.

When an event00 packet is bypassed through a DataPump the PCLKSTB signal is asserted for a duration of approximately 48ns, then deasserted. The through time taken to bypass the packet is also measured by counting SCI clock cycles in the DP_CLKTHRU register described in Section 4.8. Other event packets are treated like event00 packets.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

4.6 Bandwidth Allocation and the Low-go Bit

Since SCI nodes are connected in ringlets, multiple transactions may be transmitted concurrently. Since there is no arbitration for access to the ringlet, certain bandwidth allocation protocols have been defined to guarantee bandwidth to all nodes.

Bandwidth allocation is managed by two mechanisms. These are; 1) transmission flow control through the use of a token, or "go" bit, in idle symbols which determine when a node is allowed to transmit, and 2) priority schemes using command.mpr, command.spr, control.tpr, and idle.ipr fields which allow unfair nodes to determine the priority for transmission of unsent packets.

The DataPump uses the idle.lg "low-go" bit for flow control but does not support any priority allocation mechanisms. Since idles are always received in aligned pairs (see Section 4.4.5) the idle.lg bits in idle pairs are essentially or'ed together.

A DataPump can only transmit a packet from its send queue when its bypass FIFO is empty and it can postpend the transmission packet to an idle pair with the low-go bit set. If no idles with low-go set are passed through to the transmitter then the DataPump is disabled from sending packets from the send queue.

The low-go bit in outgoing idles is blocked (set to zero) whenever the DataPump's bypass FIFO is not empty. This prevents a downstream DataPump from transmitting and will eventually free up ringlet bandwidth so the blocked DataPump can empty its bypass FIFO.

While the DataPump is blocking low-go, any idle received with low-go set will set the saved value of low-go which will be transmitted in the first pair of idle symbols after the DataPump is no longer blocked (i.e. bypass FIFO is emptied).

Finally, if a node is not blocked and an idle pair with low-go set is transmitted, the idle pair immediately following will also have their low-gos set. These low-go extensions will prevent a complete loss of low-go bits in a ringlet.

If all low-go bits in all idles are cleared (thus no one is allowed to transmit), the scrubber will detect this and set an error condition.

4.6.1 Allocation Count and Circulation Count

Two flag bits in the idle symbols, idle.ac and idle.cc, keep track of when all producers have had the opportunity to transmit and when an idle symbol has circulated completely around the ringlet. These bits are controlled by the scrubber.

Circulation count is passed by non-scrubber nodes and toggled by the scrubber. When a non-scrubber node is transmitting idles, the value of idle.cc on output is set to the most recently received idle.cc value. In a scrubber node, output idle.cc is the complement of the most recently received idle.cc value. Thus, idle.cc acts as a token which takes roughly the latency of the ringlet to circulate.

The allocation count bit, idle.ac, is similar to idle.cc except it is a token which indicates when all nodes have had an opportunity to transmit. When the idle.ac bit toggles, a time interval has elapsed in which all nodes have had the opportunity to transmit.

The idle.ac bit is passed by a node when its output is not blocked. When its output is blocked (i.e. transmitting from its send queue or emptying bypass FIFO), it must output the idle.ac value saved from the last idle received before transmission began. Therefore, idle.ac cannot change at the output of a blocked node. It can only change after the node has recovered and is enabled for transmission. A non-scrubber node will pass the value of idle.ac which was last received. The scrubber will output the complement of this value. The allocation count and circulation count flags are used to detect error conditions or set time-outs as described in following sections.

4.7 Queue Allocation

Bandwidth allocation protocols guarantee that all nodes get bandwidth to transmit packets. However, if the queues on a consumer node become filled, then new send packets for this node are echoed with a "busy" status and must be re-sent until queue space is available and they are accepted. Queue allocation protocol includes a simple reservation scheme insuring all producer's packets eventually get accepted by consumers.

This reservation mechanism is described in the SCI standard. There are four queue allocation states which the ringlet may

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

be in at any time. These are SERVE_NA, SERVE_A, SERVE_NB, and SERVE_B. Also, send packets and busy echo packets contain a phase field which has values NOTRY, DOTRY, RETRY_A, and RETRY_B.

The DataPump implements this A/B aging protocol. In states SERVE_NA or SERVE_NB, basically all packets are accepted if queue space is available. If any packets are busied, the queue will transition to SERVE_A or SERVE_B.

In SERVE_A and SERVE_B states, the DataPump will only accept packets with phase RETRY_A or RETRY_B respectively. Exiting either SERVE_A or SERVE_B is handled by maintaining counters for phase A and B which accumulate the number of busied packets for retry and decrementing the counter every time a packet is accepted (the count is incremented every time a packet sent with DOTRY phase is returned busied of type A or B). When the count is back to zero, the queue transitions back to SERVE_NA or SERVE_NB.

The busied counter is 4-bits. If the number of busied packets is less than 15 then the DataPump will exit SERVE_A or SERVE_B when all busied packets have been accepted.

If the number of busied packets equals or exceeds 15, exiting SERVE_A or SERVE_B will be accomplished by a time-out on the allocation count counter. The allocation counter increments each time all nodes on the ringlet have had the opportunity to transmit. If no more RETRY_A or RETRY_B packets have been received within 4 allocation intervals, the queue times out and transitions to SERVE_NA or SERVE_NB.

A time-out on the allocation counter when the busied packet counter did not overflow is a ringlet state error and is described in the error handling Section 4.9.

Separate state machines for request and response queues process packet reservations separately to avoid deadlock situations.

For more detailed descriptions of the A/B reservation scheme or the com.phase, com.bsy, or idle.ac fields refer to the SCI standard, Section 3.7.

Queue selection protocol specifies that for a node with multiple request or response send queue entries only one packet at a time may be sent with DOTRY phase for each queue type, request or response. Also, the producer should alternate transmission order between request and response so that both queues are serviced equally. For more information on queue selection, see Section 5.3 Send Queue SCI Transmission Order.

4.8 Control and Status Support Registers

SCI follows the CSR (Control and Status Register) architecture as defined in IEEE Std 1212-1991. An SCI node must have the defined registers and behavior as outlined in the standards.

The DataPump does not contain any CSRs but has configuration and status registers modeled after the CSR standard to provide the functionality required to implement full CSR compliance on the node interface logic.

CSR registers are software visible to both the processor elements connected to a node and to remote nodes through the SCI links. CSRs are address mapped into memory address space as defined by the 64-bit fixed address model, which allocates the most significant 256Mbyte space to registers.

Remote access through SCI is provided by read and write request transactions to that address space. The DataPump does not, however, decode requests to CSR address space. It simply queues request packets and sends them up to the node interface logic. The node interface logic is responsible for servicing the CSR requests (both local and remote).

The node interface logic must use the support registers on the DataPump to set state and values in the node CSRs. For example, the DP_STATE register indicates the operational state of the DataPump. If the DataPump has been reset from the SCI link, it will enter the "initializing" state reflected by the DP_STATE resetlinc bit. The logical SCI node's STATE CSR is then the combination of the DataPump state and the rest of the node interface logic state.

The CSR support registers can also be accessed through the Test Access Port scan port. See 9.0 Test Access Port for details.

These registers are initialized by toggling the NRESET pin. A complete description of chip reset is given in Section 4.10 DataPump Reset. A summary of the CSR support registers on the DataPump is given in Table 4.6.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Table 4.6: DataPump CSR Support Registers

Register	Offset	Description
DP_STATE	000_{16}	The DP_STATE register contains the current status of the DataPump. It also contains the chip revision number.
DP_NODEID	008_{16}	Contains nodeId value and distanceId value from reset initialization.
DP_CLKTHRU	070_{16}	Contains the time interval between arrival and departure of the clockStrobe transaction. Measured in 4ns increments.
DP_ERRLOG	184_{16}	Contains error codes and diagnostic information.
DP_SENDQ_TAG[3:0]	300_{16},304_{16}, 308_{16},$30C_{16}$	Provide read access to the send queue tranId and targetId values for each packet in the queue.
DP_PC_CONFIG0	320_{16}	PPC0 output pin selection/configuration and mask/compare values for performance counters.
DP_PC_CONFIG1	328_{16}	PPC1 output pin selection/configuration.

These registers are detailed in the following sections. Tables for each register indicate the register functionality and use the following code for access restrictions; RO - read only; RW - read or write value to 1 or 0; RC - read or clear value on write. The clear function is NOT a bitwise clear. Just writing the CSR register with any data value will clear the register if it is RC.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

4.8.1 DP_STATE (Offset 000_{16})

Table 4.7: DP_STATE Register

Name	Bits	Access	Description
rev_num[0:3]	0:3	RO	Hardwired chip revision number; 0010 for this chip.
reserved	4:7	-	(Test usage only: timeadj[7:4])
stayrun	8	RW	Stay in running state if a fatal link error occurs when set.
moretry	9	RW	When asserted, enables Queue Allocation as described in Section 4.7. When negated compatible with 7201A.
reserved	10:11	-	(Test usage only: timeadj[2:0])
insync[0:1]	12:13	RO	Count of the number of valid sync pakets received during initialization state INIT.
not used	15:14	-	
gotodead	16	RW	When asserted forces DataPump to DEAD state.
not used	17	-	
ruup	18	RO	When true indicates SCI input strobe is running.
runlinc	19	RO	When asserted indicates DataPump is in the RUN state.
initlinc	20	RO	When asserted indicates DataPump is in the INIT state.
sngerr	21	RW	Capture first fatal error only (single error; see Section 4.9.)
not used	22	-	
sigrst	23	RW	Used during initialization.
deadlinc	24	RO	When asserted indicates DataPump is in the DEAD state.
resetlinc	25	RO	When asserted indicates DataPump is in the RESET state.
inscrub	26	RO	Indicates this DataPump is the scrubber when asserted.
not used	27:30	-	

Name	Bits	Access	Description
dreq	31	RW	Disable requests. DataPump will not accept packets from the NIBus when set.

4.8.2 DP_NODEID (Offset 008_{16})

Table 4.8: DP_NODEID Register

Name	Bits	Access	Description
nodeid[0:15]	0:15	RW	Node ID value for this DataPump. Should be set before normal operation.
not used	16:25	-	
clkthru[0:5]	26:31	RO	Count of SCI clock cycles elapsed while an event00 packet was bypassed through a clockStrobe slave DataPump (see Section 5.1.11 on the PCLKSTB signal)

4.8.3 DP_CLKTHRU (Offset 070_{16})

Table 4.9: DP_CLKTHRU Register

Name	Bits	Access	Description
nodeid[0:15]	0:15	RO	Node ID value for this DataPump, read-only.
not used	16:25	-	

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Name	Bits	Access	Description
clkthru[0:5]	26:31	RO	Count of SCI clock cycles elapsed while an event00 packet was bypassed through a clockStrobe slave DataPump (see Section 5.1.11 on the PCLKSTB signal)

4.8.4 DP_ERRLOG (Offset 184_{16})

NOTE: A CSR register write to this address clears all register bits.

Table 4.10: DP_ERRLOG Register

Name	Bits	Access	Description
not used	0:2	-	
sqfrz[0:3]	3:6	RC	Send queue slot "freeze" bits. A set bit indicates an error occurred on the packet in that send queue slot and it was "frozen". Slot is freed when bit is cleared. Tag information for each slot is contained in the DP_SENDQ_TAG registers. The correspondence is as follows; sqfrz0 -> DP_SENDQ_TAG0 sqfrz1 -> DP_SENDQ_TAG1 sqfrz2 -> DP_SENDQ_TAG2 sqfrz3 -> DP_SENDQ_TAG3
tcode[0:7]	7:14	RC	NIBus transfer errors - (see Table 4.18) bit 7 (sendQfull) - load attempted to full send Q bit 8 (dupTranId) - send req packet had a duplicate tranId bit 9 (sizeErr) - send packet size error bit 10 (NIregPar) - NIBus parity error on register read/write bit 11 (NIsendqPar) - NIBus parity error on send Q load bit 12 - not used bit 13 (rcvQempty) - read attempted from empty receive Q bit 14 (regAddrErr) - register access to bad address
notruup	15	RC	Lost input SCI strobe signal.
flgerr	16	RC	Received packet had a flag error
not used	17	-	
strperr[0:6]	18:24	RC	SCI input linc stripper errors - bit 18 (badIdle) - idle symbol had a parity error bit 19 (badThruCrc) - bad CRC in bypassed packet bit 20 (badStrpCrc) - bad CRC in stripped packet bit 21 (strpTooLong) - stripped packet greater than 48 symbols bit 22 (tossClkStb) - received extra clockStrobe packet bit 23 (reqAcTmo) - request Q reservation early ac timeout bit 24 (rspAcTmo) - response Q reservation early ac timeout

Name	Bits	Access	Description
fatalerr[0:6]	25:31	RC	Fatal linc errors - (except for tooLong which is NOT fatal) bit 25 (noInSync) - elastic buffer lost synchronization bit 26 (tooLong) - bypassed packet greater than 48 symbols bit 27 (scrbLgTmr) - scrubber detected no low-go bits in ringlet bit 28 (reqRsvErr) - bad retry phase on request Q bit 29 (rspRsvErr) - bad retry phase on response Q bit 30 (acFail) - ac toggled more than once while output blocked bit 31 (echoUnkn) - stripped echo doesn't match any in send Q

4.8.5 DP_SENDQ_TAG[0:3]

Table 4.11: DP_SENDQ_TAG3 Register Offset 300_{16} (Request Queue Slot 0)

Name	Bits	Access	Description
not used	0	-	
reserved	1:4	RO	(Test usage only: rqvld_s[3:0])
reserved	5	RO	(Test usage only: sqvld_s0)
echoNo	6	RO	target node for packet doesn't reply, packet scrubbed
sndQpar	7	RO	parity error occurred on packet transmission from send Q
sndQto	8	RO	cc timeout on packet, no echo received in 4 cc times
validbit	9	RO	valid bit for send Q slot 3
not used	10:25	-	
tranid[0:5]	26:31	RO	transaction ID in send Q slot 3 (request slot 0)

Table 4.12: DP_SENDQ_TAG2 Register Offset 304_{16} (Request Queue Slot 1)

Name	Bits	Access	Description
not used	0	-	
reserved	1:4	RO	(Test usage only: rqvld_s[7:4])
reserved	5	RO	(Test usage only: sqvld_s1)
echoNo	6	RO	target node for packet doesn't reply, packet scrubbed
sndQpar	7	RO	parity error occurred on packet transmission from send Q
sndQto	8	RO	cc timeout on packet, no echo received in 4 cc times
validbit	9	RO	valid bit for send Q slot 2
not used	10:25	-	
tranid[0:5]	26:31	RO	transaction ID in send Q slot 2 (request slot 1)

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Table 4.13: DP_SENDQ_TAG1 Register Offset 308_{16} (Response Queue Slot 0)

Name	Bits	Access	Description
not used	0:5	-	
reserved	1:4	RO	(Test usage only: rqvld_sf[3:0])
reserved	5	RO	(Test usage only: sqvld_s2)
echoNo	6	RO	target node for packet doesn't reply, packet scrubbed
sndQpar	7	RO	parity error occurred on packet transmission from send Q
sndQto	8	RO	cc timeout on packet, no echo received in 4 cc times
validbit	9	RO	valid bit for send Q slot 1
not used	10:25	-	
tranid[0:5]	26:31	RO	transaction ID in send Q slot 1 (response slot 0)

Table 4.14: DP_SENDQ_TAG0 Register Offset $30C_{16}$ (Response Queue Slot 1)

Name	Bits	Access	Description
not used	0:5	-	
reserved	1:4	RO	(Test usage only: rqvld_f[7:4])
reserved	5	RO	(Test usage only: sqvld_s3)
echoNo	6	RO	target node for packet doesn't reply, packet scrubbed
sndQpar	7	RO	parity error occurred on packet transmission from send Q
sndQto	8	RO	cc timeout on packet, no echo received in 4 cc times
validbit	9	RO	valid bit for send Q slot 0
not used	10:25	-	
tranid[0:5]	26:31	RO	transaction ID in send Q slot 0 (response slot 1)

4.8.6 DP_PC_CONFIG0 (offset 320_{16})

The detailed function of these register bits is described in Section 8.0 Performance Counters.

Table 4.15: DP_PC_CONFIG0 Register

Name	Bits	Access	Description
pc0mask[0:4]	0:4	RW	Selector for the PPC0 output pin. Assert only one at a time; bit 0 (pktSlot0Rs) - PPC0 toggles when a packet is transmitted from the send-response Q slot 0 and the packet matches the command mask/ compare value. bit 1 (pktSlot0Rq) -PPC0 toggles when a packet is transmitted from the send-request Q slot 0 and the packet matches the comand mask/compare value. bit 2 (pktRcv) - PPC0 toggles when a packet is received and targeted to this node. bit 3 (pktByp) - PPC0 toggles when a packet is bypassed. bit 4 (pktSent) - PPC0 toggles when a packet is sent.
rcvVldMask	5	RW	Used to qualify the PPC1 output pin RcvPkt when a received packet is actually stripped and the packet is entered into the receive Q (as opposed to tossed for Q reservations or other reasons).
extMask[0:4]	6:10	RW	Masks the command symbol bits 4,5,7,8,9 from the extended comparison for PPC1 matching. These bits correspond to the command phase, ech, bsy, and res bits as follows; bit 6 (extMask0) - com.phase0 bit 7 (extMask1) - com.phase1 bit 8 (extMask2) - com.ech bit 9 (extMask3) - com.bsy (for echo packets) bit 10 (extMask4) - com.res (for echo packets)
extCmp[0:4]	11:15	RW	Compare values for command symbol bits 4,5,7,8,9 as described above.
cmdSns	16	RW	Command sense bit which, when asserted, inverts the cmd field match signal to allow qualifying PPC1 counters on mis-match instead of match.
cmdMask[0:6]	17:23	RW	Masks the command symbol bits 9:15 from the command comparison for PPC1 matching. These bits correspond to the com.cmd field of send packets (not echos).
cmdCmp[0:6]	24:30	RW	Comparison value for the com.cmd bits.
refMask	31		When asserted forces the PPC0 pktRcv signal to be qualified with the ext mask/compare value. If the ext mask/compare is set to match echos, then refMask is used to cause pktRcv to count only echo packets or only send packets.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

4.8.7 DP_PC_CONFIG1 (Offset 328_{16})

DP_PC_CONFIG1 is the selector for driving the PPC1 pin. The following table lists the function of the PPC1 pin when that bit of DP_PC_CONFIG1 is asserted. Only one bit of this register should be set at a time. Setting more than one bit will produce undefined results. See Section 8.0 Performance Counters for a more complete description of these bits.

Table 4.16: DP_PC_CONFIG1 Register

Name	Bits	Access	Description
not used	0:18	-	
rsRsvPhs	19	RW	PPC1 asserted while the receive-response reservation state machine state phase matches ext bits 0:1 (extMask[0:4] set to 11000 and extCmp[0:1] set to the desired phase)
rqRsvPhs	20	RW	PPC1 asserted while the receive-request reservation state machine state phase matches ext bits 0:1
rsSlot0Tm	21	RW	PPC1 asserted while a packet matching the cmd mask/compare is entered in send-response queue slot 0.
rqSlot0Tm	22	RW	PPC1 asserted while a packet matching the cmd mask/compare is entered in send-request queue slot 0.
outputCc	23	RW	PPC1 toggles each time the idle.cc bit changes at the SCI output link.
outputAc	24	RW	PPC1 toggles each time the idle.ac bit changes at the SCI output link.
rRspDepth	25	RW	PPC1 toggles at a rate proportional to the number of q entries in the receive-response queue (see Section 8 Performance Counters)
rReqDepth	26	RW	PPC1 toggles at a rate proportional to the number of q entries in the receive-request queue (see Section 8 Performance Counters)
sRspDepth	27	RW	PPC1 toggles at a rate proportional to the number of q entries in the send-response queue (see Section 8 Performance Counters)
sReqDepth	28	RW	PPC1 toggles at a rate proportional to the number of q entries in the send-response queue (see Section 8 Performance Counters)
RcvPkt	29	RW	PPC1 toggles each time a packet matching both ext and cmd mask/ compare values is received and targeted to this node.
BypPkt	30	RW	PPC1 toggles each time a packet matching both ext and cmd mask/ compare values is bypassed through this node.
SndPkt	31	RW	PPC1 toggles each time a packet matching both ext and cmd mask/ compare values is transmitted by this node.

4.9 Error Handling

The DataPump checks and maintains an error log which records various error types. Errors fall into two categories; 1) SCI link related errors logged in the DP_ERRLOG fields notruup, flgerr, and strperr[0:6]; and 2) NIBus (node interface bus) transfer errors. SCI link related errors are logged in the DP_ERRLOG fields notruup, flgerr, strperr[0:6], and fatalerr[0:6]. NIBus errors are logged in the DP_ERRLOG.tcode[0:7] field. Multiple errors can occur and will all be logged unless the DP_STATE.sngerr bit is set.

4.9.1 SCI Link Related Errors

Errors in the SCI link occur due to protocol or ringlet state errors. Some of these are fatal and cause the link to enter the "dead" state. Fatal errors are listed in Table 4.17. Going to "dead" on fatal errors can be prevented by setting the DP_STATE.stayrun bit. Others are not fatal and will be logged while the SCI link remains in the "running" state.

Any SCI link error will assert the node interface logic interrupt pin NINTRNI. It remains asserted until the DP_ERRLOG register is cleared by a CSR write.

Table 4.17: Fatal SCI Link Errors

Error	Bit	Description
notruup	15	Lost input SCI strobe signal.
echoUnkn	31	Received echo with no corresponding send packet in send queue.
acFail	30	Allocation count bit flipped more than once while node was blocking.
rspRsvErr	29	Retry packet received with wrong reservation phase for response queue.
reqRsvErr	28	Retry packet received with wrong reservation phase for request queue.
scrbLgTmr	27	No idle.lg bits set in ringlet detected by the scrubber.
noInSync	25	Lost input sync (elastic buffer).

Non-fatal SCI link related errors include errors detected at the input link and send queue related errors. Non-fatal input link errors are listed in Table 4.18. Multiple errors can occur and will be logged unless the DP_STATE.sngerr bit is set.

Table 4.18: Non-fatal Input Link Errors

Error	Bit	Description
flgerr	16	Received packet was not properly framed.
tooLong	26	Bypassed packet greater than 48 symbols. Packet is truncated to 48 and correctly framed. Outgoing CRC is stomped.
badIdle	18	Received idle with bad parity. Replaced with last good idle symbol.
badThruCrc	19	Bypassed packet had bad CRC. Outgoing CRC is stomped.
badStrpCrc	20	Stripped packet for receive queue had bad CRC. Packet tossed and echo CRC stomped.
strpTooLong	21	Stripped packet for receive queue greater than 48 symbols. Was tossed and echo CRC stomped.
tossClkStb	22	Received clockStrobe while still bypassing a clockStrobe. Packet tossed.
rspAcTmo	23	Allocation counter timed out (acTmr=4) on response Q reservation before all outstanding reservations completed.
reqAcTmo	24	Allocation counter timed out (acTmr=4) on request Q reservation before all outstanding reservations completed.

The send queue releated errors are of three types listed in Table 4.19. These errors are non-fatal and are indicated by bits in the appropriate DP_SENDQ_TAG registers listed in 4.8.5. Also given in the tag register is the valid bit for the slot indicating whether the packet is actually valid, plus the packet's tranId value allowing identification of the packet with the error.

If a send queue related error occurs on a packet in a queue slot, the packet will be held ("frozen") in the slot so that its DP_SENDQ_TAG register can be examined. This "freeze" indication is provided in the DP_ERRLOG.sqfrz[0:3] field. These send queue errors also assert the NINTRNI pin.

Table 4.19: Send Q Tag Error Flags

SendQ error	Bit	Description
sndQto	8	Send queue packet time out. Occurs when packet has waited 4 cc counts without receiving a matching echo.
sndQpar	7	Send queue parity error. Upon transmission from send queue if a parity error is detected on the packet data.
echoNo	6	Echo NONE status returned. If packet receives echo with NONE status indicating targetId addressed no node.

These errors are or'ed together to assert the NERRNI signal pin. This pin will remain asserted until all DP_ERRLOG tcode bits are cleared by a register write.

Transfer errors are summarized in Table 4.20.

Table 4.20: NI Transfer Errors

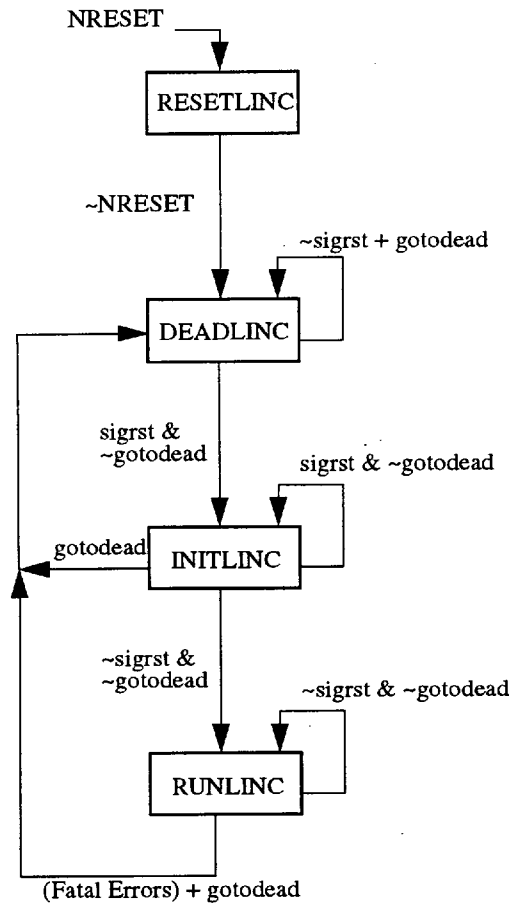
Error	Bit	Description
sendQfull	7	Tried to transfer to full send queue.
dupTranId	8	Send req packet has duplicate tranId to packet already in the req queue.
sizeErr	9	Node interface transfer length not multiple of 8 symbols or greater than max 48 symbols.
NIregPar	10	Parity error detected on register write transfer to DataPump.
NIsendqPar	11	Parity error detected on send queue transfer to DataPump.
rcvQempty	13	Tried to transfer from empty receive queue.
regAddrErr	14	Register offset address supplied from node interface on register read or write doesn't match any register.

4.10 DataPump Reset and Initialization

SCI auto-initialization as described in the IEEE SCI specification is not supported on this DataPump chip. Instead, a greatly simplified initialization scheme has been implemented which requires some software intelligence to start the linc. The DataPump sequences through four valid chip states described below:

- RESETLINC - A hard reset of all registers in the DataPump is performed (NRESET asserted).
- DEADLINC - Internal chip clocks are running and the DataPump is waiting for the upstream strobe to start. In this state, the state-machines in the linc portion of the chip is frozen. Valid idles are sent from the linc.
- INITLINC - In this state, the upstream strobe has been received, and we are waiting for upstream synchronization. The bypass FIFO has now been re-synchronized but is locked so that no bypass traffic is let through. The output unit sends valid sync and idle packets to the downstream linc. Also, valid and freeze bits are cleared to free up used queue slots (i.e. for a warm restart).
- RUNLINC - DataPump initialization sequence complete, ready for normal chip operation. The initialization flow Diagram for the VSC7201C is shown in fig. 4.10. The "~" used in the figure indicates the false value of the signal. The "&" means the logical AND and the "+" means the logical OR of the listed signals. The only signals required to cycle through initialization are the NRESET pin and the DP_STATE register bits sigrst and gotodead. However, the following sequence is recommended for reliable synchronization.

Figure 4.10 : Initialization Flow Diagram



741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

1. The NRESET pin should be asserted for at least 8 NICLK cycles then deasserted. The DataPump then proceeds to DEADLINC.
2. The DP_STATE.ruup bit is monitored which, when set, indicates that upstream clock is running. When ruup is detected, DP_STATE.sigrst is set. The DataPump state then advances from DEADLINC to INITLINC. Note that DP_STATE.gotodead was cleared by NRESET and should not be set or else the DataPump will remain in DEADLINC.
3. In the INITLINC state, software should monitor the DP_STATE.insync value which indicates that the upstream block is running and has detected a certain number of sync packets. Once DP_STATE.insync reaches value "11" (it is a 2 bit field and indicates three sync packets received), then DP_STATE.sigrst is de-asserted and the DataPump enters the RUNLINC state.

Setting of the NodeID values and scrubber selection should also be done during initialization. The DP_NODEID value can be set anytime after RESETLINC but MUST be done for all DataPumps in the ringlet before any packets are loaded into any send queues for transmission.

The ringlet should also have one and ONLY one scrubber selected (by asserting the NSCRUB pin) before entering the RUNLINC state.

The DP_STATE register contains many useful signals related to chip state, initialization, and error control (stayrun and gotodead). See 4.8.1 for a description of this register.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

5.0 Node Interface Bus (NIBus)

The DataPump communicates with the node controller via a bidirectional GTL interface called the NIBus. The DataPump informs the node controller when receive queues contain data or when send queues have available slots for transmission. The node controller can initiate various commands with the DataPump to read and write packets across this interface.

Signal names are prepended with an N or P indicating negative true or positive true signals.

5.1 Interface Protocol and Signal Description

The NIBus is a synchronous 64-bit data bus plus 8 bits of byte parity plus control signals. All signals are sampled or driven on the rising edge of CLKNI.

The bus master is assumed to be the node interface logic. The DataPump chip is a slave on this bus controlled by asserting chip select, NDPSEL, and a 3-bit command on the PCMND pins.

PCMND selected transfers of data can be from 2 to 12 sequential 64-bit data transfers for SCI packets and register reads and writes.

5.1.1 PDATA[0:63], PPARITY[0:7]

PDATA is bidirectional data bus for use in transferring data to and from the DataPump. PPARITY indicates byte parity on PDATA. Good parity is odd (all ones give parity of one).

PDATA and PPARITY are floated at the end of the transfer or one cycle after NDPSEL goes false and remain floated until NDPSEL is asserted.

PDATA[0:63] are also floated by the DataPump starting one cycle after PCMND is sampled, when it responds to a transfer command from the node interface to the DataPump (sendReq, sendResp, and regWrite). PDATA[32:63] are floated similarly when the DataPump responds to a regRead transfer.

5.1.2 NDPSEL- Datapump Chip Select

The NDPSEL input is used to chip select the DataPump and allow it to execute transfers and drive the data and control signals. The cycle in which NDPSEL is asserted, the DataPump will sample its PCMND inputs and begin executing commands.

If NDPSEL goes false during any clock cycle of a DataPump command, the DataPump will abort the command cleanly without an error condition. This is an acceptable protocol for aborting DataPump command sequences.

One cycle after NDPSEL goes false, all outputs will be floated except PRCVREQ, PRCVRSP, PSNDREQ, PSNDRSP, PCLKSTB, NERRNI, and NINTRNI.

5.1.3 PCMND[0:2] - Transfer Command

The PCMND inputs select the DataPump transfer command as listed in Table 5.1. The PCMND value is sampled in the cycle NDPSEL is asserted.

If NDPSEL remains asserted after a transfer has completed, the DataPump will wait until NDPSEL is toggled before sampling a new PCMND. Therefore, NDPSEL acts as a strobe for latching a new PCMND value

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Table 5.1: .PCMND Commands

Command	PCMND [0:2]	Description
rcvReq	000	Transfer from the receive-request queue (5-15 cycles).
rcvResp	001	Transfer from the receive-response queue (5-15 cycles).
regRead	010	Read contents of target register (4 cycles).
reserved	011	Reserved for future use.
sendReq	100	Transfer to the send-request queue (5-15 cycles).
sendResp	101	Transfer to the send-response queue (5-15 cycles).
regWrite	110	Write contents of target CSR (4 clock cycles, 64 bits max.).
reserved	111	Reserved for future use.

5.1.4 NNIACK - Node Interface Acknowledge

NNIACK is an input used to indicate valid end of transfer on send, receive, and register transfers.

Transfers from the node interface to the DataPump are acknowledged valid by asserting NNIACK simultaneously with the last data transfer cycle.

The packet size is determined by the DataPump based on receiving NNIACK with last data. The NNIRDY flow control can be used to delay NNIACK from the end of data if necessary.

Aborting a transfer from the node interface to the DataPump should be done with NDPSEL, not NNIACK.

Transfers from the DataPump to the node interface are acknowledged received as valid by asserting NNIACK one cycle after the last data transfer. If this acknowledge is provided, the DataPump will remove the packet from its queue after receiving NNIACK.

NNIACK false one cycle after the last data transfer indicates invalid transfer and abort without error condition. In this case, the DataPump will not remove the transferred packet from the receive queue.

5.1.5 NDPACK - DataPump Acknowledge

NDPACK is an output used to indicate valid end of transfer by the DataPump on send, receive, and register transfers.

Transfers from the DataPump to the Node Interface are acknowledged valid by asserting NDPACK in the same cycle as the last data transfer.

NDPACK false in this case indicates an error condition and the transfer should be disregarded. If the packet had a parity error, it will be discarded after the last cycle of the transfer (based on packet size). If NDPSEL was deasserted aborting the transfer before the end, the packet will not be removed from the queue regardless of parity error.

Data transfer length must be determined by the node interface by saving the "size" value from the header field in the packet and counting cycles. See Section 5.3.1 Condensed Request/Response Packets for more on the "size" value.

Transfers to the DataPump from the node interface are acknowledged valid by asserting NDPACK two cycles after the last data transfer cycle. The extra cycle is necessary for parity checking.

NDPACK false in this case indicates an error condition and that the transferred packet cannot be accepted and was not queued. See Section 4.9 for interface error conditions.

NDPACK is floated at the end of the transfer or starting one cycle after NDPSEL goes false and remains floated until NDPSEL is asserted.

5.1.6 NNIRDY - Node Interface Flow Control

NNIRDY is an input to the DataPump. Flow control is provided through NNIRDY.

NNIRDY applies to data transfers and NNIACK acknowledge of transfers in either direction. A false NNIRDY will cause the DataPump to hold its state in the next cycle. It will continue to hold until NNIRDY goes true.

The DataPump also qualifies PCMND sampling with NNIRDY.

5.1.7 PRCVREQ, PRCVRSP - Rcv Queue Flags

These receive queue flags are outputs from the DataPump indicating the presence of a received packet from the SCI link. These queue flags will not be asserted until the entire packet has been received and the CRC has checked without error.

The flags go false (queue empty) two cycles after receiving a receive queue transfer command if the packet being transferred is the last in the queue.

If the transfer doesn't complete (aborted or error detected) and the packet is not freed from the receive queue, the queue flag will be asserted again at the end of the transfer.

These receive flags are always driven and do not float when the DataPump is deselected.

5.1.8 PSNDREQ, PSNDRSP - Send Queue Flags

The send queue flags are outputs from the DataPump indicating at least one available queue slot in each queue type (request or response) when asserted true.

They will go false (indicating queue full) two cycles after receiving a send queue transfer command and if the queue will fill up as a result of the transfer. If the transfer doesn't complete and the packet isn't queued, the queue flag will go true again at the end of the transfer.

They will go true indicating that a queue slot has freed up only after receiving a normal echo packet from the target SCI node of the send packet which was freed from the queue.

The send queue flags are always driven and do not float when the DataPump is deselected.

5.1.9 NINTRNI - Interrupt Node Interface

NINTRNI is an output of the DataPump and is asserted whenever an error occurs synchronous to the SCI link. These errors are logged in the DP_ERRLOG register not including the tcode error field. NINTRNI remains asserted until the DP_ERRLOG is cleared.

NINTRNI is always driven and is not floated when the DataPump is deselected.

5.1.10 NERRNI - NIBus Error Flag

NERRNI is an output of the DataPump which is asserted whenever an NIBus transfer error occurs. These errors are listed in Section 4.9.

NERRNI remains asserted until the DP_ERRLOG tcode field is cleared.

5.1.11 PCLKSTB - Clock Strobe

PCLKSTB is an output of the DataPump and is asserted in response to an event00 packet and whether the DataPump is the clockStrobe master or slave.

The DataPump is the clockStrobe master when the event00 packet is loaded into its send request queue for transmission. This packet should be targeted to the clock strobe master's DataPump targetId address so the event00 makes a complete trip around the ringlet and is stripped and tossed by the clockStrobe master.

Other DataPumps in the ringlet which pass the event00 through are clockStrobe slaves.

The clockStrobe master will assert PCLKSTB when the event00 is transmitted on its output link and will deassert PCLKSTB when the packet is stripped at its input link.

The clockStrobe slave will assert PCLKSTB for 13 CLKNI cycles when it receives and bypasses the event00 packet. The

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

slave will also start the DP_CLKTHRU register to count 4ns SCI clock cycles when the event00 is received and will stop the counter when the event00 is transmitted on its output link.

PCLKSTB is always driven and is not floated when the DataPump is deselected.

5.1.12 NRESET

NRESET is an input to the DataPump. Reset initialization of the DataPump occurs when the NRESET pin is driven from low to high. This rising edge transition must be synchronous to the NIBus clock, CLKNI. NRESET must be asserted for at least 8 CLKNI cycles.

For a description of reset initialization see sections 4.10 DataPump Reset and 5.7 NIBus Reset and State Sync to the SCI Link.

5.1.13 NSYNCRQ - Send Sync Packet Request

The NSYNCRQ pin is used to cause the DataPump to transmit a SYNC packet on its output. A SYNC packet will be scheduled for transmission if a one to zero transition is registered on the NSYNCRQ pin. Additional SYNC packets will only be scheduled after the first is transmitted and if, after that transmission, another one to zero transition is registered on the NSYNCRQ pin.

Therefore, NSYNCRQ is intended to be asserted on the order of milliseconds in periodicity. Since the DataPump does not implement any pin-to-pin deskewing, no SYNC packets are required periodically and no NSYNCRQ assertions are required.

5.1.14 NSCRUB - Scrubber Selection

The NSCRUB pin is used to select the DataPump as the scrubber in an SCI ringlet. For a description of scrubber selection and initialization see Section 4.10.

For information on scrubber operation see the SCI standard Section 3.9.2 Scrubber maintenance.

5.1.15 PPC[0,1] - Performance Counters

The PPC output pins provide performance counter signals for monitoring internal events in the DataPump. See Section 8.0 Performance Counters for details.

These outputs are synchronous to the internal 250 MHz SCI clock (however, maximum toggle frequency is 66MHz). They are not synchronized in the DataPump to the NIBus clock, CLKNI.

5.1.16 CLKNI - Node Interface Clock

CLKNI is the interface clock supplied by the node interface logic. All NIBus signals are synchronous with this clock.

5.1.17 CLKHI, PCKHSEL, PCLK250, PCKHMPY[0:1], SCI Link Clock and Control

CLKHI (and NCLKHI) is the master differential clock input for generating the internal 250Mhz SCI clock. It is not required to be phase or frequency related to the node interface clock, CLKNI.

The DataPump has an onboard PLL for frequency multiplication to generate 250MHz. The CLKHI reference frequency for the PLL is 100MHz.

The internal SCI clock is intended to be 250MHz but can be programmed to different values using the PCKHSEL, PCLK250, and PCKHMPY[0:1] inputs for debug and diagnostic purposes. The values higher than 250MHz are not supported.

PCKHSEL is a PLL bypass control input. When asserted the PLL is bypassed and the internal SCI clock is connected directly to the CLKHI input. Any frequency up to 250MHz can be provided in bypass mode.

PCLK250 and PCKHMPY[0:1] provide divisions from the PLL generated 500MHz value. The divide options and available internal SCI clock rates are given in Table 5.2. The values higher than 250MHz are documented for completeness but are not supported.

Table 5.2: SCI Clock Values

PCLK250	PCKHMPY [0:1]	Divide Value	SCI clk (MHz)
1	10	1	500
1	00	2	250
1	01	4	125

5.2 Transfer Packet Formats

The packet formats for transfers between the DataPump and node interface differ slightly from the sequential SCI packet format mainly for ease of data alignment to the 64-bit interface. These formats define the sequence of transfer data pertaining to a particular PCMND request from the node interface logic.

5.2.1 Condensed Request/Response Packets

There are four condensed transfer packet formats corresponding to the PCMND[0:2] transfer command requests- rcvReq, rcvResp, sendReq, sendResp. These condensed packet formats are shown in figures 5.2 - 5.5.

Send Queue transfer packets are always transferred from the node interface logic to the DataPump sequentially from header (octlet 0, figs. 5.2, 5.4) to last octlet. Receive-request queue packet transfers from the DataPump to the node interface logic are also always sequential from header to tail.

The header information has been condensed to the lower 48-bits of the PDATA bus so that if the node interface logic is split into separate controller and datapath elements the controller need only connect to those 48-bits.

In both send packets, only the targetId of the destination node is required since the sourceId is provided by the DataPump and cannot be changed.

In the receive packets, the targetId is provided in the most significant 16-bits as a matter of completeness, but this value will always be the nodeId of the DataPump.

The control symbol contains the fields tranId, tpr, trc, todExp, and todMant. Only the tranId value is really required and used by the DataPump. The trace bit, trc, tpr, time of death count, todExp and todMant, are provided for completeness and are passed but not used by the DataPump.

Finally, the packet size, "size", in pairs of octlets (16-bytes) is given in the header of the receive packets to allow the node interface to determine packet length for the transfer.

This size value is not required on the send side since the DataPump computes packet length based on when the node interface finishes the send queue transfer with NNIACK asserted.

5.2.2 Register Read and Write Formats

Registers on the DataPump are accessed using register read and write CMND transfer commands.

On regRead commands, the most-significant 32-bits of the data bus, PDATA[0:31], are driven by the DataPump with the contents of the register specified by the 12-bit register address offset supplied on PDATA[36:47]. In this case, the DataPump floats its PDATA[32:63] pins to allow the node interface logic to drive register address offset on PDATA[36:47]. Of course PDATA[32:35], PDATA[48:63], and PPARITY[6:7] must be driven to ones by the node interface logic to give correct parity.

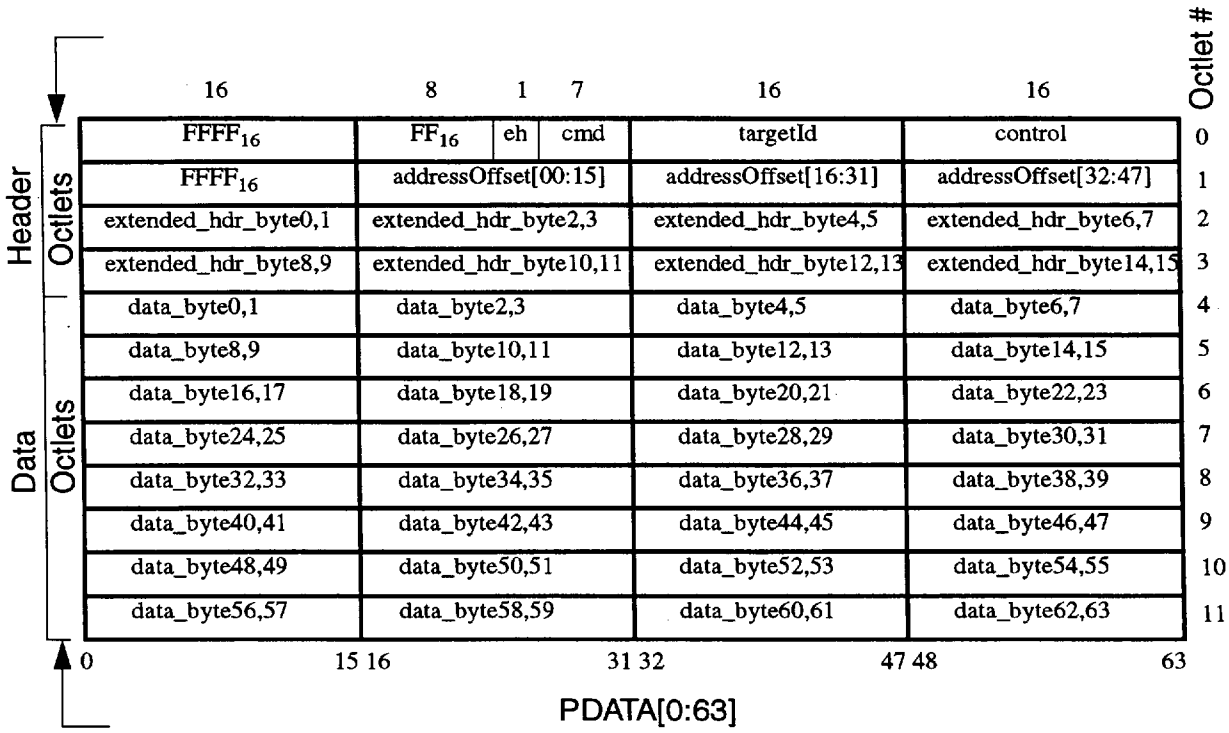
On regWrite commands, PDATA[0:31] specify the write data and PDATA[36:47] specify the register address offset. Again PDATA[32:35], PDATA[48:63], and PPARITY[6:7] must be driven to ones by the node interface logic to give correct parity.

The effects of the write vary depending on the register. See Section 4.8 Control and Status Support Registers.

The data transfer formats for register commands are summarized in Figure 5.6.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.2 : Send-Request Condensed Transfer Packet Format



Field Descriptions:

targetId - target node ID;

eh - extended header flag from control symbol;
cmd - command field from control symbol;

control - control symbol, tranId (required), tpr, trc, todExp, todMant (all optional);

addressOffset[00:47] - 48-bit address;

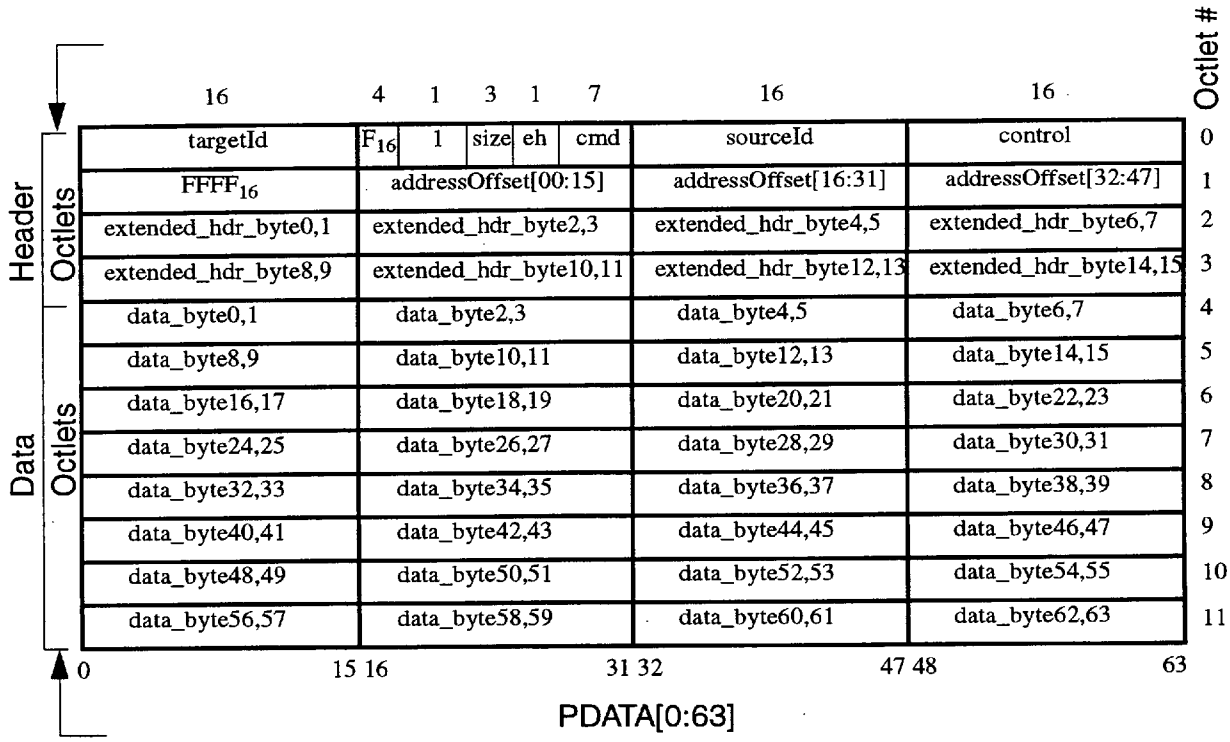
extended_hdr - either 0 or 16 bytes depending on value of eh flag bit

data_byte0-63 - data packet either 0, 16, 32, 48, or 64 bytes

Fields set to ones (e.g. $FFFF_{16}$) are specified to give known parity. These fields are reserved.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.3 : Receive Request Condensed Transfer Packet Format



Field Descriptions:

targetId - target node ID, i.e. this node;

size - size of complete packet in pairs of octlets, from 1 to 6;
eh - extended header flag from control symbol;
cmd - command field from control symbol;
sourceId - node ID of producer of this packet;
control - control symbol, tranId (required), tpr, trc, todExp, todMant (all optional);

addressOffset[00:47] - 48-bit address;

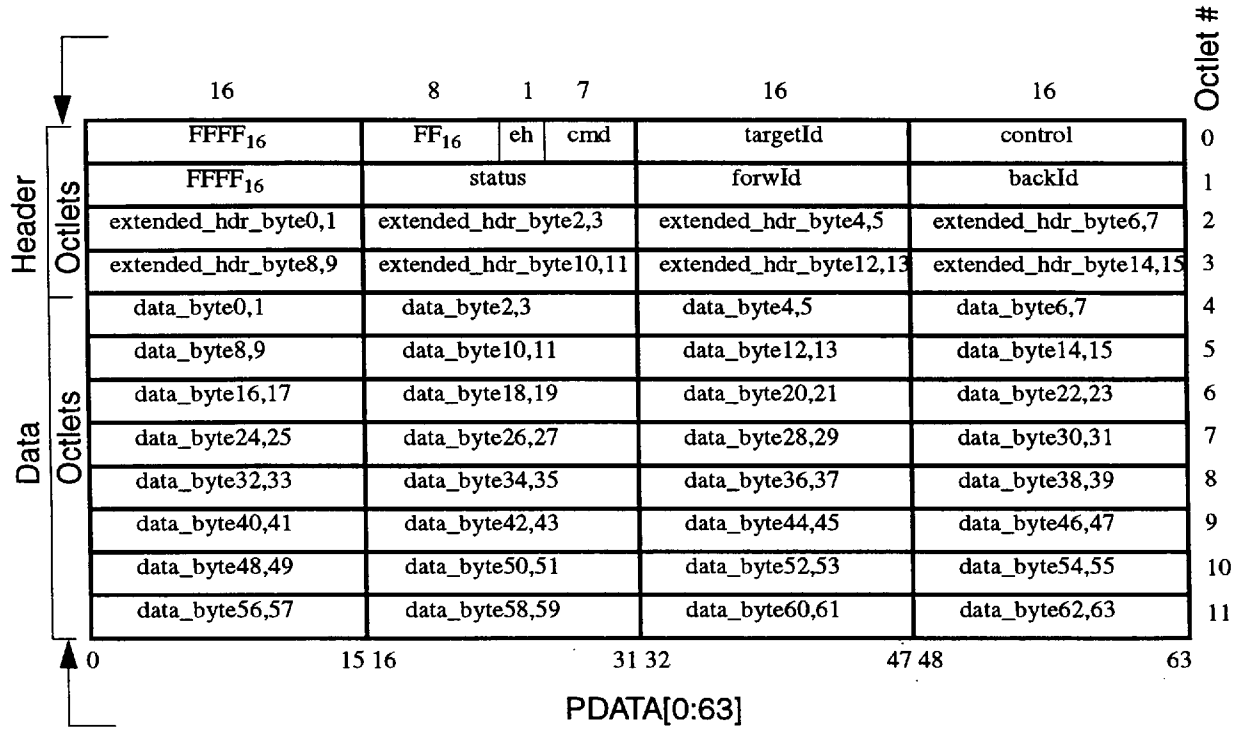
extended_hdr - either 0 or 16 bytes depending on value of eh flag bit

data_byte0-63 - data packet either 0, 16, 32, 48, or 64 bytes

Fields set to ones (e.g. $FFFF_{16}$) are specified to give known parity. These fields are reserved.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.4 : Send-Response Condensed Transfer Packet Format



Field Descriptions:

targetId - target node ID;

eh - extended header flag from control symbol;
cmd - command field from control symbol;

control - control symbol, tranId (required), tpr, trc, todExp, todMant (all optional);
status - response packet status;
forwId, backId - for cache coherency control; transmitted but not used by DataPump

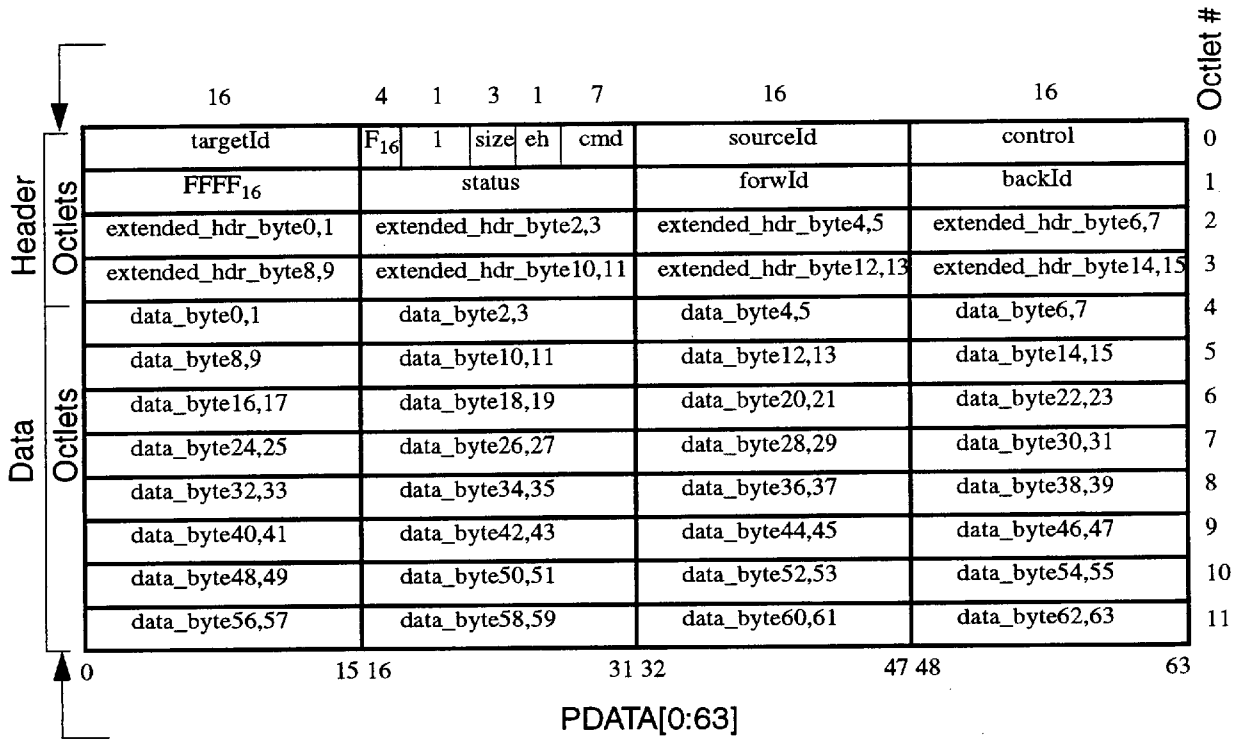
extended_hdr - either 0 or 16 bytes depending on value of eh flag bit

data_byte0-63 - data packet either 0, 16, 32, 48, or 64 bytes

Fields set to ones (e.g. $FFFF_{16}$) are specified to give known parity. These fields are reserved.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.5 : Receive-Response Condensed Transfer Packet Format



Field Descriptions:

targetId - target node ID, i.e. this node;

size - size of complete packet in pairs of octlets, from 1 to 6;
eh - extended header flag from control symbol;
cmd - command field from control symbol;
sourceId - node ID of producer of this packet;
control - control symbol, tranId (required), tpr, trc, todExp, todMant (all optional);
status - response packet status;
forwId, backId - for cache coherency control; transmitted but not used by DataPump

extended_hdr - either 0 or 16 bytes depending on value of eh flag bit

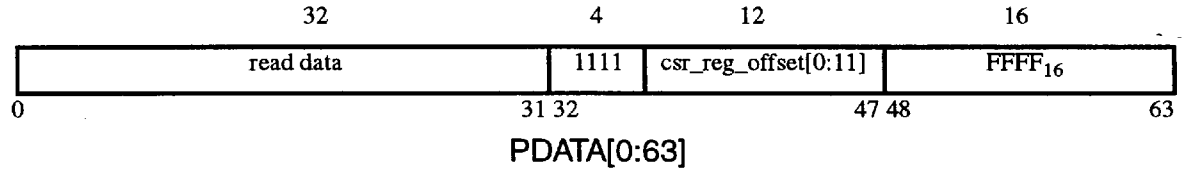
data_byte0-63 - data packet either 0, 16, or 64 bytes

Fields set to ones (e.g. $FFFF_{16}$) are specified to give known parity. These fields are reserved.

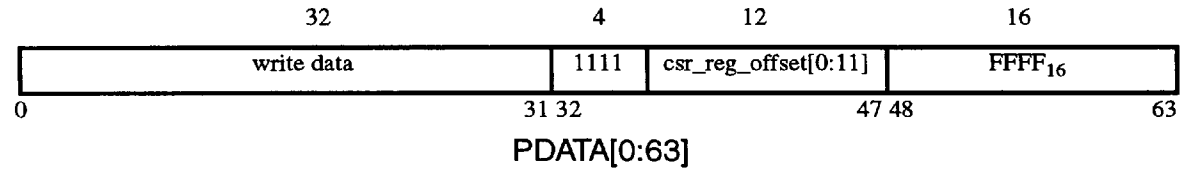
741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.6 : CSR Transfer Formats

Register Address Transfer Format:



Register Read/Write Transfer Format:



csr_reg_offset[0:11]:

meaning: Register offset address driven from the node interface logic on both register reads and writes.

read data:

meaning: Register read data driven by DataPump.

write data:

meaning: Register write data driven by node interface logic.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

5.2.3 32-Byte Line Packet Support

SCI provides the capability for sending a 32-byte payload in a 64-byte packet. This does not mean that the 64-byte packet contains a first 32-bytes which is meaningful and another 32-bytes of padding out to 64-bytes.

The payload of a 64-byte command can in fact be any multiple of 16-bytes and still be SCI compliant. The DataPump supports these payload sizes.

To send 32-byte packets (can be any of the coherent or non-coherent commands) the node interface logic simply loads the packet into the send queue and indicates the end of the data transfer with the NNIACK signal. The DataPump will check for legal payload size (0, 16, 32, 48, or 64 bytes) and send the packet normally.

To receive 32-byte packets, the node interface logic unloads the receive queue and determines data length by looking at the "size" field in the first 64-bit transfer.

If 32-byte lines are used, one of the addressOffset bits should be allocated to select one of two 32-byte lines within a 64-byte data size.

5.3 Send Queue SCI Transmission Order

Once either of the send queues has been loaded with a packet, the DataPump is responsible for getting the packet sent out on the SCI link. Each request and response queue has up to two packet entries, which may be unsent. The queues may also contain packets which have been sent but have not been received as acknowledged by an echo with DONE or NONE (i.e. non-busy) returned phase.

To determine the order of transmission of the queue entries, the DataPump keeps three state bits and an age bit for each request and response queue entry. Table 5.3 summarizes these bits.

The highest transmission priority for unsent packets is the one with tryReserve set. The selection of request or response queues is done in alternating order so that both queues are serviced equally. The next priority of ordering for transmission is based on packetAge. For more information on send packet transmissions and flow control see 4.6 Bandwidth Allocation and 4.7 Queue Allocation.

Table 5.3: Send Queue State Tags

Tag Bit	Description
ready	After a packet is entered into the send queue by the node interface logic, it is ready for transmission with phase NOTRY.
tryReserve	One packet from each queue is allowed to be transmitted with DOTRY phase in order to reserve receive queue space. If no packet has tryReserve set, the oldest packet based on packetAge will get tryReserve set. TryReserve remains set on a packet until the packet is removed from the queue.
packetAge	Each packet entry will have an age which gets updated whenever packets are removed from the queue or when new packets are loaded by the node interface logic.
check	This flag indicates packet has been sent and is waiting for echo.
freeze	This flag indicates the packet has experienced either a parity error on transmission, has timed out waiting for echo, or received echo with NONE status. See 4.9.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

5.4 Receive Queue Transfer Order

Data transfers from the DataPump receive queues to the node controller can start if one of the queue flags is true (PRCVREQ, PRCVRSP), after the node controller has selected the DataPump for receive packet transfer.

If more than one packet is in the queues, the order which the DataPump will unload them in is in the order they were received on the SCI link for that packet type, request or response.

Since the order packets are received from SCI nodes cannot be guaranteed, the node interface logic must check the tranId of the packet to determine which outstanding request it is for.

5.5 NIBus State Machine

The basic flow diagram for the NIBus is shown in Figure 5.7. This is an approximation to the actual logic and should be used for general understanding only. Refer to the bus cycles in figures 5.8 to 5.25 for actual signal transitions.

The DataPump starts in the IDLE state and waits for NDPSEL falling edge and NNIRDY true. Once this is true, the PCMND value is sampled and the DataPump enters the CMD state.

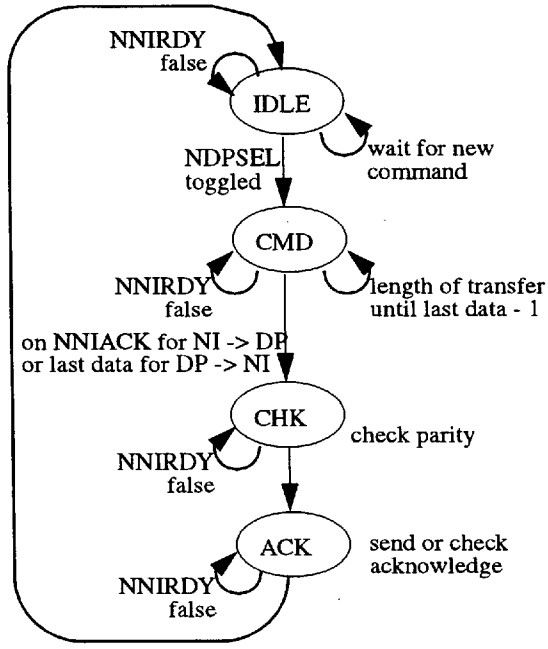
In the CMD state, data is either read out of or written into registers or queues up to but not including the last data transfer. While in the CMD state, NNIRDY is sampled every cycle and if asserted false, the state will freeze the next cycle.

Also, if NDPSEL is de-asserted, the DataPump goes back to IDLE and aborts the transfer without error. No packets are affected and no configuration registers are written.

The parity check state, CHK, is entered on receiving NNIACK for node interface to DataPump transfers or when driving the last data on receive queue transfers or on reading configuration registers on regRead transfers.

The ACK state is always entered after CHK if NNIRDY is true. In ACK the DataPump will drive NDPACK on node interface to DataPump transfers and check NNIACK on DataPump to node interface transfers.Finally, ACK goes to IDLE if NNIRDY was true.

Figure 5.7 : Basic NIBus Flow Diagram



5.6 Bus Cycles and Use of Signals

Send queue transfers from the node interface to the DataPump are shown in figures 5.8 through 5.17. While these examples show accesses to the send request queue, the transfers are identical for the send response queue.

Also, bus cycles with wait states using NNIRDY and aborts using NDPSEL are shown. These examples are applicable to receive queue and register transfers also.

Receive queue transfers are shown in figures 5.18 through 5.21. Register reads and writes are shown in figures 5.22 through 5.25.

These are not the complete set of bus cycle possibilities but indicate the operation of all the signals for all the various operations.

5.6.1 Send Queue Transfers and Use of NDPSEL

Figures 5.8 and 5.9 illustrate basic send queue transfer bus cycles plus the use of NDPSEL to initiate transfer commands.

Figure 5.8 specifically shows a send request queue packet transfer to the DataPump. Firstly, the queue flag PSNDREQ goes true indicating a slot has opened up in the send request queue for a new packet.

The node interface logic can then initiate a sendReq packet transfer to that queue. It does so in cycle T2 by driving NDPSEL from high to low while supplying the command code for sendReq, 100_2, on the PCMND pins. The DataPump transitions from the IDLE state to the CMD state.

The transfer commences (CMD state) as long as NDPSEL remains low up until NNIACK is asserted indicating end of data. This is done in cycle T7. The DataPump then completes checking the packet (CHK state) for correct size and length and correct parity.

These checks are completed and NDPACK is asserted (ACK state) in cycle T8 indicating successful transfer and the packet is validated in the send queue.

NDPSEL is allowed to transition from low to high in the acknowledge cycle, ACK. If it remains low, as shown in Figure 5.9 in cycle T9, the DataPump remains in the IDLE state and will not accept new commands.

NDPSEL must be driven high for a cycle then low again simultaneously with a new command, as shown in Figure 5.8 cycle T10 or Figure 5.9 cycle T11, to start a new transfer.

5.6.2 Using NNIRDY to Add Wait Cycles

NNIRDY false freezes the state of the DataPump. This is illustrated in figures 5.10 through 5.13.

While these diagrams are shown for sendReq transfers, the same behavior relative to NNIRDY occurs for all transfers.

Figures 5.10 and 5.11 show wait cycles in the CMD state. Figure 5.12 shows a wait cycle in the CHK state. Figure 5.13 show a wait cycle in the ACK state.

5.6.3 Using NDPESL to Abort Transfers

Figures 5.14, 5.15, 5.16, and 5.20 illustrate using NDPSEL to abort a transfer command before it is finished.

Aborting a send queue transfer in the CMD or CHK states will cause the DataPump not to enter the packet into the send queue and sends it into the IDLE state.

Aborting a register write in the CMD or CHK states will cause the DataPump not to write the selected register and sends it into the IDLE state.

Aborting a register read in the CMD or CHK states causes the DataPump to go into IDLE and not drive out the read data.

Figure 5.14 show aborting a send queue transfer during the CMD state. The cycle in which NDPSEL goes high, T7, the DataPump goes to the IDLE state.

Figure 5.15 shows aborting during the CHK state. NDPSEL goes high in T8 and the DataPump goes to the IDLE state.

Aborting a send queue transfer, a register write, or a register read during the ACK state has the effect of sending the DataPump into the IDLE state regardless of the value of NNIRDY. This is shown in Figure 5.16.

Whenever NDPSEL goes high on send queue transfers or register reads or writes, NNIRDY and NNIACK are not sampled and are don't cares.

Aborting a receive queue transfer in the CMD, CHK, or ACK states causes the DataPump not to remove the packet being transferred from the receive queue and to enter the IDLE state. The abort from the ACK state is shown in Figure 5.20.

In the case of receive queue transfers, NNIRDY and NNIACK must be true in the ACK cycle in order to remove the packet just

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

transferred from the receive queue upon entering the IDLE state. This is shown in Figure 5.18, cycle T8. NDPSEL may be going high at this time, as shown.

5.6.4 Transfer Errors Indicated by NDPACK

Errors in the transfer detected by the DataPump will be indicated by not asserting NDPACK when it is supposed to be asserted. This is shown in Figure 17 for send queue transfers, Figure 21 for receive queue transfers, Figure 23 for register writes, and Figure 25 for register reads.

In each of these cases, NDPACK has a particular cycle in which it should be asserted and which the node interface logic can determine before hand.

In the case of send queue transfers, NDPACK should occur in the ACK cycle as shown in cycle T8 of Figure 5.17. This location may move out depending on wait cycles inserted by NNIRDY false.

In the case of the receive queue transfers, NDPACK should be asserted with the last data transfer. This cycle is determined by the node interface logic by counting data transfer cycles until the "size" value is reached. "size" is the packet size in octlet pairs given in the header octlet (i.e. the first octlet transferred, see figures 5.3 and 5.5). Figure 5.21 shows the error indication for receive queue transfers.

Register transfers always have the same number of cycles (if NNIRDY always true) and NDPACK should always be asserted in the appropriate cycle. Figures 5.23 and 5.25 show the cycle timing for error conditions in the register reads and writes.

5.6.5 Receive Queue Transfers

A basic receive queue transfer from the DataPump to the node interface logic is shown in Figure 5.18. This example shows the receive request queue going from empty to having received one packet in cycle T1 as indicated by PRCVREQ. The node interface responds with a rcvReq transfer command in T2. The PRCVREQ queue flag goes false again when the DataPump begins the transfer. A packet with 16-bytes of data is transferred and ends successfully in cycle T8. The node interface logic could issue a new command as soon as T9.

5.6.6 Register Read and Write Bus Cycles

A basic Register write transfer from the node interface to the DataPump is shown in Figure 5.22. The node interface sends the regWrite command in cycle T2 and write data on PDATA[0:31] (writes to registers only use the most significant 32-bits) and write address on PDATA[36:47] in cycle T3. The node interface must also drive the other bits in PDATA[32:63] and correct parity on PPARITY[4:7] to ensure that a false parity error doesn't occur. The DataPump acknowledges the successful write completion in cycle T5. A new command could be issued in T6.

A basic register read transfer is shown in Figure 5.24. The node interface sends the regRead command in cycle T2 and read address on PDATA[36:47] in cycle T3. The DataPump sends read data back on PDATA[0:31] in cycle T5. A new command could be issued in T6.

Note that the PDATA bus direction is split into different directions on the upper 32-bits and lower 32-bits in this transfer. This is the only transfer sequence where this happens

Also note that when the node interface drives address on PDATA[36:47] it must also drive all other bits in PDATA[32:63] and correct parity on PPARITY[4:7] or else a false parity error might be detected.

5.7 NIBus Reset and State Sync to the SCI Link

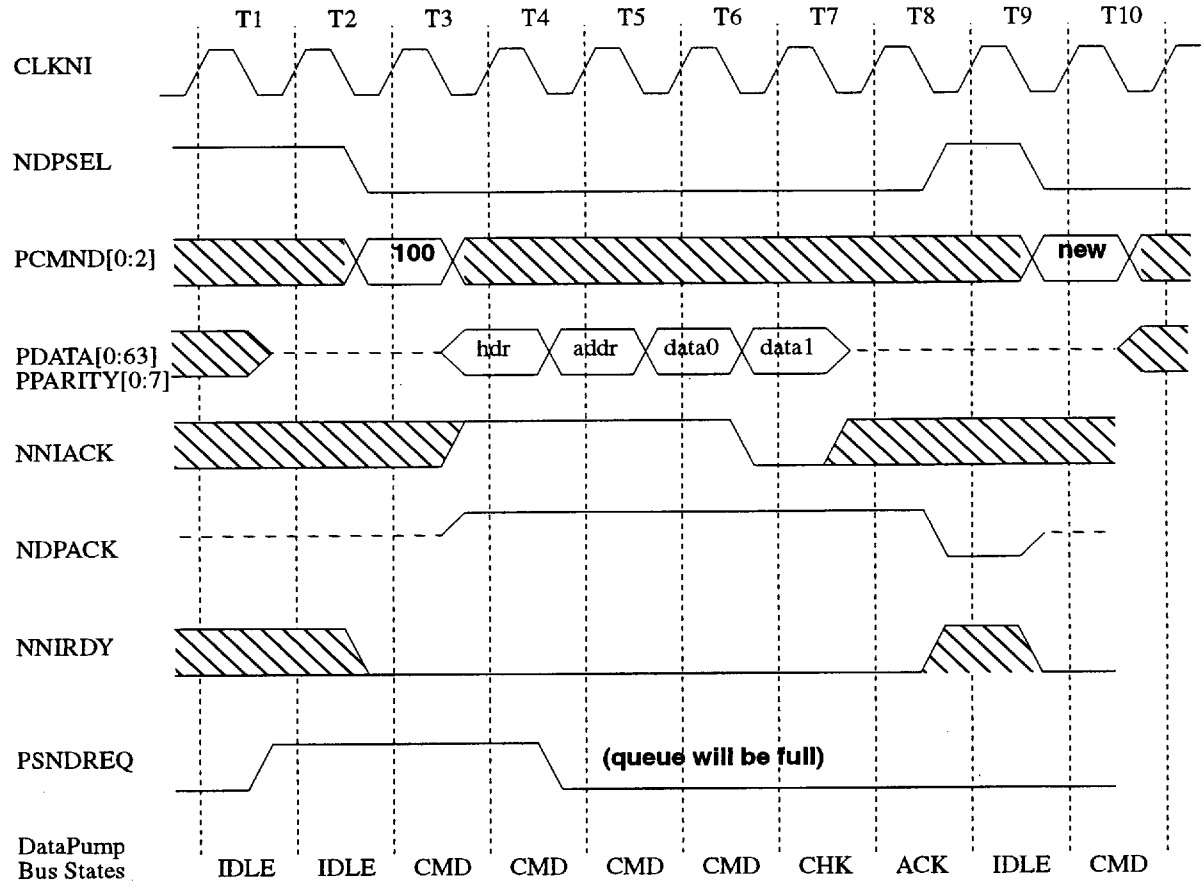
After toggling NRESET the NIBus state machine immediately enters the IDLE state, aborting any transfer it might have been in. At this point, new commands will be accepted from the node interface logic. Access to the queues and registers is not blocked, so the state of the SCI link should be checked before packets are put into the send queue (see 4.10, 4.11, and 4.12 sections on link initialization).

5.7.1 Node Interface Transfer Errors

Errors which occur directly related to transfers or protocol between the DataPump and node interface logic are logged by the DP_ERRLOG tcode field.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.8 : SendReq Queue Transfer Initiated by PSNDREQ Flag



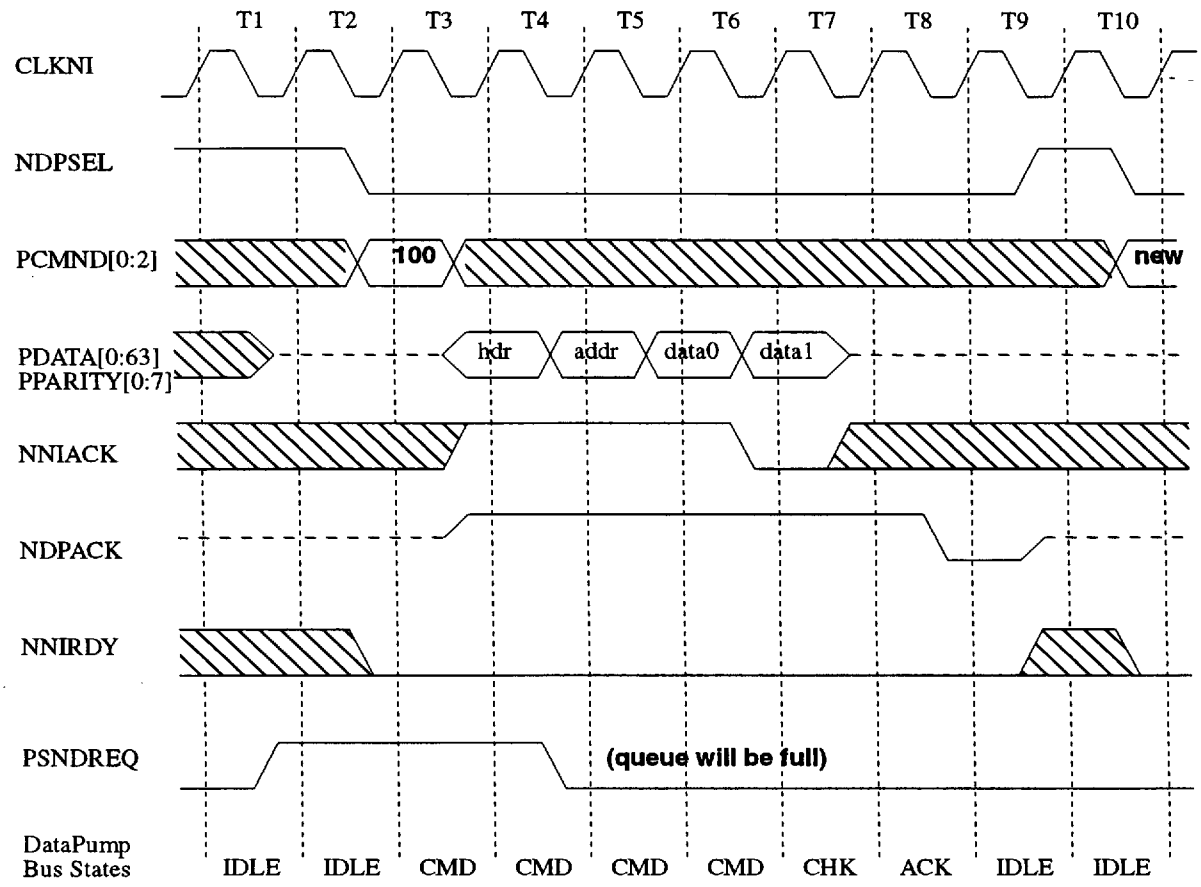
This diagram shows a normal send packet transfer to the DataPump. In this case, the packet contains the required header information and 16 bytes of data. Larger data payloads simply extend the number of cycles. NNIACK is used to indicate the end of transfer.

The DataPump never drives PDATA or PPARITY during a send queue transfer.

This diagram illustrates minimum cycle timing of all signals with no hold cycles.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.9 : SendReq Queue Transfer Followed by Extra IDLE Cycle Due to NDPSEL Low

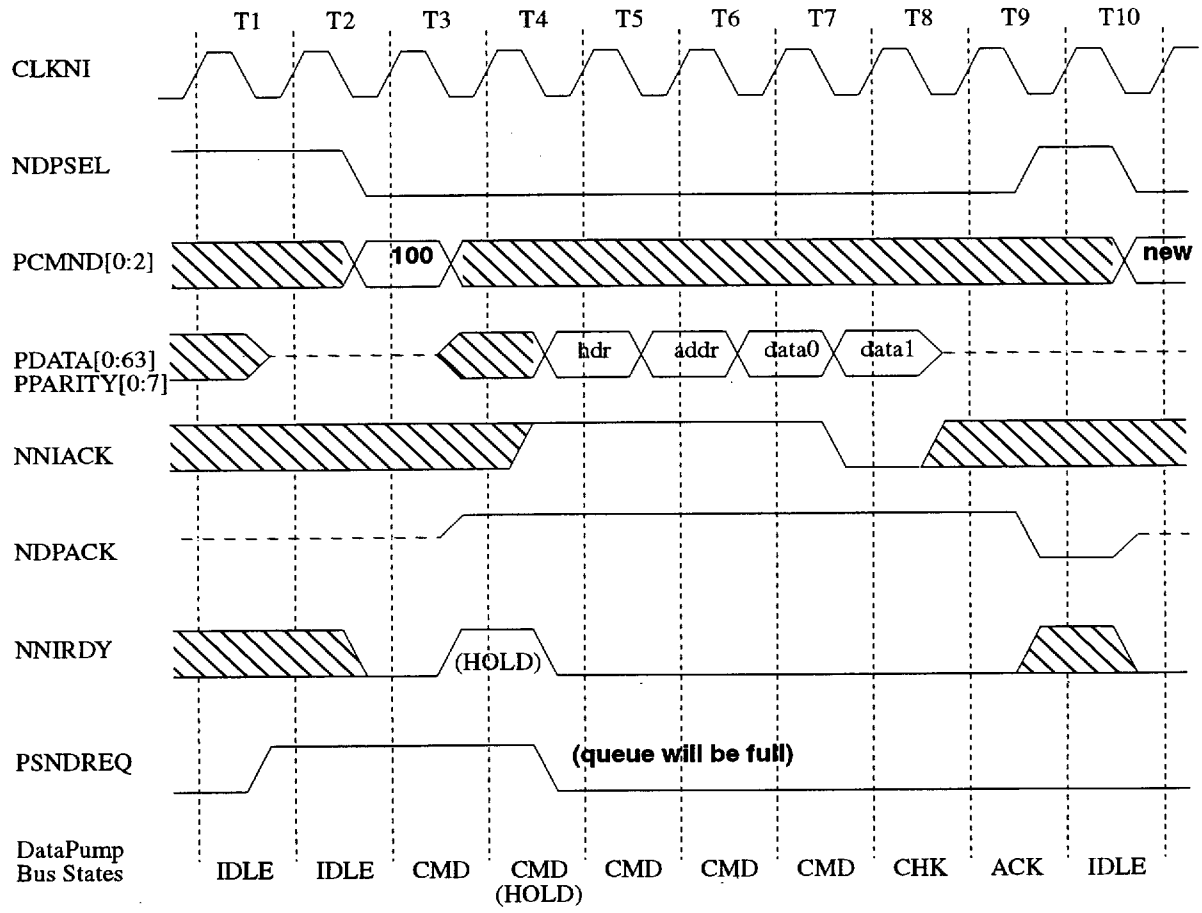


This diagram illustrates the end of cycle transition to the IDLE state in cycle T9. The DataPump remains in IDLE until NDPSEL makes a high to low transition as shown in T10. In T11 the DataPump enters the CMD state as a result of NDPSEL transitioning from high to low in T10 and meeting input low set-up time before the transition to T11.

In order to load in a new command, NDPSEL must be driven high and low again as shown in this diagram. Otherwise, the DataPump remains in the IDLE state.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.10 : SendReq Queue Transfer with Hold on First Data Transfer

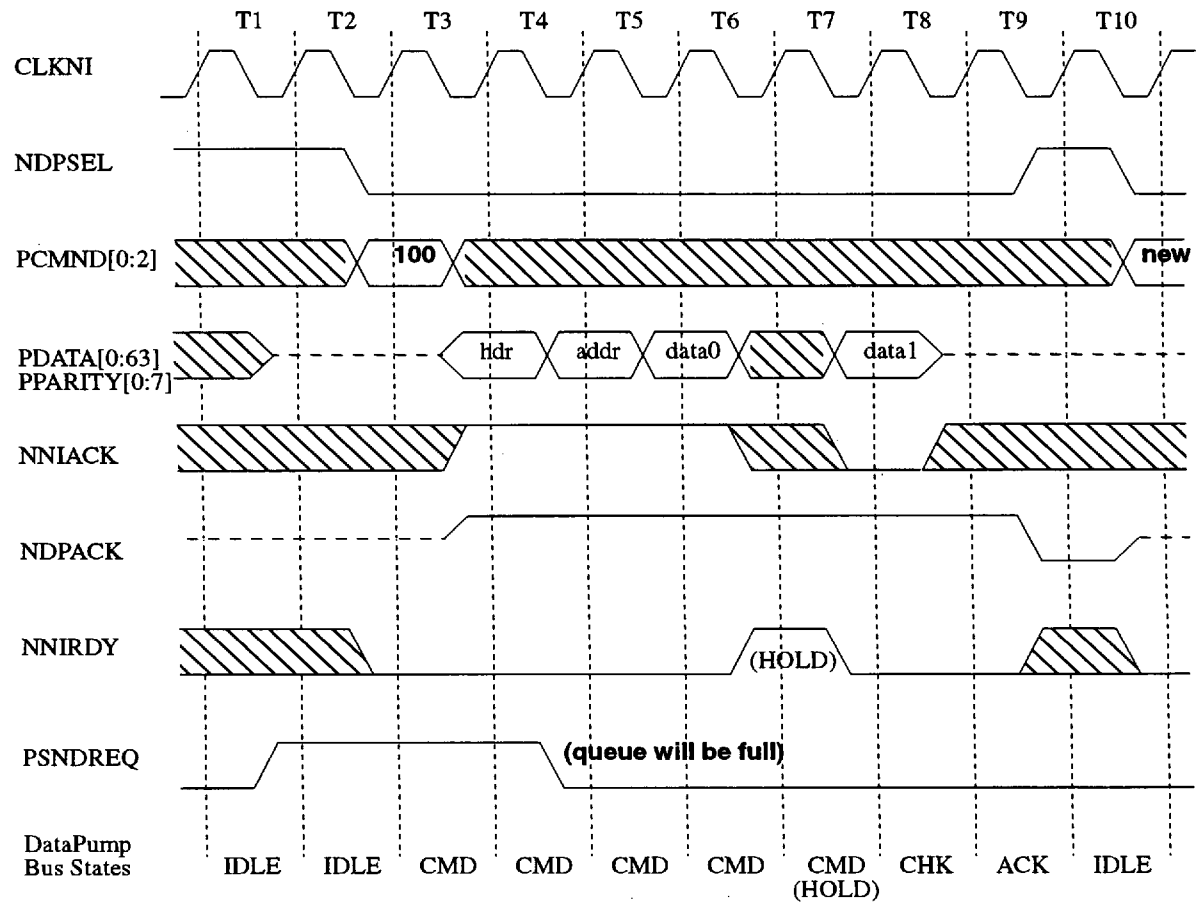


This diagram illustrates the use of NNIRDY for flow control on providing data to the DataPump. NNIRDY false in the cycle first data is expected, T4, will cause the DataPump to hold for as many cycles as it remains false. In the cycle NNIRDY goes true again, T5, the DataPump begins loading the first data, i.e. "hdr".

Of course, NNIRDY false can be used to hold any cycle during the transfer.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.11 : SendReq Queue Transfer with Hold on Last Data Transfer



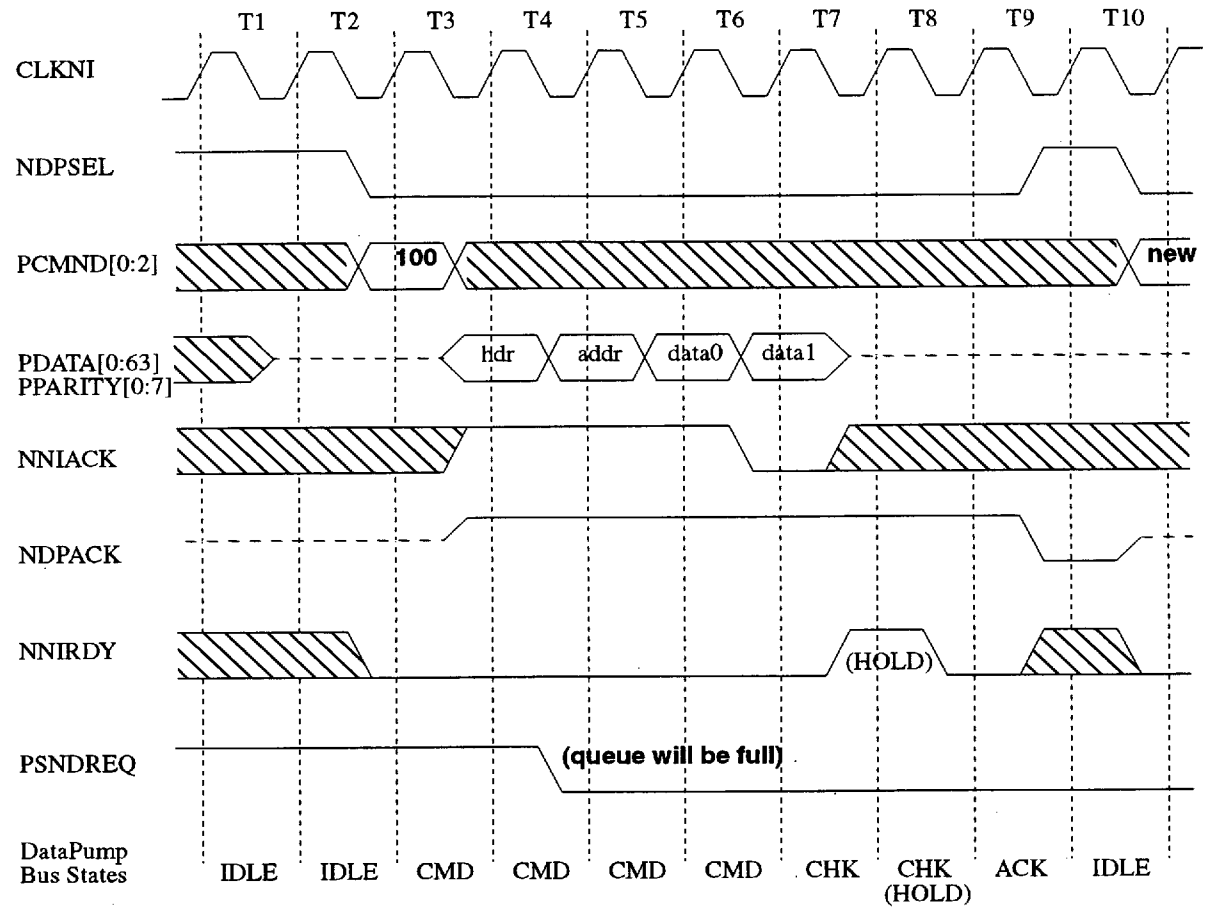
This diagram illustrates the use of NNIRDY to delay providing the last piece of data of the packet. NNIRDY false in T7 causes the DataPump to hold that cycle. Once NNIRDY goes true again in T8 the DataPump expects to see last data in T8 along with NNIACK.

This is a way to delay the assertion of NNIACK at the end of a packet if more time is needed in the node interface logic to compute NNIACK.

Of course, NNIRDY false can be used to hold any cycle during the transfer.

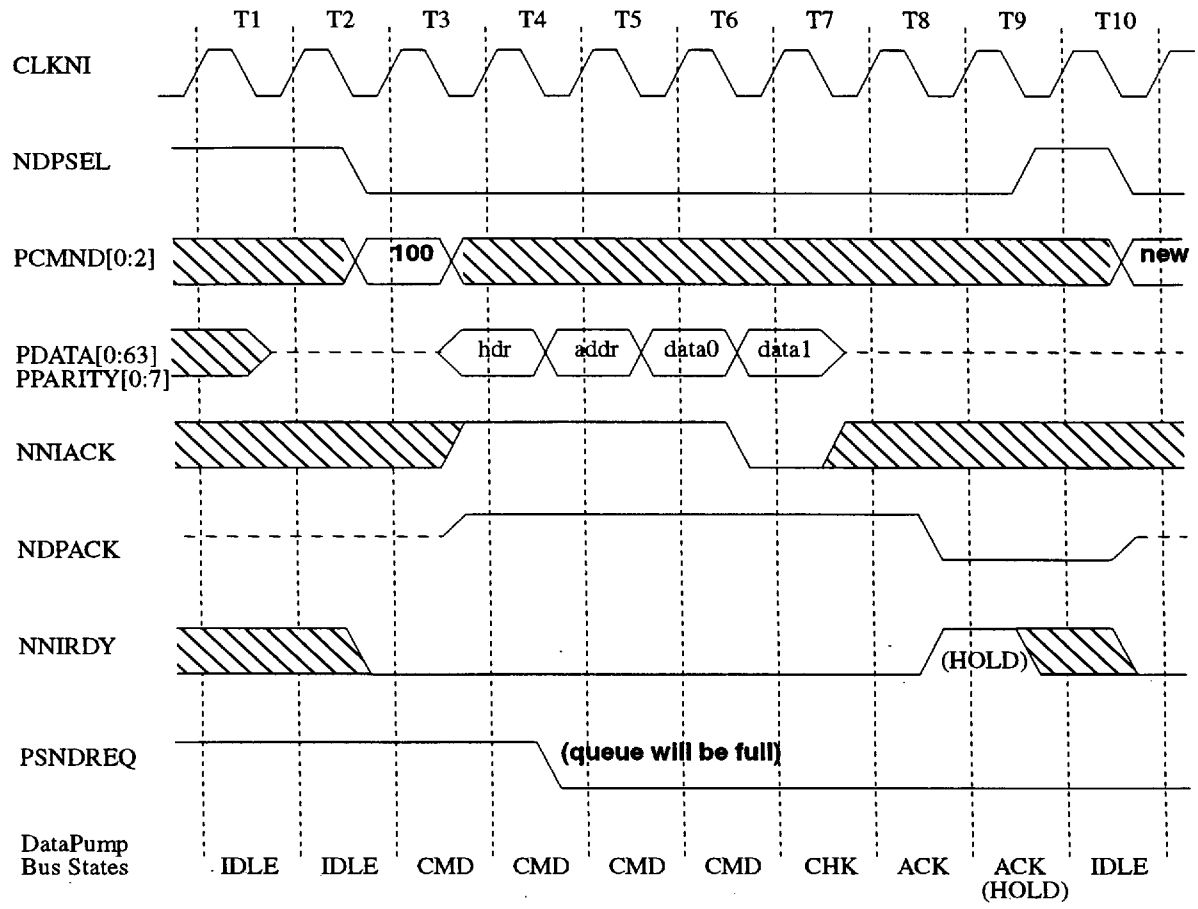
741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.12 : SendReq Queue Transfer with Hold Before NDPACK.



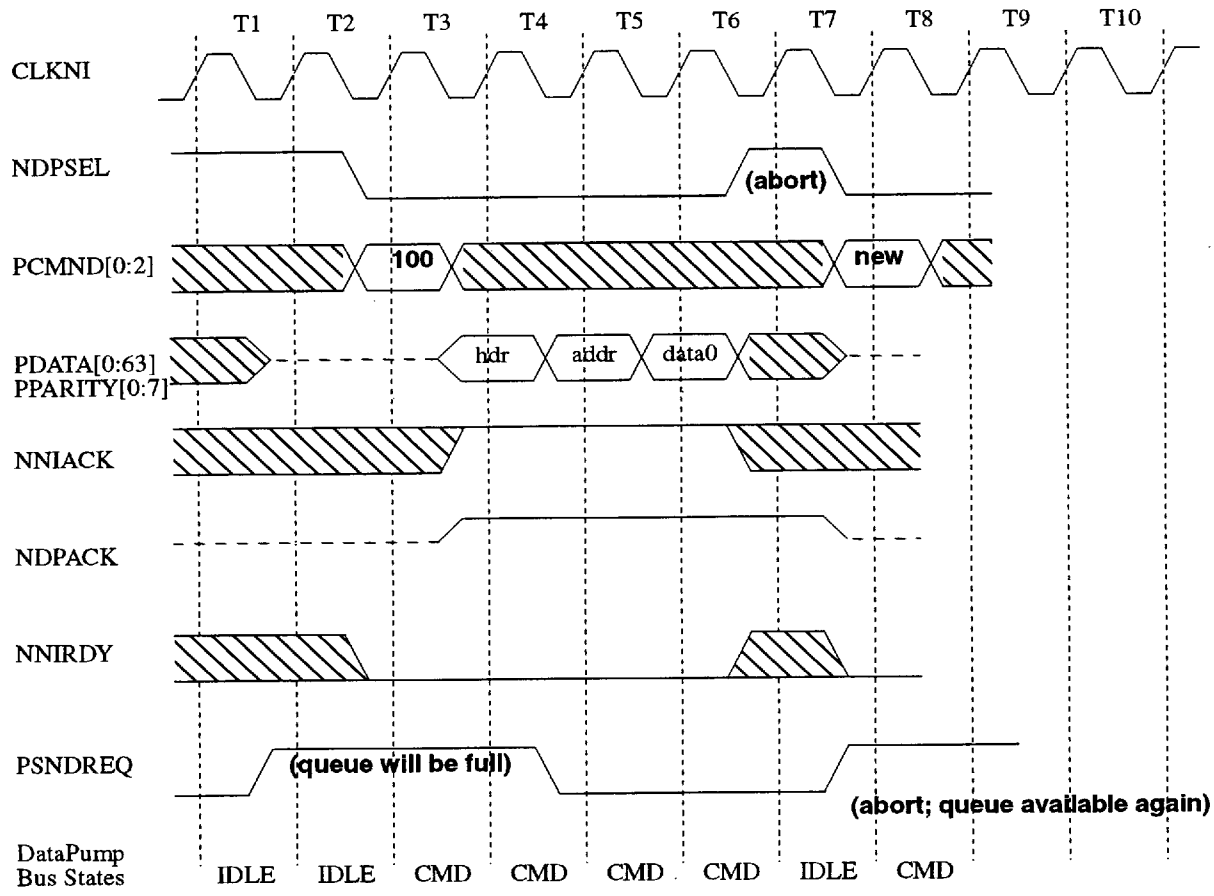
This diagram illustrates hold using NNIRDY false in the CHK cycle to hold off the assertion of NDPACK.

Figure 5.13 : SendReq Queue Transfer with Hold on NDPACK



This diagram illustrates hold using NNIRDY false in the ACK cycle to extend NDPACK one cycle.

Figure 5.14 : SendReq Queue Transfer Abort During CMD State



This diagram illustrates the use of NDPSEL driven false during a transfer to abort the transfer. During the transfer the DataPump is in the CMD state. A normal end of packet transfer occurs when NNIACK is asserted. The DataPump then transitions from the CMD state to the CHK state, then from CHK to ACK and IDLE again.

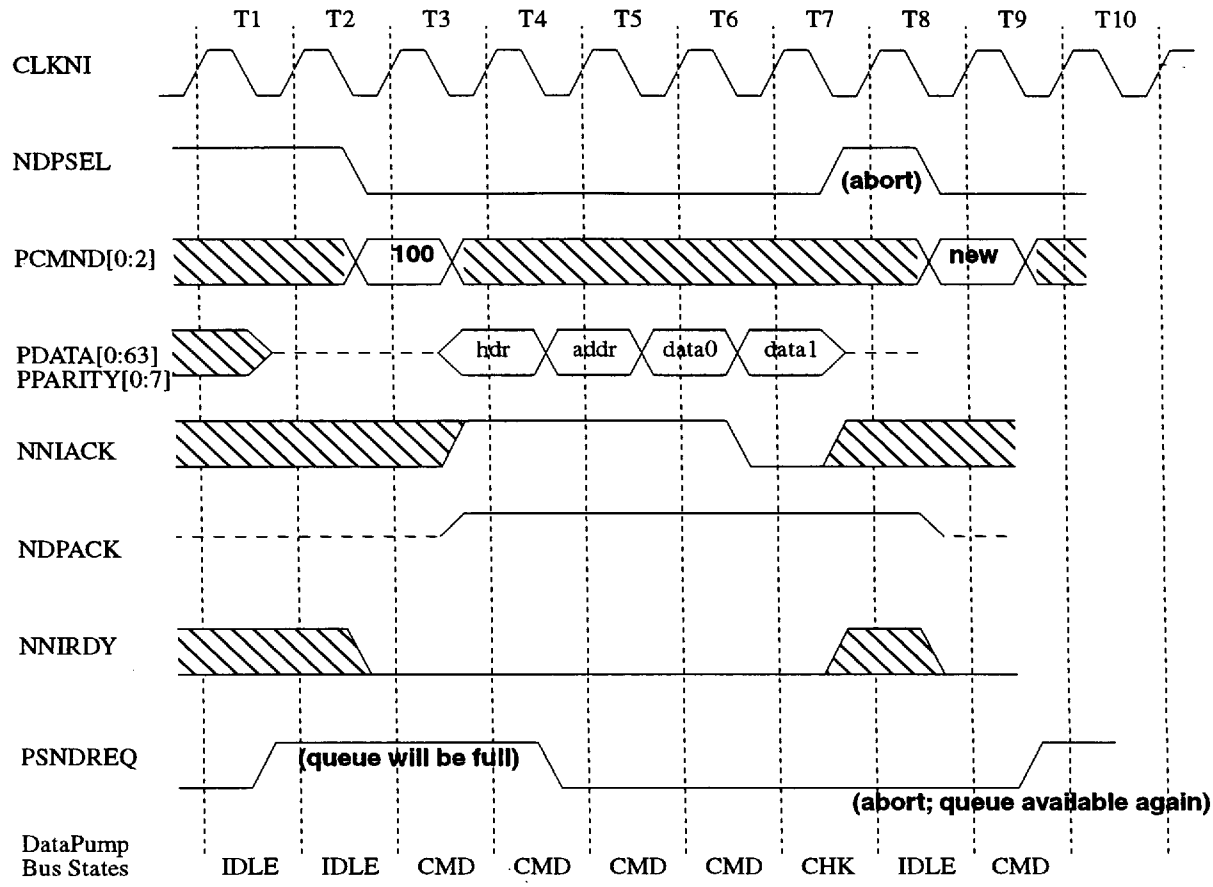
If NDPSEL is driven false during CMD state as shown in cycle T6, the DataPump aborts and goes to IDLE. The packet is not entered into the send queue.

If the packet transfer being executed would have filled the queue, the queue flag goes false as shown in cycle T4. Since the abort frees the queue slot again, the queue flag goes true again. It does take an additional cycle after the abort, as shown in cycle T8, to drive the queue flag true again.

If a new sendReq command is attempted in T8 (where "new" is indicated) then a sendQfull transfer error will occur since the queue is not logically available, due to the abort, until T9.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.15 : SendReq Queue Transfer Abort During CHK State



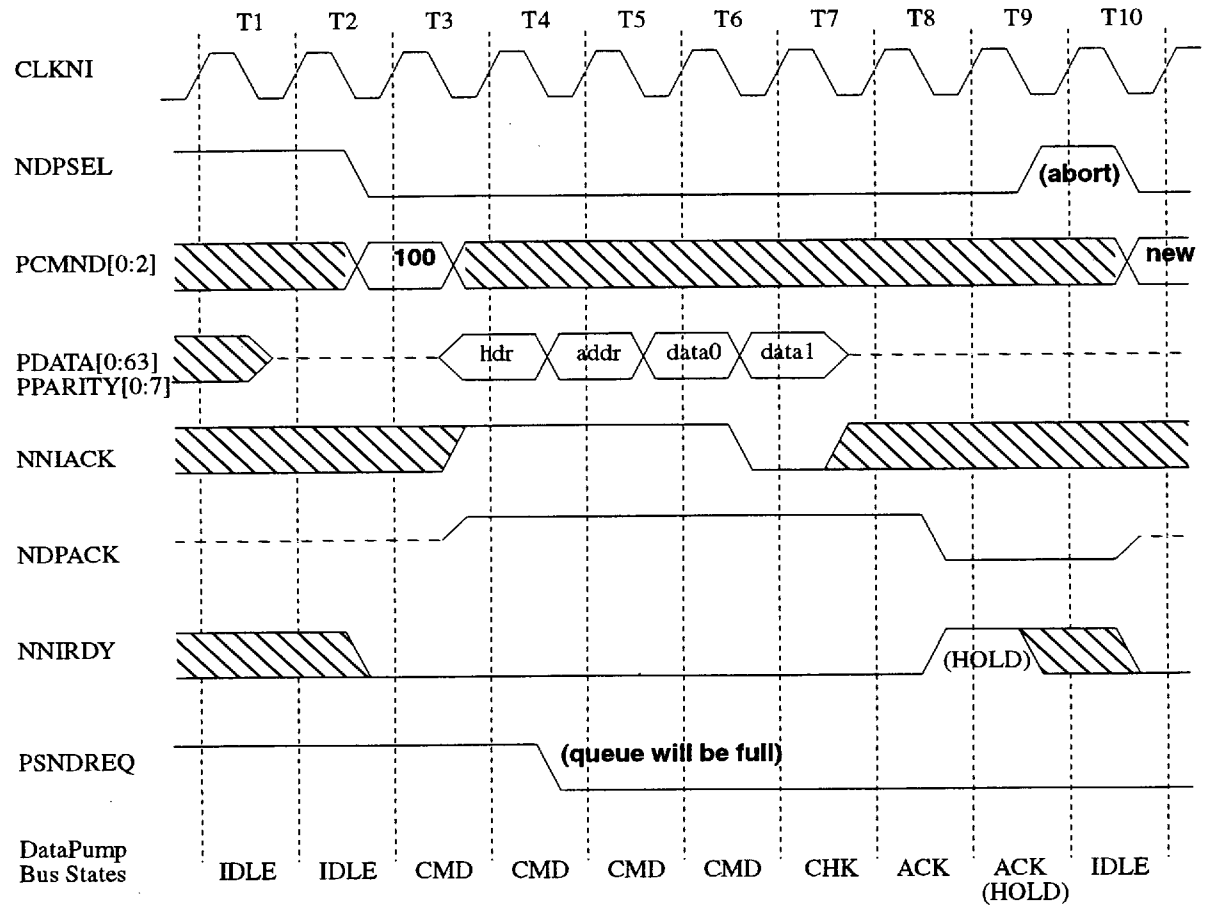
This diagram illustrates the use of NDPSEL driven false after the end of transfer to abort the entry into the queue. During the transfer the DataPump is in the CMD state. A normal end of packet transfer occurs when NNIACK is asserted. The DataPump then transitions from the CMD state to the CHK state, as shown in cycle T7.

If NDPSEL is driven false during CHK state as shown in cycle T7, the DataPump aborts and goes to IDLE. The packet is not entered into the send queue.

If the packet transfer being executed would have filled the queue, the queue flag goes false as shown in cycle T4. Since the abort frees the queue slot again, the queue flag goes true again. It does take an additional cycle after the abort, as shown in cycle T9, to drive the queue flag true again.

If a new sendReq command is attempted in T9 (where "new" is indicated) then a sendQfull transfer error will occur since the queue is not logically available, due to the abort, until T10.

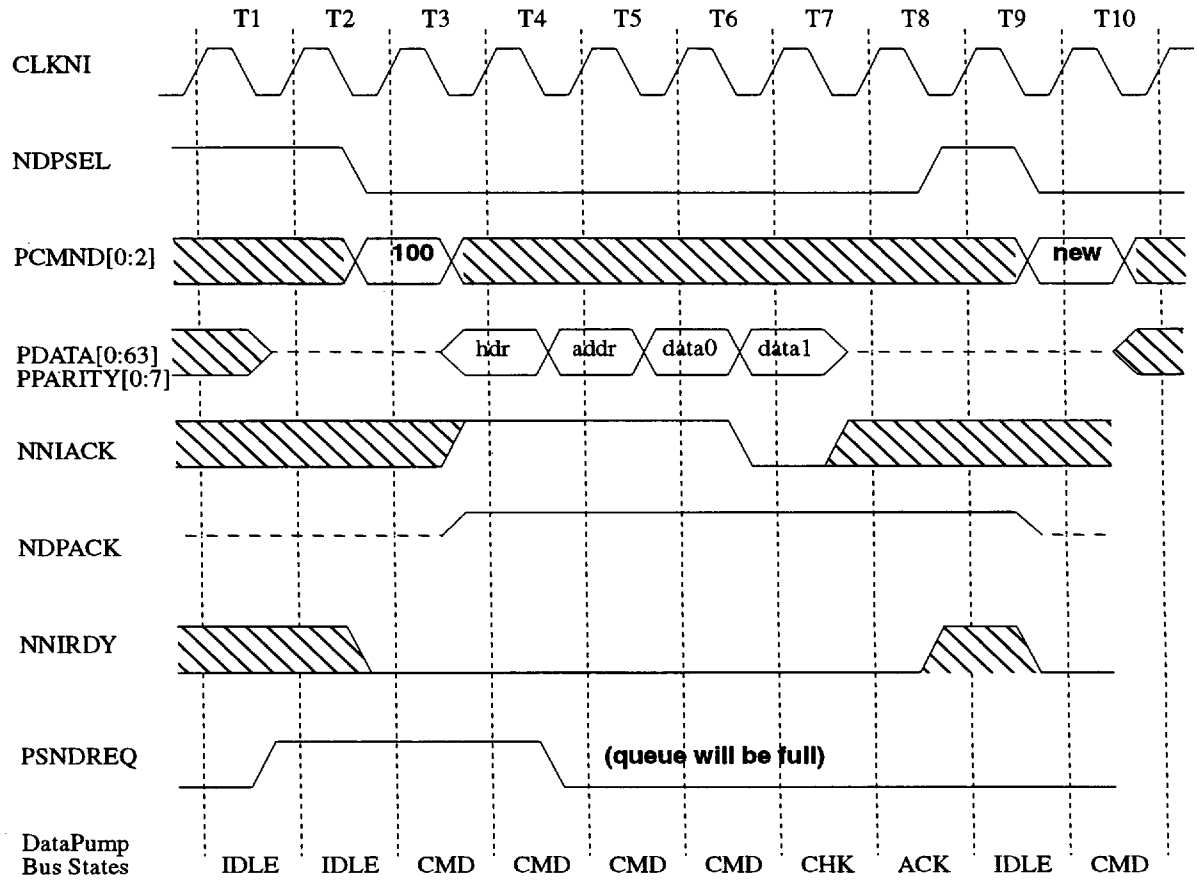
Figure 5.16 : SendReq Queue Transfer with Abort During ACK Hold



This diagram illustrates how NDPSEL going high during ACK always aborts and sends the DataPump to IDLE. In the case of send queue transfers, however, there is no side effect of this abort. The send queue packet has already been validated in the queue by T8. NDPSEL going high after that only has the effect of overriding NNIRDY false (in T9) and sending the DataPump to IDLE.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.17 : SendReq Queue Transfer with DataPump Detected Transfer Error



This diagram shows a send packet transfer to the DataPump with a DataPump detected error in the transfer. The error could be due to bad parity in the transfer, trying to write a full queue (not in this case since the PSNDREQ queue flag shows room), packet size not a multiple of two octlets or greater than 12 octlets (this example shows a valid packet size and even number of octlets), or request packet has a duplicate tranId to a request packet already in the queue.

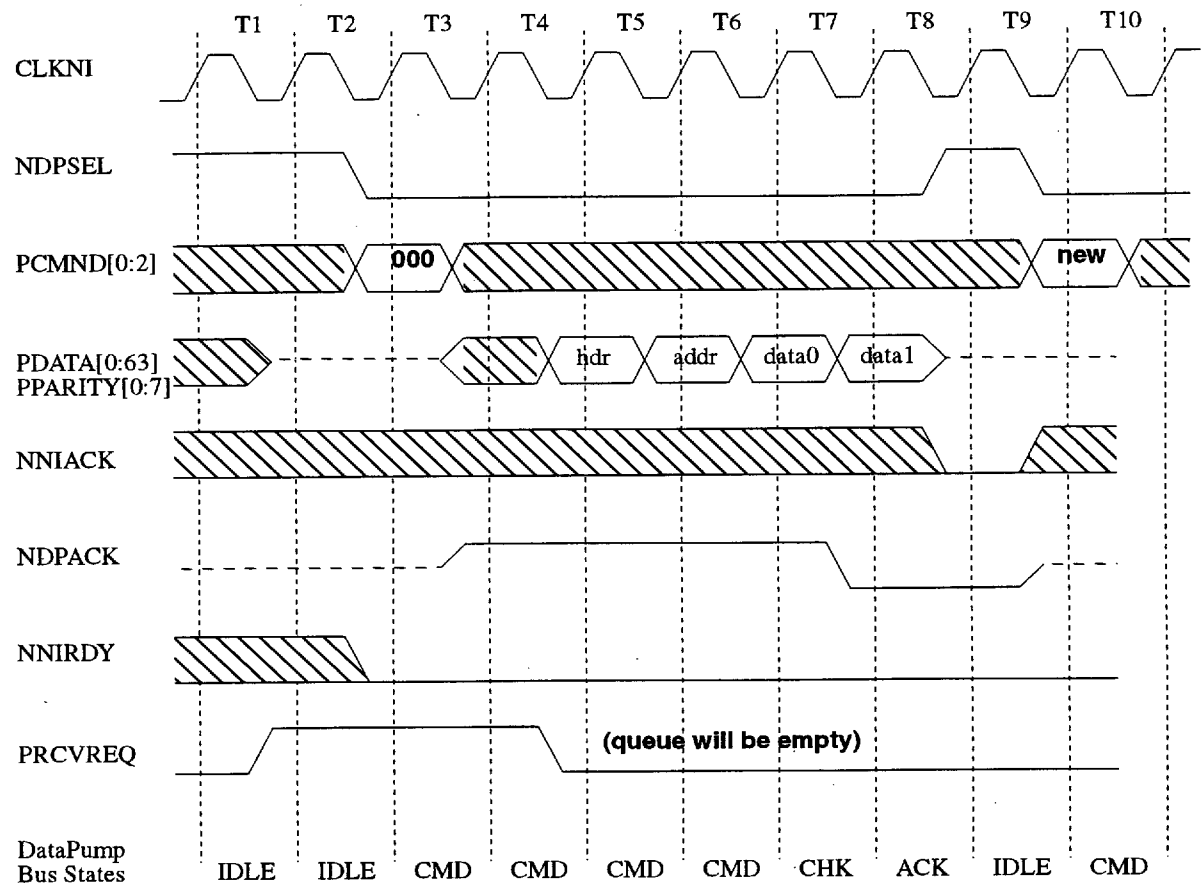
The error is indicated by NDPACK false in cycle T8 when it should have been asserted true. The NDPACK signal always follows two cycles after NNIACK true (in the ACK state) unless NNIRDY is used to add wait cycles.

The packet is not entered into the queue and the DP_ERRLOG register records the transfer error and should be inspected by the node interface logic.

This diagram illustrates minimum cycle timing of all signals with no hold cycles.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.18 : RcvReq Queue Transfer Initiated by PRCVREQ Flag

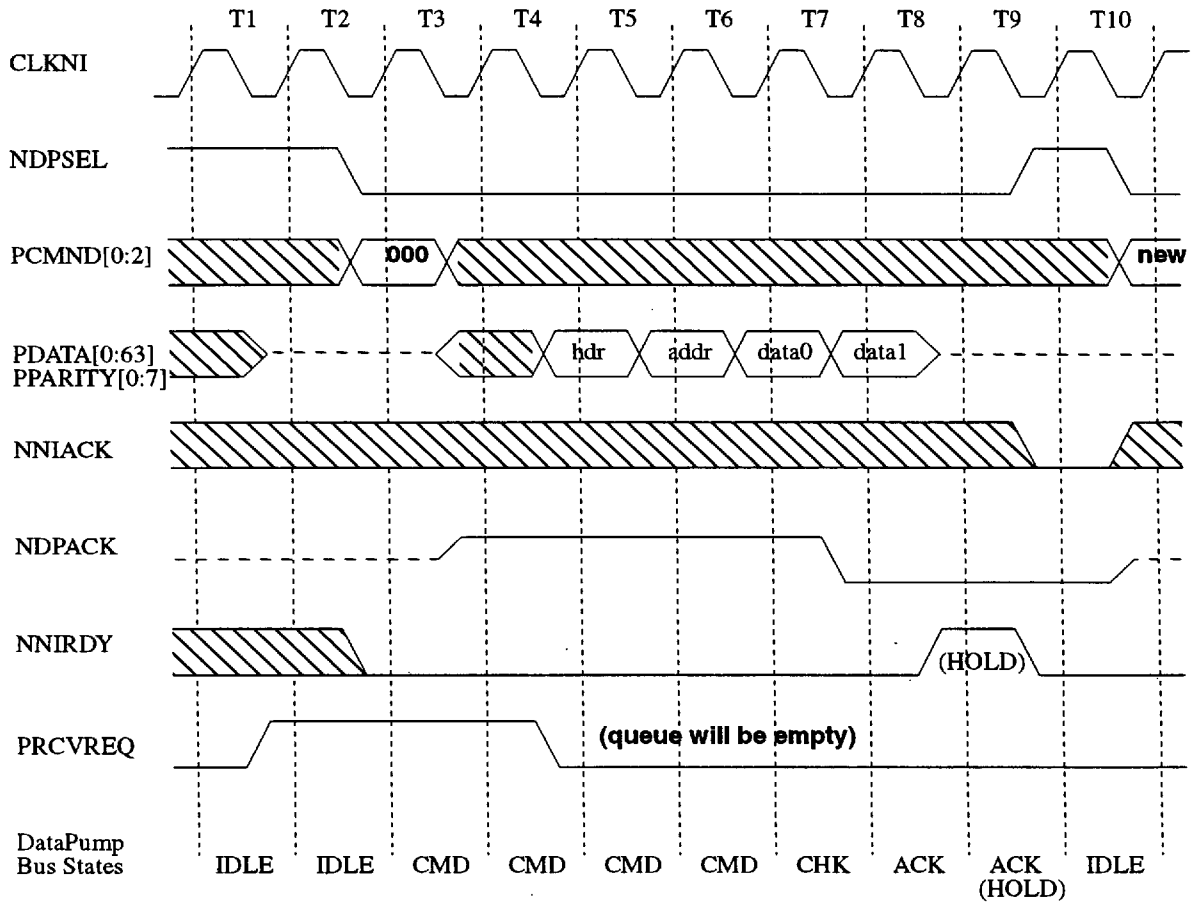


This diagram shows a normal packet transfer from the DataPump to the node interface logic. In this case, the packet contains the required header information and 16 bytes of data. Larger data payloads simply extend the number of cycles. NDPACK is used to indicate the end of transfer.

This diagram illustrates minimum cycle timing of all signals with no hold cycles.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.19 : RcvReq Queue Transfer with Hold for NNIACK

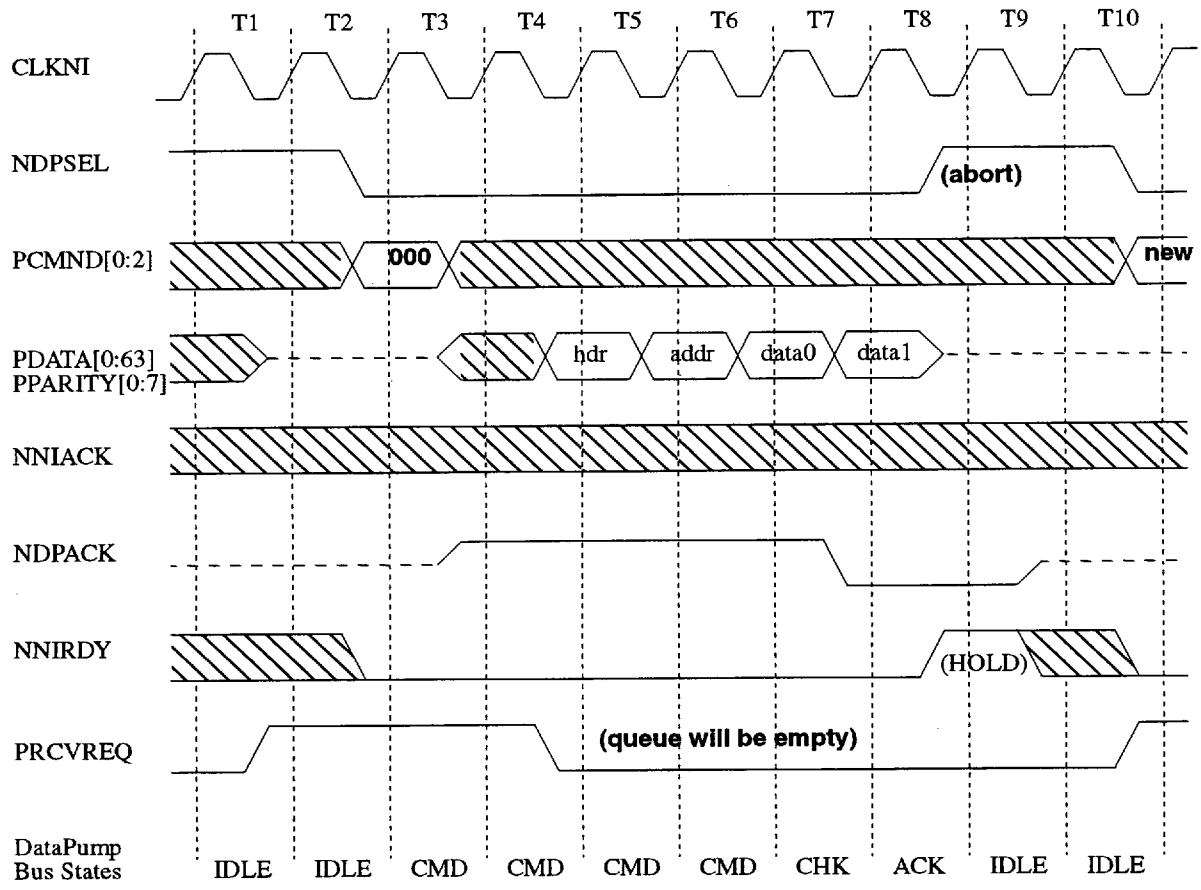


This diagram shows the use of NNIRDY to hold off the assertion of NNIACK for one cycle at the end of a transfer to the node interface logic. This may occur if the node interface logic needs an extra clock cycle to verify the transfer and check parity before sending acknowledge, NNIACK. NNIRDY false in T9 causes the DataPump to hold in the ACK state and wait on sampling NNIACK until NNIRDY goes true again in T10. Once the DataPump sees NNIACK in T10, it goes into the IDLE state and removes the packet just transferred from the receive queue.

If NNIACK had not been asserted in T10, the DataPump would not remove the packet from its receive queue and the PRCVREQ flag would go true again in cycle T11.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.20 : RcvReq Queue Transfer with Abort During ACK Hold

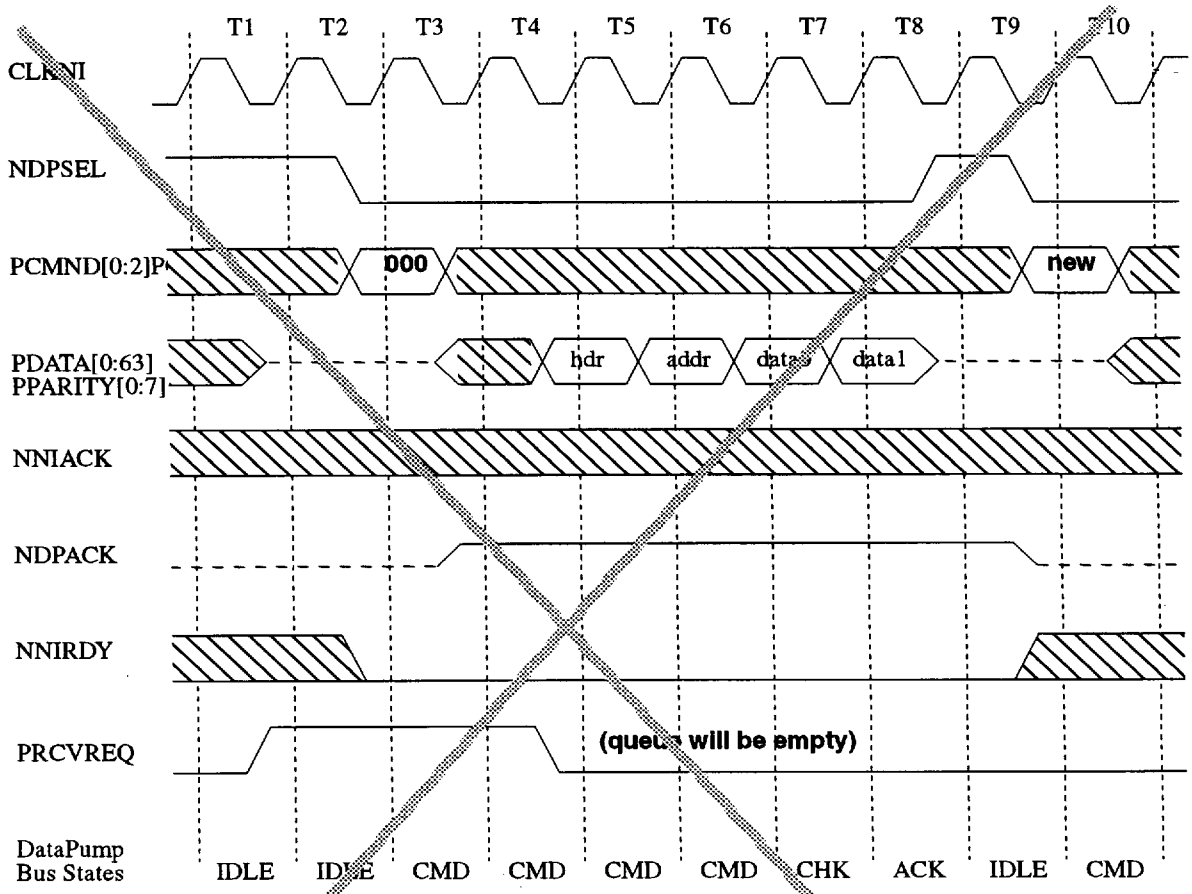


This diagram shows how NDPSEL going high in the ACK state overrides NNIRDY false (as in T8). At the end of a receive queue transfer the DataPump checks for NNIRDY true and NNIACK true in the ACK cycle (beginning T9) in order to remove the just transferred receive packet from the receive queue.

Since NDPSEL aborted the end of the transfer in T9 before the DataPump received NNIACK and NNIRDY true, the packet would not be removed from the queue. This is reflected by the PRCVREQ flag going true again in T10 indicating a packet still present in the queue.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.21 : RcvReq queue Transfer with DataPump Detected Transfer Error



This diagram shows a packet transfer from the DataPump with a DataPump detected error in the transfer. The error could be due to bad parity in the packet in the receive queue or trying to read an empty queue (not in this case since the PRCVREQ queue flag shows a packet to be read).

The error is indicated by NDPACK false beginning in cycle T8 when it should have been asserted true. The NDPACK signal always occurs in the last data transfer cycle as indicated by counting octlets transferred up to the "size" value provided in the header octlet.

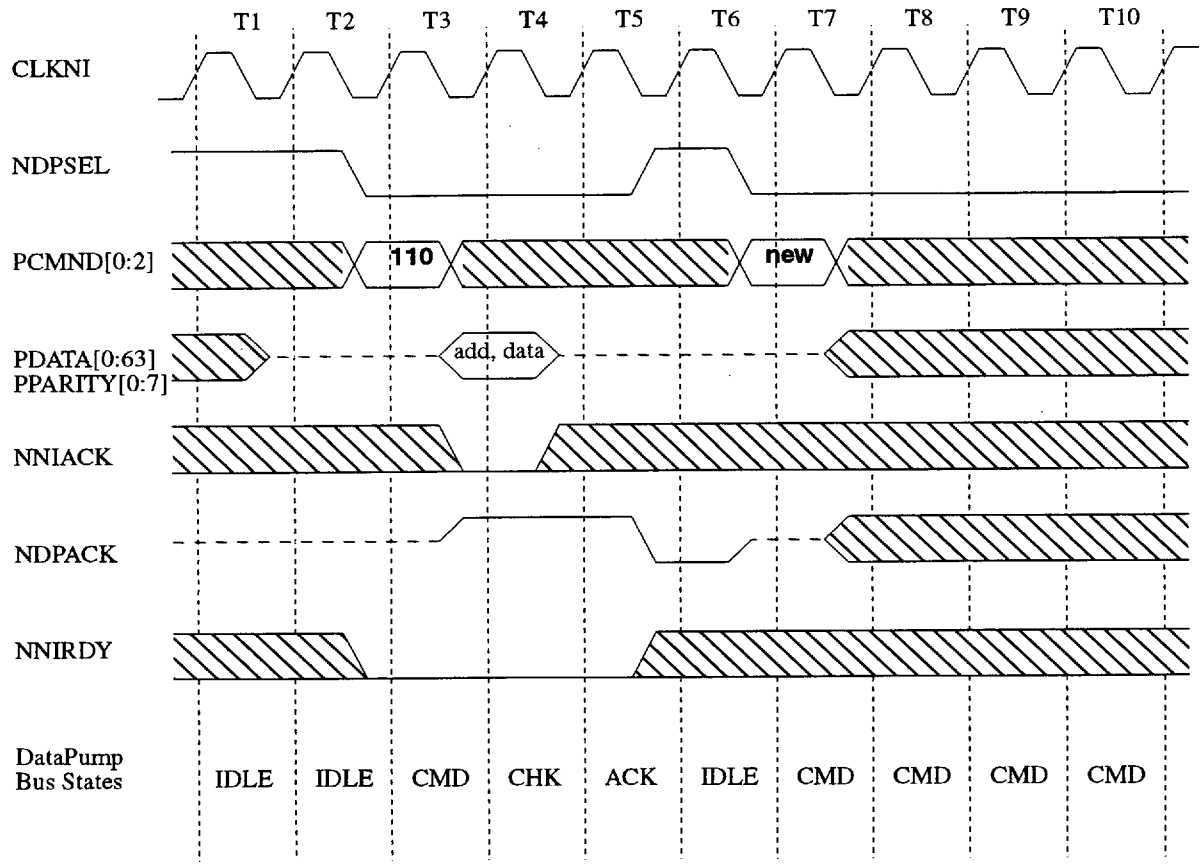
If the error was due to a parity error in the packet, it is removed from the receive queue in the ACK cycle even if the bad parity were detected in cycle T4, first data. However, if NDPSEL goes false anytime before ACK the transfer will abort and the packet will be retained in the queue even if it had bad parity. This gives the same behavior due to abort if an error occurred or not. The DP_ERRLOG register records the transfer error and should be inspected by the node interface logic.

This diagram illustrates minimum cycle timing of all signals with no hold cycles.

THIS FUNCTION IS NOT IMPLEMENTED. NO PARITY CHECKING IN QUEUES.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.22 : RegWrite Transfer

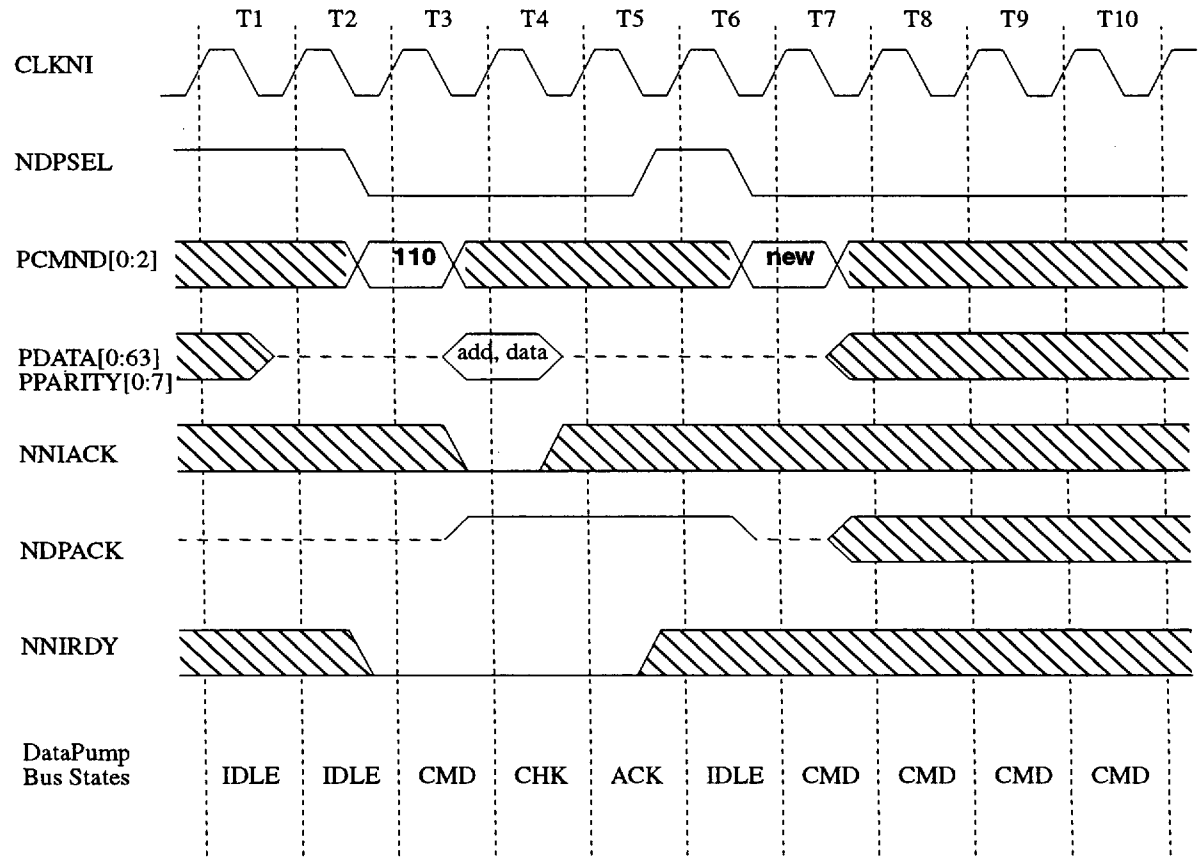


This diagram shows a normal register write transfer from the node interface logic to the DataPump. In this case, register address offset and data are provided in T4 along with NNIACK. NDPACK is used to acknowledge the end of transfer in T6.

This diagram illustrates minimum cycle timing of all signals with no hold cycles.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.23 : RegWrite Transfer with DataPump Detected Transfer Error



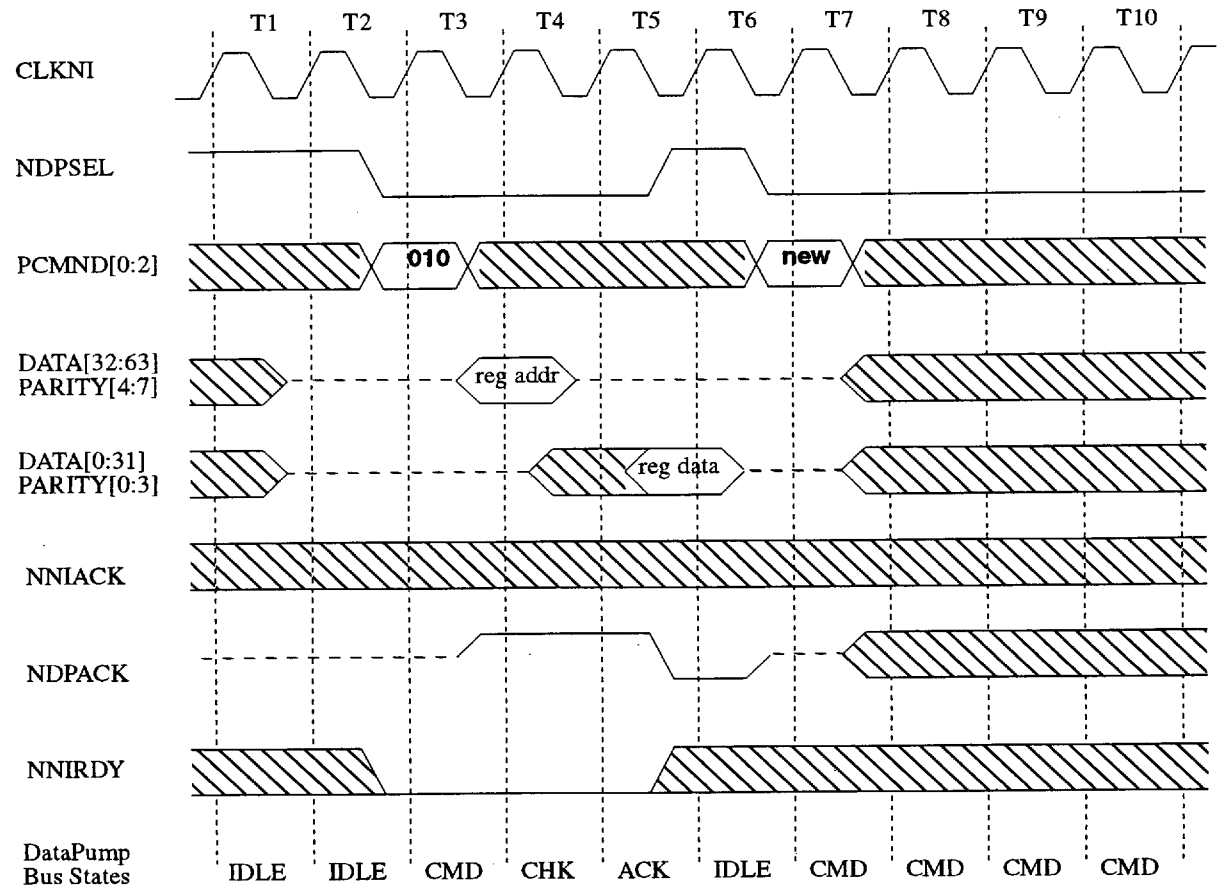
This diagram shows a register write transfer from the node interface logic to the DataPump with a DataPump detected transfer error. Errors of this type could be bad parity in the transfer or bad register offset address. In either case, the error is indicated by NDPACK false in T5 when it should have been asserted true.

No register is written and the DP_ERRLOG register records the transfer error and should be inspected by the node interface logic.

This diagram illustrates minimum cycle timing of all signals with no hold cycles.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.24 : RegRead Transfer

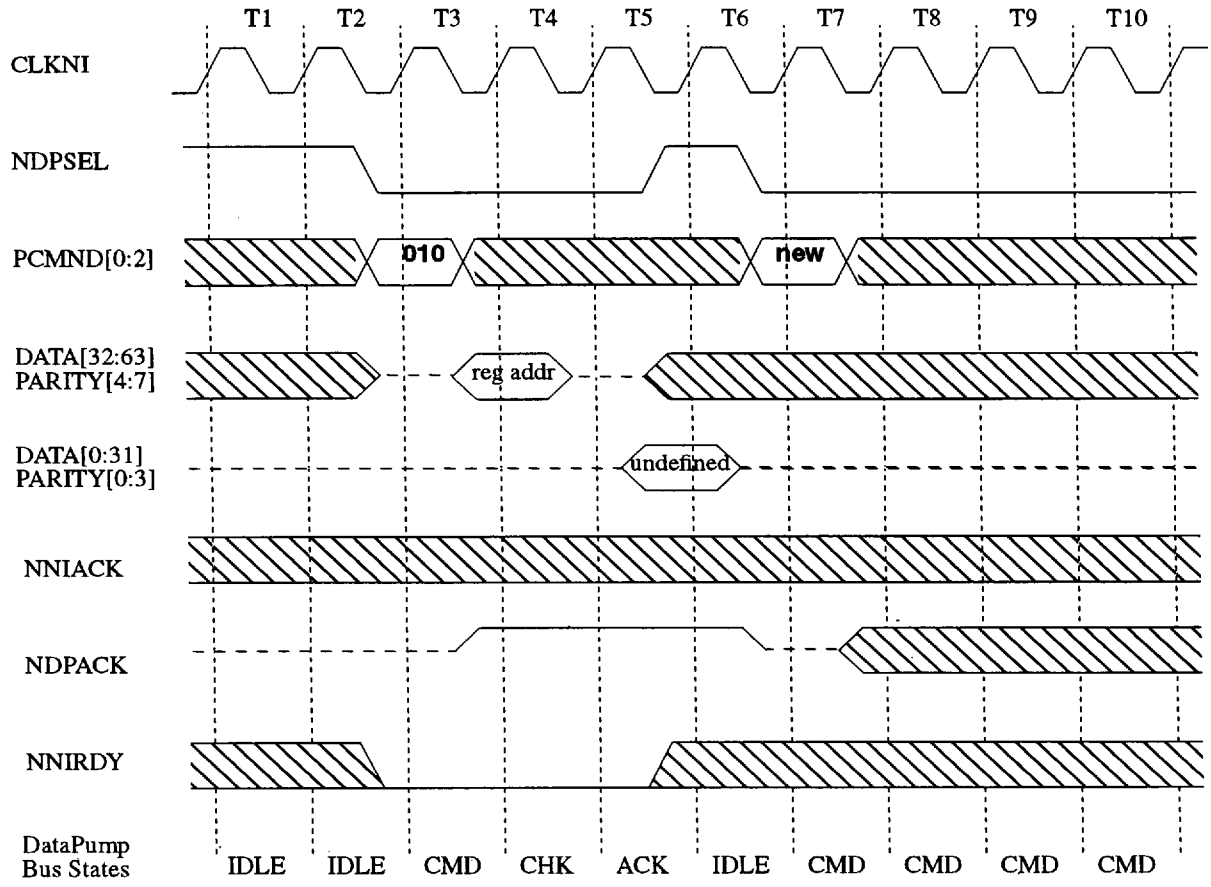


This diagram shows a normal register read transfer from the node interface logic to the DataPump. In this case, register address offset is provided in T4. The DataPump provides register read data back in cycle T5 along with NDPACK. NNIACK is not required in this case since the DataPump would take no action based on a true or false NNIACK.

This diagram illustrates minimum cycle timing of all signals with no hold cycles.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 5.25 : RegRead transfer with DataPump detected error



This diagram shows a register read transfer with a DataPump detected transfer error. This type of error could only be a bad register offset address. The error is indicated by NDPACK false in T5 when it should have been asserted true. The data provided by the DataPump in T5 is undefined.

The DP_ERRLOG register records the transfer error and should be inspected by the node interface logic.

This diagram illustrates minimum cycle timing of all signals with no hold cycles.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

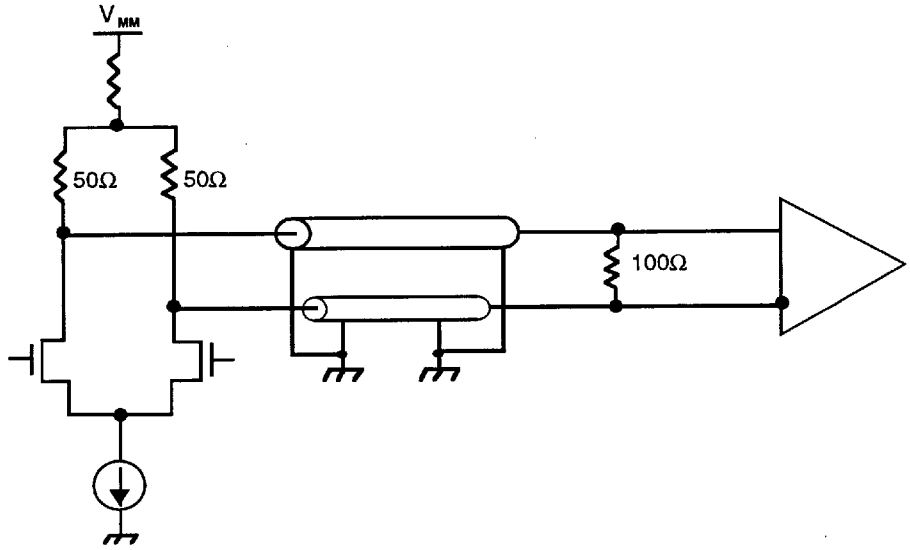
6.0 SCI High Speed Link Signals

The high speed interface on the DataPump chip complies with the Low Voltage Differential Signals proposed IEEE Std 1596.4. These are unidirectional, fully differential signals. The DC and AC electrical specifications are given in Section 11.0 Electrical Specifications.

6.1 PSCIDI[0:15] and PSCIDO[0:15] - Data

These are the 16-bit parallel data in and data output signals. The outputs have 50 ohm output impedance in each differential output. The inputs are terminated between true and complement inputs with on-chip nominal 100 ohms as shown in Figure 6-1.

Figure 6.1: PSCIDI and PSCIDO Terminations



6.2 PSCISI, PSCISO - SCI Strobe

This is the input and output strobe signal. The strobe transitions at the same time as data and flag and is used to latch data and flag into the DataPump on both edges. The DataPump has internal delay between strobe and data, flag to generate a clock signal for the input registers.

6.3 PSCIFI, PSCIFO - Flag Bit Encoding

Flag is asserted at the beginning of a packet to indicate packet start.

Flag is de-asserted near the end of a packet to indicate packet end. The number of cycles flag remains asserted depends on packet type.

If flag remains asserted at least 4 cycles, the packet must be a send packet (or an ABORT packet in which flag stays high 6 cycles and causes a noInSync error). The minimum send packet is 8 symbols. Flag will be deasserted 4 symbols before the end of the packet (indicating when to begin creating an echo packet if necessary).

An echo packet has a sequence of 3 cycles when flag is asserted. Flag goes false on the last cycle of the echo packet.

Sync packets have 8 symbols and flag is asserted on the first symbol only.

The ABORT packet (see Section 4.4.4 Special Packets and Flag Encoding) is 8 symbols long and the flag is asserted for the first 6 symbols.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

7.0 Latency

(this section intentionally left blank)

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

8.0 Performance Counters

Two output pins, PPC0 and PPC1, are provided for monitoring certain internal events in the DataPump and which can be used along with external counters to monitor various performance aspects. There are two CSR registers which control the events which drive the PPC0 and PPC1 signals. Performance metrics are usually calculated by accumulating a reference count and dividing that into a qualified count with the ratio representing the performance metric. PPC0 is the output pin used to generate the reference count and PPC1 is used for the qualified count. For example, PPC0 may be set up to count all packets sent by the DataPump and PPC1 may be set to count all packets sent with a retry phase. The ratio of PPC1 to PPC0 is then the percentage of transmissions by this DataPump which were retrys. Another example would be to set PPC0 to count all bypassed packets and PPC1 to count bypassed packets which were coherent memory read requests. The ratio of PPC1 to PPC0 would then represent the percentage of those type packets out of all packets travelling to other nodes around the ringlet.

8.1 Command Field Mask/Compare

There are three 7-bit mask/compare functions located in the DataPump for performing a match on a packet's command field value. There is one comparator in the SCI input block, one in the SCI transmission block, and one at the send queue. A match out of these comparators occurs according to the following logic;

cmdMatch = ((packet.cmd[9:15] & cmdMask[0:6]) ==

(cmdCmp[0:6] & cmdMask[0:6]))

Thus, a bitwise equal comparison is made between the cmdCmp value in the DP_PC_CONFIG0 register and the packet's command field cmd value (bits 9 through 15). The cmdMask value masks off the comparison on a bitwise basis if the corresponding cmdMask bit is set to one. "cmdMatch" then goes true if all bits of the packet's command.cmd value is equal to the cmdCmp value for the bits which are not masked off by cmdMask.

A DP_PC_CONFIG0 register bit is also provided to invert the sense of "cmdMatch" called cmdSns. If cmdSns is set to one, "cmdMatch" will go true for mismatches instead of matches.

These comparators located in the SCI input, SCI transmission block, and the send queue are used to qualify the counting of packets to match user specified command values. For example, the "cmdMatch" result will go true only for response packets if cmdCmp is set to 1111100 and cmdMask is set to 0000011 and cmdSns is 0 since response packets can have only the command field values of 1111100, 111101, 1111110, or 1111111. If cmdSns is set then to 1 then "cmdMatch" will go true for non-response packets (requests, moves, and events). In this way, the command comparator can be set up to match any type of packets.

8.2 Extended Command Field Mask/Compare

In addition to the command field mask/compare functions there are two extended command field mask/compare functions. One is located in the SCI input block and one in the SCI transmission block. The extended command field comparator is 5-bits wide and operates on the command symbol bits 4,5,7,8, and 9. Bits 4 and 5 are the com.phase0 and com.phase1 bits which are used by the queue reservation protocol. Bit 7 indicates the packet is an echo when set. Bits 8 and 9 are the com.bsy and com.res bits respectively when the packet is an echo. Therefore, the extended command field mask/compare is useful for looking at the phase of packets to check for retry traffic and also for distinguishing echos from send packets. The logic for this mask/compare is as follows;

extMatch = ((packet.cmd[4,5,7-9] & extMask[0:4]) ==

(extCmp[0:4] & extMask[0:4]))

The values of extMask[0:4] and extCmp[0:4] are put into the DP_PC_CONFIG0 register. The mask/compare functions exactly like the command field mask/compare except on different bits of the command field.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

8.3 PPC0 Functionality

The reference counter PPC0 can be set to monitor one of five events. The PPC0 output will toggle every time one of these events occurs. A resetable external counter should be connected to the PPC0 pin to accumulate a reference count over some time interval. These countable events are;

1. pktRcv - Packets received (i.e. target ID of packet matches this DataPump's nodeid) including echoes. Echoes can be excluded from this count by using the DP_PC_CONFIG0 refMask bit. This bit enables the extended command field mask/compare result to qualify the pktRcv PPC0 signal. Therefore, to exclude echoes from the pktRcv count the ref-Mask bit is set, the extMask value is set to 11011 and the extCmp value is set to 00000. This will cause pktRcv to toggle only on received send packets.
2. Packets bypassed, including echo packets.
3. Packets sent, not including generated echoes.
4. Number of packets transmitted from the send-request queue.
5. Number of packets transmitted from the send-response queue.

Thus, the counter connected to PPC0 can be used to count all packets received, sent, or bypassed by this DataPump over a particular time interval or can count the number of packets transmitted from either of the send queues over a time interval. This count value will be used as the refernce value to measure some other qualified count value against. The qualified count value is generated by the PPC1 signal.

8.4 PPC1 Functionality

The qualified counter PPC1 can be set to monitor one of 13 different events. These events are broken down into the following categories;

1. Toggle whenever a matching packet is received, bypassed, or sent. These are used to monitor the packet traffic and bandwidth utilization on the SCI links.
2. Toggle at a rate proportional to the current number of packets in one of the four queues. This can be used to measure average queue depth and utilization.
3. Toggle whenever cc or ac changes. These give a measure of time around the ringlet and allocation time around the ring-let.
4. Assert whenever a matching packet is entered into one of the send queues. This is used to measure latency of a packet in the send queue; that is, the amount of time a packet spends in the send queue from the time it is entered until it is finally accepted at the other end and removed from the queue.
5. Assert whenever the current reservation phase matches the extended mask/compare value in the phase bits.

This is used to measure the amount of time the DataPump spends in each of the reservation phases.

The DP_PC_CONFIG1 register selects which of the 13 signals drives the PPC1 output pin. A value of one in that register corresponding to a particular signal will selct that signal to drive PPC1. If more than one bit of DP_PC_CONFIG1 is set to one the PPC1 output will be unpredictable.

8.4.1 Toggle on Matching Packet

There are three signals which will toggle when detecting a matching packet. These are;

1. RcvPkt - toggle when a matching packet is received at the SCI input link and the target ID matches this DataPump. This includes received echo packets but not generated echo packets. An additional qualifier bit in the DP_PC_CONFIG0 register is the rcvVldMask bit. When set to one, RcvPkt will toggle only on matching receive packets which in fact get entered into the receive queue. This would eliminate toggling on receive matching packets which are tossed because the receive queue is full or for other reasons such as bad CRC, etc.
2. BypPkt - toggle when a matching packet is bypassed from the input link to the bypass FIFO. This includes received echo packets but not generated echos.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

3. SndPkt - toggle when a matching packet is transmitted from the send queue to the SCI output link, e.g. this does not include echos generated by the input link on retry echos.

All three of these signals use both the cmdMatch and extMatch results to qualify the toggling of the signal. The result of these comparators is anded together to generate the qualifier. There is a cmdMatch and extMatch comparator in both the SCI input link (used to qualify RcvPkt and BypPkt) and the SCI output link (used to qualify SndPkt).

8.4.2 Average Queue Depth Measurement

When PPC1 is set to monitor one of the four queue depth signals (rRspDepth, rReqDepth, sRspDepth, sReqDepth) it will toggle at a rate proportional to the number of packets in that queue. If there are four packets in the queue PPC1 will rise and fall 4 times during an 8 cycle CLKNI period. If there are three packets it will rise and fall 3 times in that same 8 cycle interval, for 2 packets it will rise and fall twice, and for 1 packet it will rise and fall only once. Therefore, a counter should be connected to PPC1 and a seperate counter to CLKNI. Average queue depth could then be calculated by first resetting both counters, letting them run for an interval while normal SCI operation occurs, then stoping them and using the following formula;

average queue depth = (PPC1 count)/((CLKNI count)/8)

In this way, the average queue depth can be determined for any queue under various loading conditions, etc.

8.4.3 CC and AC Ringlet Time Measurement

PPC1 can also be connected to signals monitoring the idle.cc and idle.ac bits going through the DataPump. When PPC1 is connected to either of these signals, it will basically reflect the value of idle.cc or idle.ac in the DataPump.

8.4.4 Send Queue Latency Measurement

The amount of time a send packet resides in the send queue can be monitored by connecting PPC1 to one of the signals, rsSlot0Tm or rqSlot0Tm. These signals are asserted as long as a packet is validated in either the request or response queue slot 0 location. Slot 0 is the first queue slot to be used after the queue starts from reset. By counting real time with one counter and elapsed time as qualified by PPC1 with another counter, the average latency of slot 0 for request or response send queues can be measured. The cmdMatch comparator is also applied to the command field of the packet in slot 0 for this signal. If the cmdMatch is false, PPC1 will not be asserted even if a packet is validated in slot 0. This allows the user to only measure latency of certain types of packets in the send queue; for example, cmdMatch coul be set to only match cache coherent traffic, thus giving latency for only cache coherent traffic.

8.4.5 Receive Queue Reservation State

The amount of time spent in one of the queue reservation states (SERVE_NA, SERVE_A, SERVE_NB, SERVE_B) can be determined by using the rsRsvPhs and rqRsvPhs signals connected to PPC1. These signals will be asserted whenever the reservation state matches the phase in extCmp[0:1]. The extMask[0:1] will also mask off either or both bits if desired. By using a real time counter and a second elapsed time counter connected to PPC1 the user can measure average time spent in a particular reservation phase, or by using extMask the average time spent in a reservation phase (SERVE_A and SERVE_B) versus a non-reservation phase (SERVE_NA and SERVE_NB).

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

9.0 Test Access Port

The DataPump conforms to the IEEE Std 1149.1 Standard Test Access Port and Boundary Scan Architecture. This allows access to all the inputs and outputs of the DataPump as well as on-line scan access to the CSR support registers using scan shadow registers. Also provided is access to internal RAMs.

The Test Access Port uses the pins PTCK, PTMS, NTRST, PTDI, and PTDO to perform serial shifting of data and control for testing the DataPump.

The internal DataPump clocks are all connected to PTCK to perform boundary scan or RAM testing. The clocks run normally for CSR shadow register access. The DataPump registers can be read and written through JTAG while the part is running normal operation.

A block diagram of the various scan chains accessible through JTAG is shown in Figure 9.1.

These registers operate like dual rank registers. That is, they have a shift register part and a parallel loading register part. In the case of boundary scan, the dual rank register is specified by the IEEE std.

The CSR shadow register is a single rank parallel load and shift register. For shadow register read operations the shadow register parallel loads from a selected internal CSR register and the data is then shifted out. For write operations the shadow register is shifted into and the data parallel loaded into the selected internal register.

The instruction register is a dual rank register made up of the shift register and a parallel load register. The outputs of the parallel load register actually provide the instruction data

9.0.1 Test-Logic_Reset State

Test logic is disabled in this state so normal operation of the DataPump can proceed unhindered. This state is entered asynchronously when NTRST is asserted or as long as PTMS is logic 1 for 5 consecutive PTCK rising edges. This state is unaffected by NRESET.

9.0.2 Run-Test/Idle State

This is the idle state between scan operations. No scan activity takes place during this state.

9.0.3 Select-DR-Scan State

This is the initial entry state for data register scan operations. No actual operations take place during this state, but it is necessary to go through this state to do data register scan operations.

9.0.4 Capture-DR State

In this state, a specific instruction register selected data scan register parallel loads from a given source.

In the case of the boundary scan EXTEST instruction the boundary scan register parallel loads from the input pins (and from chip internal signals on output pins).

In the case of a CSR read the internal register selected for read parallel loads into the shadow register.

No shifting of data takes place in this state.

9.0.5 Shift-DR State

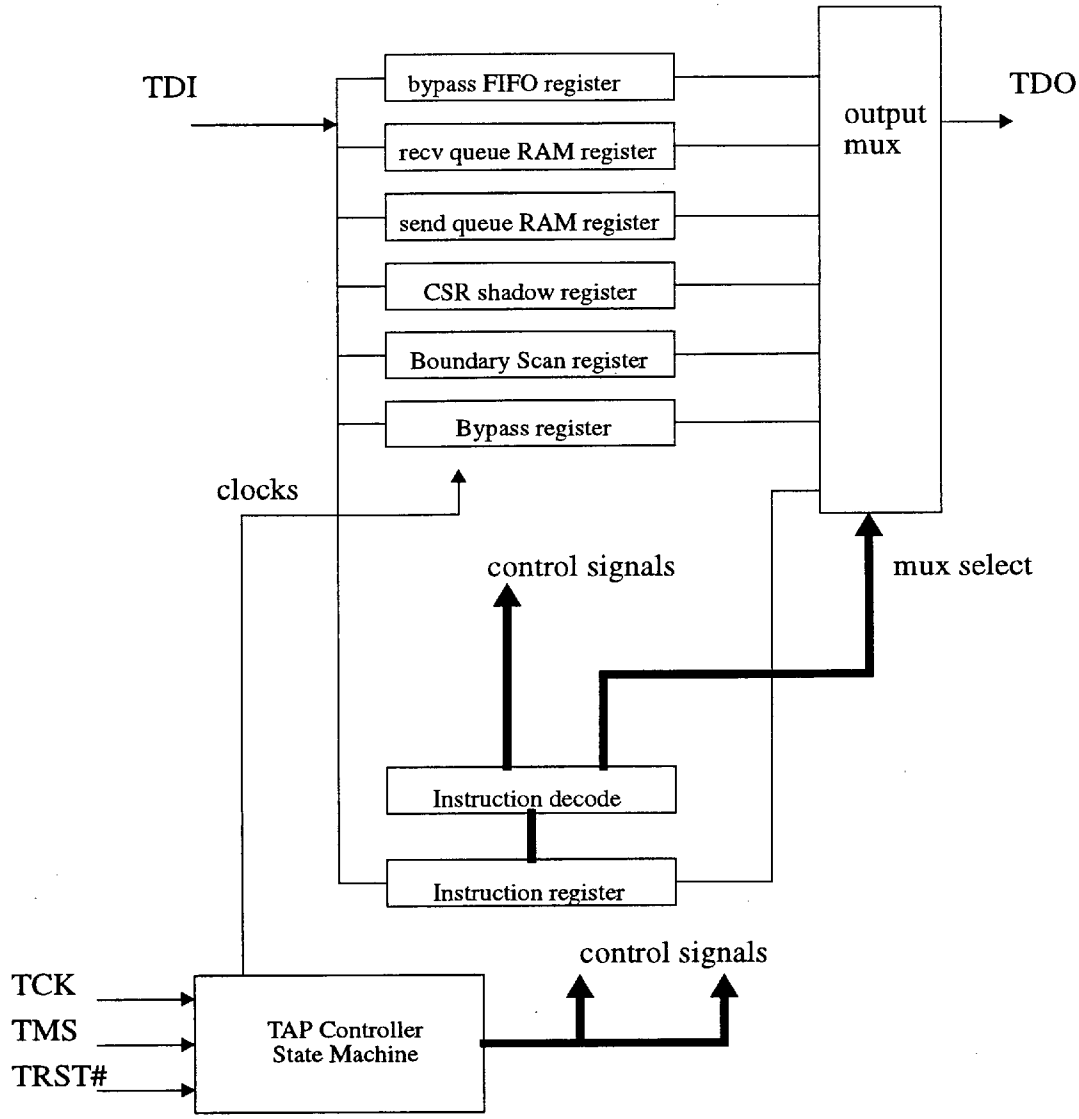
In this state, the data register currently connected between PTDI and PTDO as defined by the instruction register is shifted one stage toward PTDO on PTCK rising edge. If PTMS remains logic 0, data will continue to shift with each PTCK rising edge.

9.0.6 Exit1-DR State

This is a temporary state during which no scan operations take place and no registers change value.

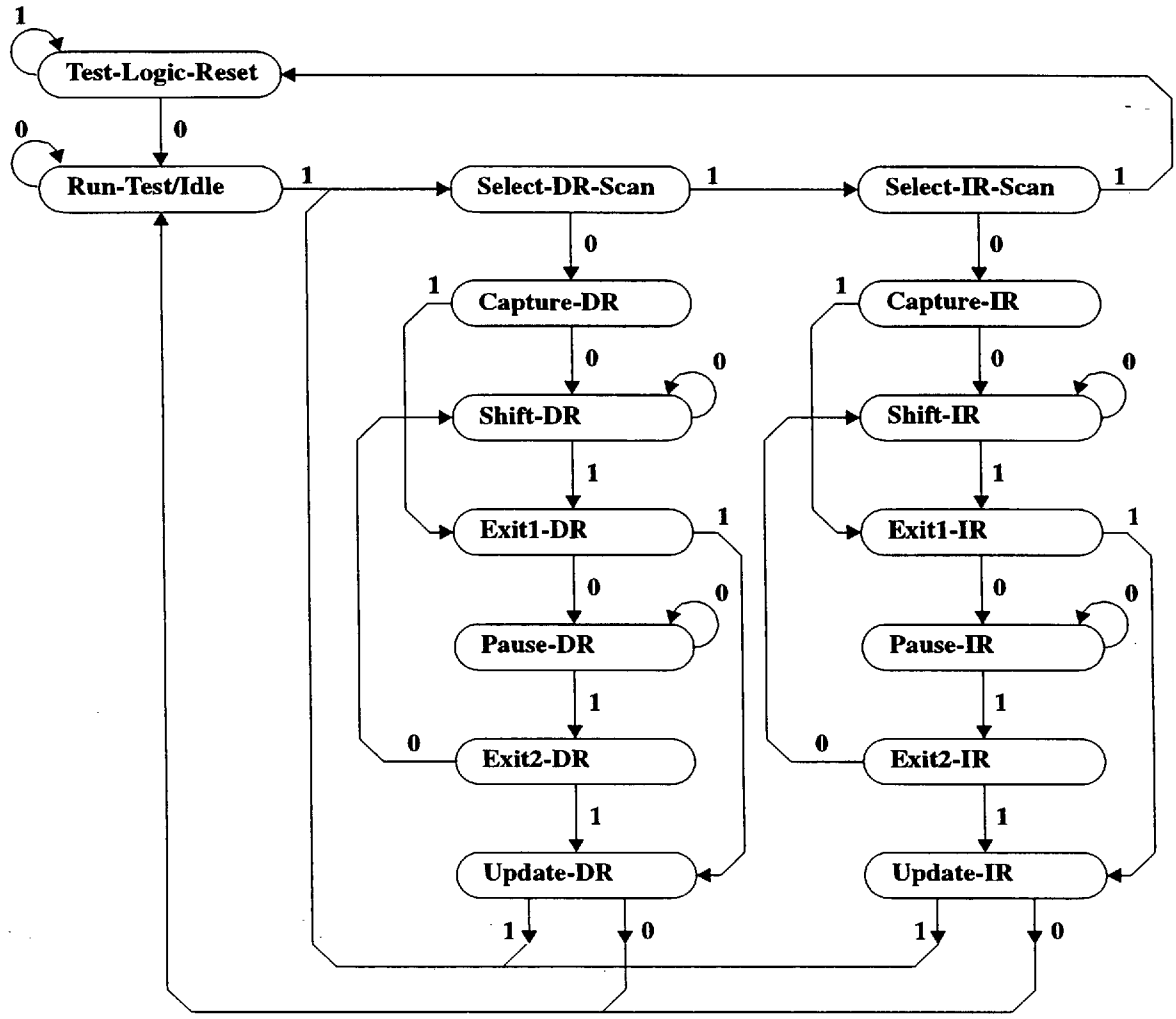
741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 9.1 : Scan Test Access Port Block Diagram



741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 9.2 : Test Access Port State Diagram



Note: The value (0 or 1) shown adjacent to each state transition represents the value of TMS sampled on the rising edge of TCK.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

9.1 TAP Controller State Machine

The TAP controller state machine is clocked by PTCK and controlled by PTMS. It is asynchronously reset by NTRST. Figure 9.2 shows the TAP controller state diagram. The various states are described below.

9.1.1 Test-Logic_Reset State

Test logic is disabled in this state so normal operation of the DataPump can proceed unhindered. This state is entered asynchronously when NTRST is asserted or as long as PTMS is logic 1 for 5 consecutive PTCK rising edges. This state is unaffected by NRESET.

9.1.2 Run-Test/Idle State

This is the idle state between scan operations. No scan activity takes place during this state.

9.1.3 Select-DR-Scan State

This is the initial entry state for data register scan operations. No actual operations take place during this state, but it is necessary to go through this state to do data register scan operations.

9.1.4 Capture-DR State

In this state, a specific instruction register selected data scan register parallel loads from a given source.

In the case of the boundary scan EXTEST instruction the boundary scan register parallel loads from the input pins (and from chip internal signals on output pins).

In the case of a CSR read the internal register selected for read parallel loads into the shadow register.

No shifting of data takes place in this state.

9.1.5 Shift-DR State

In this state, the data register currently connected between PTDI and PTDO as defined by the instruction register is shifted one stage toward PTDO on PTCK rising edge. If PTMS remains logic 0, data will continue to shift with each PTCK rising edge.

9.1.6 Exit1-DR State

This is a temporary state during which no scan operations take place and no registers change value.

9.1.7 Pause-DR State

This state allows the TAP controller to temporarily halt the shifting of data through the data register connected between PTDI and PTDO. If PTMS remains logic 0, the TAP controller will remain paused in this state.

This is a temporary state during which no scan operations take place and no registers change value.

9.1.8 Exit2-DR State

This is a temporary state during which no scan operations take place and no registers change value.

9.1.9 Update-DR State

In this state a specific instruction register selected data register parallel loads.

In the case of the boundary scan EXTEST instruction the output pin drivers parallel load from the boundary scan chain.

In the case of a CSR write instruction, the selected internal register parallel loads from the shadow register.

No shifting of data takes place during this state.

9.1.10 Select-IR-Scan State

This is the initial entry state for instruction register scan operations. No actual operations take place during this state, but it is necessary to go through this state to do instruction register scan operations.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

9.1.11 Capture-IR State

In this state, the shift register in the instruction register is loaded with the previous instruction value on the rising edge of PTCK. The instruction register retains its previous value.

9.1.12 Shift-IR State

In this state, the shift register in the instruction register is connected between PTDI and PTDO and is shifted one stage toward PTDO on PTCK rising edge. If PTMS remains logic 0, data will continue to shift with each PTCK rising edge. The current instruction register value retains its previous value.

9.1.13 Exit1-IR State

This is a temporary state during which no scan operations take place and no registers change value.

9.1.14 Pause-IR State

This state allows the TAP controller to temporarily halt the shifting of data through the instruction shift register connected between PTDI and PTDO. If PTMS remains logic 0, the TAP controller will remain paused in this state.

This is a temporary state during which no scan operations take place and no registers change value.

9.1.15 Exit2-IR State

This is a temporary state during which no scan operations take place and no registers change value.

9.1.16 Update-IR State

The new instruction shifted into the instruction shift register is parallel loaded into the instruction register on the rising edge of PTCK in this state. This value becomes the current instruction.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

9.2 TAP Controller Instructions

Instructions are shifted into the instruction register using the IR (instruction register) state machine operations. The instructions supported include TAP mandatory BYPASS, SAMPLE/PRELOAD, and EXTEST. DataPump specific instructions are also included.

The instruction register is 8-bits long and each instruction code (inst[7:0]) is given in hex form in Table 9.1. The shift order for inst[0:15] is inst7 to inst0.

The basic instructions are listed in Table 9.1.

Table 9.1: TAP Controller Instructions

Instruction	Inst [0:15]	Description
EXTEST	00_{16}	Provides external pin and board testing.
SAMPLE/ PRELOAD	01_{16}	Allows sampling inputs and preloading outputs without affecting normal operation.
RD_STATE	05_{16}	Read DP_STATE register
WR_STATE	06_{16}	Write DP_STATE register
RD_NODEID	15_{16}	Read DP_NODEID register
WR_NODEID	16_{16}	Write DP_NODEID register
RD_CLKTHRU	25_{16}	Read DP_CLKTHRU
RD_ERRLOG	45_{16}	Read DP_ERRLOG
WR_ERRLOG	46_{16}	Write DP_ERRLOG
RD_PC_CONFIG	55_{16}	Read DP_PC_CONFIG
WR_PC_CONFIG	56_{16}	Write DP_PC_CONFIG
RD_SQTAG0	65_{16}	Read DP_SENDQ_TAG0
RD_SQTAG1	75_{16}	Read DP_SENDQ_TAG1
RD_SQTAG2	85_{16}	Read DP_SENDQ_TAG2
RD_SQTAG3	95_{16}	Read DP_SENDQ_TAG3
BYPASS	FF_{16}	Connects TDI to bypass register to TDO.

9.3 BYPASS Instruction

The BYPASS instruction selects the single stage BYPASS shift register to be connected between PTDI and PTDO. This allows for a minimum delay shift path through the DataPump in the Shift-DR state. Capture-DR and Update-DR have no effect during this instruction. Execution of this instruction do not affect normal DataPump operation.

9.4 CSR Register Read/Write Instructions

These instructions provide shadow register access to the CSR support registers. The CSR support registers are a maximum of 32-bits in size. A 40-bit shift register (called a shadow register) is provided which is connected to PTDI and PTDO when CSR read/write instructions are loaded into the instruction register.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

The register read instructions parallel load data from the selected CSR register into the shadow shift register on Capture-DR. The CSR register data is parallel loaded in with bit 0 corresponding to bit 0 of the shadow register. The data is then shifted out on Shift-DR starting with bit 39.

Bits 33-39 are always 0. Bit 32 is the NIbusy bit and indicates that the Node Interface is currently reading a CSR register and using the register read multiplexers. If this bit is set, the read data loaded into the shadow register is undefined and the JTAG CSR read must be tried again until NIbusy is observed 0. The remaining bit ordering shifted out (bits 31 to 0) is the same order as specified for each CSR register in Section 4.8.

The register write instructions also parallel load data from the selected register on Capture-DR the same as a register read. During Shift-DR this read data is shifted out while the data to be written is shifted in. The write data is applied to the selected register on Update-DR.

The write data should be shifted in starting with 8 zeros (shadow register bits 32-39) followed by the write data starting with bit 31.

The write functionality applied on Update-DR is the same as if the data were written from the NIBus with a CSR write transaction. That is, if a bit of a register is read-only, the Update-DR won't change the bit. If a bit is clear-able, it will be cleared on Update-DR if the corresponding write data bit is set to one. Finally, if the bit is write-able, that bit will take the value of the corresponding bit in the write data on Update-DR. Full specification of each register bit and its access types are given in Section 4.8.

9.5 EXTEST Instruction

The EXTEST instruction allows testing of off-chip connections to the I/O and PC-board interconnections. Data at the input pins is captured in the boundary scan shift register in the Capture-DR state.

Data is shifted in the boundary scan shift register in the Shift-DR state.

The boundary scan output registers are parallel loaded and the outputs of the DataPump driven on the rising edge of PTCK in the Update-DR state.

9.6 SAMPLE/PRELOAD Instruction

The SAMPLE/PRELOAD instruction operates identically to the EXTEST instruction in the DataPump.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

9.7 Boundary Scan Register and Ordering

The boundary scan register consists of a bit for each I/O plus three extra bits which control the tristate and bi-directional outputs. These three control bits are called Q_TRIDPACK, Q_TRI6332, and Q_TRI310.

PDATA[0:63] and PPARITY[0:7] are bidirectional I/O and NDPACK is a tri-stateable output.

Table 9.2 gives the boundary scan ordering.

Signal pin TSTMD must be grounded during boundary scan testing. Signal pin PTDI connects to the head of the boundary scan list. PTDI feeds Q_TRIDPACK. Signal pin PTDO is fed from the tail of the boundary scan list. Signal PSCIDO15 feeds PTDO.

The boundary scan ordering is as follows: PTDI → Q_TRIDPACK →.....→ PSCIDO15→ PTDO.

I/O's not sampled in the boundary scan mode are the following: PTCLK, PTMS, PTDI, NTRST, CLKHI, NCLKHI, and PTDO.

Table 9.2: Boundary Scan Chain Ordering

Name	Type	Position in B.S. Chain	Control Pin
Q_TRIDPACK	Control	1	—
Q_TRI6332	Control	2	—
Q_TRI310	Control	3	—
NO CONNECT 1	—	4	—
NO CONNECT 2	—	5	—
NO CONNECT 3	—	6	—
NO CONNECT 4	—	7	—
NDPSEL	Input	8	—
PCMND2	Input	9	—
PCMND1	Input	10	—
PCMND0	Input	11	—
NNIACK	Input	12	—
NNIRDY	Input	13	—
PCKHMPY0	Input	14	—
PCKHMPY1	Input	15	—
NRESET	Input	16	—
PSTOP	Input	17	—
CLKNI	Input	18	—
PCKHSEL	Input	19	—
PTSTMD	Input	20	—
PSTEP	Input	21	—
PMAINTMD0	Input	22	—
PMAINTMD1	Input	23	—
PCLK250	Input	24	—
NSYNCRQ	Input	25	—
NSCRUB	Input	26	—
NDPACK	Output	27	Q_TRIDPACK

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Name	Type	Position in B.S. Chain	Control Pin
PRCVREQ	Output	28	—
PRCVRSP	Output	29	—
PSNDREQ	Output	30	—
PSNDRSP	Output	31	—
NINTRNI	Output	32	—
PCLKSTB	Output	33	—
PPC0	Output	34	—
PPC1	Output	35	—
NERRNI	Output	36	—
PPARITY0	I/O	37	Q_TRI6332
PPARITY1	I/O	38	Q_TRI6332
PPARITY2	I/O	39	Q_TRI6332
PPARITY3	I/O	40	Q_TRI6332
PPARITY4	I/O	41	Q_TRI310
PPARITY5	I/O	42	Q_TRI310
PPARITY6	I/O	43	Q_TRI310
PPARITY7	I/O	44	Q_TRI310
PDATA0	I/O	45	Q_TRI6332
PDATA1	I/O	46	Q_TRI6332
PDATA2	I/O	47	Q_TRI6332
PDATA3	I/O	48	Q_TRI6332
PDATA4	I/O	49	Q_TRI6332
PDATA5	I/O	50	Q_TRI6332
PDATA6	I/O	51	Q_TRI6332
PDATA7	I/O	52	Q_TRI6332
PDATA8	I/O	53	Q_TRI6332
PDATA9	I/O	54	Q_TRI6332
PDATA10	I/O	55	Q_TRI6332
PDATA11	I/O	56	Q_TRI6332
PDATA12	I/O	57	Q_TRI6332
PDATA13	I/O	58	Q_TRI6332
PDATA14	I/O	59	Q_TRI6332
PDATA15	I/O	60	Q_TRI6332
PDATA16	I/O	61	Q_TRI6332
PDATA17	I/O	62	Q_TRI6332
PDATA18	I/O	63	Q_TRI6332
PDATA19	I/O	64	Q_TRI6332
PDATA20	I/O	65	Q_TRI6332
PDATA21	I/O	66	Q_TRI6332

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Name	Type	Position in B.S. Chain	Control Pin
PDATA22	I/O	67	Q_TRI6332
PDATA23	I/O	68	Q_TRI6332
PDATA24	I/O	69	Q_TRI6332
PDATA25	I/O	70	Q_TRI6332
PDATA26	I/O	71	Q_TRI6332
PDATA27	I/O	72	Q_TRI6332
PDATA28	I/O	73	Q_TRI6332
PDATA29	I/O	74	Q_TRI6332
PDATA30	I/O	75	Q_TRI6332
PDATA31	I/O	76	Q_TRI6332
PDATA32	I/O	77	Q_TRI310
PDATA33	I/O	78	Q_TRI310
PDATA34	I/O	79	Q_TRI310
PDATA35	I/O	80	Q_TRI310
PDATA36	I/O	81	Q_TRI310
PDATA37	I/O	82	Q_TRI310
PDATA38	I/O	83	Q_TRI310
PDATA39	I/O	84	Q_TRI310
PDATA40	I/O	85	Q_TRI310
PDATA41	I/O	86	Q_TRI310
PDATA42	I/O	87	Q_TRI310
PDATA43	I/O	88	Q_TRI310
PDATA44	I/O	89	Q_TRI310
PDATA45	I/O	90	Q_TRI310
PDATA46	I/O	91	Q_TRI310
PDATA47	I/O	92	Q_TRI310
PDATA48	I/O	93	Q_TRI310
PDATA49	I/O	94	Q_TRI310
PDATA50	I/O	95	Q_TRI310
PDATA51	I/O	96	Q_TRI310
PDATA52	I/O	97	Q_TRI310
PDATA53	I/O	98	Q_TRI310
PDATA54	I/O	99	Q_TRI310
PDATA55	I/O	100	Q_TRI310
PDATA56	I/O	101	Q_TRI310
PDATA57	I/O	102	Q_TRI310
PDATA58	I/O	103	Q_TRI310
PDATA59	I/O	104	Q_TRI310
PDATA60	I/O	105	Q_TRI310

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Name	Type	Position in B.S. Chain	Control Pin
PDATA61	I/O	106	Q_TRI310
PDATA62	I/O	107	Q_TRI310
PDATA63	I/O	108	Q_TRI310
PSCIDI0	Input	109	—
PSCIDI1	Input	110	—
PSCIDI2	Input	111	—
PSCIDI3	Input	112	—
PSCIDI4	Input	113	—
PSCIDI5	Input	114	—
PSCIDI6	Input	115	—
PSCIDI7	Input	116	—
PSCIDI8	Input	117	—
PSCIDI9	Input	118	—
PSCIDI10	Input	119	—
PSCIDI11	Input	120	—
PSCIDI12	Input	121	—
PSCIDI13	Input	122	—
PSCIDI14	Input	123	—
PSCIDI15	Input	124	—
PSCIFI	Input	125	—
PSCISI	Input	126	—
PSCISO	Output	127	—
PSCIFO	Output	128	—
PSCIDO0	Output	129	—
PSCIDO1	Output	130	—
PSCIDO2	Output	131	—
PSCIDO3	Output	132	—
PSCIDO4	Output	133	—
PSCIDO5	Output	134	—
PSCIDO6	Output	135	—
PSCIDO7	Output	136	—
PSCIDO8	Output	137	—
PSCIDO9	Output	138	—
PSCIDO10	Output	139	—
PSCIDO11	Output	140	—
PSCIDO12	Output	141	—
PSCIDO13	Output	142	—
PSCIDO14	Output	143	—
PSCIDO15	Output	144	—

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

10.0 Internal Scan Test and Diagnostics

Scan test of on-chip RAMs through JTAG can be performed but is not documented here. No other internal chip state is available for scan. There is no built-in self test either.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

11.0 Electrical Specifications

11.1 DC Specifications

Table 11.1: Absolute Maximum Ratings

Symbol	Rating	Limit
V_{MM}	+2 V power supply voltage	-0.5 to 2.6 V
V_{TTL}	+3.3 V power supply voltage	-0.5 to 4.0 V
VinA	Any pad voltage except LVDS outputs and GTL input or output	-1.0 to V_{TTL} + 1.0 V
VinB	LVDS output or GTL I/O pad voltage	-1.0 to V_{DD} + 1.0 V
I_{OUTT}	TTL output short circuit current	TBD
I_{OUTG}	GTL output short circuit current (V_{GTLOUT} = 3 V)	TBD
Tc	Case temperature under bias	-55C to +125C
Tstg	Storage temperature	-65C to +150C

Table 11.2: Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units
V_{MM}	Supply voltage +2.0 V	1.9	2.0	2.1	V
V_{TTL}	Supply voltage +3.3 V	3.1	3.3	3.5	V
Tc	Operating case temperature	0	-	75	deg C

Table 11.3: Power Dissipation

Symbol	Parameter	Min	Typ	Max	Units
I_{MM}	Power supply current from V_{MM}	-	-	12.9	A
I_{TTL}	Power supply current from V_{TTL}	-	-	0.3	A
Pd	Total Power Dissipation	-	22	28.1	W

Table 11.4: TTL Input/Output DC Electrical Specifications *(Over recommended operating conditions.)*

Symbol	Parameter	Min	Typ	Max	Units
V_{OH}	Output high, I_{OH} = -2.4 mA	2.4	-	-	V
V_{OL}	Output low, I_{OL} = 8 mA	-	-	0.4	V
V_{IH}	Input high voltage [Not 5V Tolerant]	2.0	-	V_{TTL} + 1.0	V
V_{IL}	Input low voltage	- 1.0	-	0.8	V
I_{IH}	Input high current, V_{IN} = 2.4 V	-	-	50	uA
I_{IL}	Input low current, V_{IN} = 0.4 V	-1000	-	-	uA

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Table 11.5: GTL Input/Output DC Electrical Specifications *(Over recommended operating conditions. Termination resistance = 25*

Symbol	Parameter	Min	Typ	Max	Units
V_{TT}	Termination voltage	1.14	1.20	1.26	V
V_{GREF}	Reference voltage	(2/3)*V_{TT} - 2%	0.8	(2/3)*V_{TT} + 2%	V
I_{GREF}	Input reference current on V_{GREF}	-	-	10	mA
V_{OL}	Output low, I_{OL} = 50 mA	-	-	0.4	V
V_{IH}	Input high voltage	V_{REF} + 0.1	0.85	-	V
V_{IL}	Input low voltage	-	0.75	V_{REF} - 0.1	V
I_{ZOL}	Output leakage current, 0.0 <= V_{OUT} <= V_{TT} (output off)	-	-	+200	uA
I_{IL}	Input leakage current, 0.0 <= V_{IN} <= V_{TT}	-10	-	+10	uA

ohms.)

Table 11.6: LVDS Output Driver DC Electrical Specifications *(Over recommended operating conditions.)*

Symbol	Parameter	Conditions	Min	Max	Units
V_{OH}	Output voltage high	Rload = 100 ohms	1125	1600	mV
V_{OL}	Output voltage low	Rload = 100 ohms	800	1275	mV
V_{OD}	Output differential voltage	Rload = 100 ohms	250	400	mV
V_{OS}	Output offset voltage	Rload = 100 ohms	1000	1400	mV
dV_{OD}	Change in differential voltage between complementary states.	Rload = 100 ohms	-	25	mV
dV_{OS}	Change in offset voltage between complementary states.	Rload = 100 ohms	-	25	mV
R_O	Output impedance	Iload = 2 to 2.5 mA	35	65	ohms
dR_O	Ro mismatch	Iload = 2 to 2.5 mA	-	10	%

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Table 11.7: LVDS Input Receiver DC Electrical Specifications *(Over recommended operating conditions.)*

Symbol	Parameter	Conditions	Min	Max	Units
V_{IH}	Input voltage high	V_{ID}= 100mV	100	2200	mV
V_{IL}	Input voltage low	V_{ID}= 100mV	0	2100	mV
+/- V_{ID}	Input differential voltage		100	500	mV
V_{ICM}	Input voltage common mode	V_{ID}= V_{IDMIN}	50	2150	mV
		V_{ID}= V_{IDMAX}	250	1950	
R_{IN}	Input impedance	0 < V_{IN} < 2.4V	80	120	ohm

11.2 AC Timing Specifications

Table 11.8 shows the timing specifications for all the NIBus signals plus miscellaneous GTL level signals. All signals in this group are synchronous to CLKNI.

The inputs include PDATA[0:63], PPARITY[0:7], NDPSEL, PCMND[0:2], NNIACK, NNIRDY, NRESET, and NSYNCRQ.

The outputs include PDATA[0:63], PPARITY[0:7], NDPACK, PRCVREQ, PRCVRSP, PSNDREQ, PSNDRSP, NINTRNI, NERRNI, and PCLKSTB.

Table 11.8: NIBus AC Specifications *(Over recommended operating conditions.)*

Symbol	Parameter	Min	Max	Units	Figure
f_{NICLK}	CLKNI frequency.	0	66.7	MHz	-
T_{PER}	CLKNI period.	15	-	nS	11.1
t_R	CLKNI rise time	300	1200	pS	11.1
t_F	CLKNI fall time	300	1200	pS	11.1
t_{PW}	CLKNI pulse width	6	-	nS	11.1
t_{SU}	NIBus input setup time	4	-	nS	11.2
t_{HLD}	NIBus input hold time	0.5	-	nS	11.2
t_{CKO}	NIBus clock to output delay	2.0	8.5	nS	11.3
t_R	NIBus output rise time	500	1500	pS	11.1
t_F	NIBus output fall time	500	1500	pS	11.1
t_{HIZ}	NIBus clock to output hi-Z	2.0	12.0	ns	11.4

Table 11.9 gives the timing specifications for the SCI LVDS links. The input CLKHI is GTL level and is used to generate the internal SCI clock.

The SCI inputs include PSCIDI[0:15], PSCISI, and PSCIFI.

The SCI outputs include PSCIDO[0:15], PSCISO, and PSCIFO.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Table 11.9: SCI Link AC Specifications *(Over recommended operating conditions.)*

Symbol	Parameter	Min	Max	Units	Figure
f_{CLKHI}	CLKHI frequency	99.5	100.5	MHz	-
T_{JCLKHI}	CLKHI jitter tolerance	-	100	pS	-
t_R	CLKHI rise time	300	1200	pS	11.1
t_F	CLKHI fall time	300	1200	pS	11.1
t_{DC}	CLKHI duty cycle	40	60	%	-
t_R	SCI output rise time	300	500	pS	11.1
t_F	SCI output fall time	300	500	pS	11.1
t_{OSKEW}	SCI output differential skew	-	50	pS	-
t_{OSKEW}	SCI output to output skew	-	100	pS	-
t_{OSKEW}	SCI output pulse distortion	-	200	pS	-
t_R	SCI input rise time	300	800	pS	11.1
t_F	SCI input fall time	300	800	pS	11.1
t_{ISKEW}	SCI input to input skew	-	500	pS	-
t_{CKO}	PSCISO to PSCIDO delay	1800	2200	ps	11.3

Table 11.10 gives the timing values for relevant signals during TAP testing and internal scan mode testing.

For TAP test mode, the relevant inputs have TTL levels and are the clock PTCK, mode select PTMS, NTRST, and data in PTDI. The relevant output has TTL levels and is data out PTDO.

The relevant output is TTL level, PTDO

Table 11.10: .Test Mode AC Specifications Over recommended operating conditions.

Symbol	Parameter	Min	Max	Units	Figure
f_{TSTCK}	PTCK, CLKNI, CLKHI, input frequency	0	25	MHz	-
T_{TSTCK}	PTCK input period	40	-	nS	11.1
t_R	Input PTCK rise time	2	8	nS	11.1
t_F	Input PTCK fall time	2	8	nS	11.1
t_{PW}	PTCK input pulse width	15	-	nS	11.1
t_{SU}	Scan control and data inputs setup time to PTCK	10	-	nS	11.2
t_{HLD}	Scan control and data inputs hold time from TCK	2	-	nS	11.2
t_{CKO}	Scan data clock to output delay during test	2	8	nS	11.3

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Figure 11.1 : Period, Pulse Width, Rise, and Fall Time Definitions

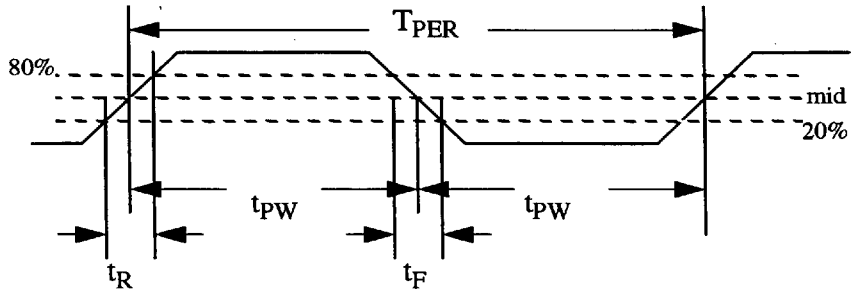


Figure 11.2: Setup and Hold Time Definitions

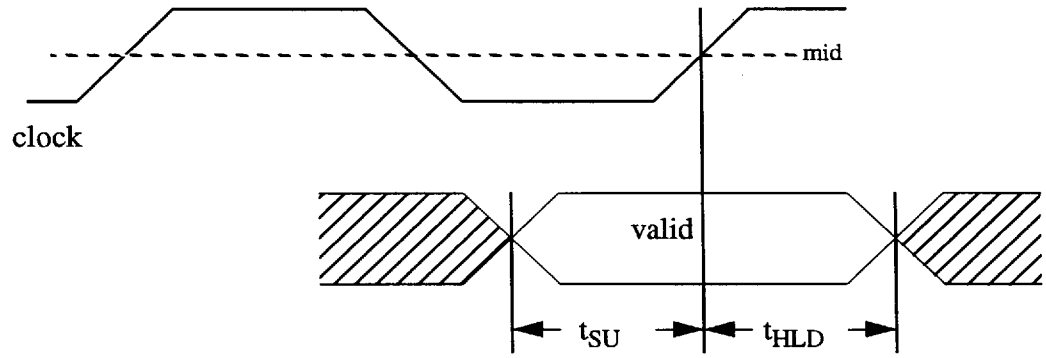
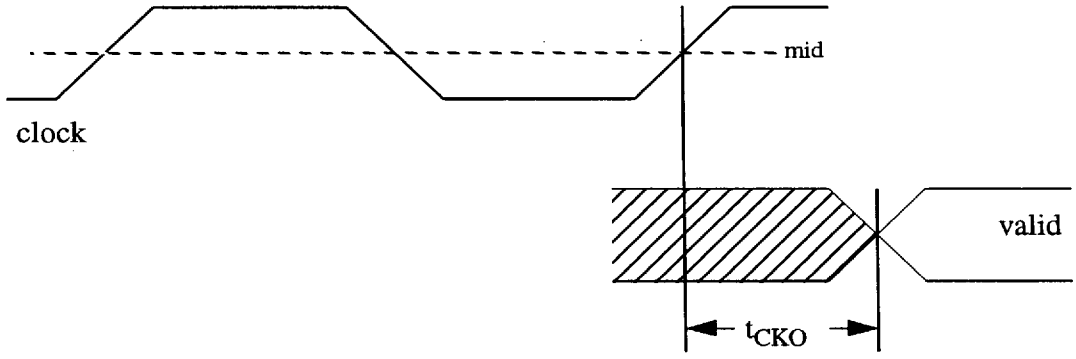


Figure 11.3 : Clock to Output Definition



741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

12.0 Package Pin Description *[No Pin at Locations A01, L01 and L21]*

Package Pin Number	Signal Name	Package Pin Number	Signal Name	Package Pin Number	Signal Name
A02	PDATA57	B17	PDATA28	D11	VMM
A03	PDATA55	B18	VCC	D12	VMM
A04	PDATA54	B19	PDATA25	D13	PDATA35
A05	PDATA50	B20	PDATA23	D14	PDATA31
A06	VCC	B21	VCC	D15	PDATA27
A07	PPARITY5	C01	VCC	D16	PPARITY2
A08	PDATA46	C02	PPARITY7	D17	VMM
A09	PDATA44	C03	PDATA63	D18	VCC
A10	PDATA43	C04	PDATA62	D19	NSCIDI0
A11	VCC	C05	PDATA59	D20	PSCIDI1
A12	PDATA40	C06	VMM	D21	VCC
A13	PPARITY4	C07	VCC	E01	NSCIDO9
A14	PDATA38	C08	VMM	E02	PSCIDO13
A15	PDATA36	C09	VMM	E03	PSCIDO15
A16	VCC	C10	VCC	E04	VMM
A17	PDATA33	C11	PDATA42	E18	VMM
A18	PDATA30	C12	VCC	E19	PSCIDI0
A19	PDATA29	C13	VMM	E20	PSCIDI2
A20	PDATA26	C14	VMM	E21	NSCIDI4
A21	VCC	C15	VCC	F01	NSCIDO12
B01	VCC	C16	VMM	F02	PSCIDO14
B02	PDATA61	C17	PDATA24	F03	NSCIDO11
B03	PDATA58	C18	PDATA22	F04	VCC
B04	VCC	C19	PDATA21	F18	VCC
B05	PPARITY6	C20	PDATA20	F19	NSCIDI1
B06	PDATA52	C21	VCC	F20	PSCIDI3
B07	PDATA51	D01	VCC	F21	NSCISI
B08	PDATA49	D02	PSCIDO11	G01	PSCIDO12
B09	PDATA47	D03	NSCIDO15	G02	PSCIDO9
B10	PDATA45	D04	VCC	G03	NSCIDO14
B11	PDATA41	D05	VMM	G04	NSCIDO13
B12	PDATA39	D06	PDATA60	G18	NSCIDI2
B13	PDATA37	D07	PDATA56	G19	NSCIDI3
B14	PDATA34	D08	PDATA53	G20	PSCIDI4
B15	PDATA32	D09	PDATA48	G21	PSCISI
B16	PPARITY3	D10	VMM	H01	NSCISO

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Package Pin Number	Signal Name	Package Pin Number	Signal Name	Package Pin Number	Signal Name
H02	NSCIDO8	N02	NSCIDO3	U21	PSCIDI15
H03	VCC	N03	PSCIDO3	V01	VCC
H04	VCC	N04	NSCIDO4	V02	PCMND2
H18	VCC	N18	NSCIDI13	V03	NDPSEL
H19	VCC	N19	PSCIDI12	V04	VCC
H20	NSCIFI	N20	NSCIDI12	V05	VMM
H21	NSCIDI6	N21	NSCIDI11	V06	VGREF
J01	PSCISO	P01	PSCIDO2	V07	NINTRNI
J02	PSCIDO10	P02	PSCIDO4	V08	PPC1
J03	NSCIDO10	P03	VCC	V09	PDATA4
J04	PSCIDO8	P04	VCC	V10	VMM
J18	PSCIFI	P18	VCC	V11	VMM
J19	NSCIDI5	P19	VCC	V12	VMM
J20	PSCIDI5	P20	PSCIDI13	V13	PDATA17
J21	PSCIDI6	P21	PSCIDI11	V14	PTDO
K01	NSCIDO6	R01	NSCIDO0	V15	PSTEP
K02	PSCIFO	R02	NSCIDO1	V16	NTRST
K03	NSCIFO	R03	NSCRUB	V17	VTTL
K04	VCC	R04	NNIACK	V18	VCC
K18	VCC	R18	PSTOP	V19	TE
K19	NSCIDI7	R19	CLKHI	V20	N/C
K20	PSCIDI7	R20	NSCIDI15	V21	VCC
K21	NSCIDI8	R21	NSCIDI14	W01	VCC
L02	NSCIDO7	T01	PSCIDO0	W02	IPNC
L03	PSCIDO6	T02	NSYNCRQ	W03	OPNC
L04	VMM	T03	PCMND1	W04	NDPACK
L18	VMM	T04	VCC	W05	PRCVRSP
L19	NSCIDI9	T18	VCC	W06	VMM
L20	PSCIDI8	T19	PCKHMPY1	W07	VCC
M01	PSCIDO7	T20	NCLKHI	W08	VMM
M02	NSCIDO5	T21	PSCIDI14	W09	VMM
M03	PSCIDO5	U01	PSCIDO1	W10	VCC
M04	VCC	U02	NNIRDY	W11	PDATA11
M18	VCC	U03	PCMND0	W12	VCC
M19	PSCIDI10	U04	VMM	W13	VMM
M20	NSCIDI10	U18	VMM	W14	VMM
M21	PSCIDI9	U19	PCKHMPY0	W15	VCC
N01	NSCIDO2	U20	NRESET	W16	VTTL

Package Pin Number	Signal Name	Package Pin Number	Signal Name	Package Pin Number	Signal Name
W17	PTCK	Y12	PDATA14	AA07	PDATA5
W18	PMAINTMD1	Y13	PPARITY1	AA08	PDATA7
W19	CLKNI	Y14	PDATA18	AA09	PDATA8
W20	N/C	Y15	N/C	AA10	PDATA9
W21	VCC	Y16	PDIV10OUT	AA11	VCC
Y01	VCC	Y17	PTSTMD	AA12	PDATA12
Y02	PRCVREQ	Y18	VCC	AA13	PDATA13
Y03	PSNDREQ	Y19	PTMS	AA14	PDATA15
Y04	VCC	Y20	PMAINTMD0	AA15	PDATA16
Y05	NERRNI	Y21	VCC	AA16	VCC
Y06	PDATA0	AA01	VCC	AA17	PDATA19
Y07	PDATA1	AA02	PSNDRSP	AA18	PCKHSEL
Y08	PDATA3	AA03	PCLKSTB	AA19	PCLK250
Y09	PDATA6	AA04	PPC0	AA20	PTDI
Y10	PPARITY0	AA05	PDATA2	AA21	VCC
Y11	PDATA10	AA06	VCC		

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

13.0 Signal Pin Description

Signal Name	Package Pin Number	Signal Name	Package Pin Number	Signal Name	Package Pin Number
CLKHI	R19	NSCIDO14	G03	PDATA18	Y14
CLKNI	W19	NSCIDO15	D03	PDATA19	AA17
IPNC	W02	NSCIDO2	N01	PDATA2	AA05
N/C	V20	NSCIDO3	N02	PDATA20	C20
N/C	W20	NSCIDO4	N04	PDATA21	C19
N/C	Y15	NSCIDO5	M02	PDATA22	C18
NCLKHI	T20	NSCIDO6	K01	PDATA23	B20
NDPACK	W04	NSCIDO7	L02	PDATA24	C17
NDPSEL	V03	NSCIDO8	H02	PDATA25	B19
NERRNI	Y05	NSCIDO9	E01	PDATA26	A20
NINTRNI	V07	NSCIFI	H20	PDATA27	D15
NNIACK	R04	NSCIFO	K03	PDATA28	B17
NNIRDY	U02	NSCISI	F21	PDATA29	A19
NRESET	U20	NSCISO	H01	PDATA3	Y08
NSCIDI0	D19	NSCRUB	R03	PDATA30	A18
NSCIDI1	F19	NSYNCRQ	T02	PDATA31	D14
NSCIDI10	M20	NTRST	V16	PDATA32	B15
NSCIDI11	N21	OPNC	W03	PDATA33	A17
NSCIDI12	N20	PCKHMPY0	U19	PDATA34	B14
NSCIDI13	N18	PCKHMPY1	T19	PDATA35	D13
NSCIDI14	R21	PCKHSEL	AA18	PDATA36	A15
NSCIDI15	R20	PCLK250	AA19	PDATA37	B13
NSCIDI2	G18	PCLKSTB	AA03	PDATA38	A14
NSCIDI3	G19	PCMND0	U03	PDATA39	B12
NSCIDI4	E21	PCMND1	T03	PDATA4	V09
NSCIDI5	J19	PCMND2	V02	PDATA40	A12
NSCIDI6	H21	PDATA0	Y06	PDATA41	B11
NSCIDI7	K19	PDATA1	Y07	PDATA42	C11
NSCIDI8	K21	PDATA10	Y11	PDATA43	A10
NSCIDI9	L19	PDATA11	W11	PDATA44	A09
NSCIDO0	R01	PDATA12	AA12	PDATA45	B10
NSCIDO1	R02	PDATA13	AA13	PDATA46	A08
NSCIDO10	J03	PDATA14	Y12	PDATA47	B09
NSCIDO11	F03	PDATA15	AA14	PDATA48	D09
NSCIDO12	F01	PDATA16	AA15	PDATA49	B08
NSCIDO13	G04	PDATA17	V13	PDATA5	AA07

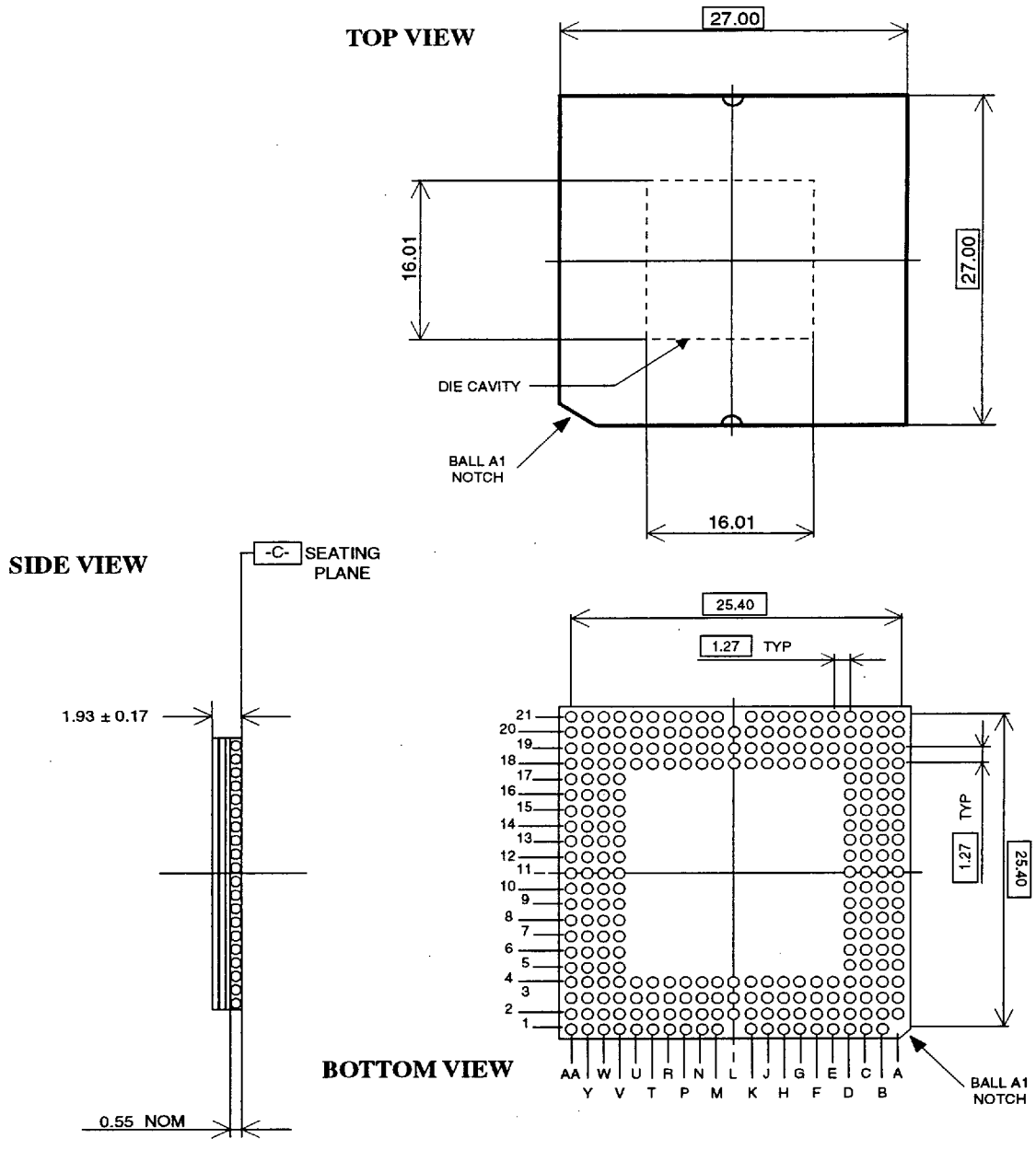
741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

Signal Name	Package Pin Number	Signal Name	Package Pin Number	Signal Name	Package Pin Number
PDATA50	A05	PSCIDI15	U21	TE	V19
PDATA51	B07	PSCIDI2	E20	VCC	A06
PDATA54	A04	PSCIDI3	F20	VCC	A11
PDATA55	A03	PSCIDI4	G20	VCC	A16
PDATA56	D07	PSCIDI5	J20	VCC	A21
PDATA57	A02	PSCIDI6	J21	VCC	B01
PDATA58	B03	PSCIDI7	K20	VCC	B04
PDATA59	C05	PSCIDI8	L20	VCC	B18
PDATA6	Y09	PSCIDI9	M21	VCC	B21
PDATA60	D06	PSCIDO0	T01	VCC	C01
PDATA61	B02	PSCIDO1	U01	VCC	C07
PDATA62	C04	PSCIDO10	J02	VCC	C10
PDATA63	C03	PSCIDO11	D02	VCC	C12
PDATA7	AA08	PSCIDO12	G01	VCC	C15
PDATA8	AA09	PSCIDO13	E02	VCC	C21
PDATA9	AA10	PSCIDO14	F02	VCC	D01
PDIV10OUT	Y16	PSCIDO15	E03	VCC	D04
PMAINTMD0	Y20	PSCIDO2	P01	VCC	D18
PMAINTMD1	W18	PSCIDO3	N03	VCC	D21
PPARITY0	Y10	PSCIDO4	P02	VCC	F04
PPARITY1	Y13	PSCIDO5	M03	VCC	F18
PPARITY2	D16	PSCIDO6	L03	VCC	H03
PPARITY3	B16	PSCIDO7	M01	VCC	H04
PPARITY4	A13	PSCIDO8	J04	VCC	H18
PPARITY5	A07	PSCIDO9	G02	VCC	H19
PPARITY6	B05	PSCIFI	J18	VCC	K04
PPARITY7	C02	PSCIFO	K02	VCC	K18
PPC0	AA04	PSCISI	G21	VCC	M04
PPC1	V08	PSCISO	J01	VCC	M18
PRCVREQ	Y02	PSNDREQ	Y03	VCC	P03
PRCVRSP	W05	PSNDRSP	AA02	VCC	P04
PSCIDI0	E19	PSTEP	V15	VCC	P18
PSCIDI1	D20	PSTOP	R18	VCC	P19
PSCIDI10	M19	PTCK	W17	VCC	T04
PSCIDI11	P21	PTDI	AA20	VCC	T18
PSCIDI12	N19	PTDO	V14	VCC	V01
PSCIDI13	P20	PTMS	Y19	VCC	V04
PSCIDI14	T21	PTSTMD	Y17	VCC	V18

Signal Name	Package Pin Number	Signal Name	Package Pin Number	Signal Name	Package Pin Number
VCC	V21	VCC	AA21	VMM	L04
VCC	W01	VGREF	V06	VMM	L18
VCC	W07	VMM	C06	VMM	U04
VCC	W10	VMM	C08	VMM	U18
VCC	W12	VMM	C09	VMM	V05
VCC	W15	VMM	C13	VMM	V10
VCC	W21	VMM	C14	VMM	V11
VCC	Y01	VMM	C16	VMM	V12
VCC	Y04	VMM	D05	VMM	W06
VCC	Y18	VMM	D10	VMM	W08
VCC	Y21	VMM	D11	VMM	W09
VCC	AA01	VMM	D12	VMM	W13
VCC	AA06	VMM	D17	VMM	W14
VCC	AA11	VMM	E04	VTTL	V17
VCC	AA16	VMM	E18	VTTL	W16

14.0 Mechanical Specifications

14.1 Package Information 269 BGA - 50 mil centers



15.0 Thermal Specifications

The VSC7201C requires a heatsink and substantial airflow for most applications. The thermal resistance from junction to case (θ jc) is 1.2°C/W. With a P_{DMAX} of 27W and a targeted junction temperature of 110°C, the case temperature should be kept under 75°C (Approximately 100°C - 28W *1.2°C/W).

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896

16.0 Ordering Information

The part number for this product is formed by a combination of the device number and the package style:

VSC7201C TE

Device Type:

VSC7201C: SCI DataPump

Package Style:

TE: 269 pin BGA; 27mm Body

Notice

This document contains information on products that are in the preproduction phase of development. The information contained in this document is based on test results and initial product characterization. Characteristic data and other specifications are subject to change without notice. Therefore, the reader is cautioned to confirm that this datasheet is current prior to placing orders.

Warning

Vitesse Semiconductor Corporation's products are not intended for use in life support appliances, devices or systems. Use of a Vitesse product in such applications without the written consent of the appropriate Vitesse officer is prohibited.

741 Calle Plano, Camarillo, CA 93012 • 805/388-3700 • FAX: 805/987-5896