DALLAS SEMICONDUCTOR

www.dalsemi.com

DS3131
BoSS Bit SynchronouS HDLC Controller

FEATURES

- 40 timing independent bit synchronous ports 40Rx & 40Tx) coupled with 40 independent Bi-directional HDLC channels
- Each port can operate up to 52 Mbps
- 132 Mbps full duplex throughput
- Diagnostic loopbacks in both directions
- Onboard Bit Error Rate Tester (BERT)
- Large 8KB FIFO in both receive and transmit directions
- Local bus allows for PCI bridging or local access
- 132 Mbps full duplex throughput
- Efficient scatter/gather DMA
- 25 MHz to 33 MHz 32-bit PCI backplane interface
- 3.3V low power CMOS with 5V tolerant I/O

DESCRIPTION

The DS3131 BoSS Bit Synchronous HDLC Controller can handle up to 40 channels of high speed bit synchronous HDLC. Unlike the DS3134 CHATEAU, the DS3131 BoSS does not contain any onboard support for channelized T1 or E1 data streams. Like the DS3134 CHATEAU, the onboard DMA has been optimized for maximum flexibility and PCI bus efficiency. A full suite of driver code for the device is available by contacting the factory.

The BoSS consists of the following blocks:

- Layer One
- HDLC
- FIFO
- DMA
- PCI Bus
- Local Bus

There are 40 High Speed Bit Synchronous HDLC Controllers (one for each port) that are capable of operating at speeds up to 52 Mbps in channelized.

Applications/Markets include:

- xDSL and Cable Modem Access Multiplexers (DSLAMs)
- High speed point to point connections like HSSI
- Clear channel (unchannelized) T1/E1
- Clear channel (unchannelized) T3/E3
- High density V.35 terminations

The device fully meets the following specifications: ANSI (American National Standards Institute) T1.403-1995 Network-to-Customer Installation DS1 Metallic Interface March 21, 1995 and PCI Local Bus Specification V2.1 June 1, 1995. ITU Q.921 March 1993 and ISO Standard 3309-1979 Data Communications – HDLC Procedures – Frame Structure.

REVISION HISTORY

Version 1 (8/6/98)
Original release.

Version 2 (1/26/99)
1. The octet synchronous functionality was removed.
2. The throughput rate was changed from 164 Mbps to 132 Mbps.
3. The maximum PCI Bus speed was changed from 50 MHz to 33 MHz.
4. The polarity of the LBPXS input signal was changed and a pull-up was added.
5. Added bit numbers to register descriptions.
6. The RCSO & TCSO outputs were removed.
7. Layer One port modes 2, 3, & 4 were removed.
8. The Master Control (MC) register was modified to allow the BERT Port Select Bits.
9. The transmit & receive DMA throttle bits were merged (see the MC register).
10. The COFA status bits were removed.
11. Changed BERT ones and zeros detector from 32 consecutive to 31 consecutive.
12. Changed BERT Bit and Error Counters to count during loss of receive synchronization.
13. Changed the HDLC channel priority scheme from high/low select to a quantized selection.
14. Added information about the DMA queues full & empty states.

Version 3 (5/1/2000)
1. Updated pin-out specifications.
2. Updated Bit 4 and Bit 5 in Master Configuration Register Description in Section 4.2.
3. Added Bits 1 to 15/Device Internal Test Bits Description in page 42.
4. Added note for Bit 0/ Force Resynchronization (RESYNC) Bit in Section 5.4 on page 49.
5. Corrected typo for BERTC1 Register Name on page 50.
6. Corrected typo in Figure 7.1A on page 50.
7. Page 54 – Corrected Transparent Mode in Table 6.1C.
8. Corrected acronym for address 0800 in Table 8.0A on page 71.
9. Corrected PCI error in Error conditions section on page 98.
10. Page 105 – Corrected bits in Register TDMAQ.
11. Corrected errors in Transmit Done Queue Descriptor in Figure 8.24A on page 106.
12. Corrected parameters in AC characteristics on page 156.

Version 4 (10/11/2000)
1. Tri-state capable was removed in Section 2.2.
2. Updated diagram in Figure 5.1A.
3. Removed the description of Bit 13 in Transmit Side Control Bits (TP[n]CR).
4. Updated the description of Octet Length Check in Table 6.1C.
5. Updated the description of Receive HDLC Channel Definition in Section 6.2.
6. Updated the information at Table 10.1A.
7. Updated figure 9.1C.

TABLE OF CONTENTS

Section 1:Introduction 5

Section 2: Signal Description 12
2.1 Overview / Signal Lead List 12
2.2 Serial Port Interface Signal Description 17
2.3 Local Bus Signal Description 18
2.4 JTAG Signal Description 21
2.5 PCI Bus Signal Description 22
2.6 Supply & Test Signal Description 25

Section 3: Memory Map 26
3.0 Introduction 26
3.1 General Configuration Registers 26
3.2 Receive Port Registers 27
3.3 Transmit Port Registers 28
3.4 HDLC Registers 29
3.5 BERT Registers 29
3.6 Receive DMA Registers 29
3.7 Transmit DMA Registers 30
3.8 FIFO Registers 30
3.9 PCI Configuration Registers for Function 0 31
3.10 PCI Configuration Registers for Function 1 32

Section 4: General Device Configuration & Status/Interrupt 32
4.1 Master Reset & ID Register Description 32
4.2 Master Configuration Register Description Interrupt 34
4.3.1 Status & Interrupt General Description 34
4.3.2 Status & Interrupt Register Description 36
4.4 Test Register Description 41

Section 5: Layer One 42
5.1 General Description 42
5.2 Port Register Description 44
5.3 BERT 46
5.4 BERT Register Description 47

Section 6: HDLC 52
6.1 General Description 52
6.2 HDLC Register Description 54

Section 7: FIFO 59
7.1 General Description & Example 59
7.2 FIFO Register Description 61

Section 8: DMA 67
8.0 Introduction 67
8.1 Receive Side 69
8.1.1 Overview 69
8.1.2 Packet Descriptors 73
8.1.3 Free Queue 75
8.1.4 Done Queue 79
8.1.5 DMA Configuration RAM 85
8.2 Transmit Side 88
8.2.1 Overview 88
8.2.2 Packet Descriptors 96
8.2.3 Pending Queue 98
8.2.4 Done Queue 101
8.2.5 DMA Configuration RAM 107

Section 9: PCI Bus 111
9.1 PCI General Description 111
9.2 PCI Configuration Register Description 118

Section 10: Local Bus 127
10.1 LocalBusGeneralDescription 147

Section 12: AC Characteristics 148

Section 13: Mechanical Dimensions 155

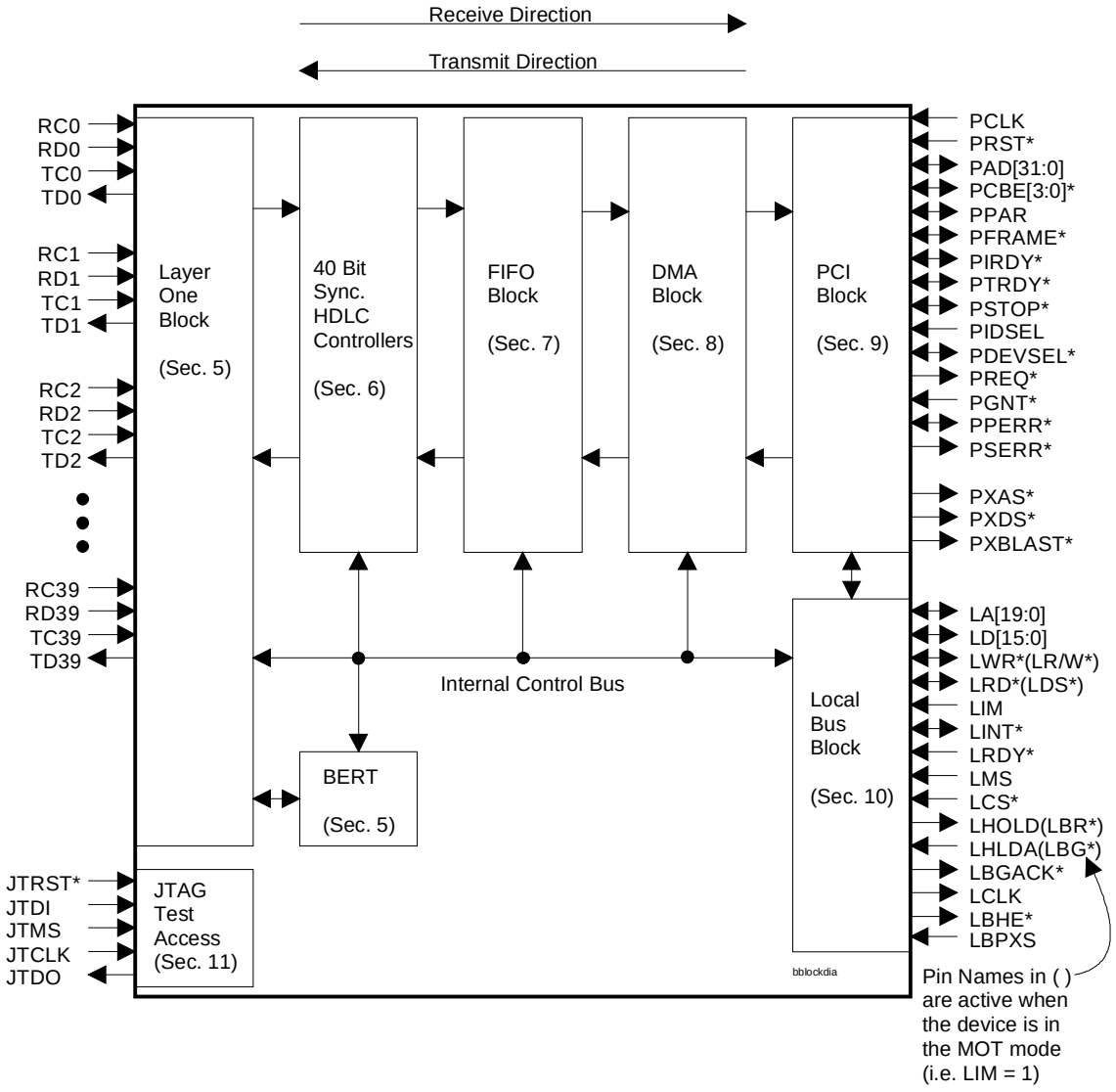
Section 14: Applications 156
14.1 T1/E1 and T3/E3 Applications 156
14.2 XDSL & Cable Modem Application 159
14.3 SONET/SDH Application 160

SECTION 1: INTRODUCTION

The DS3131 BoSS HDLC Controller is based on Dallas Semiconductor's DS3134 CHATEAU HDLC Controller. BoSS & CHATEAU share the same DMA & FIFO structure and the signal locations for the Local Bus and the PCI Bus are the same. The primary difference between the two devices is in the Layer One functionality. CHATEAU supports channelized T1/E1 whereas BoSS does not. Hence, the Layer One functions in CHATEAU that support channelized interfaces does not exist in BoSS.

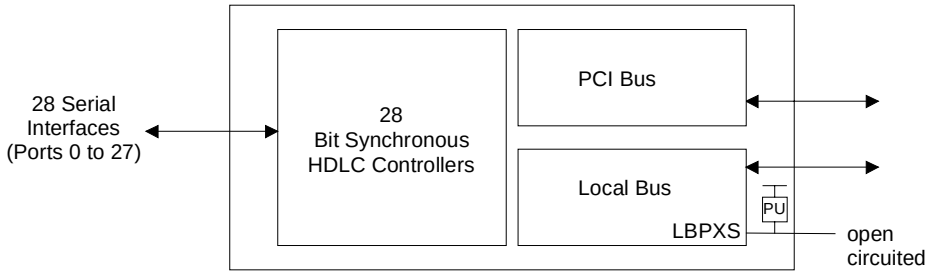
The DS3131 BoSS consists of 40 Bit Synchronous HDLC controllers. The primary features of the device are listed in Table 1A. This data sheet is split in Sections along the major blocks of the device as shown in Figure 1A. Throughout the data sheet, certain terms will be used and these terms are defined in Table 1B. The DS3131 device is designed to meet certain specifications and a listing of these governing specifications is shown in Table 1C. The DS3131 can be operated in two configurations depending on whether the Local Bus is enabled or not. See Figure 1B.

DS3131 BLOCK DIAGRAM Figure 1A

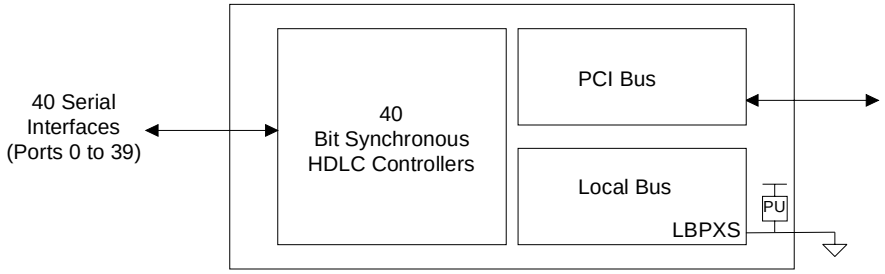


DS3131 CONFIGURATION OPTIONS Figure 1B

28 Bit Synchronous Ports / Local Bus Enabled (Default State)



40 Bit Synchronous Ports / Local Bus Disabled



DS3131 FEATURE LIST Table 1A

Layer One
- 40 Independent Bit Synchronous Physical Ports all Capable of Speeds Up to 52 Mbps
- Each Port can be Independently Configured
- Loopback in Both Directions (receive to transmit & transmit to receive)
- Onboard BERT Generation and Detection

HDLC
- 40 Independent Full Duplex HDLC Channels
- 132 Mbps throughput in both the Receive and Transmit Directions
- Transparent Mode
- Automatic Flag Detection and Generation
- Shared Opening and Closing Flag
- Interframe Fill
- Zero Stuffing and Destuffing
- CRC16/32 Checking and Generation
- Abort Detection and Generation
- CRC Error and Long/Short Frame Error Detection
- Bit Flip
- Invert Data

FIFO
- Large 8 kb Receive and 8 kb Transmit Buffers Maximize PCI Bus Efficiency
- Small Block Size of 16 bytes Allows Maximum Flexibility
- Programmable Low and High Water Marks
- Programmable HDLC Channel Priority Setting

DMA
- Efficient Scatter-Gather DMA Minimizes PCI Bus Accesses (same as the DS3134 Chateau)
- Programmable Small and Large Buffer Sizes Up to 8191 bytes & Algorithm Select
- Descriptor Bursting to Conserve PCI Bus Bandwidth
- Programmable Packet Storage Address Offset
- Identical Receive & Transmit Descriptors Minimize Host Processing in Store-and-Forward
- Automatic Channel Disabling and Enabling on Transmit Errors
- Receive Packets are Timestamped
- Transmit Packet Priority Setting

PCI Bus
- 32-Bit 33 MHz
- Version 2.1 Compliant
- Contains Extension Signals that Allow Adoption to Custom Buses
- Can Burst Up to 256 32-Bit Words to Maximize Bus Efficiency

Local Bus
- Can Operate as a Bridge from the PCI Bus or a Configuration Bus
- In Bridge Mode, Can Arbitrate for the Bus
- 8 or 16 Bits Wide
- In Bridge Mode, Supports a 1M Byte Address Space
- Supports Both Intel and Motorola Bus Timing

JTAG Test Access

3.3V Low Power CMOS with 5V Tolerant Inputs and Outputs

256 Lead Plastic BGA Package (27mm x 27mm)

DATA SHEET DEFINITIONS Table 1B

Note: The ports on the device are numbered 0 to 39 while the HDLC channels are numbered 1 to 40. HDLC Channel #1 is always associated with Port #0, HDLC Channel # 2 with Port #1, and so on.

Acronym Or Term	Definition
BERT	Bit Error Rate Tester.
Descriptor	A message passed back and forth between the DMA and the Host.
Dword	Double Word. A 32-bit data entity.
DMA	Direct Memory Access.
FIFO	First In First Out. A temporary memory storage scheme.
HDLC	High level Data Link Control.
Host	The main controller that resides on the PCI Bus.
n/a	Not Assigned.

GOVERNING SPECIFICATIONS Table 1C

PCI Local Bus Specification V2.1 June 1, 1995.

ITU Q.921 March 1993.

ISO Standard 3309-1979 Data Communications – HDLC Procedures – Frame Structure.

GENERAL DESCRIPTION

The Layer One Block handles the physical input and output of serial data to and from the DS3131. The DS3131 is capable of operating in a number of modes and can be used in many applications requiring high density and high speed HDLC termination. Section 14 details a few common applications for the DS3131. The Layer One Block prepares the incoming data for the HDLC Block and grooms data from the HDLC Block for transmission. The Layer One Block interfaces directly to the Bit Error Rate Tester (BERT) Block. The BERT Block can generate and detect both pseudorandom and repeating bit patterns and it is used to test and stress data communication links. The BERT Block is a global chip resource that can be assigned to any one of the 40 Bit Synchronous ports.

There are 40 Bit Synchronous HDLC Controllers (one for each port), each of that is capable of operating at speeds up to 52 Mbps. The Bit Synchronous HDLC Controller s has serial interfaces. The HDLC controllers perform all of the Layer 2 processing which include, zero stuffing and destuffing, flag generation and detection, CRC generation and checking, abort generation and checking.

In the receive path, the following process occurs. The HDLC controllers collect the incoming data and then signal the FIFO that the controller has data to transfer to the FIFO. The 40 ports are priority decoded (Port 0 gets the highest priority) for the transfer of data from the HDLC controllers to the FIFO Block. The priority of transfer between the HDLC controllers and the FIFO is of no real concern however since the DS3131 has been designed to handle up to 132 Mbps in both the receive and transmit directions without any potential loss of data due to priority conflicts.

The FIFO transfers data from the HDLC Engines into the FIFO and checks to see if the FIFO has filled to beyond the programmable High Water Mark. If it has, then the FIFO signals to the DMA that data is ready to be burst read from the FIFO to the PCI Bus. The FIFO Block controls the DMA Block and it tells the DMA when to transfer data from the FIFO to the PCI Bus. Since the DS3131 can handle multiple HDLC channels, it is quite possible that at any one time, several HDLC channels may need to have data transferred from the FIFO to the PCI Bus. The FIFO determines which HDLC channel the DMA will handle next via a Host configurable algorithm, which allows the selection to be either round robin or priority, decoded (with HDLC Channel 1 getting the highest priority). Depending on the application, the selection of this algorithm can be quite important. The DS3131 cannot control when it will be granted PCI Bus access and if bus access is restricted, then the Host may wish to prioritize which HDLC channels get top priority access to the PCI Bus when it is granted to the DS3131.

When the DMA transfers data from the FIFO to the PCI Bus, it burst reads all available data in the FIFO (even if the FIFO contains multiple HDLC packets) and tries to empty the FIFO. If an incoming HDLC packet is not large enough to fill the FIFO to the High Water Mark, then the FIFO will not wait for more data to enter the FIFO, it will signal the DMA that a End Of Frame (EOF) was detected and that data is ready to be transferred from the FIFO to the PCI Bus by the DMA.

In the transmit path, a very similar process occurs. As soon as a HDLC channel is enabled, the HDLC (Layer 2) Engines begin requesting data from the FIFO. Like the receive side, the 40 ports are priority decoded with Port 0 (HDLC Channel #1) getting the highest priority. Hence, if multiple ports are requesting packet data, the FIFO will first satisfy the requirements on all the enabled HDLC channels in the lower numbered ports before moving on to the higher numbered ports. Again there is no potential loss of data as long as the transmit throughput maximum of 132 Mbps is not exceeded. When the FIFO detects that a HDLC Engine needs data, it then transfers the data from the FIFO to the HDLC Engines. If the FIFO detects that the FIFO is below the Low Water Mark, it then checks with the DMA to see if there is any data available for that HDLC Channel. The DMA will know if any data is available because the Host on the PCI Bus will have informed it of such via the Pending Queue Descriptor. When the DMA detects that data is available, it informs the FIFO and then the FIFO decides which HDLC channel gets the highest priority to the DMA to transfer data from the PCI Bus into the FIFO. Again, since the DS3131 can handle multiple HDLC channels, it is quite possible that at any one time, several HDLC channels may need the DMA to burst data from the PCI Bus into the FIFO. The FIFO determines which HDLC channel the DMA will handle next via a Host configurable algorithm, which allows the selection to be either round robin or priority, decoded (with HDLC Channel 1 getting the highest priority).

When the DMA begins burst-writing data into the FIFO, it will try to completely fill the FIFO with HDLC packet data even if it that means writing multiple packets. Once the FIFO detects that the DMA has filled it to beyond the Low Water Mark (or an EOF is reached), the FIFO will begin transferring data to the HDLC controller.

One of the unique attributes of the DS3131 is the structure of the DMA. The DMA has been optimized to maintain maximum flexibility yet reduce the number of bus cycles required to transfer packet data. The DMA uses a flexible scatter/gather technique, which allows that packet data to be place anywhere within the 32-bit address space. The user has the option on the receive side of two different buffer sizes which are called “large” and “small” but that can be set to any size up to 8191 bytes. The user has the option to store the incoming data either, only in the large buffers, only in the small buffers, or fill a small buffer first and then fill large buffers as needed. The varying buffer storage options allow the user to make the best use of the available memory and to be able to balance the tradeoff between latency and bus utilization.

The DMA uses a set of descriptors to know where to store the incoming HDLC packet data and where to obtain HDLC packet data that is ready to be transmitted. The descriptors are fixed size messages that are handed back and forth from the DMA to the Host. Since this descriptor transfer utilizes bus cycles, the DMA has been structured to minimize the number of transfers required. For example on the receive side, the DMA obtains descriptors from the Host to know where in the 32-bit address space to place the incoming packet data. These descriptors are known as Free Queue Descriptors. When the DMA reads these descriptors off of the PCI Bus, they contain all the information that the DMA needs to know where to store the incoming data. Unlike other existing scatter/gather DMA architectures, the DS3131 DMA does not need to use any more bus cycles to determine where to place the data. Other DMA architectures tend to use pointers, which require them to go back onto the bus to obtain more information and hence use more bus cycles.

Another technique that the DMA uses to maximize bus utilization is the ability to burst read and write the descriptors. The device can be enabled to read and write the descriptors in bursts of 8 or 16 instead of one at a time. Since there is fixed overhead associated with each bus transaction, the ability to burst read and write descriptors allows the device to share the bus overhead among 8 or 16 descriptor transactions which reduces the total number of bus cycles needed.

The DMA can also burst up to 256 dwords (1024 bytes) onto the PCI Bus. This helps to minimize bus cycles by allowing the device to burst large amounts of data in a smaller number of bus transactions which reduces bus cycles by reducing the amount of fixed overhead that is placed on the bus.

When the Local Bus is enabled, ports 28 to 39 (HDLC Channels 29 to 40) are disabled to make room for the signals needed by the Local Bus. The Local Bus Block has two modes of operation. It can used as either a Bridge from the PCI Bus in which case it is a bus master or it can be used as a Configuration Bus in which case it is a bus slave. The Bridge Mode allows the Host on the PCI Bus to access the local bus. The DS3131 will map data from the PCI Bus to the local bus. In the Configuration Mode, the local bus is used only to control and monitor the DS3131 while the HDLC packet data will still be transferred to the Host via the PCI Bus.

RESTRICTIONS

In creating the overall system architecture, the user must balance the port, throughput, and HDLC channel restrictions of the DS3131. Table 1D lists all of the upper bound maximum restrictions on the DS3131.

DS3131 RESTRICTIONS Table 1D

Port	maximum of 40 physical ports maximum data rate of 52 Mbps
Throughput	maximum receive: 132 Mbps maximum transmit: 132 Mbps
HDLC	maximum of 40 channels

INTERNAL DEVICE CONFIGURATION REGISTERS

All of the internal device configuration registers (with the exception of the PCI Configuration Registers, which are 32-bit registers,) are 16 bits wide and they are not byte addressable. When the Host on the PCI Bus accesses these registers, the particular combination of byte enables (i.e. PCBE* signals) is not important but at least one of the byte enables must be asserted for a transaction to occur. All the registers are read/write registers unless otherwise noted. Not assigned bits (identified as n/a in the data sheet) should be set to zero when written to allow for future upgrades to the device. These bits have no meaning and could be either zero or one when read.

INITIALIZATION

On a system reset (which can be invoked by either hardware action via the PRST* signal or software action via the RST control bit in the Master Reset and ID register), all of the internal device configuration register are set to zero (0000h). Please note that the Local Bus Bridge Mode Control register (LBBMC) is not affected by a software invoked system reset, it will be forced to all zeros only by a hardware reset. The internal registers within that are accessed indirectly (these are listed as "indirect registers" in the data sheet and consist of the Port DS0 Configuration registers in the Layer One Block, the DMA Configuration RAMs, the HDLC Configuration registers, and the FIFO registers) are not affected by a system reset and they must be configured on power-up by the Host to a proper state. Figure 1C lists the ordered steps to initialize the DS3131.

Note: After device power up and reset, it takes 768 RC or TC clocks to get a port up and operating. In other words, the ports must have a minimum of 768 clocks before packet data can be processed.

INITIALIZATION STEPS Figure 1C

Initialization Step	Comments
1: Initialize the PCI Configuration Registers	Achieved by asserting the PIDSEL signal.
2: Initialize All Indirect Registers	It is recommended that all of the indirect registers be set to 0000h. Please note that only certain sections of the DMA Configuration Registers can be written to by the Host (details are in Section 8). See Table 1E.
3: Configure the Device for Operation	Program all the necessary registers, which includes the Layer One, HDLC, FIFO, and DMA registers.
4: Load the DMA Descriptors	Indicate to the DMA where packet data can be written and where pending data (if any) resides
5: Enable the DMAs	Done via the RDE and TDE control bits in the Master Configuration (MC) register.
6: Enable DMA for each HDLC Channel	Done via the Channel Enable bit in the Receive & Transmit Configuration RAM

INDIRECT REGISTERS Table 1E

Register Name (Acronym)	**Number of Indirect Registers**
Receive HDLC Channel Definition register (RHCD)	40 (one for each HDLC Channel)
Transmit HDLC Channel Definition register (THCD)	40 (one for each HDLC Channel)
Receive DMA Configuration register (RDMAC)	240 (three for each HDLC Channel) Note 1
Transmit DMA Configuration register (TDMAC)	480 (six for each HDLC Channel) Note 1
Receive FIFO Staring Block Pointer register (RFSBP)	40 (one for each HDLC Channel)
Receive FIFO Block Pointer register (RFBP)	512 (one for each FIFO Block)
Receive FIFO High Water Mark register (RFHWM)	40 (one for each HDLC Channel)
Transmit FIFO Staring Block Pointer register (TFSBP)	40 (one for each HDLC Channel)
Transmit FIFO Block Pointer register (TFBP)	512 (one for each FIFO Block)
Transmit FIFO Low Water Mark register (TFLWM)	40 (one for each HDLC Channel)

NOTE:

1. On device initialization, the Host only needs to write to one of the Receive and one of the Transmit DMA registers. See Sections 8.1.5 & 8.2.5 for details.

SECTION 2: SIGNAL DESCRIPTION

2.1 OVERVIEW / SIGNAL LEAD LIST

This section describes the input and output signals on the DS3131. Signal names follow a convention that is shown below. Table 2.1 lists all of the signals, their signal type, description, and lead location.

SIGNAL NAMING CONVENTION

First Letter	Signal Category	Section
R	Receive Serial Port	2.2
T	Transmit Serial Port	2.2
L	Local Bus	2.3
J	JTAG Test Port	2.4
P	PCI Bus	2.5

LXXX-T/RXXX Multiplexed local bus with extended ports controlled by LBPXS.

SIGNAL DESCRIPTION Table 2.1

Lead	Symbol	Type	Signal Description
W20	TC0	I	Transmit Serial Clock for Port 0.
U19	TC1	I	Transmit Serial Clock for Port 1.
T17	TC2	I	Transmit Serial Clock for Port 2
U20	TC3	I	Transmit Serial Clock for Port 3.
T19	TC4	I	Transmit Serial Clock for Port 4.
R18	TC5	I	Transmit Serial Clock for Port 5.
R19	TC6	I	Transmit Serial Clock for Port 6.
P18	TC7	I	Transmit Serial Clock for Port 7.
P20	TC8	I	Transmit Serial Clock for Port 8.
N19	TC9	I	Transmit Serial Clock for Port 9.
M17	TC10	I	Transmit Serial Clock for Port 10.
M19	TC11	I	Transmit Serial Clock for Port 11.
L19	TC12	I	Transmit Serial Clock for Port 12.
L20	TC13	I	Transmit Serial Clock for Port 13.
K19	TC14	I	Transmit Serial Clock for Port 14.
J20	TC15	I	Transmit Serial Clock for Port 15.
J18	TC16	I	Transmit Serial Clock for Port 16.
H19	TC17	I	Transmit Serial Clock for Port 17.
G20	TC18	I	Transmit Serial Clock for Port 18.
F20	TC19	I	Transmit Serial Clock for Port 19.
F19	TC20	I	Transmit Serial Clock for Port 20.
G17	TC21	I	Transmit Serial Clock for Port 21.
E19	TC22	I	Transmit Serial Clock for Port 22.
E18	TC23	I	Transmit Serial Clock for Port 23.
C20	TC24	I	Transmit Serial Clock for Port 24.
D18	TC25	I	Transmit Serial Clock for Port 25.
B20	TC26	I	Transmit Serial Clock for Port 26.
B19	TC27	I	Transmit Serial Clock for Port 27.
V19	TD0	O	Transmit Serial Data for Port 0.
U18	TD1	O	Transmit Serial Data for Port 1.
V20	TD2	O	Transmit Serial Data for Port 2.
T18	TD3	O	Transmit Serial Data for Port 3.
T20	TD4	O	Transmit Serial Data for Port 4.
P17	TD5	O	Transmit Serial Data for Port 5.
R20	TD6	O	Transmit Serial Data for Port 6.
P19	TD7	O	Transmit Serial Data for Port 7.
N18	TD8	O	Transmit Serial Data for Port 8.
N20	TD9	O	Transmit Serial Data for Port 9.
M18	TD10	O	Transmit Serial Data for Port 10.
M20	TD11	O	Transmit Serial Data for Port 11.
L18	TD12	O	Transmit Serial Data for Port 12.
K20	TD13	O	Transmit Serial Data for Port 13.
K18	TD14	O	Transmit Serial Data for Port 14.
J19	TD15	O	Transmit Serial Data for Port 15.
H20	TD16	O	Transmit Serial Data for Port 16.

Lead	Symbol	Type	Signal Description
H18	TD17	O	Transmit Serial Data for Port 17.
G19	TD18	O	Transmit Serial Data for Port 18.
G18	TD19	O	Transmit Serial Data for Port 19.
E20	TD20	O	Transmit Serial Data for Port 20.
F18	TD21	O	Transmit Serial Data for Port 21.
D20	TD22	O	Transmit Serial Data for Port 22.
D19	TD23	O	Transmit Serial Data for Port 23.
E17	TD24	O	Transmit Serial Data for Port 24.
C19	TD25	O	Transmit Serial Data for Port 25.
C18	TD26	O	Transmit Serial Data for Port 26.
A20	TD27	O	Transmit Serial Data for Port 27.
W9	NC	-	No Connect. Do not connect any signal to this lead.
U14	NC	-	No Connect. Do not connect any signal to this lead.
V17	PAD0	I/O	PCI Multiplexed Address & Data Bit 0.
U16	PAD1	I/O	PCI Multiplexed Address & Data Bit 1.
Y18	PAD2	I/O	PCI Multiplexed Address & Data Bit 2.
W17	PAD3	I/O	PCI Multiplexed Address & Data Bit 3.
V16	PAD4	I/O	PCI Multiplexed Address & Data Bit 4.
Y17	PAD5	I/O	PCI Multiplexed Address & Data Bit 5.
W16	PAD6	I/O	PCI Multiplexed Address & Data Bit 6.
V15	PAD7	I/O	PCI Multiplexed Address & Data Bit 7.
W15	PAD8	I/O	PCI Multiplexed Address & Data Bit 8.
V14	PAD9	I/O	PCI Multiplexed Address & Data Bit 9.
Y15	PAD10	I/O	PCI Multiplexed Address & Data Bit 10.
W14	PAD11	I/O	PCI Multiplexed Address & Data Bit 11.
Y14	PAD12	I/O	PCI Multiplexed Address & Data Bit 12.
V13	PAD13	I/O	PCI Multiplexed Address & Data Bit 13.
W13	PAD14	I/O	PCI Multiplexed Address & Data Bit 14.
Y13	PAD15	I/O	PCI Multiplexed Address & Data Bit 15.
V9	PAD16	I/O	PCI Multiplexed Address & Data Bit 16.
U9	PAD17	I/O	PCI Multiplexed Address & Data Bit 17.
Y8	PAD18	I/O	PCI Multiplexed Address & Data Bit 18.
W8	PAD19	I/O	PCI Multiplexed Address & Data Bit 19.
V8	PAD20	I/O	PCI Multiplexed Address & Data Bit 20.
Y7	PAD21	I/O	PCI Multiplexed Address & Data Bit 21.
W7	PAD22	I/O	PCI Multiplexed Address & Data Bit 22.
V7	PAD23	I/O	PCI Multiplexed Address & Data Bit 23.
U7	PAD24	I/O	PCI Multiplexed Address & Data Bit 24.
V6	PAD25	I/O	PCI Multiplexed Address & Data Bit 25.
Y5	PAD26	I/O	PCI Multiplexed Address & Data Bit 26.
W5	PAD27	I/O	PCI Multiplexed Address & Data Bit 27.
V5	PAD28	I/O	PCI Multiplexed Address & Data Bit 28.
Y4	PAD29	I/O	PCI Multiplexed Address & Data Bit 29.
Y3	PAD30	I/O	PCI Multiplexed Address & Data Bit 30.
U5	PAD31	I/O	PCI Multiplexed Address & Data Bit 31.
Y16	PCBE0*	I/O	PCI Bus Command / Byte Enable Bit 0.

Lead	Symbol	Type	Signal Description
V12	PCBE1*	I/O	PCI Bus Command / Byte Enable Bit 1.
Y9	PCBE2*	I/O	PCI Bus Command / Byte Enable Bit 2.
W6	PCBE3*	I/O	PCI Bus Command / Byte Enable Bit 3.
Y2	PCLK	I	PCI & System Clock. A 25 MHz to 33 MHz clock is applied here.
Y11	PDEVSEL*	I/O	PCI Device Select.
W10	PFRAME*	I/O	PCI Cycle Frame.
W4	PGNT*	I	PCI Bus Grant.
Y6	PIDSEL	I	PCI Initialization Device Select.
W18	PINT*	O	PCI Interrupt.
V10	PIRDY*	I/O	PCI Initiator Ready.
W12	PPAR	I/O	PCI Bus Parity.
V11	PPERR*	I/O	PCI Parity Error.
V4	PREQ*	O	PCI Bus Request.
W3	PRST*	I	PCI Reset.
Y12	PSERR*	O	PCI System Error.
W11	PSTOP*	I/O	PCI Stop.
Y10	PTRDY*	I/O	PCI Target Ready.
V18	PXAS*	O	PCI Extension Signal: Address Strobe.
Y20	PXBLAST*	O	PCI Extension Signal: Burst Last.
W19	PXDS*	O	PCI Extension Signal: Data Strobe.
Y1	RC0	I	Receive Serial Clock for Port 0.
V3	RC1	I	Receive Serial Clock for Port 1.
V2	RC2	I	Receive Serial Clock for Port 2.
T4	RC3	I	Receive Serial Clock for Port 3.
U2	RC4	I	Receive Serial Clock for Port 4.
U1	RC5	I	Receive Serial Clock for Port 5.
R3	RC6	I	Receive Serial Clock for Port 6.
T1	RC7	I	Receive Serial Clock for Port 7.
P3	RC8	I	Receive Serial Clock for Port 8.
P2	RC9	I	Receive Serial Clock for Port 9.
N3	RC10	I	Receive Serial Clock for Port 10.
N1	RC11	I	Receive Serial Clock for Port 11.
M2	RC12	I	Receive Serial Clock for Port 12.
L3	RC13	I	Receive Serial Clock for Port 13.
L1	RC14	I	Receive Serial Clock for Port 14.
K3	RC15	I	Receive Serial Clock for Port 15.
J1	RC16	I	Receive Serial Clock for Port 16.
J3	RC17	I	Receive Serial Clock for Port 17.
H1	RC18	I	Receive Serial Clock for Port 18.
H3	RC19	I	Receive Serial Clock for Port 19.
G2	RC20	I	Receive Serial Clock for Port 20.
F1	RC21	I	Receive Serial Clock for Port 21.
G4	RC22	I	Receive Serial Clock for Port 22.
E1	RC23	I	Receive Serial Clock for Port 23.
E3	RC24	I	Receive Serial Clock for Port 24.
C1	RC25	I	Receive Serial Clock for Port 25.

Lead	Symbol	Type	Signal Description
D3	RC26	I	Receive Serial Clock for Port 26.
C2	RC27	I	Receive Serial Clock for Port 27.
W2	RD0	I	Receive Serial Data for Port 0.
W1	RD1.	I	Receive Serial Data for Port 1.
U3	RD2	I	Receive Serial Data for Port 2.
V1	RD3	I	Receive Serial Data for Port 3.
T3	RD4	I	Receive Serial Data for Port 4.
T2	RD5	I	Receive Serial Data for Port 5.
P4	RD6	I	Receive Serial Data for Port 6.
R2	RD7	I	Receive Serial Data for Port 7.
R1	RD8	I	Receive Serial Data for Port 8.
P1	RD9	I	Receive Serial Data for Port 9.
N2	RD10	I	Receive Serial Data for Port 10.
M3	RD11	I	Receive Serial Data for Port 11.
M1	RD12	I	Receive Serial Data for Port 12,
L2	RD13	I	Receive Serial Data for Port 13.
K1	RD14	I	Receive Serial Data for Port 14.
K2	RD15	I	Receive Serial Data for Port 15.
J2	RD16	I	Receive Serial Data for Port 16.
J4	RD17	I	Receive Serial Data for Port 17.
H2	RD18	I	Receive Serial Data for Port 18.
G1	RD19	I	Receive Serial Data for Port 19.
G3	RD20	I	Receive Serial Data for Port 20.
F2	RD21	I	Receive Serial Data for Port 21.
F3	RD22	I	Receive Serial Data for Port 22.
E2	RD23	I	Receive Serial Data for Port 23.
D1	RD24	I	Receive Serial Data for Port 24.
E4	RD25	I	Receive Serial Data for Port 25.
D2	RD26	I	Receive Serial Data for Port 26.
B1	RD27	I	Receive Serial Data for Port 27.
A19	JTMS	I	JTAG IEEE 1149.1 Test Mode Select.
D16	JTDO	O	JTAG IEEE 1149.1 Test Serial Data Output.
B18	JTCLK	I	JTAG IEEE 1149.1 Test Serial Clock.
B17	JTRST	I	JTAG IEEE 1149.1 Test Reset.
C17	JTDI	I	JTAG IEEE 1149.1 Test Serial Data Input.
C10	NC	-	No Connect. Do not connect any signal to this lead.
C8	LA0-RD37	I/O-I	Local Bus Address Bit 0 - Receive Serial Data for Port 37.
A7	LA1-RC37	I/O -I	Local Bus Address Bit 1- Receive Serial Clock for Port 37.
B7	LA2-RD36	I/O-I	Local Bus Address Bit 2-Receive Serial Data for Port 36.
A6	LA3-RC36	I/O-I	Local Bus Address Bit 3-Receive Serial Clock for Port 36.
C7	LA4-RD35	I/O-I	Local Bus Address Bit 4 - Receive Serial Data for Port 35.
B6	LA5-RC35	I/O -I	Local Bus Address Bit 5 – Receive Serial Clock for Port 35.
A5	LA6-RD34	I/O-I	Local Bus Address Bit 6-Receive Serial Data for Port 34.
D7	LA7-RC34	I/O-I	Local Bus Address Bit 7-Receive Serial Clock for Port 34.
C6	LA8-RD33	I/O-I	Local Bus Address Bit 8 - Receive Serial Data for Port 33.
B5	LA9-RC33	I/O -I	Local Bus Address Bit 9 – Receive Serial Clock for Port 33.

Lead	Symbol	Type	Signal Description
A4	LA10-RD32	I/O-I	Local Bus Address Bit 10-Receive Serial Data for Port 32.
C5	LA11-RC32	I/O-I	Local Bus Address Bit 11-Receive Serial Clock for Port 32.
B4	LA12-RD31	I/O-I	Local Bus Address Bit 12 - Receive Serial Data for Port 31.
A3	LA13-RC31	I/O -I	Local Bus Address Bit 13 – Receive Serial Clock for Port 31.
D5	LA14-RD30	I/O-I	Local Bus Address Bit 14-Receive Serial Data for Port 30.
C4	LA15-RC30	I/O-I	Local Bus Address Bit 15-Receive Serial Clock for Port 30.
B3	LA16-RD29	I/O-I	Local Bus Address Bit 16 - Receive Serial Data for Port 29.
B2	LA17-RC29	I/O -I	Local Bus Address Bit 17 – Receive Serial Clock for Port 29.
A2	LA18-RD28	I/O-I	Local Bus Address Bit 18-Receive Serial Data for Port 28.
C3	LA19-RC28	I/O-I	Local Bus Address Bit 19-Receive Serial Clock for Port 28.
A18	LD0-TC28	I/O-I	Local Bus Data Bit 0 - Transmit Serial Clock for Port 28.
A17	LD1-TD28	I/O-O	Local Bus Data Bit 1 -Transmit Serial Data for Port 28.
C16	LD2-TC29	I/O-I	Local bus Data Bit 2 –Transmit Serial Clock for Port 29.
B16	LD3-TD29	I/O-O	Local Bus Data Bit 3 - Transmit Serial Data for Port 29.
A16	LD4-TC30	I/O-I	Local Bus Data Bit 4 - Transmit Serial Clock for Port 30.
C15	LD5-TD30	I/O-O	Local Bus Data Bit 5 –Transmit Serial Data for Port 30.
D14	LD6-TC31	I/O-I	Local Bus Data Bit 6-Transmit Serial Clock for Port 31.
B15	LD7-TD31	I/O-O	Local Bus Data Bit 7 - Transmit Serial Data for Port 31.
A15	LD8-TC32	I/O-I	Local Bus Data Bit 8 - Transmit Serial Clock for Port 32.
C14	LD9-TD32	I/OO	Local Bus Data Bit 9 –Transmit Serial Data for Port 32.
B14	LD10-TC33	I/O-I	Local Bus Data Bit 10-Transmit Serial Clock for Port 33.
A14	LD11-TD33	I/O-O	Local Bus Data Bit 11 - Transmit Serial Data for Port 33.
C13	LD12-TC34	I/O-I	Local Bus Data Bit 12-Transmit Serial Clock for Port 34.
B13	LD13-TD34	I/O-O	Local Bus Data Bit 13 –Transmit Serial Data for Port 34.
A13	LD14-TC35	I/O-I	Local Bus Data Bit 14-Transmit Serial Clock for Port 35.
D12	LD15-TD35	I/O-O	Local Bus Data Bit 15 - Transmit Serial Data for Port 35.
C9	LMS-RD39	I-I	Local Bus Mode Select - Receive Serial Data for Port 39.
B11	LBHE-TD37	O-O	Local Bus Byte High Enable - Transmit Serial Data for Port 37.
B10	LHOLD-TD39	O-O	Local Bus Hold (Local Bus Request) - Transmit Serial Data for Port 39.
A9	LWR-RC39	I/O-I	Local Bus Write Enable (Local Bus Read/Write Select) – Receive Serial Clock for Port 39.
C12	LIM-TC36	I-I	Local Bus Intel/Motorola Bus Select – Transmit Serial Clock for Port 36.
C11	LHLDA-TC38	I-I	Local Bus Hold Acknowledge (Local Bus Grant) – Transmit Serial Clock for Port 38.
B12	LCLK-TD36	O-O	Local bus Clock –Transmit Serial Data for Port 36.
A11	LBGACK-TD38	O-O	Local buses Grant Acknowledge –Transmit Serial Data for Port 38.
A8	LRD-RD38	I/O-I	Local Bus Read Enable (Local Bus Data Strobe) –Receive Serial Data for port 38.
A12	LINT-TC37	I/O-I	Local bus Interrupt-Transmit Serial Clock for Port 37.
A10	LRDY-TC39	I-I	Local Bus PCI Bridge Ready -Transmit Serial Clock for Port 39.
B9	LBPXS	I	Local Bus Port Extension Select. Leave open to enable Local Bus.
B8	LCS-RC38	I-I	Local Bus Chip Select –Receive Serial Data for Port 38.
Y19	TEST	I	Test. Factory test signals; leave open circuited.

Lead	Symbol	Type	Signal Description
D6	VDD	-	Positive Supply. 3.3V (+/- 10%).
D10	VDD	-	Positive Supply. 3.3V (+/- 10%).
D11	VDD	-	Positive Supply. 3.3V (+/- 10%).
D15	VDD	-	Positive Supply. 3.3V (+/- 10%).
F4	VDD	-	Positive Supply. 3.3V (+/- 10%).
F17	VDD	-	Positive Supply. 3.3V (+/- 10%).
K4	VDD	-	Positive Supply. 3.3V (+/- 10%).
K17	VDD	-	Positive Supply. 3.3V (+/- 10%).
L4	VDD	-	Positive Supply. 3.3V (+/- 10%).
L17	VDD	-	Positive Supply. 3.3V (+/- 10%).
R4	VDD	-	Positive Supply. 3.3V (+/- 10%).
R17	VDD	-	Positive Supply. 3.3V (+/- 10%).
U6	VDD	-	Positive Supply. 3.3V (+/- 10%).
U10	VDD	-	Positive Supply. 3.3V (+/- 10%).
U11	VDD	-	Positive Supply. 3.3V (+/- 10%).
U15	VDD	-	Positive Supply. 3.3V (+/- 10%).
A1	VSS	-	Ground Reference.
D4	VSS	-	Ground Reference.
D8	VSS	-	Ground Reference.
D9	VSS	-	Ground Reference.
D13	VSS	-	Ground Reference.
D17	VSS	-	Ground Reference.
H4	VSS	-	Ground Reference.
H17	VSS	-	Ground Reference.
J17	VSS	-	Ground Reference.
M4	VSS	-	Ground Reference.
N4	VSS	-	Ground Reference.
N17	VSS	-	Ground Reference.
U4	VSS	-	Ground Reference.
U8	VSS	-	Ground Reference.
U12	VSS	-	Ground Reference.
U13	VSS	-	Ground Reference.
U17	VSS	-	Ground Reference.

2.2 SERIAL PORT INTERFACE SIGNAL DESCRIPTION

Signal Name: **RC0 to RC39**
Signal Description: **Receive Serial Clock**
Signal Type: **Input**
Data can be clocked into the device either on falling edges (normal clock mode) or rising edges (inverted clock mode) of RC. This is programmable on a per port basis. RC can operate at speeds from DC to 52 MHz. Clock gapping is acceptable. If not used, should this signal should be tied low.

Signal Name: **RD0 to RD39**
Signal Description: **Receive Serial Data**
Signal Type: **Input**
Can be sampled either on the falling edge of RC (normal clock mode) or the rising edge of RC (inverted clock mode). If not used, this signal should be tied low.

Signal Name: **TC0 to TC39**
Signal Description: **Transmit Serial Clock**
Signal Type: **Input**
Data will be clocked out of the device at TD either on rising edges (normal clock mode) or falling edges (inverted clock mode) of TC. This is programmable on a per port basis. TC can operate at speeds from DC to 52 MHz. Clock gapping is acceptable. If not used, this signal should be tied low.

Signal Name: **TD0 to TD39**
Signal Description: **Transmit Serial Data**
Signal Type: **Output**
Can be updated either on the rising edge of TC (normal clock mode) or the falling edge of TC (inverted clock mode). TD can be forced either high or low via the TP[n]CR register. See Section 5.1 for details.

2.3 LOCAL BUS SIGNAL DESCRIPTION

Note: The signals listed in this section are only active when the Local Bus is enabled.

Signal Name: **LMS**
Signal Description: **Local Bus Mode Select**
Signal Type: **Input**
This signal should be tied low either when the device is to be operated with no Local Bus access or if the Local Bus will be used to act as a bridge from the PCI bus. This signal should be tied high if the Local Bus is to be used by an external host to configure the device.

0 = Local Bus is in the **PCI Bridge Mode** (master)
1 = Local Bus is in the **Configuration Mode** (slave)

Signal Name: **LIM**
Signal Description: **Local Bus Intel/Motorola Bus Select**
Signal Type: **Input**
The signal determines whether the Local Bus will operate in the Intel Mode (LIM = 0) or the Motorola Mode (LIM = 1). The signal names in parenthesis are operational when the device is in the Motorola Mode.

0 = Local Bus is in the **Intel Mode**
1 = Local Bus is in the **Motorola Mode**

Signal Name: **LBPXS**
Signal Description: **Local Bus or Port Extension Select**
Signal Type: **Input (with internal 10k pull up)**

This signal must be left open circuited (or tied high) to activate and enable the Local Bus. When this signal is tied low, the Local Bus is disabled and the Local Bus signals are redefined to support 12 Bit Synchronous HDLC Controllers on Ports 28 to 39. See Table 2.1C for details.

0 = Local Bus Disabled
1 (or open circuited) = Local Bus Enabled

Signal Name: **LD0 to LD15**
Signal Description: **Local Bus Non-Multiplexed Data Bus**
Signal Type: **Input / Output (tri-state capable)**

In PCI Bridge Mode (LMS = 0), data from/to the PCI bus can be transferred to/from these signals. When writing data to the Local Bus, these signals will be outputs and updated on the rising edge of LCLK. When reading data from the Local Bus, these signals will be inputs, which will be sampled on the rising edge of LCLK. Depending on the assertion of the PCI Byte Enables (PCBE0 to PCBE3) and the Local Bus Width (LBW) control bit in the Local Bus Bridge Mode Control Register (LBBMC), this data bus will utilize all 16-bits (LD[15:0]) or just the lower 8-bits (LD[7:0]) or the upper 8-bits (LD[15:8]). If the upper LD bits (LD[15:8]) are used, then the Local Bus High Enable signal (LBHE*) will be asserted during the bus transaction. If the Local Bus is not currently involved in a bus transaction, then all 16 signals will be tri-stated. In the Configuration Mode (LMS = 1), the external host will configure the device and obtain real time status information about the device via these signals. When reading data from the Local Bus, these signals will be outputs that are updated on the rising edge of LCLK. When writing data to the Local Bus, these signals will become inputs, which will be sampled on the rising edge of LCLK. In the Configuration Mode, only the 16-bit bus width is allowed (i.e. byte addressing is not available).

Signal Name: **LA0 to LA19**
Signal Description: **Local Bus Non-Multiplexed Address Bus**
Signal Type: **Input / Output (tri-state capable)**

In the PCI Bridge Mode (LMS = 0), these signals are outputs that will be asserted on the rising edge of LCLK to indicate which address to be written to or read from. If bus arbitration is enabled via the Local Bus Arbitration (LARBE) control bit in the Local Bus Bridge Mode Control Register (LBBMC), then these signals will be tri-stated when the Local Bus is not currently involved in a bus transaction and driven when a bus transaction is active. When bus arbitration is disabled, these signals are always driven. In the Configuration Mode (LMS = 1), these signals are inputs and only the bottom 16 (LA[15:0]) are active, the upper four (LA[19:16]) are ignored and should be tied low. These signals will be sampled on the rising edge of LCLK to determine the internal device configuration register that the external host wishes to access.

Signal Name: **LWR* (LR/W*)**
Signal Description: **Local Bus Write Enable (Local Bus Read/Write Select)**
Signal Type: **Input / Output (tri-state capable)**

In the PCI Bridge Mode (LMS = 0), this output signal is asserted on the rising edge of LCLK. In Intel Mode (LIM = 0) it will be asserted when data is to be written to the Local Bus. In Motorola Mode (LIM = 1), this signal will determine whether a read or write is to occur. If bus arbitration is enabled via the Local Bus Arbitration (LARBE) control bit in the Local Bus Bridge Mode Control Register (LBBMC), then this signal will be tri-stated when the Local Bus is not currently involved in a bus transaction and driven when a bus transaction is active. When bus arbitration is disabled, this signal is always driven. In the Configuration Mode (LMS = 1), this signal is sampled on the rising edge of LCLK. In Intel Mode (LIM = 0) it will determine when data is to be written to the device. In Motorola Mode (LIM = 1), this signal will be used to determine whether a read or write is to occur.

Signal Name: **LRD* (LDS*)**
Signal Description: **Local Bus Read Enable (Local Bus Data Strobe)**
Signal Type: **Input / Output (tri-state capable)**

In the PCI Bridge Mode (LMS = 0), this active low output signal is asserted on the rising edge of LCLK. In Intel Mode (LIM = 0) it will be asserted when data is to be read from the Local Bus. In Motorola Mode (LIM = 1), the rising edge will be used to write data into the slave device. If bus arbitration is enabled via the Local Bus Arbitration (LARBE) control bit in the Local Bus Bridge Mode Control Register (LBBMC), then this signal will be tri-stated when the Local Bus is not currently involved in a bus transaction and driven when a bus transaction is active. When bus arbitration is disabled, this signal is always driven. In the Configuration Mode (LMS = 1), this signal is an active low input which is sampled on the rising edge of LCLK. In Intel Mode (LIM = 0) it will determine when data is to be read from the device. In Motorola Mode (LIM = 1), the rising edge will be used to write data into the device.

Signal Name: **LINT***
Signal Description: **Local Bus Interrupt**
Signal Type: **Input / Output (open drain)**

In the PCI Bridge Mode (LMS = 0), this active low signal is an input which sampled on the rising edge of LCLK. If asserted and unmasked, this signal will cause an interrupt at the PCI bus via the PINTA* signal. If not used in the PCI Bridge Mode, this signal should be tied high. In the Configuration Mode (LMS = 1) this signal is an open drain output which will be forced low if one or more unmasked interrupt sources within the device is active. The signal will remain low until either the interrupt is serviced or masked.

Signal Name: **LRDY***
Signal Description: **Local Bus PCI Bridge Ready [PCI Bridge Mode Only]**
Signal Type: **Input**

This active low signal is sampled on the rising edge of LCLK to determine when a bus transaction is complete. This signal is only examined when a bus transaction is taking place. This signal is ignored when the Local Bus is in the Configuration Mode (LMS = 1) and should be tied high.

Signal Name: **LHLDA (LBG*)**
Signal Description: **Local Bus Hold Acknowledge (Local Bus Grant) [PCI Bridge Mode Only]**
Signal Type: **Input**
This input signal is sampled on the rising edge of LCLK to determine when the device has been granted access to the bus. In Intel Mode (LIM = 0) this is an active high signal and in Motorola Mode (LIM = 1) this is an active low signal. This signal is ignored and should be tied high when the Local Bus is in the Configuration Mode (LMS = 1). Also, in the PCI Bridge Mode (LMS = 0), this signal should be tied deasserted when the Local Bus Arbitration is disabled via the Local Bus Bridge Mode Control Register.

Signal Name: **LHOLD (LBR*)**
Signal Description: **Local Bus Hold (Local Bus Request) [PCI Bridge Mode Only]**
Signal Type: **Output**
This active low signal will be asserted when the Local Bus is attempting to take control of the bus. It will be deasserted in the Intel Mode (LIM = 0) when the bus access is complete. It will be deasserted in the Motorola Mode (LIM = 1) when the Local Bus Hold Acknowledge/Grant signal (LHLDA/LBG*) has been detected. This signal is tri-stated when the Local Bus is in the Configuration Mode (LMS = 1) and also in the PCI Bridge Mode (LMS = 0) when the Local Bus Arbitration is disabled via the Local Bus Bridge Mode Control Register.

Signal Name: **LBGACK***
Signal Description: **Local Bus Grant Acknowledge [PCI Bridge Mode Only]**
Signal Type: **Output (tri-state capable)**
This active low signal is asserted when the Local Bus Hold Acknowledge/Bus Grant signal (LHLDA/LBG*) has been detected and it continues it's assertion for a programmable (32 to 1048576) number of LCLKs based upon the Local Bus Arbitration Timer setting in the Local Bus Bridge Mode Control Register (LBBMC) register. This signal is tri-stated when the Local Bus is in the Configuration Mode (LMS = 1).

Signal Name: **LBHE***
Signal Description: **Local Bus Byte High Enable [PCI Bridge Mode Only]**
Signal Type: **Output (tri-state capable)**
This active low output signal is asserted when all 16-bits of the data bus (LD[15:0]) are active. It will remain high if only the lower 8-bits (LD [7:0)] are active. If bus arbitration is enabled via the Local Bus Arbitration (LARBE) control bit in the Local Bus Bridge Mode Control Register (LBBMC), then this signal will be tri-stated when the Local Bus is not currently involved in a bus transaction and driven when a bus transaction is active. When bus arbitration is disabled, this signal is always driven. This signal will remain in tri-state when the Local Bus is not currently involved in a bus transaction and when the Local Bus is in the Configuration Mode (LMS = 1).

Signal Name: **LCLK**
Signal Description: **Local Bus Clock [PCI Bridge Mode Only]**
Signal Type: **Output (tri-state capable)**
This signal outputs a buffered version of the clock applied at the PCLK input. All Local Bus signals are generated and sampled from this clock. This output is tri-stated when the Local Bus is in the Configuration Mode (LMS = 1). It can be disabled in the PCI Bridge Mode via the Local Bus Bridge Mode Control Register (LBBMC).

Signal Name: **LCS***
Signal Description: **Local Bus Chip Select [Configuration Mode Only]**
Signal Type: **Input**
This active low signal must be asserted for the device to accept a read or write command from an external host. This signal is ignored in the PCI Bridge Mode (LMS = 0) and should be tied high.

2.4 JTAG SIGNAL DESCRIPTION

Signal Name: **JTCLK**
Signal Description: **JTAG IEEE 1149.1 Test Serial Clock**
Signal Type: **Input**
This signal is used to shift data into JTDI on the rising edge and out of JTDO on the falling edge. If not used, this signal should be pulled high.

Signal Name: **JTDI**
Signal Description: **JTAG IEEE 1149.1 Test Serial Data Input**
Signal Type: **Input (with internal 10k pull up)**
Test instructions and data are clocked into this signal on the rising edge of JTCLK. If not used, this signal should be pulled high. This signal has an internal pull-up.

Signal Name: **JTDO**
Signal Description: **JTAG IEEE 1149.1 Test Serial Data Output**
Signal Type: **Output**
Test instructions are clocked out of this signal on the falling edge of JTCLK. If not used, this signal should be left open circuited.

Signal Name: **JTRST***
Signal Description: **JTAG IEEE 1149.1 Test Reset**
Signal Type: **Input (with internal 10k pull up)**
This signal is used to synchronously reset the test access port controller. At power up, JTRST must be set low and then high. This action will set the device into the boundary scan bypass mode allowing normal device operation. If boundary scan is not used, this signal should be held low. This signal has an internal pull-up.

Signal Name: **JTMS**
Signal Description: **JTAG IEEE 1149.1 Test Mode Select**
Signal Type: **Input (with internal 10k pull up)**
This signal is sampled on the rising edge of JTCLK and is used to place the test port into the various defined IEEE 1149.1 states. If not used, this signal should be pulled high. This signal has an internal pull-up.

2.5 PCI BUS SIGNAL DESCRIPTION

Signal Name: **PCLK**
Signal Description: **PCI & System Clock**
Signal Type: **Input (Schmitt triggered)**
This clock input is used to provide timing for the PCI bus and to the internal logic of the device. A 25 MHz to 33 MHz clock with a nominal 50% duty cycle should be applied here.

Signal Name: **PRST***
Signal Description: **PCI Reset**
Signal Type: **Input**
This active low input is used to force an asynchronous reset to both the PCI bus and the internal logic of the device. When forced low, this input forced all the internal logic of the device into its default state and it forces the PCI outputs into tri-state and the TD[39:0] output port data signals high.

Signal Name: **PAD0 to PAD31**
Signal Description: **PCI Address & Data Multiplexed Bus**
Signal Type: **Input / Output (tri-state capable)**
Both Address and Data information are multiplexed onto these signals. Each bus transaction consists of an address phase followed by one or more data phases. Data can be either read or written in bursts. During the first clock cycle of a bus transaction, the address is transferred. When the Little-Endian format is selected, PAD[31:24] is the msb of the DWORD, when Big-Endian is selected, PAD[7:0] contain the msb. When the device is an initiator, these signals are always outputs during the address phase. They remain outputs for the data phase(s) in a write transaction and become inputs for a read transaction. When the device is a target, these signals are always inputs during the address phase. They remain inputs for the data phase(s) in a read transaction and become outputs for a write transaction. When the device is not involved in a bus transaction, these signals remain tri-stated. These signals are always updated and sampled on the rising edge of PCLK.

Signal Name: **PCBE0* / PCBE1* / PCBE2* / PCBE3***
Signal Description: **PCI Bus Command and Byte Enable**
Signal Type: **Input / Output (tri-state capable)**
Bus Command and Byte Enables are multiplexed onto the same PCI signals. During an address phase, these signals define the Bus Command. During the data phase, these signals as used as Bus Enables. During data phases, PCBE0 refers to the PAD[7:0] and PCBE3 refers to PAD[31:24]. When this signal is high, the associated byte is invalid, when low, the associated byte is valid. When the device is an initiator, this signal is an output and is updated on the rising edge of PCLK. When the device is a target, this signal is an input and is sampled on the rising edge of PCLK. When the device is not involved in a bus transaction, these signals are tri-stated.

Signal Name: **PPAR**
Signal Description: **PCI Bus Parity**
Signal Type: **Input / Output (tri-state capable)**
This signal provides information on even parity across both the PAD address/data bus and the PCBE bus command/byte enable bus. When the device is an initiator, this signal is an output for writes and an input for reads and is updated on the rising edge of PCLK. When the device is a target, this signal is an input for writes and an output for reads and is sampled on the rising edge of PCLK. When the device is not involved in a bus transaction, PPAR is tri-stated.

Signal Name: **PFRAME***
Signal Description: **PCI Cycle Frame**
Signal Type: **Input / Output (tri-state capable)**
This active low signal is created by the bus initiator and is used to indicate the beginning and duration of a bus transaction. PFRAME* is asserted by the initiator during the first clock cycle of a bus transaction and it will remain asserted until the last data phase of a bus transaction. When the device is an initiator, this signal is an output and is updated on the rising edge of PCLK. When the device is a target, this signal is an input and is sampled on the rising edge of PCLK. When the device is not involved in a bus transaction, PFRAME* is tri-stated.

Signal Name: **PIRDY***
Signal Description: **PCI Initiator Ready**
Signal Type: **Input / Output (tri-state capable)**
This active low signal is created by the initiator to signal the target that it is ready to send/accept or to continue sending/accepting data. This signal handshakes with the PTRDY* signal during a bus transaction to control the rate at which data transfers across the bus. During a bus transaction, PIRDY* is deasserted when the initiator cannot temporarily accept or send data and a wait state is invoked. When the device is an initiator, this signal is an output and is updated on the rising edge of PCLK. When the device is a target, this signal is an input and is sampled on the rising edge of PCLK. When the device is not involved in a bus transaction, PIRDY* is tri-stated.

Signal Name: **PTRDY***
Signal Description: **PCI Target Ready**
Signal Type: **Input / Output (tri-state capable)**
This active low signal is created by the target to signal the initiator that it is ready to send/accept or to continue sending/accepting data. This signal handshakes with the PIRDY* signal during a bus transaction to control the rate at which data transfers across the bus. During a bus transaction, PTRDY* is deasserted when the target cannot temporarily accept or send data and a wait state is invoked. When the device is an target, this signal is an output and is updated on the rising edge of PCLK. When the device is a initiator, this signal is an input and is sampled on the rising edge of PCLK. When the device is not involved in a bus transaction, PTRDY* is tri-stated.

Signal Name: **PSTOP***
Signal Description: **PCI Stop**
Signal Type: **Input / Output (tri-state capable)**
This active low signal is created by the target to signal to the initiator that it requests the initiator stop the current bus transaction. When the device is an target, this signal is an output and is updated on the rising edge of PCLK. When the device is a initiator, this signal is an input and is sampled on the rising edge of PCLK. When the device is not involved in a bus transaction, PSTOP* is tri-stated.

Signal Name: **PIDSEL**
Signal Description: **PCI Initialization Device Select**
Signal Type: **Input**
This input signal is used as a chip select during configuration read and write transactions. **This signal is disabled when the Local Bus is set in the Configuration Mode (LMS = 1).** When PIDSEL is set high during the address phase of a bus transaction and the Bus Command signals (PCBE0 to PCBE3) indicate a register read or write, then the device allows access to the PCI configuration registers and the PDEVSEL* signal is asserted during the PCLK cycle. PIDSEL is sampled on the rising edge of PCLK.

Signal Name: **PDEVSEL***
Signal Description: **PCI Device Select**
Signal Type: **Input / Output (tri-state capable)**
This active low signal is created by the target when it has decoded the address sent to it by the initiator as it's own to indicate that that the address is valid. If the device is an initiator and does not see the this signal asserted within six PCLK cycles, then the bus transaction is aborted and the PCI Host is alerted. When the device is an target, this signal is an output and is updated on the rising edge of PCLK. When the device is a initiator, this signal is an input and is sampled on the rising edge of PCLK. When the device is not involved in a bus transaction, PDEVSEL* is tri-stated.

Signal Name: **PREQ***
Signal Description: **PCI Bus Request**
Signal Type: **Output (tri-state capable)**
This active low signal is asserted by the initiator to request that the PCI bus arbiter allow it access to the bus. PREQ* is updated on the rising edge of PCLK.

Signal Name: **PGNT***
Signal Description: **PCI Bus Grant**
Signal Type: **Input**
This active low signal is asserted by the PCI bus arbiter to indicate to the PCI requesting agent that access to the PCI bus has been granted. The device samples PGNT* on the rising edge of PCLK and if detected, will initiate a bus transaction when it has sensed that the PFRAME* signal has been deasserted.

Signal Name: **PPERR***
Signal Description: **PCI Parity Error**
Signal Type: **Input / Output (tri-state capable)**
This active low signal reports parity errors that occur. PPERR* can be enabled and disabled via the PCI Configuration Registers. This signal is updated on the rising edge of PCLK.

Signal Name: **PSERR***
Signal Description: **PCI System Error**
Signal Type: **Output (open drain)**
This active low signal reports any parity errors that occur during the address phase. PSERR* can be enabled and disabled via the PCI Configuration Registers. This signal is updated on the rising edge of PCLK.

Signal Name: **PINTA***
Signal Description: **PCI Interrupt**
Signal Type: **Output (open drain)**
This active low (open drain) signal is asserted low asynchronously when the device is requesting attention from the device driver. PINTA will be deasserted when the device-interrupting source has been service or masked. This signal is updated on the rising edge of PCLK.

PCI EXTENSION SIGNALS

These signals are not part of the normal PCI Bus signal set. There are additional signals that are asserted when BoSS is an Initiator on the PCI Bus to help users interpret the normal PCI Bus signal set and connect them to a non-PCI environment like an Intel i960 type bus. The timing for these signals is shown below.

Signal Name: **PXAS***
Signal Description: **PCI Extension Address Strobe**
Signal Type: **Output**
This active low signal is asserted low on the same clock edge as PFRAME* and is deasserted after one clock period. This signal will only be asserted when the device is an initiator. This signal is an output and is updated on the rising edge of PCLK.

Signal Name: **PXDS***
Signal Description: **PCI Extension Data Strobe**
Signal Type: **Output**
This active low signal is asserted when the PCI bus either contains valid data to be read from the device or can accept valid data that is written into the device. This signal will only be asserted when the device is an initiator. This signal is an output and is updated on the rising edge of PCLK.

Signal Name: **PXBLAST***
Signal Description: **PCI Extension Burst Last**
Signal Type: **Output**
This active low signal is asserted on the same clock edge as PFRAME* is deasserted and is deasserted on the same clock edge as PIRDY* is deasserted. This signal will only be asserted when the device is an initiator. This signal is an output and is updated on the rising edge of PCLK.

2.6 SUPPLY & TEST SIGNAL DESCRIPTION

Signal Name: **TEST**
Signal Description: **Factory Test Input**
Signal Type: **Input (with internal 10k pull up)**
This input should be left open circuited by the user.

Signal Name: **VDD**
Signal Description: **Positive Supply**
Signal Type: **n/a**
3.3V (+/- 10%). All VDD signals should be tied together.

Signal Name: **VSS**
Signal Description: **Ground Reference**
Signal Type: **n/a**
All VSS signals should be tied to the local ground plane.

SECTION 3: MEMORY MAP

3.0 INTRODUCTION

All addresses within the memory map on dword boundaries even though all of the internal device configuration registers are only one word (16 bits) wide. The memory map consumes an address range of 4KB (12 bits). When the PCI Bus is the Host (i.e. the Local Bus is in the Bridge Mode), the actual 32-bit PCI Bus addresses of the internal device configuration registers is obtained by adding the DC Base Address value in the PCI Device Configuration Memory Base Address Register (see Section 9.2 for details) to the *offset* listed in Sections 3.1 to 3.10. When an external host is configuring the device via the Local Bus (i.e. the Local Bus is in the Configuration Mode), the offset is 0h and the Host on the Local Bus will use the 16-bit *addresses* listed in Sections 3.1 to 3.10.

MEMORY MAP ORGANIZATION Table 3.0A

Sec.	Register Name	PCI Host [offset from DC Base]	Local Bus Host (16-bit address)
3.1	General Configuration Registers	(0x000)	(00xx)
3.2	Receive Port Registers	(0x1xx)	(01xx)
3.3	Transmit Port Registers	(0x2xx)	(02xx)
3.4	HDLC Registers	(0x4xx)	(04xx)
3.5	BERT Registers	(0x5xx)	(05xx)
3.6	Receive DMA Registers	(0x7xx)	(07xx)
3.7	Transmit DMA Registers	(0x8xx)	(08xx)
3.8	FIFO Registers	(0x9xx)	(09xx)
3.9	PCI Configuration Registers for Function 0	(PIDSEL)	(0Axx)
3.10	PCI Configuration Registers for Function 1	(PIDSEL)	(0Bxx)

3.1 GENERAL CONFIGURATION REGISTERS (0xx)

Offset/Address	Acronym	Register Name	Section
0000	MRID	Master Reset & ID Register.	4.1
0010	MC	Master Configuration.	4.2
0020	SM	Master Status Register.	4.3.2
0024	ISM	Interrupt Mask Register for SM.	4.3.2
0028	SDMA	Status Register for DMA.	4.3.2
002C	ISDMA	Interrupt Mask Register for SDMA.	4.3.2
0040	LBBMC	Local Bus Bridge Mode Control Register.	10.2
0050	TEST	Test Register.	4.4

3.2 RECEIVE PORT REGISTERS (1xx)

Offset/Address	Acronym	Register Name	Section
0100	RP0CR	Receive Port 0 Control Register.	5.2
0104	RP1CR	Receive Port 1 Control Register.	5.2
0108	RP2CR	Receive Port 2 Control Register.	5.2
010C	RP3CR	Receive Port 3 Control Register.	5.2
0110	RP4CR	Receive Port 4 Control Register.	5.2
0114	RP5CR	Receive Port 5 Control Register.	5.2
0118	RP6CR	Receive Port 6 Control Register.	5.2
011C	RP7CR	Receive Port 7 Control Register.	5.2
0120	RP8CR	Receive Port 8 Control Register.	5.2
0124	RP9CR	Receive Port 9 Control Register.	5.2
0128	RP10CR	Receive Port 10 Control Register.	5.2
012C	RP11CR	Receive Port 11 Control Register.	5.2
0130	RP12CR	Receive Port 12 Control Register.	5.2
0134	RP13CR	Receive Port 13 Control Register.	5.2
0138	RP14CR	Receive Port 14 Control Register.	5.2
013C	RP15CR	Receive Port 15 Control Register.	5.2
0140	RP16CR	Receive Port 16 Control Register.	5.2
0144	RP17CR	Receive Port 17 Control Register.	5.2
0148	RP18CR	Receive Port 18 Control Register.	5.2
014C	RP19CR	Receive Port 19 Control Register.	5.2
0150	RP20CR	Receive Port 20 Control Register.	5.2
0154	RP21CR	Receive Port 21 Control Register.	5.2
0158	RP22CR	Receive Port 22 Control Register.	5.2
015C	RP23CR	Receive Port 23 Control Register.	5.2
0160	RP24CR	Receive Port 24 Control Register.	5.2
0164	RP25CR	Receive Port 25 Control Register.	5.2
0168	RP26CR	Receive Port 26 Control Register.	5.2
016C	RP27CR	Receive Port 27 Control Register.	5.2
0170	RP28CR	Receive Port 28 Control Register.	5.2
0174	RP29CR	Receive Port 29 Control Register.	5.2
0178	RP30CR	Receive Port 30 Control Register.	5.2
017C	RP31CR	Receive Port 31 Control Register.	5.2
0180	RP32CR	Receive Port 32 Control Register.	5.2
0184	RP33CR	Receive Port 33 Control Register.	5.2
0188	RP34CR	Receive Port 34 Control Register.	5.2
018C	RP35CR	Receive Port 35 Control Register.	5.2
0190	RP36CR	Receive Port 36 Control Register.	5.2
0194	RP37CR	Receive Port 37 Control Register.	5.2
0198	RP38CR	Receive Port 38 Control Register.	5.2
019C	RP39CR	Receive Port 39 Control Register.	5.2

3.3 TRANSMIT PORT REGISTERS (2xx)

Offset/Address	Acronym	Register Name	Section
0200	TP0CR	Transmit Port 0 Control Register.	5.2
0204	TP1CR	Transmit Port 1 Control Register.	5.2
0208	TP2CR	Transmit Port 2 Control Register.	5.2
020C	TP3CR	Transmit Port 3 Control Register.	5.2
0210	TP4CR	Transmit Port 4 Control Register.	5.2
0214	TP5CR	Transmit Port 5 Control Register.	5.2
0218	TP6CR	Transmit Port 6 Control Register.	5.2
021C	TP7CR	Transmit Port 7 Control Register.	5.2
0220	TP8CR	Transmit Port 8 Control Register.	5.2
0224	TP9CR	Transmit Port 9 Control Register.	5.2
0228	TP10CR	Transmit Port 10 Control Register.	5.2
022C	TP11CR	Transmit Port 11 Control Register.	5.2
0230	TP12CR	Transmit Port 12 Control Register.	5.2
0234	TP13CR	Transmit Port 13 Control Register.	5.2
0238	TP14CR	Transmit Port 14 Control Register.	5.2
023C	TP15CR	Transmit Port 15 Control Register.	5.2
0240	TP16CR	Transmit Port 16 Control Register.	5.2
0244	TP17CR	Transmit Port 17 Control Register.	5.2
0248	TP18CR	Transmit Port 18 Control Register.	5.2
024C	TP19CR	Transmit Port 19 Control Register.	5.2
0250	TP20CR	Transmit Port 20 Control Register.	5.2
0254	TP21CR	Transmit Port 21 Control Register.	5.2
0258	TP22CR	Transmit Port 22 Control Register.	5.2
025C	TP23CR	Transmit Port 23 Control Register.	5.2
0260	TP24CR	Transmit Port 24 Control Register.	5.2
0264	TP25CR	Transmit Port 25 Control Register.	5.2
0268	TP26CR	Transmit Port 26 Control Register.	5.2
026C	TP27CR	Transmit Port 27 Control Register.	5.2
0270	TP28CR	Transmit Port 28 Control Register.	5.2
0274	TP29CR	Transmit Port 29 Control Register.	5.2
0278	TP30CR	Transmit Port 30 Control Register.	5.2
027C	TP31CR	Transmit Port 31 Control Register.	5.2
0280	TP32CR	Transmit Port 32 Control Register.	5.2
0284	TP33CR	Transmit Port 33 Control Register.	5.2
0288	TP34CR	Transmit Port 34 Control Register.	5.2
028C	TP35CR	Transmit Port 35 Control Register.	5.2
0290	TP36CR	Transmit Port 36 Control Register.	5.2
0294	TP37CR	Transmit Port 37 Control Register.	5.2
0298	TP38CR	Transmit Port 38 Control Register.	5.2
029C	TP39CR	Transmit Port 39 Control Register.	5.2

3.4 HDLC REGISTERS (4xx)

Offset/Address	Acronym	Register Name	Section
0400	RHCDIS	Receive HDLC Channel Definition Indirect Select.	6.2
0404	RHCD	Receive HDLC Channel Definition.	6.2
0410	RHPL	Receive HDLC Maximum Packet Length.	6.2
0480	THCDIS	Transmit HDLC Channel Definition Indirect Select.	6.2
0484	THCD	Transmit HDLC Channel Definition.	6.2

3.5 BERT REGISTERS (5xx)

Offset/Address	Acronym	Register Name	Section
0500	BERTC0	BERT Control 0.	5.6
0504	BERTC1	BERT Control 1.	5.6
0508	BERTRP0	BERT Repetitive Pattern Set 0 (lower word).	5.6
050C	BERTRP1	BERT Repetitive Pattern Set 1 (upper word).	5.6
0510	BERTBC0	BERT Bit Counter 0 (lower word).	5.6
0514	BERTBC1	BERT Bit Counter 1 (upper word).	5.6
0518	BERTEC0	BERT Error Counter 0 (lower word).	5.6
051C	BERTEC1	BERT Error Counter 1 (upper word).	5.6

3.6 RECEIVE DMA REGISTERS (7xx)

Offset/Address	Acronym	Register Name	Section
0700	RFQBA0	Receive Free Queue Base Address 0 (lower word).	8.1.3
0704	RFQBA1	Receive Free Queue Base Address 1 (upper word).	8.1.3
0708	RFQEA	Receive Free Queue End Address.	8.1.3
070C	RFQSBSA	Receive Free Queue Small Buffer Start Address.	8.1.3
0710	RFQLBWP	Receive Free Queue Large Buffer Host Write Pointer.	8.1.3
0714	RFQSBWP	Receive Free Queue Small Buffer Host Write Pointer.	8.1.3
0718	RFQLBRP	Receive Free Queue Large Buffer DMA Read Pointer.	8.1.3
071C	RFQSBRP	Receive Free Queue Small Buffer DMA Read Pointer.	8.1.3
0730	RDQBA0	Receive Done Queue Base Address 0 (lower word).	8.1.4
0734	RDQBA1	Receive Done Queue Base Address 1 (upper word).	8.1.4
0738	RDQEA	Receive Done Queue End Address.	8.1.4
073C	RDQRP	Receive Done Queue Host Read Pointer.	8.1.4
0740	RDQWP	Receive Done Queue DMA Write Pointer.	8.1.4
0744	RDQFFT	Receive Done Queue FIFO Flush Timer.	8.1.4
0750	RDBA0	Receive Descriptor Base Address 0 (lower word).	8.1.2
0754	RDBA1	Receive Descriptor Base Address 1 (upper word).	8.1.2
0770	RDMACIS	Receive DMA Configuration Indirect Select.	8.1.5
0774	RDMAC	Receive DMA Configuration.	8.1.5
0780	RDMAQ	Receive DMA Queues Control.	8.1.3/.4
0790	RLBS	Receive Large Buffer Size.	8.1.1
0794	RSBS	Receive Small Buffer Size.	8.1.1

3.7 TRANSMIT DMA REGISTERS (8xx)

Offset/Address	Acronym	Register Name	Section
0800	TPQBA0	Transmit Pending Queue Base Address 0 (lower word).	8.2.3
0804	TPQBA1	Transmit Pending Queue Base Address 1 (upper word).	8.2.3
0808	TPQEA	Transmit Pending Queue End Address.	8.2.3
080C	TPQWP	Transmit Pending Queue Host Write Pointer.	8.2.3
0810	TPQRP	Transmit Pending Queue DMA Read Pointer.	8.2.3
0830	TDQBA0	Transmit Done Queue Base Address 0 (lower word).	8.2.4
0834	TDQBA1	Transmit Done Queue Base Address 1 (upper word).	8.2.4
0838	TDQEA	Transmit Done Queue End Address.	8.2.4
083C	TDQRP	Transmit Done Queue Host Read Pointer.	8.2.4
0840	TDQWP	Transmit Done Queue DMA Write Pointer.	8.2.4
0844	TDQFFT	Transmit Done Queue FIFO Flush Timer.	8.2.4
0850	TDBA0	Transmit Descriptor Base Address 0 (lower word).	8.2.2
0854	TDBA1	Transmit Descriptor Base Address 1 (upper word).	8.2.2
0870	TDMACIS	Transmit DMA Configuration Indirect Select.	8.2.5
0874	TDMAC	Transmit DMA Configuration.	8.2.5
0880	TDMAQ	Transmit DMA Queues Control.	8.2.3/.4

3.8 FIFO REGISTERS (9xx)

Offset/Address	Acronym	Register Name	Section
0900	RFSBPIS	Receive FIFO Starting Block Pointer Indirect Select.	7.2
0904	RFSBP	Receive FIFO Starting Block Pointer.	7.2
0910	RFBPIS	Receive FIFO Block Pointer Indirect Select.	7.2
0914	RFBP	Receive FIFO Block Pointer.	7.2
0920	RFHWMIS	Receive FIFO High Water Mark Indirect Select.	7.2
0924	RFHWM	Receive FIFO High Water Mark.	7.2
0980	TFSBPIS	Transmit FIFO Starting Block Pointer Indirect Select.	7.2
0984	TFSBP	Transmit FIFO Starting Block Pointer.	7.2
0990	TFBPIS	Transmit FIFO Block Pointer Indirect Select.	7.2
0994	TFBP	Transmit FIFO Block Pointer.	7.2
09A0	TFLWMIS	Transmit FIFO Low Water Mark Indirect Select.	7.2
09A4	TFLWM	Transmit FIFO Low Water Mark.	7.2

3.9 PCI CONFIGURATION REGISTERS FOR FUNCTION 0 (PIDSEL/Axx)

Offset/Address	Acronym	Register Name	Section
0x000/0A00	PVID0	PCI Vendor ID / Device ID 0.	9.2
0x004/0A04	PCMD0	PCI Command Status 0.	9.2
0x008/0A08	PRCC0	PCI Revision ID / Class Code 0.	9.2
0x00C/0A0C	PLTH0	PCI Cache Line Size / Latency Timer / Header Type 0.	9.2
0x010/0A10	PDCM	PCI Device Configuration Memory Base Address.	9.2
0x03C/0A3C	PINTL0	PCI Interrupt Line & Pin / Min. Grant / Max. Latency 0.	9.2

3.10 PCI CONFIGURATION REGISTERS FOR FUNCTION 1 (PIDSEL/Bxx)

Offset/Address	Acronym	Register Name	Section
0x100/0B00	PVID1	PCI Vendor ID / Device ID 1.	9.2
0x104/0B04	PCMD1	PCI Command Status 1.	9.2
0x108/0B08	PRCC1	PCI Revision ID / Class Code 1.	9.2
0x10C/0B0C	PLTH1	PCI Cache Line Size / Latency Timer / Header Type 1.	9.2
0x110/0B10	PLBM	PCI Device Local Base Memory Base Address.	9.2
0x13C/0B3C	PINTL1	PCI Interrupt Line & Pin / Min. Grant / Max. Latency 1.	9.2

SECTION 4: GENERAL DEVICE CONFIGURATION & STATUS/INTERRUPT

4.1 MASTER RESET & ID REGISTER DESCRIPTION

The Master Reset & ID (MRID) register can be used to globally reset the device. When the RST bit is set to one, all of the internal registers (except the PCI configuration registers) will be placed into their default state, which is 0000h. The Host must set the RST bit back to zero before the device can be programmed for normal operation. The RST bit does not force the PCI outputs to tri-state as does the hardware reset which is invoked via the PRST* pin. A reset invoked by the PRST* pin will force the RST bit to zero as well as the rest of the internal configuration registers. See Section 1 for more details on device initialization.

The upper byte of the MRID register is read only and it can be read by the Host to determine the chip revision. Contact the factory for specifics on the meaning of the value read from the ID0 to ID7 bits.

Register Name: **MRID**
Register Description: **Master Reset and ID Register**
Register Address: **0000h**

7	6	5	4	3	2	1	0
n/a	n/a	n/a	n/a	n/a	n/a	n/a	RST
15	14	13	12	11	10	9	8
ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Master Software Reset (RST).
0 = normal operation
1 = force all internal registers (except LBBMC) to their default value of 0000h

Bits 8 to 15 / Chip Revision ID Bit 0 to 7 (ID0 to ID7). Read only. Contact the factory for details on the meaning of the ID bits.

4.2 MASTER CONFIGURATION REGISTER DESCRIPTION

The Master Configuration (MC) register is used by the Host to enable the receive and transmit DMAs as well as to control their PCI Bus bursting attributes and to select which port the BERT is to be dedicated to.

Register Name: **MC**
Register Description: **Master Configuration Register**
Register Address: **0010h**

7	6	5	4	3	2	1	0
BPS0	PBO	RFPC1	RFPC0	TDE	DT1	DT0	RDE
15	14	13	12	11	10	9	8
TFPC1	TFPC0	n/a	BPS5	BPS4	BPS3	BPS2	BPS1

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Receive DMA Enable (RDE). This bit is used to enable the receive DMA. When it is set to zero, the receive DMA will not pass any data from the receive FIFO to the PCI Bus even if there is one or more HDLC channels enabled. On device initialization, the Host should fully configure the receive DMA before enabling it via this bit.

0 = receive DMA is disabled
1 = receive DMA is enabled

Bit 1 / DMA Throttle Select Bit 0 (DT0).

Bit 2 / DMA Throttle Select Bit 1 (DT1).
These two bits select the maximum burst length that the receive and transmit DMA is allowed on the PCI Bus. The DMA can be restricted to a maximum burst length of just 32 dwords (128 bytes) or it can be incrementally adjusted up to 256 dwords (1024 bytes). The Host will select the optimal length based on a number of factors including the system environment for the PCI Bus, the number of HDLC channels being used, and the trade off between channel latency and bus efficiency.

00 = burst length maximum is 32 dwords
01 = burst length maximum is 64 dwords
10 = burst length maximum is 128 dwords
11 = burst length maximum is 256 dwords

Bit 3 / Transmit DMA Enable (TDE). This bit is used to enable the transmit DMA. When it is set to zero, the transmit DMA will not pass any data from the PCI Bus to the transmit FIFO even if there is one or more HDLC channels enabled. On device initialization, the Host should fully configure the transmit DMA before enabling it via this bit.

0 = transmit DMA is disabled
1 = transmit DMA is enabled

Bit 4 / Receive FIFO Priority Control Bit 0 (RFPC0).

Bit 5 / Receive FIFO Priority Control Bit 1 (RFPC1).
These two bits select the algorithm the FIFO will use to determine which HDLC Channel gets the highest priority to the DMA to transfer data from the FIFO to the PCI Bus. In the priority-decoded scheme, the lower the HDLC channels number, the higher the priority.

00 = all HDLC channels are serviced Round Robin
01 = HDLC Channels 1 & 2 are Priority Decoded; other HDLC Channels are Round Robin
10 = HDLC Channels 1 to 8 are Priority Decoded; other HDLC Channels are Round Robin
11 = HDLC Channels 1 to 16 are Priority Decoded; other HDLC Channels are Round Robin

Bit 6 / PCI Bus Orientation (PBO).
This bit selects whether HDLC packet data on the PCI Bus will operate in either Little Endian format or Big Endian formats. Little Endian byte ordering places the least significant byte at the lowest address while Big Endian places the least significant byte at the highest address. This bit setting only affects HDLC data on the PCI Bus. All other PCI Bus transactions to the internal device configuration registers, PCI configuration registers, and Local Bus, are always in Little Endian format.

0 = HDLC Packet Data on the PCI Bus is in Little Endian format
1 = HDLC Packet Data on the PCI Bus is in Big Endian format

Bits 7 to 12 / BERT Port Select Bits 0 to 5 (BPS0 to BPS5). These 6 bits select which port has the dedicated resources of the BERT.

000000 (00h) = Port 0
000001 (01h) = Port 1
000010 (02h) = Port 2
100110 (26h) = Port 38
100111 (27h) = Port 39
101000 (28h) = illegal setting
111111 (3 Fh) = illegal setting

Bit 14 / Transmit FIFO Priority Control Bit 0 (TFPC0).

Bit 15 / Transmit FIFO Priority Control Bit 1 (TFPC1).
These two bits select the algorithm the FIFO will use to determine which HDLC Channel gets the highest priority to the DMA to transfer data from the PCI Bus to the FIFO. In the priority-decoded scheme, the lower the HDLC channel numbers, the higher the priority.

00 = all HDLC channels are serviced Round Robin
01 = HDLC Channels 1 & 2 are Priority Decoded; other HDLC Channels are Round Robin
10 = HDLC Channels 1 to 4 are Priority Decoded; other HDLC Channels are Round Robin
11 = HDLC Channels 1 to 16 are Priority Decoded; other HDLC Channels are Round Robin

4.3 STATUS & INTERRUPT

4.3.1 STATUS & INTERRUPT GENERAL DESCRIPTION OF OPERATION

There are two status register in the device, Status Master (SM) and Status for DMA (SDMA). Both registers report events in real time as they occur by setting a bit within the register to a one. Each bit has the ability to generate an interrupt at the PCI Bus via the PINTA* output signal pin and if the Local Bus is in the Configuration Mode, then an interrupt will also be created at the LINT* output signal pin. Each status register has an associated Interrupt Mask Register, which can allow/deny interrupts from being generated on a bit-by-bit basis. All status remains active even if the associated Interrupt is disabled.

SM REGISTER

The Status Master (SM) register reports events that occur at the Port Interface, at the BERT receiver, at the PCI Bus and at the Local Bus. See Figure 4.3.1A for details.

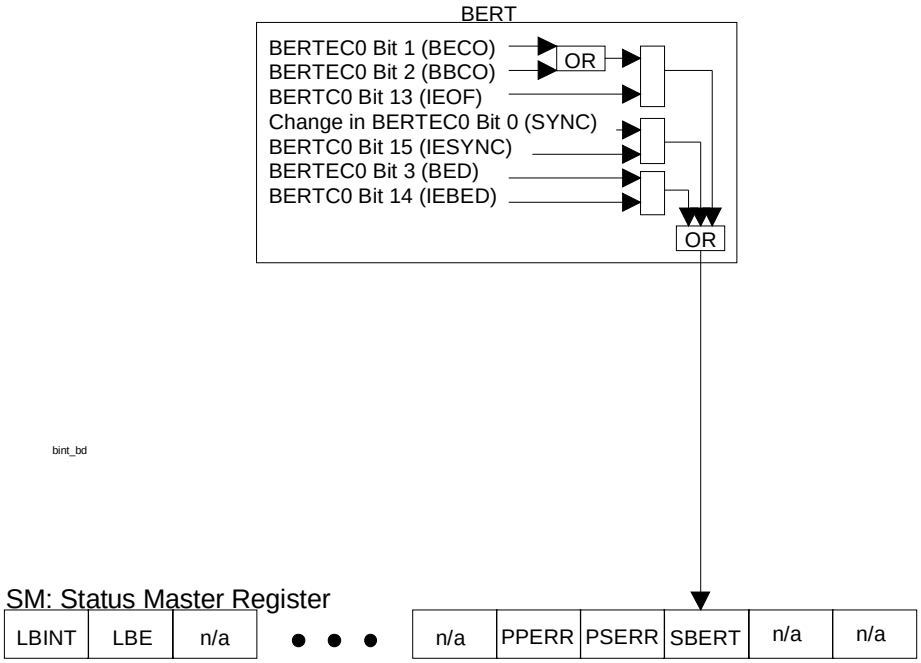
The BERT receiver will report three events, a change in the receive synchronizer status, a bit error being detected, and if either the Bit Counter or the Error Counter overflows. Each of these events can be masked within the BERT function via the BERT Control Register (BERTC0). If the software detects that the BERT has reported an event has occurred, then the software must read the BERT Status Register (BERTEC0) to determine which event(s) has occurred.

The SM register also reports events as they occur in the PCI Bus and the Local Bus. There are no control bits to stop these events from being reported in the SM register. When the Local Bus is operated in the PCI Bridge Mode, SM reports any interrupts detected via the Local Bus LINT* input signal pin and if any timing errors occur because of the use of the external timing signal LRDY*. When the Local Bus is operated in the Configuration Mode, the LBINT and LBE bits are meaningless and should be ignored.

SDMA REGISTER

The Status for DMA (SDMA) register reports events that occur regarding the Receive and Transmit DMA blocks as well as the receive HDLC controller and FIFO. The SDMA will report when the DMA reads from either the Receive Free Queue or Transmit Pending Queue or writes to the Receive or Transmit Done Queues. Also reported are error conditions that might occur in the access of one of these queues. The SDMA will report if any of the HDLC channels experiences a FIFO overflow/underflow condition and if the receive HDLC controller encounters a CRC error, abort signal, or octet length problem on any of the HDLC channels. The Host can determine which specific HDLC channel incurred a FIFO overflow/underflow, CRC error, octet length error or abort by reading the status bits as reported in Done Queues which are created by the DMA. There are no control bits to stop these events from being reported in the SDMA register.

STATUS REGISTER BLOCK DIAGRAM FOR SM Figure 4.3.1A



4.3.2 STATUS & INTERRUPT REGISTER DESCRIPTION

Register Name: **SM**
Register Description: **Status Master Register**
Register Address: **0020h**

7	6	5	4	3	2	1	0
n/a	n/a	n/a	PPERR	PSERR	SBERT	n/a	n/a
15	14	13	12	11	10	9	8
LBINT	LBE	n/a	n/a	n/a	n/a	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 2 / Status Bit for Change of State in BERT (SBERT). This status bit will be set to a one if there is a major change of state in the BERT receiver. A major change of state is defined as either a change in the receive synchronization (i.e. the BERT has gone into or out of receive synchronization), a bit error has been detected, or an overflow has occurred in either the Bit Counter or the Error Counter. The Host must read the status bits of the BERT in the BERT Status Register (BERTEC0) to determine the change of state. The SBERT bit will be cleared when the BERT Status Register is read and will not be set again until the BERT has experienced another change of state. If enabled via the SBERT bit in the Interrupt Mask for SM (ISM), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 3 / Status Bit for PCI System Error (PSERR). This status bit is a software version of the PCI Bus hardware pin PSERR. It will be set to a one if the PCI Bus detects an address parity error or other PCI Bus error. The PSERR bit will be cleared when read and will not be set again until another PCI Bus error has occurred. If enabled via the PSERR bit in the Interrupt Mask for SM (ISM), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode. This status bit is also reported in the Control/Status register in the PCI Configuration registers, see Section 9 for more details.

Bit 4 / Status Bit for PCI System Error (PPERR). This status bit is a software version of the PCI Bus hardware pin PPERR. It will be set to a one if the PCI Bus detects parity errors on the PAD and PCBE* buses as experienced or reported by a target. The PPERR bit will be cleared when read and will not be set again until another parity error has been detected. If enabled via the PPERR bit in the Interrupt Mask for SM (ISM), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode. This status bit is also reported in the Control/Status register in the PCI Configuration registers, see Section 9 for more details.

Bit 14 / Status Bit for Local Bus Error (LBE). This status bit applies to the Local Bus when it is operated in the PCI Bridge Mode. It will be set to a one when the Local Bus LRDY* signal is not detected within 9 LCLK periods. This indicates to the Host that an aborted Local Bus access has occurred. If enabled via the LBE bit in the Interrupt Mask for SM (ISM), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode. The LBE bit is meaningless when the Local Bus is operated in the configuration mode and should be ignored.

Bit 15 / Status Bit for Local Bus Interrupt (LBINT). This status bit will be set to a one if the Local Bus LINT* signal has been detected as asserted. This status bit is only valid when the Local Bus is operated in the PCI Bridge Mode. The LBINT bit will be cleared when read and will not be set again until once again the LINT* signal pin has been detected as asserted. If enabled via the LBINT bit in the Interrupt Mask for SM (ISM), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin. The LBINT bit is meaningless when the Local Bus is operated in the configuration mode and should be ignored.

Register Name: **ISM**
Register Description: **Interrupt Mask Register for SM**
Register Address: **0024h**

7	6	5	4	3	2	1	0
n/a	n/a	n/a	PPERR	PSERR	SBERT	n/a	n/a
15	14	13	12	11	10	9	8
LBINT	LBE	n/a	n/a	n/a	n/a	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 2 / Status Bit for Change of State in BERT (SBERT).
0 = interrupt masked
1 = interrupt unmasked

Bit 3 / Status Bit for PCI System Error (PSERR).
0 = interrupt masked
1 = interrupt unmasked

Bit 4 / Status Bit for PCI System Error (PPERR).
0 = interrupt masked
1 = interrupt unmasked

Bit 14 / Status Bit for Local Bus Error (LBE).
0 = interrupt masked
1 = interrupt unmasked

Bit 15 / Status Bit for Local Bus Interrupt (LBINT).
0 = interrupt masked
1 = interrupt unmasked

Register Name: **SDMA**
Register Description: **Status Register for DMA**
Register Address: **0028h**

7	6	5	4	3	2	1	0
RLBRE	RLBR	ROVFL	RLENC	RABRT	RCRCE	n/a	n/a
15	14	13	12	11	10	9	8
TDQWE	TDQW	TPQR	TUDFL	RDQWE	RDQW	RSBRE	RSBR

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 2 / Status Bit for Receive HDLC CRC Error (RCRCE). This status bit will be set to a one if any of the receive HDLC channels experiences a CRC check sum error. The RCRCE bit will be cleared when read and will not be set again until another CRC check sum error has occurred. If enabled via the RCRCE bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 3 / Status Bit for Receive HDLC Abort Detected (RABRT). This status bit will be set to a one if any of the receive HDLC channels detects an abort. The RABRT bit will be cleared when read and will not be set again until another abort has been detected. If enabled via the RABRT bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 4 / Status Bit for Receive HDLC Length Check (RLENC). This status bit will be set to a one if any of the HDLC channels:
- exceeds the octet length count (if so enabled to check for octet length)
- receives a HDLC packet that does not meet the minimum length criteria
- experiences a non-integral number of octets in between opening and closing flags.

The RLENC bit will be cleared when read and will not be set again until another length violation has occurred. If enabled via the RLENC bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 5 / Status Bit for Receive FIFO Overflow (ROVFL). This status bit will be set to a one if any of the HDLC channels experiences an overflow in the receive FIFO. The ROVFL bit will be cleared when read and will not be set again until another overflow has occurred. If enabled via the ROVFL bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 6 / Status Bit for Receive DMA Large Buffer Read (RLBR). This status bit will be set to a one each time the Receive DMA completes a single read or a burst read of the Large Buffer Free Queue. The RLBR bit will be cleared when read and will not be set again, until another read of the Large Buffer Free Queue has occurred. If enabled via the RLBR bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 7 / Status Bit for Receive DMA Large Buffer Read Error (RLBRE). This status bit will be set to a one each time the Receive DMA tries to read the Large Buffer Free Queue and it is empty. The RLBRE bit will be cleared when read and will not be set again, until another read of the Large Buffer Free Queue detects that it is empty. If enabled via the RLBRE bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 8 / Status Bit for Receive DMA Small Buffer Read (RSBR). This status bit will be set to a one each time the Receive DMA completes a single read or a burst read of the Small Buffer Free Queue. The RSBR bit will be cleared when read and will not be set again, until another read of the Small Buffer Free Queue has occurred. If enabled via the RSBR bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 9 / Status Bit for Receive DMA Small Buffer Read Error (RSBRE). This status bit will be set to a one each time the Receive DMA tries to read the Small Buffer Free Queue and it is empty. The RSBRE bit will be cleared when read and will not be set again, until another read of the Small Buffer Free Queue detects that it is empty. If enabled via the RSBRE bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 10 / Status Bit for Receive DMA Done Queue Write (RDQW). This status bit will be set to a one when the Receive DMA writes to the Done Queue. Based of the setting of the Receive Done Queue Threshold Setting (RDQT0 to RDQT2) bits in the Receive DMA Queues Control (RDMAQ) register, this bit will be set either after each write or after a programmable number of writes from 2 to 128. See Section 8.1.4 for more details. The RDQW bit will be cleared when read and will not be set again until another write to the Done Queue has occurred. If enabled via the RDQW bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 11 / Status Bit for Receive DMA Done Queue Write Error (RDQWE). This status bit will be set to a one each time the Receive DMA tries to write to the Done Queue and it is full. The RDQWE bit will be cleared when read and will not be set again until another write to the Done Queue detects that it is full. If enabled via the RDQWE bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 12 / Status Bit for Transmit FIFO Underflow (TUDFL). This status bit will be set to a one if any of the HDLC channels experiences an underflow in the transmit FIFO. The TUDFL bit will be cleared when read and will not be set again until another underflow has occurred. If enabled via the TUDFL bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 13 / Status Bit for Transmit DMA Pending Queue Read (TPQR). This status bit will be set to a one each time the Transmit DMA reads the Pending Queue. The TPQR bit will be cleared when read and will not be set again until another read of the Pending Queue has occurred. If enabled via the TPQR bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 14 / Status Bit for Transmit DMA Done Queue Write (TDQW). This status bit will be set to a one when the Transmit DMA writes to the Done Queue. Based of the setting of the Transmit Done Queue Threshold Setting (TDQT0 to TDQT2) bits in the Transmit DMA Queues Control (TDMAQ) register, this bit will be set either after each write or after a programmable number of writes from 2 to 128. See Section 8.2.4 for more details. The TDQW bit will be cleared when read and will not be set again until another write to the Done Queue has occurred. If enabled via the TDQW bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Bit 15 / Status Bit for Transmit DMA Done Queue Write Error (TDQWE). This status bit will be set to a one each time the Transmit DMA tries to write to the Done Queue and it is full. The TDQWE bit will be cleared when read and will not be set again until another write to the Done Queue detects that it is full. If enabled via the TDQWE bit in the Interrupt Mask for SDMA (ISDMA), the setting of this bit will cause a hardware interrupt at the PCI Bus via the PINTA* signal pin and also at the LINT* if the Local Bus is in the Configuration Mode.

Register Name: **ISDMA**
Register Description: **Interrupt Mask Register for SDMA**
Register Address: **002Ch**

7	6	5	4	3	2	1	0
RLBRE	RLBR	ROVFL	RLENC	RABRT	RCRCE	n/a	n/a
15	14	13	12	11	10	9	8
TDQWE	TDQW	TPQR	TUDFL	RDQWE	RDQW	RSBRE	RSBR

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 2 / Status Bit for Receive HDLC CRC Error (RCRCE).

0 = interrupt masked
1 = interrupt unmasked

Bit 3 / Status Bit for Receive HDLC Abort Detected (RABRT).
0 = interrupt masked
1 = interrupt unmasked

Bit 4 / Status Bit for Receive HDLC Length Check (RLENC).
0 = interrupt masked
1 = interrupt unmasked

Bit 5 / Status Bit for Receive FIFO Overflow (ROVFL).
0 = interrupt masked
1 = interrupt unmasked

Bit 6 / Status Bit for Receive DMA Large Buffer Read (RLBR).
0 = interrupt masked
1 = interrupt unmasked

Bit 7 / Status Bit for Receive DMA Large Buffer Read Error (RLBRE).
0 = interrupt masked
1 = interrupt unmasked

Bit 8 / Status Bit for Receive DMA Small Buffer Read (RSBR).
0 = interrupt masked
1 = interrupt unmasked

Bit 9 / Status Bit for Receive DMA Small Buffer Read Error (RSBRE).
0 = interrupt masked
1 = interrupt unmasked

Bit 10 / Status Bit for Receive DMA Done Queue Write (RDQW).
0 = interrupt masked
1 = interrupt unmasked

Bit 11 / Status Bit for Receive DMA Done Queue Write Error (RDQWE).
0 = interrupt masked
1 = interrupt unmasked

Bit 12 / Status Bit for Transmit FIFO Underflow (TUDFL).
0 = interrupt masked
1 = interrupt unmasked

Bit 13 / Status Bit for Transmit DMA Pending Queue Read (TPQR).
0 = interrupt masked
1 = interrupt unmasked

Bit 14 / Status Bit for Transmit DMA Done Queue Write (TDQW).
0 = interrupt masked
1 = interrupt unmasked

Bit 15 / Status Bit for Transmit DMA Done Queue Write Error (TDQWE).
0 = interrupt masked
1 = interrupt unmasked

4.4 TEST REGISTER DESCRIPTION

Register Name: **TEST**
Register Description: **Test Register**
Register Address: **0050h**

7	6	5	4	3	2	1	0
n/a	n/a	n/a	n/a	n/a	n/a	n/a	FT
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Factory Test (FT).
This bit is used by the factory to place the DS3131 into the test mode. For normal device operation, this bit should be set to zero whenever this register is written to.

Bit 1 to 15 / Device internal test bits. Bits 1 to 15 shown in the above table is for BoSS internal (Dallas Semiconductor) tests use, not user test mode controls. Values of these bits should always be “0”. If any of these bits are set to “1” device will not function properly

SECTION 5: LAYER ONE

5.1 GENERAL DESCRIPTION

Each port on the DS3131 contains a dedicated Bit Synchronous HDLC Controller for that port. The Layer One Block Diagram in Figure 5.1A provides a block level description of the Layer One circuitry on each port. Depending on the configuration, the DS3131 can have either 28 or 40 Bit Synchronous HDLC interfaces. See Table 5.1A. Figure 1B in Section 1 details the configurations shown in Table 5.1A.

DS3131 PORT CONFIGURATION OPTIONS Table 5.1A

Local BusEnabled?	Number of Bit Synchronous Ports Available
Yes	28
No	40

Each of the 40 ports can be independently configured into a different mode. The ports are capable of operating at speeds up to 52 MHz and are gapped clock tolerant. There are no restrictions on clock gapping as long as the minimum clock period and high and low times listed in Section 13 are not violated. Each port is a simple synchronous serial interface where data is clock into the device using the RC input and clocked out of the device using the TC input. The transmit and receive timing is completely independent. Each port has an associated Receive Port Control Register (RP[n]CR where n = 0 to 39) and a Transmit Port Control Register (TP[n]CR where n = 0 to 39). These control registers are defined in detail in Section 5.2 and they control all of the circuitry in the Layer One Block.

HDLC Channel Assignment

HDLC channel numbers are assigned as shown below in Table 5.1C.

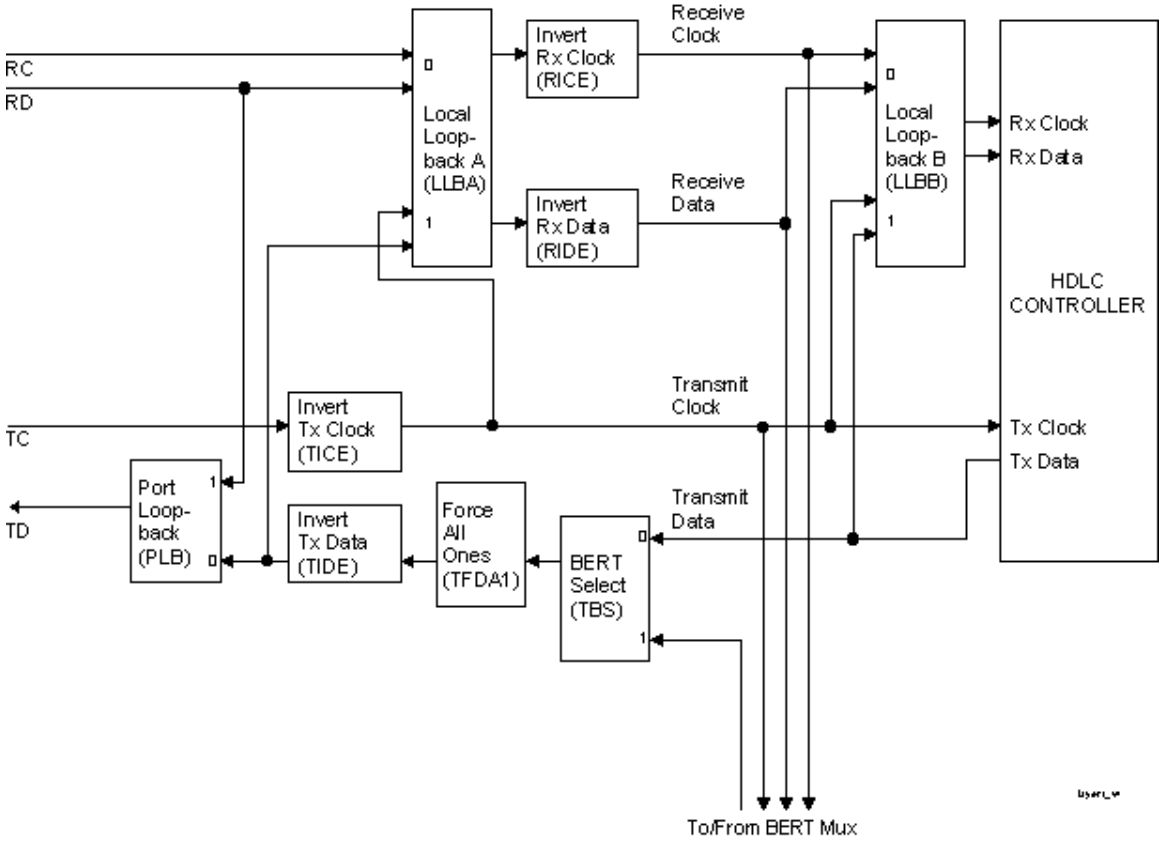
HDLC CHANNEL ASSIGNMENT Table 5.1C

Port Number	HDLC Channel Number
0	1
1	2
2	3
3	4
4	5
37	38
38	39
39	40

BERT Operation

The DS3131 contains an onboard full-featured Bit Error Rate Tester (BERT) function, which is capable of generating and detecting both pseudorandom and repeating serial bit patterns. The BERT function is a shared resource among the 40 ports on the DS3131 and it can only be assigned to one port at a time. The details on the BERT function are covered in Section 5.3.

LAYER ONE PORT INTERFACE BLOCK DIAGRAM Figure 5.1A



Note: All blocks are controlled by either the RP[n]CR or TP[n]CR registers. See Section 5.2.

5.2 PORT REGISTER DESCRIPTIONS

Receive Side Control Bits (one each for all 40 ports)

Register Name: **RP[n]CR where n = 0 to 39 for each Port**
Register Description: **Receive Port [n] Control Register**
Register Address: **See the Register Map in Section 3**

7	6	5	4	3	2	1	0
n/a	n/a	n/a	n/a	n/a	n/a	RIDE	RICE
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	LLBB	LLBA	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Invert Receive Clock Enable (RICE).
0 = do not invert clock (normal mode)
1 = invert clock (inverted clock mode)

Bit 1 / Invert Receive Data Enable (RIDE).
0 = do not invert data (normal mode)
1 = invert data (inverted data mode)

Bit 10 / Local Loopback A Enable (LLBA). This loopback routes transmit data back to the receive port close to the ports pins on the device. See Figure 5.1A.
0 = loopback disabled
1 = loopback enabled

Bit 11 / Local Loopback B Enable (LLBB). This loopback route transmits data as it leaves the bit synchronous HDLC controller back into the HDLC controller. See Figure 5.1A.
0 = loopback disabled
1 = loopback enabled

Transmit Side Control Bits (One Each For All 40 Ports)

Register Name: **TP[n]CR where n = 0 to 39 for each Port**
Register Description: **Transmit Port [n] Control Register**
Register Address: **See the Register Map in Section 3**

7	6	5	4	3	2	1	0
n/a	n/a	n/a	n/a	TFDA1	n/a	TIDE	TICE
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	tbs	plb	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Invert Transmit Clock Enable (TICE).
0 = do not invert clock (normal clock mode)
1 = invert clock (inverted clock mode)

Bit 1 / Invert Transmit Data Enable (TIDE).
0 = do not invert data (normal data mode)
1 = invert data (inverted data mode)

Bit 3 / Force Data All 1's (TFDA1).

0 = force all data at TD to be one
1 = allow data to be transmitted normally

Bit 10 / Port Loopback Enable (PLB). See Figure 5.1A for details. This loopback routes the data incoming at the RD pin to the TD pin.

0 = loopback disabled
1 = loopback enabled

Bit 11 / BERT Select (TBS). See Figure 5.1A for details. These select controls whether data is to be sourced from the HDLC Controller or from the BERT Block. See Section 5.3 for details on how to configure the operation of the BERT.

0 = source transmit data from the HDLC controller
1 = source transmit data from the BERT block

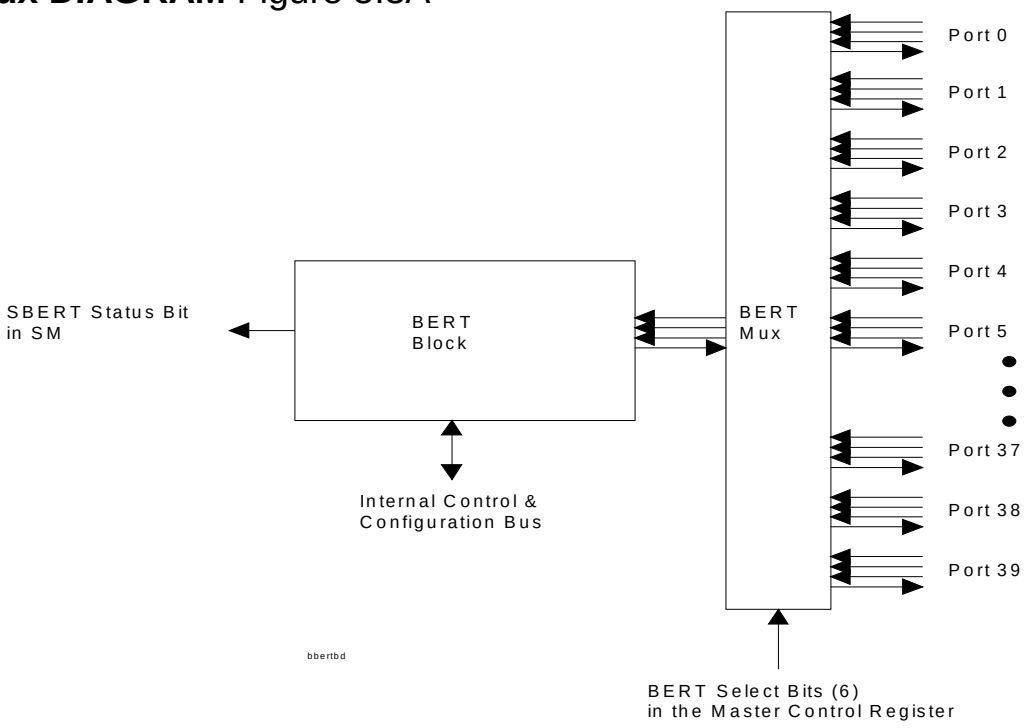
5.3 BERT

The BERT Block is capable of generating and detecting the following patterns:

- the pseudorandom patterns 2E7, 2E11, 2E15, and QRSS
- a repetitive pattern from 1 to 32 bits in length
- alternating (16-bit) words which flip every 1 to 256 words

The BERT receiver has a 32-bit Bit Counter and a 24-bit Error Counter. It can generate interrupts on detecting a bit error, a change in synchronization, or if an overflow occurs in the Bit and Error Counters. See Section 4 for details on status bits and interrupts from the BERT Block. To activate the BERT Block, the Host must configure the BERT mux via the Master Control register (see Figure 5.3A) and to have the BERT data appear at the TD pin, the TBS select bit in the TP[n]CR register must be set to a one (see Section 5.2).

BERT Mux DIAGRAM Figure 5.3A



5.4 BERT REGISTER DESCRIPTION

BERT REGISTER SET Figure 5.4A

BERTC0: BERT Control 0 lsb

n/a	TINV	RINV	PS2	PS1	PS0	LC	RESYNC

msb

IESYNC	IEBED	IEOF	n/a	RPL3	RPL2	RPL1	RPL0

BERTC1: BERT Control 1 lsb

EIB2	EIB1	EIB0	SBE	n/a	n/a	n/a	TC

msb

Alternating Word Count

BERTRP0: BERT Repetitive Pattern Set 0 (lower word) lsb

BERT Repetitive Pattern Set (lower byte)

msb

BERT Repetitive Pattern Set

BERTRP1: BERT Repetitive Pattern Set 1 (upper word) lsb

BERT Repetitive Pattern Set

msb

BERT Repetitive Pattern Set (upper byte)

BERTBC0: BERT Bit Counter 0 (lower word) lsb

BERT 32-Bit Bit Counter (lower byte)

msb

BERT 32-Bit Bit Counter

BERTBC1: BERT Bit Counter 0 (upper word) lsb

BERT 32-Bit Bit Counter

msb

BERT 32-Bit Bit Counter (upper byte)

BERTEC0: BERT Error Counter 0 / Status lsb

n/a	RA1	RA0	RLOS	BED	BBCO	BECO	SYNC

msb

BERT 24-Bit Error Counter (lower byte)

BERTEC1: BERT Error Counter 1 (upper word) lsb

BERT 24-Bit Error Counter

msb

BERT 24-Bit Error Counter (upper byte)

Register Name: **BERTC0**
Register Description: **BERT Control Register 0**
Register Address: **0500h**

7	6	5	4	3	2	1	0
n/a	TINV	RINV	PS2	PS1	PS0	LC	RESYNC
15	14	13	12	11	10	9	8
IESYNC	IEBED	IEOF	n/a	RPL3	RPL2	RPL1	RPL0

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Force Resynchronization (RESYNC). A low to high transition will force the receive BERT synchronizer to resynchronize to the incoming data stream. This bit should be toggled from low to high whenever the host wishes to acquire synchronization on a new pattern. Must be cleared and set again for a subsequent resynchronization.
Note: Pattern selects bits PS0~2 must be set before Resync.

Bit 1 / Load Bit and Error Counters (LC). A low to high transition latches the current bit and error counts into the host accessible registers BERTBC and BERTEC and clears the internal count. This bit should be toggled from low to high whenever the host wishes to begin a new acquisition period. Must be cleared and set again for a subsequent loads.

Bit 2 / Pattern Select Bit 0 (PS0).
Bit 3 / Pattern Select Bit 1 (PS1).
Bit 4 / Pattern Select Bit 2 (PS2).

000 = Pseudorandom Pattern 2E7 - 1
001 = Pseudorandom Pattern 2E11 - 1
010 = Pseudorandom Pattern 2E15 - 1
011 = Pseudorandom Pattern QRSS (2E20 - 1 with a one forced if the next 14 positions are zero)
100 = Repetitive Pattern
101 = Alternating Word Pattern
110 = illegal state
111 = illegal state

Bit 5 / Receive Invert Data Enable (RINV).

0 = do not invert the incoming data stream
1 = invert the incoming data stream

Bit 6 / Transmit Invert Data Enable (TINV).

0 = do not invert the outgoing data stream
1 = invert the outgoing data stream

Bit 8 / Repetitive Pattern Length Bit 0 (RPL0).
Bit 9 / Repetitive Pattern Length Bit 1 (RPL1).
Bit 10 / Repetitive Pattern Length Bit 2 (RPL2).
Bit 11 / Repetitive Pattern Length Bit 3 (RPL3).

RPL0 is the LSB and RPL3 is the MSB of a nibble that describes the how long the repetitive pattern is. The valid range is 17 (0000) to 32 (1111). These bits are ignored if the receive BERT is programmed for a pseudorandom pattern. To create repetitive patterns less than 17 bits in length, the user must set the length to an integer number of the desired length that is less than or equal to 32. For example, to create a 6 bit pattern, the user can set the length to 18 (0001) or to 24 (0111) or to 30 (1101).

REPETITIVE PATTERN LENGTH MAP

Length	Code	Length	Code	Length	Code	Length	Code
17 Bits	0000	18 Bits	0001	19 Bits	0010	20 Bits	0011
21 Bits	0100	22 Bits	0101	23 Bits	0110	24 Bits	0111
25 Bits	1000	26 Bits	1001	27 Bits	1010	28 Bits	1011
29 Bits	1100	30 Bits	1101	31 Bits	1101	32 Bits	1111

Bit 13 / Interrupt Enable for Counter Overflow (IEOF). Allows the receive BERT to cause an interrupt if either the Bit Counter or the Error Counter overflows.

0 = interrupt masked
1 = interrupt enabled

Bit 14 / Interrupt Enable for Bit Error Detected (IEBED). Allows the receive BERT to cause an interrupt if a bit error is detected.

0 = interrupt masked
1 = interrupt enabled

Bit 15 / Interrupt Enable for Change of Synchronization Status (IESYNC). Allows the receive BERT to cause an interrupt if there is a change of state in the synchronization status (i.e. the receive BERT either goes into or out of synchronization).

0 = interrupt masked
1 = interrupt enabled

Register Name: **BERTC1**
Register Description: **BERT Control Register 1**
Register Address: **0504h**

7	6	5	4	3	2	1	0
EIB2	EIB1	EIB0	SBE	n/a	n/a	n/a	TC
15	14	13	12	11	10	9	8
Alternating Word Count							

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Transmit Pattern Load (TC). A low to high transition loads the pattern generator with Repetitive or Pseudorandom pattern that is to be generated. This bit should be toggled from low to high whenever the host wishes to load a new pattern. Must be cleared and set again for a subsequent loads.

Bit 4 / Single Bit Error Insert (SBE). A low to high transition will create a single bit error. Must be cleared and set again for a subsequent bit error to be inserted.

Bit 5 / Error Insert Bit 0 (EIB0).
Bit 6 / Error Insert Bit 1 (EIB1).
Bit 7 / Error Insert Bit 2 (EIB2).
Will automatically insert bit errors at the prescribed rate into the generated data pattern. Useful for verifying error detection operation.

EIB2	EIB1	EIB0	Error Rate Inserted
0	0	0	no errors automatically inserted
0	0	1	10E-1
0	1	0	10E-2
0	1	1	10E-3
1	0	0	10E-4
1	0	1	10E-5
1	1	0	10E-6
1	1	1	10E-7

Bits 8 to 15 / Alternating Word Count Rate. When the BERT is programmed in the alternating word mode, the words will repeat for the count loaded into this register then flip to the other word and again repeat for the number of times loaded into this register. The valid count range is from 05h to FFh.

Register Name: **BERTBRP0**
Register Description: **BERT Repetitive Pattern Set 0**
Register Address: **0508h**

Register Name: **BERTBRP1**
Register Description: **BERT Repetitive Pattern Set 1**
Register Address: **050Ch**

BERTRP0: BERT Repetitive Pattern Set 0 (lower word)

7	6	5	4	3	2	1	0
BERT Repetitive Pattern Set (lower byte)							
15	14	13	12	11	10	9	8
Bert Repetitive Pattern Set							

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

BERTRP1: BERT Repetitive Pattern Set 1 (upper word)

23	22	21	20	19	18	17	16
BERT Repetitive Pattern Set							
31	30	29	28	27	26	25	24
Bert Repetitive Pattern Set (upper byte)							

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 31 / BERT Repetitive Pattern Set (BERTRP0 and BERTRP1). These registers must be properly loaded for the BERT to properly generate and synchronize to either a repetitive pattern, a pseudorandom pattern, or a alternating word pattern. For a repetitive pattern that is less than 32 bits, then the pattern should be repeated so that all 32 bits are used to describe the pattern. For example if the pattern was the repeating 5-bit pattern …01101… (where right most bit is one sent first and received first) then PBRP0 should be loaded with xB5AD and PBRP1 should be loaded with x5AD6. For a pseudorandom pattern, both registers should be loaded with all ones (i.e. xFFFF). For an alternating word pattern, one word should be placed into PBRP0 and the other word should be placed into PBRP1. For example, if the DDS stress pattern "7E" is to be described, the user would place x0000 in PBRP0 and x7E7E in PBRP1 and the alternating word counter would be set to 50 (decimal) to allow 100 bytes of 00h followed by 100 bytes of 7 Eh to be sent and received.

Register Name: **BERTBC0**
Register Description: **BERT 32-Bit Bit Counter (lower word)**
Register Address: **0510h**

Register Name: **BERTBC1**
Register Description: **BERT 32-Bit Bit Counter (upper word)**
Register Address: **0514h**

BERTBC0: BERT Bit Counter 0 (lower word)

7	6	5	4	3	2	1	0
BERT 32-Bit Bit Counter (lower byte)							
15	14	13	12	11	10	9	8
Bert 32-bit Bit Counter							

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

BERTBC1: BERT Bit Counter 0 (upper word)

23	22	21	20	19	18	17	16
BERT 32-Bit Bit Counter							
31	30	29	28	27	26	25	24
Bert 32-bit Bit Counter (upper byte)							

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 31 / BERT 32-Bit Bit Counter (BERTBC0 and BERTBC1). This 32-bit counter will increment for each data bit (i.e. clock) received. This counter is not disabled when the receive BERT loses synchronization. This counter will be loaded with the current bit count value when the LC control bit in the BERTC0 register is toggled from a low (0) to a high (1). When full, this counter will saturate and set the BBCO status bit.

Register Name: **BERTEC0**
Register Description: **BERT 24-Bit Error Counter (lower) & Status Information**
Register Address: **0518h**

7	6	5	4	3	2	1	0
n/a	RA1	RA0	RLOS	BED	BBCO	BECO	SYNC
15	14	13	12	11	10	9	8
BERT 24-Bit Error Counter (lower byte)							

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Real Time Synchronization Status (SYNC). Real time status of the synchronizer (this bit is not latched). Will be set when the incoming pattern matches for 32 consecutive bit positions. Will be cleared when 6 or more bits out of 64 are received in error.

Bit 1 / BERT Error Counter Overflow (BECO). A latched bit which is set when the 24-bit BERT Error Counter (BEC) overflows. Cleared when read and will not be set again until another overflow occurs.

Bit 2 / BERT Bit Counter Overflow (BBCO). A latched bit which is set when the 32-bit BERT Bit Counter (BBC) overflows. Cleared when read and will not be set again until another overflow occurs.

Bit 3 / Bit Error Detected (BED). A latched bit which is set when a bit error is detected. The receive BERT must be in synchronization for it detect bit errors. Cleared when read.

Bit 4 / Receive Loss Of Synchronization (RLOS). A latched bit which is set whenever the receive BERT begins searching for a pattern. Once synchronization is achieved, this bit will remain set until read.

Bit 5 / Receive All Zeros (RA0). A latched bit which is set when 31 consecutive zeros are received. Allowed to be cleared once a one is received.

Bit 6 / Receive All Ones (RA1). A latched bit which is set when 31 consecutive ones are received. Allowed to be cleared once a zero is received.

Bits 8 to 15 / BERT 24-Bit Error Counter (BEC). Lower word of the 24-bit error counter. See the BERTEC1 register description for details.

Register Name: **BERTEC1**
Register Description: **BERT 24-Bit Error Counter (upper)**
Register Address: **051Ch**

7	6	5	4	3	2	1	0
BERT 24-BIT ERROR COUNTER							
15	14	13	12	11	10	9	8
BERT 24-Bit Error Counter (upper byte)							

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 15 / BERT 24-Bit Error Counter (BEC). Upper two words of the 24-bit error counter. This 24-bit counter will increment for each data bit received in error. This counter is not disabled when the receive BERT loses synchronization. This counter will be loaded with the current bit count value when the LC control bit in the BERTC0 register is toggled from a low (0) to a high (1). When full, this counter will saturate and set the BECO status bit.

SECTION 6: HDLC

6.1 GENERAL DESCRIPTION

Each port on the DS3131 has a dedicated Bit Synchronous HDLC Controller associated that can operate up to 52 Mbps. See Figures 1B & 5.1A. HDLC channel numbers are assigned as shown below in Table 6.1A.

HDLC CHANNEL ASSIGNMENT Table 6.1A

Port Number	HDLC Channel Number
0	1
1	2
2	3
3	4
4	5
.	.
.	.
.	.
37	38
38	39
39	40

HDLC OPERATION

The HDLC Controllers are capable of handling all the normal real-time tasks required. Table 6.1C lists all of the functions supported by the Receive HDLC Controller and Table 6.1D lists all of the functions supported by the Transmit HDLC Controller. Each of the 40 HDLC channels within BoSS are configured by the Host via the Receive HDLC Channel Definition (RHCD) and Transmit HDLC Channel Definition (THCD) registers. There is a separate RHCD and THCD register for each HDLC channel. The Host can access the RHCD and THCD registers indirectly via the RHCDIS indirect select and THCDIS indirect select registers. See Section 6.2 for details.

On the receive side, when the HDLC Block is processing a packet, one of the outcomes shown in Table 6.1B will occur. For each packet, one of these outcomes will be reported in the Receive Done Queue Descriptor (see Section 8.1.4 for details). On the transmit side, when the HDLC Block is processing a packet, an error in the PCI Block (parity or target abort) or transmit FIFO underflow will cause the HDLC Block to send an Abort sequence (8 ones in a row) followed continuously by the selected Interfill (either 7 Eh or FFh) until the HDLC channel is reset by the transmit DMA Block (see Section 8.2.1 for details). This same sequence of events will occur even if the transmit HDLC channel is being operated in the transparent mode. If the FIFO is empty, then the interfill byte (either 7 Eh or FFh) is sent until an outgoing packet is ready for transmission.

RECEIVE BIT SYNCHRONOUS HDLC PACKET PROCESSING OUTCOMES
Table 6.1B

Outcome	Criteria
EOF / Normal Packet	Integral number of packets > min. & < max. is received & CRC is okay
EOF / Bad FCS	Integral number of packets > min. & < max. is received & CRC is bad
Abort Detected	Seven or more ones in a row detected
EOF / Too Few Bytes	Less than the packet minimum is received (if detection enabled)
Too Many Bytes	Greater than the packet maximum is received (if detection enabled)
EOF / Bad # of Bits	Not an integral number of bytes received
FIFO Overflow	Tried to write a byte into an already full FIFO

Note: EOF = End Of Frame which means that this outcome is not determined until a closing flag is detected.

If any of the 40 receive HDLC channels detects an abort sequence, a FCS checksum error, or if the packet length was incorrect, then the appropriate status bit in the Status Register for DMA (SDMA) will be set. If enabled, the setting of any of these statuses can cause a hardware interrupt to occur. See Section 4.3.2 for details on the operation of these status bits.

RECEIVE BIT SYNCHRONOUS HDLC FUNCTIONS Table 6.1C

Zero Destuff	• This operation is disabled if the channel is set to transparent mode.
Flag Detection & Byte Alignment	• Okay to have two packets separated by only one flag or by two flags sharing a zero. • This operation is disabled if the channel is set to transparent mode.
Octet Length Check	• The maximum check is programmable up to 65,536 bytes via the RHPL register. • The maximum check can be disabled via the ROLD control bit in the RHCD register. • The minimum and maximum counts include the FCS. • An error is also reported if a non-integer number of octets occur between flags.
CRC Check	• Can be either set to CRC-16 or CRC-32 or none. • The CRC can be passed through to the PCI bus or not. • The CRC check is disabled if the channel is set to transparent mode.
Abort Detection	• Checks for seven or more ones in a row.
Invert Data	• All data (including the flags & FCS) is inverted before HDLC processing. • Also available in the transparent mode.
Bit Flip	• The first bit received becomes either the LSB (normal mode) or the MSB (telecom mode) of the byte stored in the FIFO. • Also available in the transparent mode.
Transparent Mode	• If enabled, flag detection, zero destuffing, abort detection, length checking, and FCS checking are disabled. • Data is passed to the PCI Bus on octet (i.e. byte) boundaries in unchannelized operation.

TRANSMIT BIT SYNCHRONOUS HDLC FUNCTIONS Table 6.1D

Zero Stuffing	• Only used in between opening and closing flags. • Will be disabled in between a closing flag and an opening flag and for sending aborts and/or interfill data. • Disabled if the channel is set to the transparent mode.
Interfill Selection	• Can be either 7 Eh or FFh.
Flag Generation	• A programmable number of flags (1 to 16) can be set in between packets. • Disabled if the channel is set to the transparent mode.
CRC Generation	• Can be either CRC-16 or CRC-32 or none. • Disabled if the channel is set to transparent mode.
Invert Data	• All data (including the flags & FCS) is inverted after processing. • Also available in the transparent mode
Bit Flip	• The LSB (normal mode) of the byte from the FIFO becomes the first bit sent or the MSB (telecom mode) becomes the first bit sent. • Also available in the transparent mode.
Transparent Mode	• If enabled, flag generation, zero stuffing, and FCS generation is disabled. • Will pass bytes from the PCI Bus to Layer 1 on octet (i.e. byte) boundaries.
Invert FCS	• When enabled, it will invert all of the bits in the FCS (useful for HDLC testing).

6.2 BIT SYNCHRONOUS HDLC REGISTER DESCRIPTION

Register Name: **RHCDIS**
Register Description: **Receive HDLC Chan5anel Definition Indirect Select**
Register Address: **0400h**

7	6	5	4	3	2	1	0
n/a	n/a	HCID5	HCID4	HCID3	HCID2	HCID1	HCID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	n/a	n/a	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 5 / HDLC Channel ID (HCID0 to HCID5).

000000 (00h) = HDLC Channel Number 1 (Port #0)
000001 (01h) = HDLC Channel Number 2 (Port #1)
000010 (02h) = HDLC Channel Number 3 (Port #2)
000011 (03h) = HDLC Channel Number 4 (Port #3)
100110 (26h) = HDLC Channel Number 39 (Port #38)
100111 (27h) = HDLC Channel Number 40 (Port #39)
101000 (28h) = illegal setting
111111 (3 Fh) = illegal setting

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the internal Receive HDLC Definition RAM, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the channel location indicated by the HCID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the RHCD register, the IAB bit will be set to zero. When the host wishes to write data to the internal Receive HDLC Definition RAM, this bit should be written to a zero by the host. This causes the device to take the data that is currently present in the RHCD register and write it to the channel location indicated by the HCID bits. When the device has completed the write, the IAB will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **RHCD**
Register Description: **Receive HDLC Channel Definition**
Register Address: **0404h**

7	6	5	4	3	2	1	0
RABTD	RCS	RBF	RID	RCRC1	RCRC0	ROLD	RTRANS
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	n/a	n/a	RZDD

Note: bits that are underlined are read only, all other bits are read-write.

Bit 0 / Receive Transparent Enable (RTRANS). When this bit is set low, the HDLC controller performs flag delineation, zero destuffing, abort detection, octet length checking (if enabled via ROLD), and FCS checking (if enabled via RCRC0/1). When this bit is set high, the HDLC controller does not perform flag delineation, zero destuffing, and abort detection, octet length checking, or FCS checking.

0 = transparent mode disabled
1 = transparent mode enabled

Bit 1 / Receive Maximum Octet Length Detection Enable (ROLD). When this bit is set low, the HDLC controller does not check to see if the octet length of the received packets exceeds the count loaded into the Receive HDLC Packet Length (RHPL) register. When this bit is set high, the HDLC controller checks to see if the octet length of the received packets exceeds the count loaded into the RHPL register. When an incoming packet exceeds the maximum length, then the packet is aborted and the remainder is discarded. This bit is ignored if the HDLC channel is set into Transparent mode (RTRANS = 1).

0 = octet length detection disabled
1 = octet length detection enabled

Bit 2 & Bit 3 / Receive CRC Selection (RCRC0/RCRC1). These two bits are ignored if the HDLC channel is set into Transparent mode (RTRANS = 1).

RCRC1	RCRC0	Action
0	0	no CRC verification performed
0	1	16-bit CRC (CCITT/ITU Q.921)
1	0	32-bit CRC
1	1	illegal state

Bit 4 / Receive Invert Data Enable (RID). When this bit is set low, the incoming HDLC packets are not inverted before processing. When this bit is set high, the HDLC controller inverts all the data (flags, information fields, and FCS) before processing the data. The data is not re-inverted before passing to the FIFO.

0 = do not invert data
1 = invert all data (including flags and FCS)

Bit 5 / Receive Bit Flip (RBF). When this bit is set low, the HDLC controller will place the first HDLC bit received in the lowest bit position of the PCI Bus bytes (i.e. PAD [0] / PAD [8] / PAD[16] / PAD[24]). When this bit is set high, the HDLC controller will place the first HDLC bit received in the highest bit position of the PCI Bus bytes (i.e. PAD [7] / PAD [15] / PAD [23] / PAD[31]).

0 = the first HDLC bit received is placed in the lowest bit position of the bytes on the PCI Bus
1 = the first HDLC bit received is placed in the highest bit position of the bytes on the PCI Bus

Bit 6 / Receive CRC Strip Enable (RCS). When this bit is set high, the FCS is not transferred through to the PCI Bus. When this bit is set low, the HDLC controller will include the 2 bytes FCS (16-bit) or 4 bytes FCS (32-bit) in the data that it transfers to the PCI Bus. This bit is ignored if the HDLC channel is set into Transparent mode (RTRANS = 1).

0 = send FCS to the PCI Bus
1 = do not send the FCS to the PCI Bus

Bit 7 / Receive Abort Disable (RABTD). When this bit is set low, the HDLC controller will examine the incoming data stream for the Abort sequence, which is seven or more consecutive ones. When this bit is set high, the incoming data stream is not examined for the Abort sequence and if an incoming Abort sequence is received, no action will be taken. This bit is ignored when the HDLC controller is configured in the Transparent Mode (RTRANS = 1).

0 = abort detection enabled
1 = abort detection disabled

Bit 8 / Receive Zero Destuffing Disable (RZDD). When this bit is set low, the HDLC controller will zero destuff the incoming data stream. When this bit is set high, the HDLC controller will not zero destuff the incoming data stream. This bit is ignored when the HDLC controller is configured in the Transparent Mode (RTRANS = 1).

0 = zero destuffer enabled
1 = zero destuffer disabled

Register Name: **RHPL**
Register Description: **Receive HDLC Maximum Packet Length**
Register Address: **0410h**

7	6	5	4	3	2	1	0
RHPL7	RHPL6	RHPL5	RHPL4	RHPL3	RHPL2	RHPL1	RHPL0
15	14	13	12	11	10	9	8
RHPL15	RHPL14	RHPL13	RHPL12	RHPL11	RHPL10	RHPL9	RHPL8

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 15 / Receive Bit Synchronous HDLC Maximum Packet Length (RHPL0 to RHPL15). If the Receive Octet Length Detection Enable (ROLD) bit is set to one, then the HDLC controller will check the number of received octets in a packet to see if they exceed the count in this register. If the length is exceeded, then the packet is aborted and the remainder is discarded. The definition of "octet length" is everything in between the opening and closing flags which includes the address field, control field, information field, and FCS.

Register Name: **THCDIS**
Register Description: **Transmit HDLC Channel Definition Indirect Select**
Register Address: **0480h**

7	6	5	4	3	2	1	0
n/a	n/a	HCID5	HCID4	HCID3	HCID2	HCID1	HCID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	n/a	n/a	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 5 / HDLC Channel ID (HCID0 to HCID5).

000000 (00h) = HDLC Channel Number 1 (Port #0)
000001 (01h) = HDLC Channel Number 2 (Port #1)
000010 (02h) = HDLC Channel Number 3 (Port #2)
000011 (03h) = HDLC Channel Number 4 (Port #3)
100110 (26h) = HDLC Channel Number 39 (Port #38)
100111 (27h) = HDLC Channel Number 40 (Port #39)
101000 (28h) = illegal setting
111111 (3 Fh) = illegal setting

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the internal Transmit HDLC Definition RAM, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the channel location indicated by the HCID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the THCD register, the IAB bit will be set to zero. When the host wishes to write data to the internal Transmit HDLC Definition RAM, this bit should be written to a zero by the host. This causes the device to take the data that is currently present in the THCD register and write it to the channel location indicated by the HCID bits. When the device has completed the write, the IAB will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **THCD**
Register Description: **Transmit HDLC Channel Definition**
Register Address: **0484h**

7	6	5	4	3	2	1	0
TABTE	TCFCS	TBF	TID	TCRC1	TCRC0	TIFS	TTRANS
15	14	13	12	11	10	9	8
n/a	n/a	n/a	TZSD	TFG3	TFG2	TFG1	TFG0

Note: bits that are underlined are read only, all other bits are read-write.

Bit 0 / Transmit Transparent Enable (TTRANS). When this bit is set low, the HDLC controller will generate flags and the FCS (if enabled via TCRC0/1) and perform zero stuffing. When this bit is set high, the HDLC controller does not generate flags or the FCS and does not perform zero stuffing.

0 = transparent mode disabled
1 = transparent mode enabled

Bit 1 / Transmit Interfill Select (TIFS).

0 = the interfill byte is 7 Eh (01111110)
1 = the interfill byte is FFh (11111111)

Bit 2 & Bit 3 / Transmit CRC Selection (TCRC0/TCRC1). These two bits are ignored if the HDLC channel is set into Transparent mode (TTRANS = 1).

TCRC1	TCRC0	Action
0	0	no CRC is generated
0	1	16-bit CRC (CCITT/ITU Q.921)
1	0	32-bit CRC
1	1	illegal state

Bit 4 / Transmit Invert Data Enable (TID). When this bit is set low, the outgoing HDLC packets are not inverted after being generated. When this bit is set high, the HDLC controller inverts all the data (flags, information fields, and FCS) after the packet has been generated.

0 = do not invert data
1 = invert all data (including flags and FCS)

Bit 5 / Transmit Bit Flip (TBF). When this bit is set low, the HDLC controller will obtain the first HDLC bit to be transmitted from the lowest bit position of the PCI Bus bytes (i.e. PAD[0] / PAD[8] / PAD[16] / PAD[24]). When this bit is set high, the HDLC controller will obtain the first HDLC bit to be transmitted from the highest bit position of the PCI Bus bytes (i.e. PAD[7] / PAD[15] / PAD[23] / PAD[31]).

0 = the first HDLC bit transmitted is obtained from the lowest bit position of the bytes on the PCI Bus
1 = the first HDLC bit transmitted is obtained from the highest bit position of the bytes on the PCI Bus

Bit 6 / Transmit Corrupt FCS (TCFCS). When this bit is set low, the HDLC controller will allow the Frame Checksum Sequence (FCS) to be transmitted as generated. When this bit is set high, the HDLC controller will invert all the bits of the FCS before transmission occurs. This is useful in debugging and testing HDLC channels at the system level.

0 = generate FCS normally
1 = invert all FCS bits

Bit 7 / Transmit Abort Enable (TABTE). When this bit is set low, the HDLC controller will perform normally only sending an Abort sequence (eight ones in a row) when an error occurs in the PCI Block or the FIFO underflows. When this bit is set high, the HDLC controller will continuously transmit an all ones pattern (i.e. an Abort sequence). This bit is still active when the HDLC controller is configured in the Transparent Mode (TTRANS = 1).

0 = normal generation of abort
1 = constant abort generated

Bits 8 to 11/ Transmit Flag Generation Bits 0 to 3 (TFG0/TFG1/TFG2/TFG3). These four bits determine how many flags and interfill bytes will be sent in between consecutive packets.

TFG3	TFG2	TFG1	TFG0	Action
0	0	0	0	share closing and opening flag
0	0	0	1	closing flag / no interfill bytes / opening flag
0	0	1	0	closing flag / 1 interfill byte / opening flag
0	0	1	1	closing flag / 2 interfill bytes / opening flag
0	1	0	0	closing flag / 3 interfill bytes / opening flag
0	1	0	1	closing flag / 4 interfill bytes / opening flag
0	1	1	0	closing flag / 5 interfill bytes / opening flag
0	1	1	1	closing flag / 6 interfill bytes / opening flag
1	0	0	0	closing flag / 7 interfill bytes / opening flag
1	0	0	1	closing flag / 8 interfill bytes / opening flag
1	0	1	0	closing flag / 9 interfill bytes / opening flag
1	0	1	1	closing flag / 10 interfill bytes / opening flag
1	1	0	0	closing flag / 11 interfill bytes / opening flag
1	1	0	1	closing flag / 12 interfill bytes / opening flag
1	1	1	0	closing flag / 13 interfill bytes / opening flag
1	1	1	1	closing flag / 14 interfill bytes / opening flag

Bit 12 / Transmit Zero Stuffing Disable (TZSD). When this bit is set low, the HDLC controller will perform zero stuffing on the outgoing data stream. When this bit is set high, the outgoing data stream is not zero stuffed. This bit is ignored when the HDLC controller is configured in the Transparent Mode (TTRANS = 1).

0 = zero stuffing enabled
1 = zero stuffing disabled

SECTION 7: FIFO

7.1 GENERAL DESCRIPTION & EXAMPLE

BoSS contains one 8k byte FIFO for the receive path and another 8k byte FIFO for the transmit path. Both of these FIFOs are organized into Blocks. A Block is defined as four dwords (i.e. 16 bytes). Hence each FIFO is made up of 512 Blocks. See the FIFO example in Figure 7.1A.

The FIFO contains a state machine that is constantly polling the 40 ports to determine if any data is ready for transfer to/from the FIFO from/to the HDLC engines. The 40 ports are priority decoded with Port 0 getting the highest priority and Port 39 getting the lowest priority. Hence, all of the enabled HDLC channels on the lower numbered ports are serviced before the higher numbered ports. As long as the maximum throughput rate of 132 Mbps is not exceeded, the DS3131 has been designed to insure that there is enough bandwidth in this transfer to prevent any loss of data in between the HDLC Engines and the FIFO.

The FIFO also controls which HDLC channel the DMA should service to read data out of the FIFO on the receive side and to write data into the FIFO on the transmit side. Which channel gets the highest priority from the FIFO is configurable via some control bits in the Master Configuration (MC) register (see Section 4.2). There are two control bits for the receive side (RFPC0 and RFPC1) and two control bits for the transmit side (TFPC0 and TFPC1) that will determine the priority algorithm as shown in Table 7.1A. When a HDLC channel is priority decoded, the lower the number of the HDLC channel, the higher the priority. Hence HDLC channel number 1 always has the highest priority in the priority decoded scheme.

FIFO PRIORITY ALGORITHM SELECT TABLE 7.1A

Option	HDLC Channels that are Priority Decoded	HDLC Channels that are Serviced Round Robin
1	none	1 to 40
2	1 to 2	3 to 40
3	1 to 4	5 to 40
4	1 to 16	17 to 40

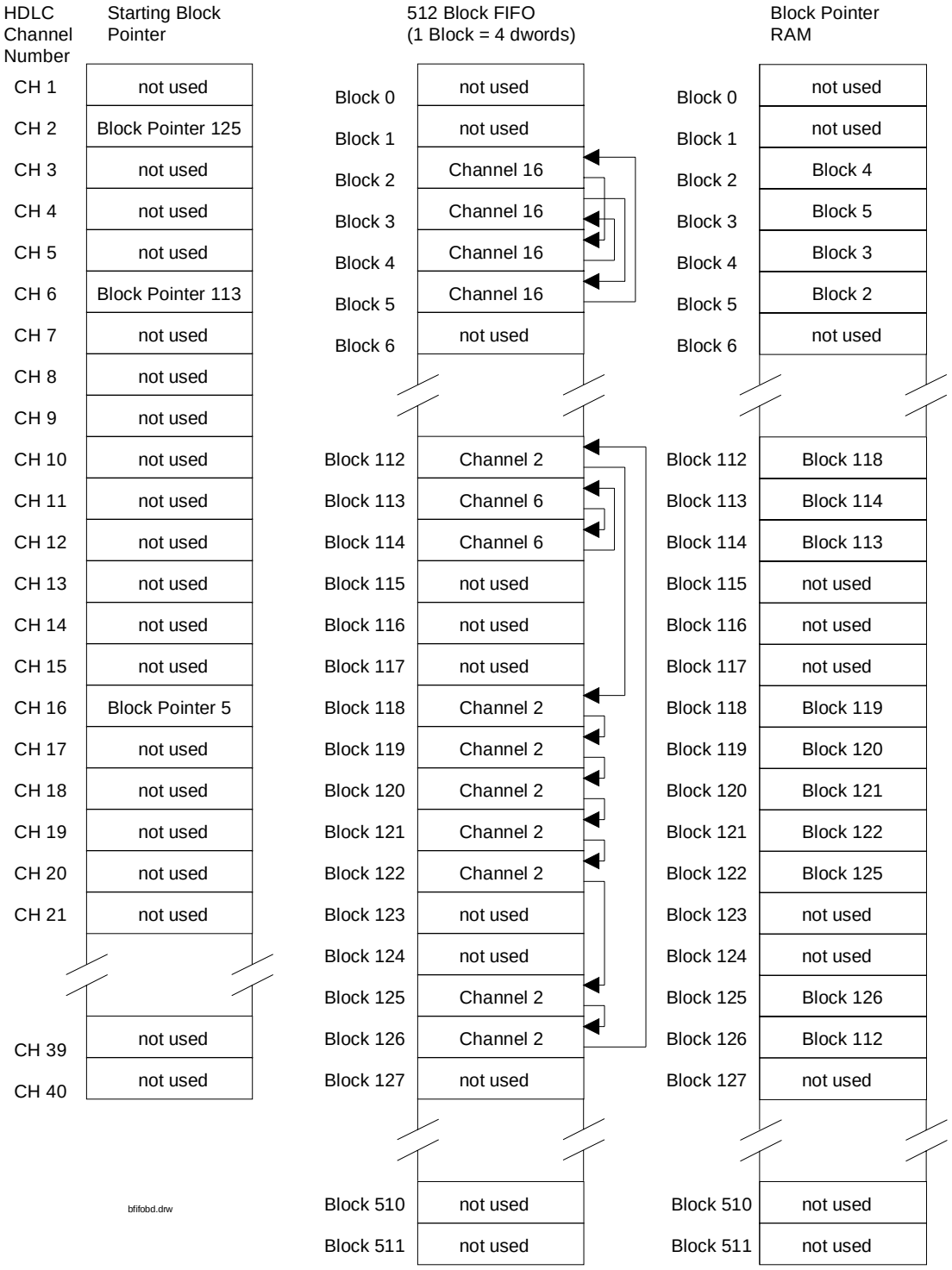
To maintain maximum flexibility for channel reconfiguration, each Block within the FIFO can be assigned to any of the 40 HDLC channels. Also, Blocks are link-listed together to form a chain whereby each Block points to the next Block in the chain. The minimum size of the link-listed chain is 4 Blocks (64 bytes) and the maximum is the full size of the FIFO which is 512 Blocks.

To assign a set of Blocks to a particular HDLC channel, the Host must configure the Starting Block Pointer and the Block Pointer RAM. The Starting Block Pointer assigns a particular HDLC channel to a set of linklisted Blocks by pointing to one of the Blocks within the chain (it does not matter which Block in the chain is pointed to). The Block Pointer RAM must be configured for each Block that is being used within the FIFO. The Block Pointer RAM indicates the next Block in the link-listed chain.

Figure 7.1A shows an example of how to configure the Starting Block Pointer and the Block Pointer RAM. In this example, only three HDLC channels are being used (channels 2, 6, and 16). The device knows that channel 2 has been assigned to the eight link-listed Blocks of 112, 118, 119, 120, 121, 122, 125, and 126 because a Block Pointer of 125 has been programmed into the channel 2 position of the Starting Block Pointer. The Block Pointer RAM tells the device how to link the eight Blocks together to form a circular chain.

The Host must set the WaterMarks for the receive and transmit paths. The receive path has a High Water Mark and the transmit path has a Low Water Mark.

FIFO EXAMPLE Figure 7.1A



RECEIVE HIGH WATER MARK

The High Water Mark indicates to the device how many Blocks should be written into the receive FIFO by the HDLC controllers before the DMA will begin sending the data to the PCI Bus. Or are other words, how full should the FIFO get before it should be emptied by the DMA. When the DMA begins reading the data from the FIFO, it will read all available data and try to completely empty the FIFO even if one or more EOF (End of Frames) are detected. As an example, if four Blocks were link-listed together and the Host programmed the High Water Mark to three Blocks, then the DMA would read the data out of the FIFO and transfer it to the PCI Bus after the HDLC controller has written three complete Blocks in succession into the FIFO and still had one Block left to fill. The DMA would not read the data out of the FIFO again until another three complete Blocks had been written into the FIFO in succession by the HDLC controller or until an EOF was detected. In this example of four Blocks being link-listed together, the High Water Mark could also be set to 1 or 2 but no other values would be allowed. If an incoming packet does not fill the FIFO enough to reach the High Water Mark before an EOF is detected, the DMA will still request that the data be sent to the PCI Bus, it will not wait for additional data to be written into the FIFO by the HDLC controllers.

TRANSMIT LOW WATER MARK

The Low Water Mark indicates to the device how many Blocks should be left in the FIFO before the DMA should begin getting more data from the PCI Bus. Or in other words, how empty should the FIFO get before it should be filled again by the DMA. When the DMA begins reading the data from the PCI Bus, it will read all available data and try to completely fill the FIFO even if one or more EOF (i.e. HDLC packets) are detected. As an example, if five Blocks were link-listed together and the Host programmed the Low Water Mark to two Blocks, then the DMA would read the data from the PCI Bus and transfer it to the FIFO after the HDLC controller has read three complete Blocks in succession from the FIFO and hence still had two blocks left before the FIFO was empty. The DMA would not read the data from the PCI Bus again until another three complete Blocks had been read from the FIFO in succession by the HDLC controllers. In this example of five Blocks being link-listed together, the Low Water Mark could also be set to any value from 1 to 4 (inclusive) but no other values would be allowed. When a new packet is written into a completely empty FIFO by the DMA, the HDLC controllers will wait until the FIFO fills beyond the Low Water Mark or until an EOF is seen before reading the data out of the FIFO.

7.2 FIFO REGISTER DESCRIPTION

Register Name: **RFSBPIS**
Register Description: **Receive FIFO Starting Block Pointer Indirect Select**
Register Address: **0900h**

7	6	5	4	3	2	1	0
n/a	n/a	HCID5	HCID4	HCID3	HCID2	HCID1	HCID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	n/a	n/a	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 5 / HDLC Channel ID (HCID0 to HCID5).
000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the current internal Receive Block Pointer, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the channel location indicated by the HCID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the RFSBP register, the IAB bit will be set to zero. When the host wishes to write data to set the internal Receive Starting Block Pointer, this bit should be written to a zero by the host. This causes the device to take the data that is currently present in the RFSBP register and write it to the channel location indicated by the HCID bits. When the device has completed the write, the IAB will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **RFSBP**
Register Description: **Receive FIFO Starting Block Pointer**
Register Address: **0904h**

7	6	5	4	3	2	1	0
RSBP7	RSBP6	RSBP5	RSBP4	RSBP3	RSBP2	RSBP1	RSBP0
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	n/a	n/a	RSBP8

Note: bits that are underlined are read only, all other bits are read-write.

Bits 0 to 8 / Starting Block Pointer (RSBP0 to RSBP8). These nine bits determine which of the 512 blocks within the receive FIFO, the host wants the device to configure as the starting block for a particular HDLC channel. Any of the blocks within a chain of blocks for a HDLC channel can be configured as the starting block. When these nine bits are read, they will report the current Block Pointer being used to write data into the Receive FIFO from the HDLC controllers.

000000000 (000h) = Use Block 0 as the Starting Block
111111111 (1FFh) = Use Block 511 as the Starting Block

Register Name: **RFBPIS**
Register Description: **Receive FIFO Block Pointer Indirect Select**
Register Address: **0910h**

7	6	5	4	3	2	1	0
BLKID7	BLKID6	BLKID5	BLKID4	BLKID3	BLKID2	BLKID1	BLKID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	n/a	n/a	n/a	BLKID8

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 8 / Block ID (BLKID0 to BLKID8).

000000000 (000h) = Block Number 0
111111111 (1FFh) = Block Number 511

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the internal Receive Block Pointer RAM, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the block location indicated by the BLKID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the RFBP register, the IAB bit will be set to zero. When the host wishes to write data to the internal Receive Block Pointer RAM, this bit should be written to a zero by the host. This causes the device to take the data that is current present in the RFBP register and write it to the channel location indicated by the BLKID bits. When the device has completed the write, the IAB will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **RFBP**
Register Description: **Receive FIFO Block Pointer**
Register Address: **0914h**

7	6	5	4	3	2	1	0
RBP7	RBP6	RBP5	RBP4	RBP3	RBP2	RBP1	RBP0
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	n/a	RBP9	RBP8

Note: bits that are underlined are read only, all other bits are read-write.

Bits 0 to 8 / Block Pointer (RBP0 to RBP8). These nine bits indicate which of the 512 blocks is the next block in the link list chain. A block is not allowed to point to itself.

000000000 (000h) = Block 0 is the Next Linked Block
111111111 (1FFh) = Block 511 is the Next Linked Block

Register Name: **RFHWMIS**
Register Description: **Receive FIFO High Water Mark Indirect Select**
Register Address: **0920h**

7	6	5	4	3	2	1	0
n/a	n/a	HCID5	HCID4	HCID3	HCID2	HCID1	HCID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	n/a	n/a	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 5 / HDLC Channel ID (HCID0 to HCID5).

000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the internal Receive High Water Mark RAM, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the channel location indicated by the HCID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the RFHWM register, the IAB bit will be set to zero. When the host wishes to write data to the internal Receive High Water Mark RAM, this bit should be written to a zero by the host. This causes the device to take the data that is currently present in the RFHWM register and write it to the channel location indicated by the HCID bits. When the device has completed the write, the IAB will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **RFHWM**
Register Description: **Receive FIFO High Water Mark**
Register Address: **0924h**

7	6	5	4	3	2	1	0
RHWM7	RHWM6	RHWM5	RHWM4	RHWM3	RHWM2	RHWM1	RHWM0
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	n/a	n/a	RHWM8

Note: bits that are underlined are read only, all other bits are read-write.

Bits 0 to 8 / High Water Mark (RHWM0 to RHWM8). These nine bits indicate the setting of the Receive High Water Mark. The High Water Mark setting is the number of successive blocks that the HDLC controller will write to the FIFO before the DMA will send the data to the PCI Bus. The High Water Mark setting must be between (inclusive) one block and one less than the number of blocks in the link-list chain for the particular channel involved. For example, if four blocks are linked together, then the High Water Mark can be set to either 1, 2 or 3.

000000000 (000h) = invalid setting
000000001 (001h) = High Water Mark is 1 Block
000000010 (002h) = High Water Mark is 2 Blocks
111111111 (1FFh) = High Water Mark is 511 Blocks

Register Name: **TFSBPIS**
Register Description: **Transmit FIFO Starting Block Pointer Indirect Select**
Register Address: **0980h**

7	6	5	4	3	2	1	0
n/a	n/a	HCID5	HCID4	HCID3	HCID2	HCID1	HCID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	n/a	n/a	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 5 / HDLC Channel ID (HCID0 to HCID5).

000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the current internal Transmit Block Pointer, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the channel location indicated by the HCID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the TFSBP register, the IAB bit will be set to zero. When the host wishes to write data to the internal Transmit Starting Block Pointer RAM, this bit should be written to a zero by the host. This causes the device to take the data that is currently present in the TFSBP register and write it to the channel location indicated by the HCID bits. When the device has completed the write, the IAB will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **TFSBP**
Register Description: **Transmit FIFO Starting Block Pointer**
Register Address: **0984h**

7	6	5	4	3	2	1	0
TSBP7	TSBP6	TSBP5	TSBP4	TSBP3	TSBP2	TSBP1	TSBP0
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	n/a	n/a	TSBP8

Note: bits that are underlined are read only, all other bits are read-write.

Bits 0 to 8 / Starting Block Pointer (TSBP0 to TSBP8). These nine bits determine which of the 512 blocks within the transmit FIFO, the host wants the device to configure as the starting block for a particular HDLC channel. Any of the blocks within a chain of blocks for a HDLC channel can be configured as the starting block. When these nine bits are read, they will report the current Block Pointer being used to read data from the Transmit FIFO by the HDLC controllers.

000000000 (000h) = Use Block 0 as the Starting Block
111111111 (1FFh) = Use Block 511 as the Starting Block

Register Name: **TFBPIS**
Register Description: **Transmit FIFO Block Pointer Indirect Select**
Register Address: **0990h**

7	6	5	4	3	2	1	0
BLKID7	BLKID6	BLKID5	BLKID4	BLKID3	BLKID2	BLKID1	BLKID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	n/a	n/a	n/a	BLKID8

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 8 / Block ID (BLKID0 to BLKID8).

000000000 (000h) = Block Number 0
111111111 (1FFh) = Block Number 511

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the internal Transmit Block Pointer RAM, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the block location indicated by the BLKID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the TFBP register, the IAB bit will be set to zero. When the host wishes to write data to the internal Transmit Block Pointer RAM, this bit should be written to a zero by the host. This causes the device to take the data that is currently present in the TFBP register and write it to the channel location indicated by the BLKID bits. When the device has completed the write, the IAB will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **TFBP**
Register Description: **Transmit FIFO Block Pointer**
Register Address: **0994h**

7	6	5	4	3	2	1	0
TBP7	TBP6	TBP5	TBP4	TBP3	TBP2	TBP1	TBP0
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	n/a	n/a	TBP8

Note: bits that are underlined are read only, all other bits are read-write.

Bits 0 to 8 / Block Pointer (TBP0 to TBP8). These nine bits indicate which of the 512 blocks is the next block in the link list chain. A block is not allowed to point to itself.

000000000 (000h) = Block 0 is the Next Linked Block
111111111 (1FFh) = Block 511 is the Next Linked Block

Register Name: **TFLWMIS**
Register Description: **Transmit FIFO Low Water Mark Indirect Select**
Register Address: **09A0h**

7	6	5	4	3	2	1	0
n/a	n/a	HCID5	HCID4	HCID3	HCID2	HCID1	HCID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	n/a	n/a	n/a	n/a

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 5 / HDLC Channel ID (HCID0 to HCID5).

000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the internal Transmit Low WaterMark RAM, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the channel location indicated by the HCID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the TFLWM register, the IAB bit will be set to zero. When the host wishes to write data to the internal Transmit Low WaterMark RAM, this bit should be written to a zero by the host. This causes the device to take the data that is currently present in the TFLWM register and write it to the channel location indicated by the HCID bits. When the device has completed the write, the IAB will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **TFLWM**
Register Description: **Transmit FIFO Low Water Mark**
Register Address: **09A4h**

7	6	5	4	3	2	1	0
TLWM7	TLWM6	TLWM5	TLWM4	TLWM3	TLWM2	TLWM1	TLWM0
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	n/a	n/a	TLWM8

Note: bits that are underlined are read only, all other bits are read-write.

Bits 0 to 8 / Low Water Mark (TLWM0 to TLWM8). These nine bits indicate the setting of the Transmit Low Water Mark. The Low Water Mark setting is the number of Blocks left in the Transmit FIFO before the DMA will get more data from the PCI Bus. The Low Water Mark setting must be between (inclusive) 1 block and one less than the number of blocks in the link list chain for the particular channel involved. For example, if five blocks are linked together, then the Low Water Mark can be set to either 1, 2, 3, or 4.

000000000 (000h) = invalid setting
000000001 (001h) = Low Water Mark is 1 Block
000000010 (002h) = Low Water Mark is 2 Blocks
111111111 (1FFh) = Low Water Mark is 511 Blocks

SECTION 8: DMA

8.0 INTRODUCTION

The DMA block (see Figure 1A) handles the transfer of packet data from the FIFO block to the PCI block and vice versa. Throughout this Section, the terms *Host* and *Descriptor* will be used. *Host* is defined as the CPU or intelligent controller that sits on the PCI Bus and instructs the device on how to handle the incoming and outgoing packet data. *Descriptor* is defined as a pre-formatted message that is passed from the Host to the DMA block or vice versa to indicate where packet data should be placed or obtained from.

On power-up, the DMA will be disabled because the RDE and TDE control bits in the Master Configuration register (see Section 4) will be set to zero. The Host must configure the DMA by writing to all of the registers listed in Table 8.0A (which includes all 40 channel locations in the Receive and Transmit Configuration RAMs) then enable the DMA by setting to the RDE and TDE control bits to one.

The structure of the DMA is such that the receive and transmit side descriptor address spaces can be shared even among multiple chips on the same bus. Via the Master Control (MC) register, the Host will determine how long the DMA will be allowed to burst onto the PCI bus. The default value is 32 dwords (128 bytes) but via the DT0 and DT1 control bits, the Host can enable the receive or transmit DMAs to burst either 64 dwords (256 bytes), 128 dwords (512 bytes), or 256 dwords (1024 bytes).

The receive and transmit Packet Descriptors have almost identical structures (see Sections 8.1.2 and 8.2.2) which provides a minimal amount of Host intervention in store-and-forward applications. In other words, the receive descriptors created by the receive DMA can be used directly by the transmit DMA.

The receive and transmit portions of the DMA are completely independent and will be discussed separately.

DMA REGISTERS THAT MUST BE CONFIGURED BY THE HOST ON POWER-UP Table 8.0A

Address	Acronym	Register	Section
0700	RFQBA0	Receive Free Queue Base Address 0 (lower word).	8.1.3
0704	RFQBA1	Receive Free Queue Base Address 1 (upper word).	8.1.3
0708	RFQEA	Receive Free Queue End Address.	8.1.3
070C	RFQSBSA	Receive Free Queue Small Buffer Start Address.	8.1.3
0710	RFQLBWP	Receive Free Queue Large Buffer Host Write Pointer.	8.1.3
0714	RFQSBWP	Receive Free Queue Small Buffer Host Write Pointer.	8.1.3
0718	RFQLBRP	Receive Free Queue Large Buffer DMA Read Pointer.	8.1.3
071C	RFQSBRP	Receive Free Queue Small Buffer DMA Read Pointer.	8.1.3
0730	RDQBA0	Receive Done Queue Base Address 0 (lower word).	8.1.4
0734	RDQBA1	Receive Done Queue Base Address 1 (upper word).	8.1.4
0738	RDQEA	Receive Done Queue End Address.	8.1.4
073C	RDQRP	Receive Done Queue Host Read Pointer.	8.1.4
0740	RDQWP	Receive Done Queue DMA Write Pointer.	8.1.4
0744	RDQFFT	Receive Done Queue FIFO Flush Timer.	8.1.4
0750	RDBA0	Receive Descriptor Base Address 0 (lower word).	8.1.2
0754	RDBA1	Receive Descriptor Base Address 1 (upper word).	8.1.2
0770	RDMACIS	Receive DMA Configuration Indirect Select.	8.1.5
0774	RDMAC	Receive DMA Configuration (all 40 channels).	8.1.5
0780	RDMAQ	Receive DMA Queues Control.	8.1.3/.4
0790	RLBS	Receive Large Buffer Size.	8.1.1
0794	RSBS	Receive Small Buffer Size.	8.1.1
0800	TPQBA0	Transmit Pending Queue Base Address 0 (lower word).	8.2.3
0804	TPQBA1	Transmit Pending Queue Base Address 1 (upper word).	8.2.3
0808	TPQEA	Transmit Pending Queue End Address.	8.2.3
080C	TPQWP	Transmit Pending Queue Host Write Pointer.	8.2.3
0810	TPQRP	Transmit Pending Queue DMA Read Pointer.	8.2.3
0830	TDQBA0	Transmit Done Queue Base Address 0 (lower word).	8.2.4
0834	TDQBA1	Transmit Done Queue Base Address 1 (upper word).	8.2.4
0838	TDQEA	Transmit Done Queue End Address.	8.2.4
083C	TDQRP	Transmit Done Queue Host Read Pointer.	8.2.4
0840	TDQWP	Transmit Done Queue DMA Write Pointer.	8.2.4
0844	TDQFFT	Transmit Done Queue FIFO Flush Timer.	8.2.4
0850	TPDBA0	Transmit Descriptor Base Address 0 (lower word).	8.2.2
0854	TPDBA1	Transmit Descriptor Base Address 1 (upper word).	8.2.2
0870	TDMACIS	Transmit DMA Configuration Indirect Select.	8.2.5
0874	TDMAC	Transmit DMA Configuration (all 40 channels).	8.2.5
0880	TDMAQ	Transmit Queues FIFO Control.	8.2.3/.4

8.1 RECEIVE SIDE

8.1.1 OVERVIEW

The receive DMA uses a scatter gather technique to write packet data into main memory. The Host will keep track of and decide where the DMA should place the incoming packet data. There are a set of descriptors that get handed back and forth between the DMA and the Host. Via these descriptors the Host can inform the DMA where to place the packet data and the DMA can tell the Host when the data is ready to be processed.

The operation of the receive DMA has three main areas as shown in Figures 8.1.1A and 8.1.1B and Table 8.1.1A. The Host will write to the Free Queue Descriptors informing the DMA where it can place the incoming packet data. Associated with each free data buffer location is a free Packet Descriptor where the DMA can write information to inform the Host about the attributes of the packet data (i.e. status information, number of bytes, etc.) that it will output. To accommodate the various needs of packet data, the Host can quantize the free data buffer space into two different buffer sizes. The Host will set the size of the buffers via the Receive Large Buffer Size (RLBS) and the Receive Small Buffer Size (RSBS) registers.

Register Name: **RLBS**
Register Description: **Receive Large Buffer Size Select**
Register Address: **0790h**

7	6	5	4	3	2	1	0
LBS7	LBS6	LBS5	LBS4	LBS3	LBS2	LBS1	LBS0
15	14	13	12	11	10	9	8
n/a	n/a	n/a	LBS12	LBS11	LBS10	LBS9	LBS8

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 12 / Large Buffer Select Bit (LBS0 to LBS12).
0000000000000 (0000h) = Buffer Size is 0 Bytes
1111111111111 (1FFFh) = Buffer Size is 8191 Bytes

Register Name: **RSBS**
Register Description: **Receive Small Buffer Size Select**
Register Address: **0794h**

7	6	5	4	3	2	1	0
SBS7	SBS6	SBS5	SBS4	SBS3	SBS2	SBS1	SBS0
15	14	13	12	11	10	9	8
n/a	n/a	n/a	SBS12	SBS11	SBS10	SBS9	SBS8

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 12 / Small Buffer Select Bit (SBS0 to SBS12).
0000000000000 (0000h) = Buffer Size is 0 Bytes
1111111111111 (1FFFh) = Buffer Size is 8191 Bytes

On a HDLC channel basis in the Receive DMA Configuration RAM, the Host will instruct the DMA on how to use the large and small buffers for the incoming packet data on that particular HDLC channel. The Host has three options (1) only use Large Buffers, (2) only use Small Buffers, and (3) first fill a Small Buffer then if the incoming packet requires more buffer space, use one or more Large Buffers for the remainder of the packet. The Host selects which option via the Size field in the Receive Configuration RAM (see Section 8.1.5). Large Buffers are best used for data intensive, time insensitive packets like graphics files whereas small buffers are best used for time sensitive information like real-time voice.

RECEIVE DMA MAIN OPERATIONAL AREAS Table 8.1.1A

Name	Section	Description
Packet Descriptors	8.1.2	A dedicated area of memory that describes the location and attributes of the packet data.
Free Queue Descriptors	8.1.3	A dedicated area of memory that the Host will write to inform the DMA where to store incoming packet data.
Done Queue Descriptors	8.1.4	A dedicated area of memory that the DMA will write to inform the Host that the packet data is ready for processing.

The Done Queue Descriptors contain information that the DMA wishes to pass to the Host. Via the Done Queue Descriptors the DMA informs the Host about the incoming packet data and where to find the Packet Descriptors that it has written into main memory. Each completed Descriptor contains the starting address of the data buffer where the packet data is stored.

If enabled, the DMA can burst read the Free Queue Descriptors and burst write the Done Queue Descriptors. This helps minimize PCI Bus accesses, freeing the PCI Bus up to do more time critical functions. See Sections 8.1.3 and 8.1.4 for more details on this feature.

RECEIVE DMA ACTIONS

A typical scenario for the Receive DMA is as follows:

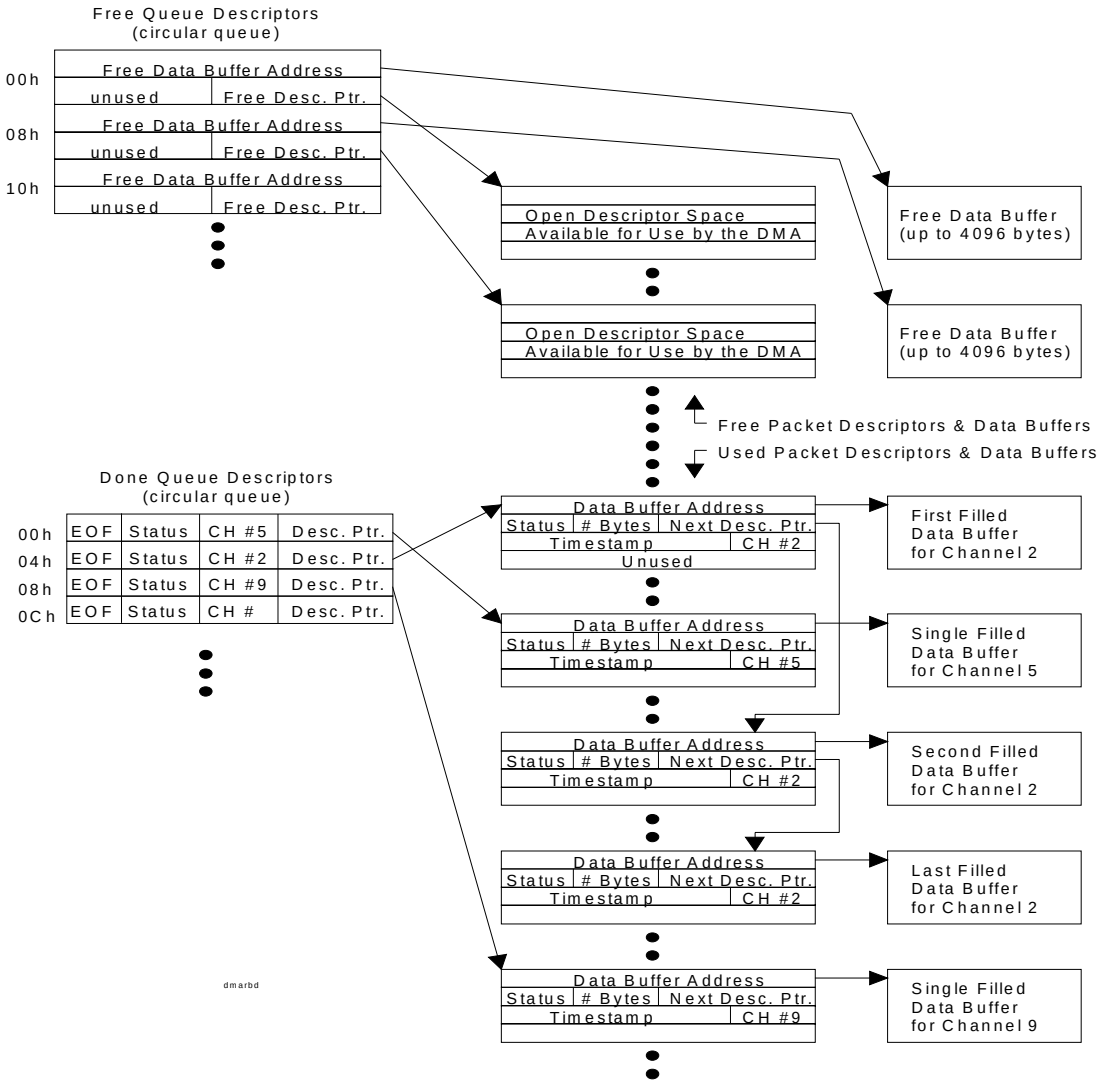
1. The receive DMA gets a request from the Receive FIFO that it has packet data that needs to be sent to the PCI Bus.
2. The receive DMA determines whether the incoming packet data should be stored in a large buffer or a small buffer.
3. The receive DMA then reads a Free Queue Descriptor (either by reading a single descriptor or a burst of descriptors) indicating where in main memory there exists some free data buffer space and where the associated free Packet Descriptor resides.
4. The receive DMA starts storing packet data in the previously free buffer data space by writing it out through the PCI Bus.
5. When the receive DMA realizes that the current data buffer is filled (by knowing the buffer size it can calculate this), it then reads another Free Queue Descriptor to find another free data buffer and Packet Descriptor location.
6. The receive DMA then writes the previous Packet Descriptor and creates a linked list by placing the current descriptor in the Next Descriptor Pointer field and then it starts filling the new buffer location. Figure 8.1.1A provides an example of Packet Descriptors being link listed together (see Channel 2).
7. This continues to all of the packet data is stored.
8. The receive DMA will either wait until a packet has been completely received or until a programmable number (from 1 to 7) of data buffers have been filled before writing the Done Queue Descriptor which indicates to the Host that packet data is ready for processing.

HOST ACTIONS

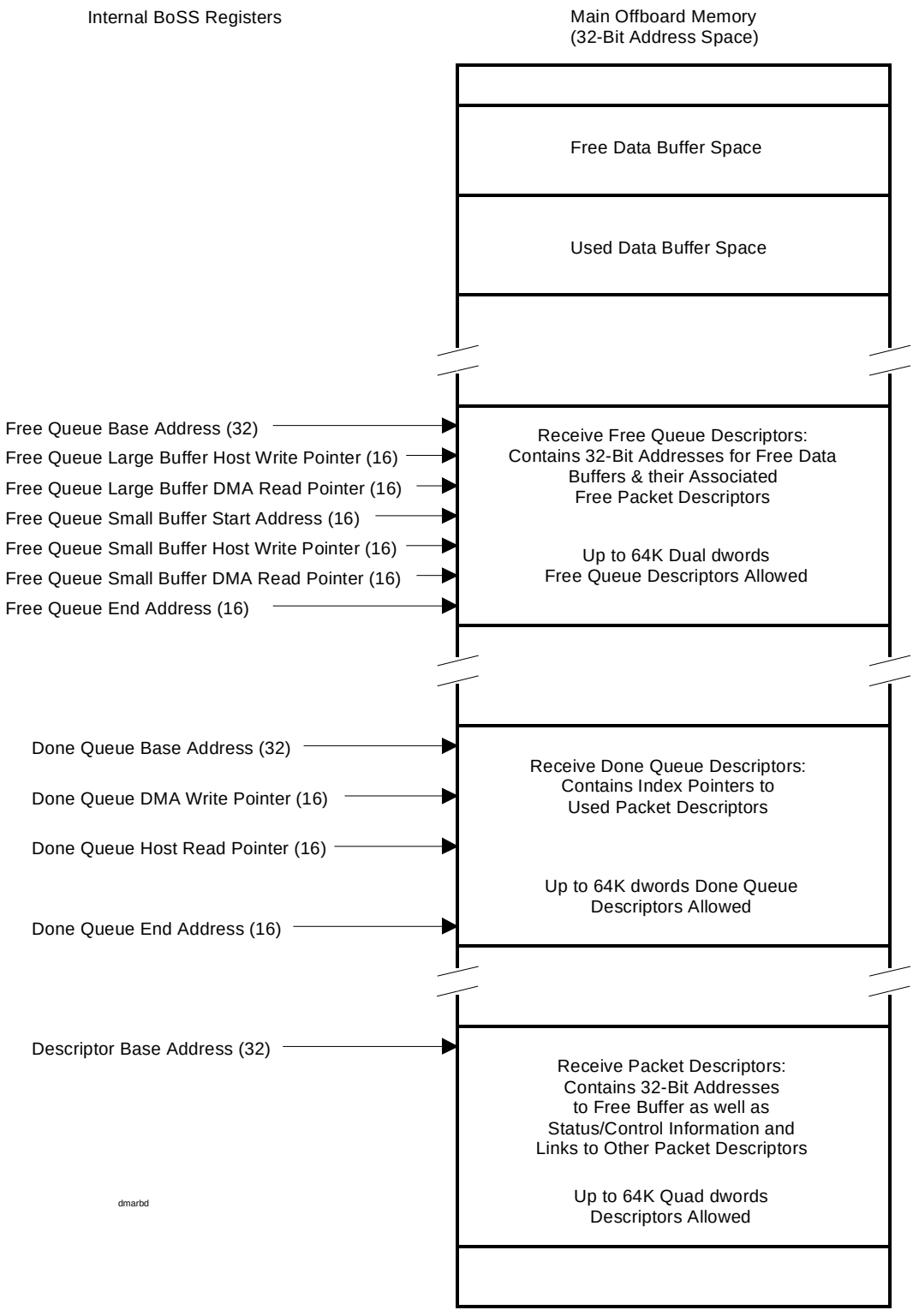
The Host will typically handle the receive DMA as follows:

1. The Host is always trying to make available free data buffer space and hence it tries to fill the Free Queue Descriptor.
2. The Host will either poll or be interrupted that some incoming packet data is ready for processing.
3. The Host then reads the Done Queue Descriptor circular queue to find out which channel has data available, what the status is, and where the receive Packet Descriptor is located.
4. The Host then reads the receive Packet Descriptor and begins processing the data.
5. The Host then reads the Next Descriptor Pointer in the link listed chain and continues this process until either a number (from 1 to 7) of descriptors have been processed or an end of packet has been reached.
6. The Host then checks the Done Queue Descriptor circular queue to see if any more data buffers are ready for processing.

RECEIVE DMA OPERATION Figure 8.1.1A



RECEIVE DMA MEMORY ORGANIZATION Figure 8.1.1B



8.1.2 PACKET DESCRIPTORS

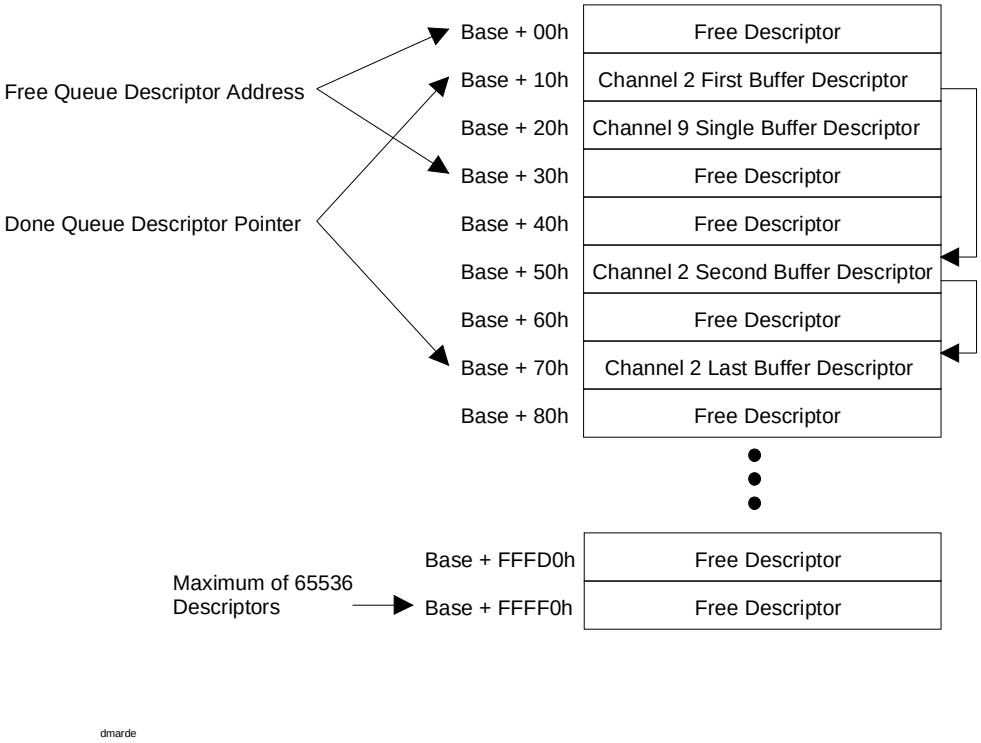
In main memory resides a contiguous section up to 65,536 quad dwords that make up the Receive Packet Descriptors. The Receive Packet Descriptors are aligned on a quad dword basis and can be placed anywhere in the 32-bit address space via the Receive Descriptor Base Address (see Table 8.1.2A). Associated with each descriptor is a data buffer. The data buffer can be up to 8191 bytes long and must be a contiguous section of main memory. The host can set two different data buffer sizes via the Receive Large Buffer Size (RLBS) and the Receive Small Buffer Size (RSBS) registers (see Section 8.1.1). If an incoming packet requires more space than the data buffer allows, then Packet Descriptors will be link-listed together by the DMA to provide a chain of data buffers. Figure 8.1.2A is an example of how three descriptors were linked together for an incoming packet on HDLC Channel 2. Figure 8.1.1A shows a similar example. Channel 9 only required a single data buffer and hence only one Packet Descriptor was used.

Packet Descriptors can be either free (i.e. available for use by the DMA) or used (i.e. currently contain data that needs to be processed by the host). Free Packet Descriptors are pointed to by the Free Queue Descriptors and used Packet Descriptors are pointed to by the Done Queue Descriptors.

RECEIVE DESCRIPTOR ADDRESS STORAGE Table 8.1.2A

Register Name	Acronym	Address
Receive Descriptor Base Address 0 (lower word)	RDBA0	0750h
Receive Descriptor Base Address 1 (upper word)	RDBA1	0754h

RECEIVE DESCRIPTOR EXAMPLE Figure 8.1.2A



RECEIVE PACKET DESCRIPTORS Figure 8.1.2B

dword 0

Data Buffer Address (32)

dword 1

BUFS (3)	Byte Count (13)	Next Descriptor Pointer (16)

dword 2

Timestamp (24)	00b	HDLC CH#(6)

dword 3

unused (32)

(Note: the organization of the Receive Descriptor is not affected by the enabling of Big Endian)

dword 0; Bits 0 to 31 / Data Buffer Address. Direct 32-bit starting address of the data buffer that is associated with this receives descriptor.

dword 1; Bits 0 to 15 / Next Descriptor Pointer. This 16-bit value is the offset from the Receive Descriptor Base Address of the next descriptor in the chain. Only valid if Buffer Status = 001 or 010.

dword 1; Bits 16 to 28 / Byte Count. Number of bytes stored in the data buffer. Maximum is 8191 bytes (0000h = 0 bytes / 1FFFh = 8191 bytes). This byte count does not include the buffer offset. The Host will determine the buffer offset (if any) via the Buffer Offset field in the Receive DMA Configuration RAM (see Section 8.1.5).

dword 1; Bits 29 to 31 / Buffer Status. Must be one of the three states listed below.

001 = first buffer of a multiple buffer packet
010 = middle buffer of a multiple buffer packet
100 = last buffer of a multiple or single buffer packet (equivalent to EOF)

dword 2; Bits 0 to 5 / HDLC Channel Number. HDLC channel number, which can be from 1 to 40.

000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

dword 2; Bits 6 to 7 / Unused. Set to 00b by the DMA.

dword 2; Bits 8 to 31 / Timestamp. When each descriptor is written into memory by the DMA, this 24-bit timestamp is provided to keep track of packet arrival times. The timestamp is based on the PCLK frequency divided by 16. For a 33 MHz PCLK, the timestamp will increment every 485 ns and will rollover every 8.13 seconds. For a 25 MHz clock, the timestamp will increment every 640 ns and will rollover every 10.7 seconds. The host can calculate the difference in arrival times of packets by knowing the PCLK frequency and then taking the difference in timestamp readings between consecutive Packet Descriptors.

dword 3; Bits 0 to 31 / Unused. Not written to by the DMA. Can be used by the host. Application Note: dword 3 is used by the Transmit DMA and in store and forward applications, the Receive and Transmit Packet Descriptors have been designed to eliminate the need for the Host to groom the descriptors before transmission. In these type of applications, the Host should not use dword 3 of the Receive Packet Descriptor.

8.1.3 FREE QUEUE

The Host will write to the Receive Free Queue, the 32-bit addresses of the available (i.e. free) data buffers and their associated Packet Descriptors. The descriptor space is indicated via a 16-bit pointer which the DMA will use along with the Receive Packet Descriptor Base Address to find the exact 32-bit address of the associated Receive Packet Descriptor.

RECEIVE FREE QUEUE DESCRIPTOR Figure 8.1.3A

dword 0

Free Data Buffer Address (32)

dword 1

Unused (16)	Free Packet Descriptor Pointer (16)

(Note: the organization of the Free Queue is not affected by the enabling of Big Endian)

dword 0; Bits 0 to 31 / Data Buffer Address. Direct 32-bit starting address of a free data buffer.

dword 1; Bits 0 to 15 / Free Packet Descriptor Pointer. This 16-bit value is the offset from the Receive Descriptor Base Address of the free descriptor space associated with the free data buffer in dword 0.

dword 1; Bits 16 to 31 / Unused. Not used by the DMA. Can be set to any value by the Host and will be ignored by the Receive DMA.

The Receive DMA will read from the Receive Free Queue Descriptor circular queue which data buffers and their associated descriptors are available for use by the DMA.

The Receive Free Queue Descriptor is actually a set of two circular queues. See Figure 8.1.3B. There is one circular queue that indicates where free large buffers and their associated free descriptors exist and there is another circular queue that indicates where free small buffers and their associated free descriptors exist.

LARGE AND SMALL BUFFER SIZE HANDLING

Via the Receive Configuration RAM Buffer Size field, the DMA knows for a particular HDLC channel, whether the incoming packets should be stored in the large or the small free data buffers. The Host informs the DMA of the size of both the large and small buffers via the Receive Large and Small Buffer Size (RLBS/RSBS) registers. For example, when the DMA knows that data is ready to be written onto the PCI Bus, it checks to see if the data is to be sent to a large buffer or a small buffer and then it goes to the appropriate Free Queue Descriptor and pulls the next available free buffer address and free descriptor pointer. If the Host wishes to have only one buffer size, then the Receive Free Queue Small Buffer Start Address will be set equal to the Receive Free Queue End Address and in the Receive Configuration RAM, none of the active HDLC channels will be configured for the small buffer size.

To keep track of the addresses of the dual circular queues in the Receive Free Queue, there are a set of internal addresses within the device that are accessed by both the Host and the DMA. On initialization, the Host will configure all of the registers shown in Table 8.1.3B. After initialization, the DMA will only write to (i.e. change) the read pointers and the Host will only write to the write pointers.

EMPTY CASE

The Receive Free Queue is considered empty when the read and write pointers are identical.

RECEIVE FREE QUEUE EMPTY STATE

	empty descriptor	
	empty descriptor	
	empty descriptor	
read pointer >	empty descriptor	< write pointer
	empty descriptor	
	empty descriptor	
	empty descriptor	

FULL CASE

The Receive Free Queue is considered full when the read pointer is ahead of the write pointer by one descriptor. Hence, one descriptor must always remain empty.

RECEIVE FREE QUEUE FULL STATE

	valid descriptor	
	valid descriptor	
	empty descriptor	< write pointer
read pointer >	valid descriptor	
	valid descriptor	
	valid descriptor	
	valid descriptor	

Table 8.1.3A describes the manner in which to calculate the absolute 32-bit address of the read and write pointers for the Receive Free Queue.

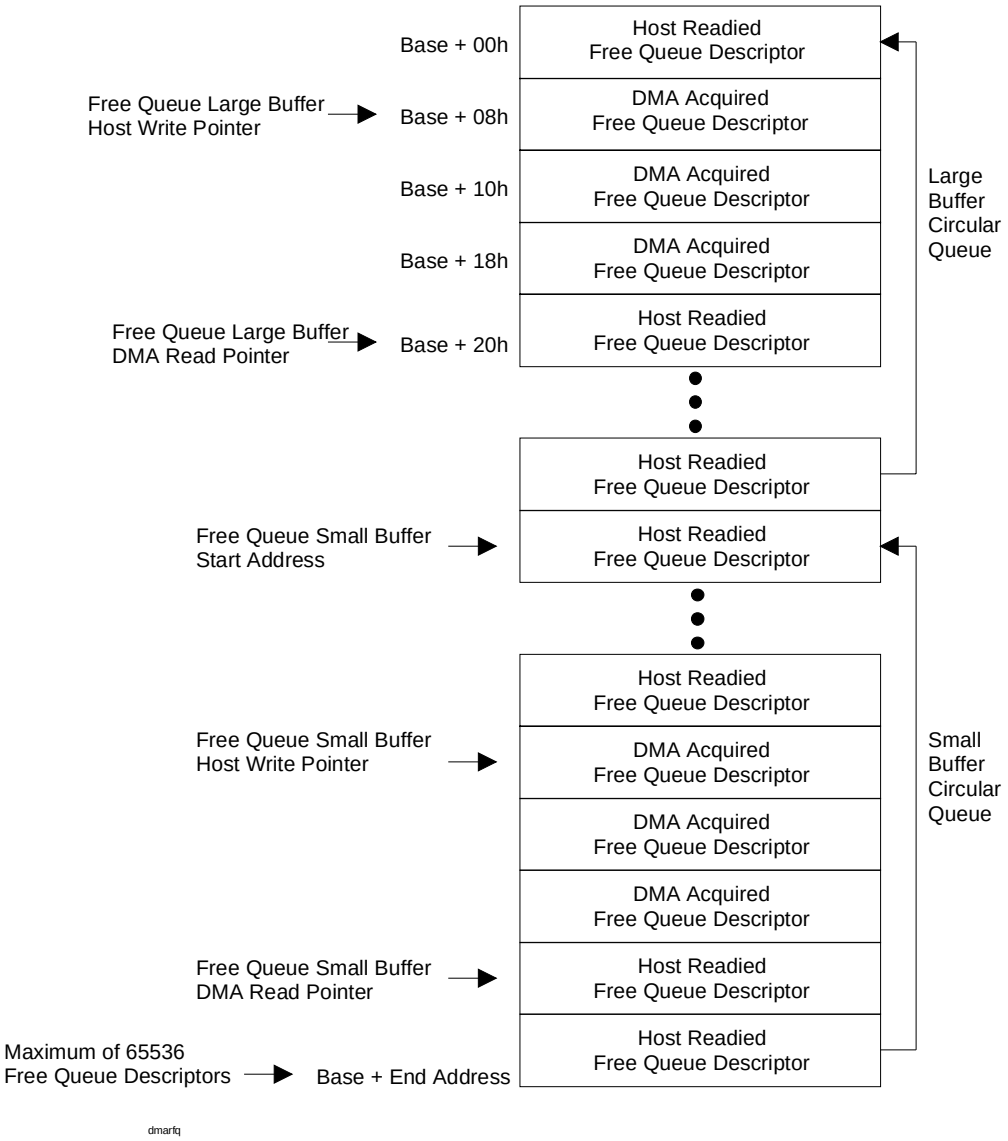
RECEIVE FREE QUEUE READ/WRITE POINTER ABSOLUTE ADDRESS CALCULATION Table 8.1.3A

Buffer	Algorithm
Large	Absolute Address = Free Queue Base + Write Pointer * 8
	Absolute Address = Free Queue Base + Read Pointer * 8
Small	Absolute Address = Free Queue Base + Small Buffer Start * 8 + Write Pointer * 8
	Absolute Address = Free Queue Base + Small Buffer Start * 8 + Read Pointer * 8

RECEIVE FREE QUEUE INTERNAL ADDRESS STORAGE Table 8.1.3B

Register Name	Acronym	Address
Receive Free Queue Base Address 0 (lower word)	RFQBA0	0700h
Receive Free Queue Base Address 1 (upper word)	RFQBA1	0704h
Receive Free Queue Large Buffer Host Write Pointer	RFQLBWP	0710h
Receive Free Queue Large Buffer DMA Read Pointer	RFQLBRP	0718h
Receive Free Queue Small Buffer Start Address	RFQSBSA	070Ch
Receive Free Queue Small Buffer Host Write Pointer	RFQSBWP	0714h
Receive Free Queue Small Buffer DMA Read Pointer	RFQSBRP	071Ch
Receive Free Queue End Address	RFQEA	0708h

RECEIVE FREE QUEUE STRUCTURE Figure 8.1.3B



Once the Receive DMA is activated (by setting the RDE control bit in the Master Configuration register; see Section 4), it can begin reading data out of the free queue. It knows where to read data out of the free queue by reading the Read Pointer and adding it to the Base Address to obtain the actual 32-bit address. Once the DMA has read the Free Queue, it increments the Read Pointer by two dwords. A check must be made to make sure the incremented address does not equal or exceed either the Receive Free Queue Small Buffer Start Address (in the case of the large buffer circular queue) or the Receive Free Queue End Address (in the case of the small buffer circular queue). If the incremented address does equal or exceed either of these addresses, then the incremented read pointer will be set equal to 0000h.

STATUS / INTERRUPTS

On each read of the Free Queue by the DMA, the DMA will set either the Status Bit for Receive DMA Large Buffer Read (RLBR) or the Status Bit for Receive DMA Small Buffer Read (RSBR) in the Status Register for DMA (SDMA). The DMA also checks the Receive Free Queue Large Buffer Host Write Pointer and the Receive Free Queue Small Buffer Host Write Pointer to make sure that a underflow does not occur. If it does occur, then the DMA will set either the Status Bit for Receive DMA Large Buffer Read Error (RLBRE) or the Status Bit for Receive DMA Small Buffer Read Error (RSBRE) in the Status Register for DMA (SDMA) and it will not read the Free Queue nor will it increment the Read Pointer. In such a scenario, the Receive FIFO may overflow if the Host does not provide Free Queue Descriptors. Each of the status bits can also (if enabled) cause an hardware interrupt to occur. See Section 4 for more details.

FREE QUEUE BURST READING

The DMA has the ability to read the Free Queue in bursts. This allows for a more efficient use of the PCI Bus. The DMA can grab messages from the Free Queue in-groups rather than one at a time, freeing up the PCI Bus for more time critical functions.

Internal to the device there is a FIFO that can store up to 16 Free Queue Descriptors (32 dwords since each descriptor occupies two dwords). If the Free Queue is operated as a dual circular queue supporting both large and small buffers, then the Receive Free Queue FIFO Control (RFQFC) bit should be set to zero and then the FIFO is cut into two 8 message FIFOs. If the Free Queue is operated as a single circular queue supporting only the large buffers, then the RFQFC bit should be set to one and then the FIFO is set up as a single 16 descriptor FIFO. The Host must configure the Free Queue FIFO for proper operation via the Receive DMA Queues Control (RDMAQ) register (see below).

When enabled via the Receive Free Queue FIFO Enable (RFQFE) bit, the Free Queue FIFO will not read the Free Queue until it reaches the Low Water Mark. When the FIFO reaches the Low Water Mark (which is two descriptors in the dual mode or four descriptors in the single mode) it will attempt to fill the FIFO with additional descriptors by burst reading the Free Queue. Before it reads the Free Queue, it checks (by examining the Receive Free Queue Host Write Pointer) to make sure that the Free Queue contains enough descriptors to fill the Free Queue FIFO. If the Free Queue does not have enough descriptors to fill the FIFO, then it will only read enough to keep from underflowing the Free Queue. If the FIFO detects that there are no Free Queue descriptors available for it to read, then it will set the either the Status Bit for Receive DMA Large Buffer Read Error (RLBRE) or the Status Bit for Receive DMA Small Buffer Read Error (RSBRE) in the Status Register for DMA (SDMA) and it will not read the Free Queue nor will it increment the Read Pointer. In such a scenario, the Receive FIFO may overflow if the Host does not provide Free Queue Descriptors. If the Free Queue FIFO can read descriptors from the Free Queue, then it will burst read them, increment the read pointer, and set either the Status Bit for Receive DMA Large Buffer Read (RLBR) or the Status Bit for Receive DMA Small Buffer Read (RSBR) in the Status Register for DMA (SDMA). See Section 4 for more details on Status Bits.

Register Name: **RDMAQ**
Register Description: **Receive DMA Queues Control**
Register Address: **0780h**

7	6	5	4	3	2	1	0
n/a	n/a	RDQF	RDQFE	RFQSF	RFQLF	RFQFC	RFQFE
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	RDQT2	RDQT1	RDQT0

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Receive Free Queue FIFO Enable (RFQFE). To enable the DMA to burst read descriptors from the Free Queue, this bit must be set to a one. If this bit is set to zero, descriptors will be read one at a time.

0 = Free Queue Burst Read Disabled
1 = Free Queue Burst Read Enabled

Bit 1 / Receive Free Queue FIFO Control (RFQFC). If the Free Queue is operated as a dual circular queue supporting both large and small buffers, then this bit should be set to zero. If the Free Queue is operated as a single circular queue only supporting large buffers, then this bit should be set to one.

0 = configure the FIFO as two 8 descriptor FIFOs (supports Large & Small Buffer reads)
1 = configure the FIFO as one 16 descriptor FIFO (supports Large Buffer reads only)

Bit 2 / Receive Free Queue Large Buffer FIFO Flush (RFQLF). When this bit is set to one, the internal Large Buffer Free Queue FIFO will be flushed (currently loaded Free Queue Descriptors are lost). This bit must be set to zero for proper operation.

0 = FIFO in normal operation
1 = FIFO is flushed

Bit 3 / Receive Free Queue Small Buffer FIFO Flush (RFQSF). When this bit is set to one, the internal Small Buffer Free Queue FIFO will be flushed (currently loaded Free Queue Descriptors are lost). This bit must be set to zero for proper operation.

0 = FIFO in normal operation
1 = FIFO is flushed

Bit 4 / Receive Done Queue FIFO Enable (RDQFE). See Section 8.1.4 for details.

Bit 5 / Receive Done Queue FIFO Flush (RDQF). See Section 8.1.4 for details.

Bits 8 to 10 / Receive Done Queue Status Bit Threshold Setting (RDQT0 to RDQT2). See Section 8.1.4 for details.

8.1.4 Done Queue

The DMA will write to the Receive Done Queue when it has filled a free data buffer with packet data and has loaded the associated Packet Descriptor with all the necessary information. The descriptor location is indicated via a 16-bit pointer which the Host will use along with the Receive Descriptor Base Address to find the exact 32-bit address of the associated Receive Descriptor.

RECEIVE DONE QUEUE DESCRIPTOR FIGURE 8.1.4A

dword 0

V	EOF	Status(3)	BUFCNT(3)	00b	HDLC CH#(6)	Descriptor Pointer (16)

(Note: the organization of the Done Queue is not affected by the enabling of Big Endian)

dword 0; Bits 0 to 15 / Descriptor Pointer. This 16-bit value is the offset from the Receive Descriptor Base Address of a Receive Packet Descriptor that has been readied by the DMA and is available for the host to begin processing.

dword 0; Bits 16 to 21 / HDLC Channel Number. HDLC channel number, which can be from 1 to 40.
000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

dword 0; Bits 22 to 23 / Unused. Set to zero by the DMA.

dword 0; Bits 24 to 26 / Buffer Count (BUFCNT). If a HDLC channel has been configured to only write to the Done Queue after a packet has been completely received (i.e. the Threshold field in the Receive DMA Configuration RAM is set to 000) then BUFCNT will always be set to 000. If the HDLC channel has been configured via the Threshold field to write to the Done Queue after a programmable number of buffers (from 1 to 7) have been filled, then BUFCNT corresponds to the number of buffers which have been written to Host memory. The BUFCNT will be less than the Threshold field value when the incoming packet does not require the number of buffers specified in the Threshold field.
000 = indicates that a complete packet has been received (only used when Threshold = 000)
001 = 1 buffer has been filled
010 = 2 buffers have been filled

111 = 7 buffers have been filled

dword 0; Bits 27 to 29 / Packet Status. These three bits report the final status of an incoming packet. They are only valid when the EOF bit is set to a one (EOF = 1).
000 = no error, valid packet received
001 = receive FIFO overflow (remainder of the packet discarded)
010 = CRC checksum error
011 = HDLC frame abort sequence detected (remainder of the packet discarded)
100 = non-aligned byte count error (not an integral number of bytes)
101 = long frame abort (max packet length exceeded; remainder of the packet discarded)
110 = PCI abort or parity data error (remainder of the packet discarded)
111 = reserved state (will never occur in normal device operation)

dword 0; Bit 30 / End Of Frame (EOF). This bit will be set to a one when this Receive Descriptor is the last one in the current descriptor chain. This indicates that a packet has been fully received or an error has been detected which has caused a premature termination.

dword 0; Bit 31 / Valid Done Queue Descriptor (V). This bit will be set to a zero by the Receive DMA. Instead of reading the Receive Done Queue Read Pointer to locate completed Done Queue Descriptors, the Host can use this bit (since the DMA will set the bit to a zero when it is written into the queue). If the latter scheme is used, the Host must set this bit to a one when the Done Queue Descriptor is read.

The Host will read from the Receive Done Queue to find which data buffers and their associated descriptors are ready for processing.

The Receive Done Queue is circular queue. To keep track of the addresses of the circular queue in the Receive Done Queue, there are a set of internal addresses within the device that accessed by both the Host and the DMA. On initialization, the Host will configure all of the registers shown in Table 8.1.4A. After initialization, the DMA will only write to (i.e. change) the write pointer and the Host will only write to the read pointer.

EMPTY CASE

The Receive Done Queue is considered empty when the read and write pointers are identical.

RECEIVE DONE QUEUE EMPTY STATE

	empty descriptor	
	empty descriptor	
	empty descriptor	
read pointer >	empty descriptor	< write pointer
	empty descriptor	
	empty descriptor	
	empty descriptor	

FULL CASE

The Receive Done Queue is considered full when the read pointer is ahead of the write pointer by one descriptor. Hence, one descriptor must always remain empty.

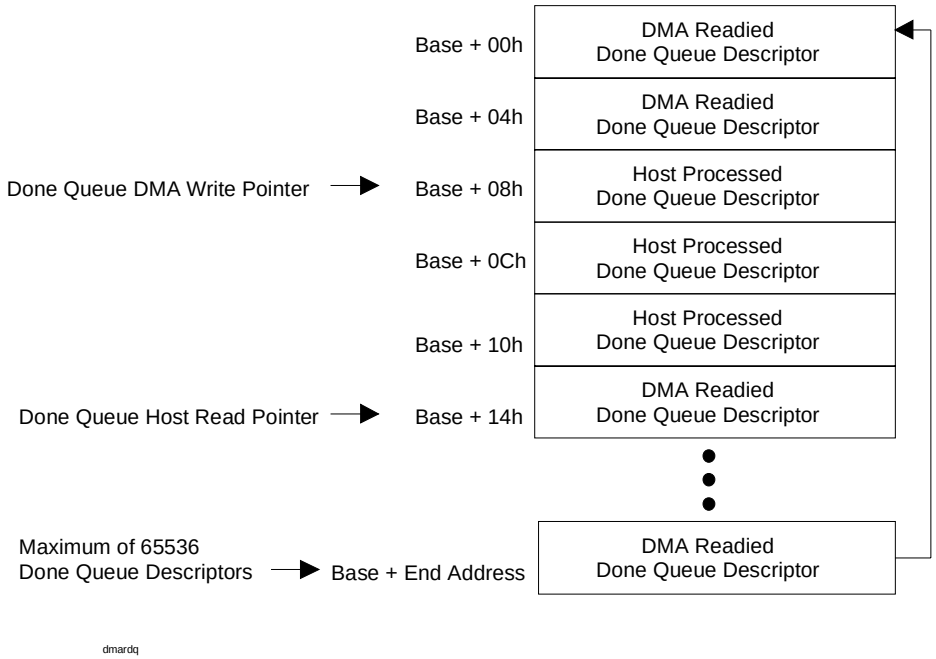
RECEIVE DONE QUEUE FULL STATE

	valid descriptor	
	valid descriptor	
	empty descriptor	< write pointer
read pointer >	valid descriptor	
	valid descriptor	
	valid descriptor	
	valid descriptor	

RECEIVE DONE QUEUE INTERNAL ADDRESS STORAGE Table 8.1.4A

Register Name	**Acronym**	**Address**
Receive Done Queue Base Address 0 (lower word)	RDQBA0	0730h
Receive Done Queue Base Address 1 (upper word)	RDQBA1	0734h
Receive Done Queue DMA Write Pointer	RDQWP	0740h
Receive Done Queue Host Read Pointer	RDQRP	073Ch
Receive Done Queue End Address	RDQEA	0738h
Receive Done Queue FIFO Flush Timer	RDQFFT	0744h

RECEIVE DONE QUEUE STRUCTURE Figure 8.1.4B



Once the Receive DMA is activated (via the RDE control bit in the Master Configuration register; see Section 4 for more details), it can begin writing data to the Done Queue. It knows where to write data into the Done Queue by reading the Write Pointer and adding it to the Base Address to obtain the actual 32-bit address. Once the DMA has written to the Done Queue, it increments the Write Pointer by one dword. A check must be made to make sure the incremented address does not exceed the Receive Done Queue End Address. If the incremented address does exceed this address, then the incremented write pointer will be set equal to 0000h (i.e. the Base Address).

STATUS BITS / INTERRUPTS

On writes to the Done Queue by the DMA, the DMA will set the Status Bit for Receive DMA Done Queue Write (RDQW) in the Status Register for DMA (SDMA). The Host can configure the DMA to either set this status bit on each write to the Done Queue or only after multiple (from 2 to 128) writes. The Host controls this by setting the RDQT0 to RDQT2 bits in the Receive DMA Queues Control (RDMAQ) register. See the description of the RDMAQ register at the end of Section 8.1.4 for more details. The DMA also checks the Receive Done Queue Host Read Pointer to make sure that an overflow does not occur. If this does occur, then the DMA will set the Status Bit for Receive DMA Done Queue Write Error (RDQWE) in the Status Register for DMA (SDMA) and it will not write to the Done Queue nor will it increment the Write Pointer. In such a scenario, packets may be lost and unrecoverable. Each of the status bits can also (if enabled) cause a hardware interrupt to occur. See Section 4 for more details.

BUFFER WRITE THRESHOLD SETTING

In the DMA Configuration RAM (see Section 8.1.5), there is a Host controlled field called Threshold (bits RDT0 to RDT2) that informs the DMA on when it should write to the Done Queue. The Host has the option to have the DMA place information in the Done Queue after a programmable number (from 1 to 7) data buffers have been filled or wait until the completed packet data has been written. The DMA will always write to the Done Queue when it has finished receiving a packet even if the threshold has not been met.

DONE QUEUE BURST WRITING

The DMA has the ability to write to the Done Queue in bursts. This allows for a more efficient use of the PCI Bus. The DMA can hand off descriptors to the Done Queue in-groups rather than one at a time, freeing up the PCI Bus for more time critical functions.

Internal to the device there is a FIFO that can store up to 8 Done Queue Descriptors (8 dwords since each descriptor occupies one dword). The Host must configure the FIFO for proper operation via the Receive DMA Queues Control (RDMAQ) register (see below).

When enabled via the Receive Done Queue FIFO Enable (RDQFE) bit, the Done Queue FIFO will not write to the Done Queue until it reaches the High Water Mark. When the Done Queue FIFO reaches the High Water Mark (which is six descriptors) it will attempt to empty the Done Queue FIFO by burst writing to the Done Queue. Before it writes to the Done Queue, it checks (by examining the Receive Done Queue Host Read Pointer) to make sure that the Done Queue has enough room to empty the Done Queue FIFO. If the Done Queue does not have enough room, then it will only burst write enough descriptors to keep from overflowing the Done Queue. If the FIFO detects that there is no room for any descriptors to be written, then it will set the Status Bit for Receive DMA Done Queue Write Error (RDQWE) in the Status Register for DMA (SDMA) and it will not write to the Done Queue nor will it increment the Write Pointer. In such a scenario, packets may be lost and unrecoverable. If the Done Queue FIFO can write descriptors to the Done Queue, then it will burst write them, increment the write pointer, and set the Status Bit for Receive DMA Done Queue Write (RDQW) in the Status Register for DMA (SDMA). See Section 4 for more details on Status bits.

DONE QUEUE FIFO FLUSH TIMER

To make sure that the Done Queue FIFO does get flushed to the Done Queue on a regular basis, the Receive Done Queue FIFO Flush Timer (RDQFFT) is used by the DMA to determine the maximum wait time in between writes. The RDQFFT is a 16-bit programmable counter that is decremented every PCLK divided by 256. It is only monitored by the DMA when the Receive Done Queue FIFO is enabled (RDQFE = 1). For a 33 MHz PCLK, the timer is decremented every 7.76 µs and for a 25 MHz clock it is decremented every 10.24 µs. Each time the DMA writes to the Done Queue it resets the timer to the count placed into it by the Host. On initialization, the Host will set a value into the RDQFFT that indicates the maximum time the DMA should wait in between writes to the Done Queue. For example, with a PCLK of 33 MHz, the range of wait times are from 7.8 µs (RDQFFT = 0001h) to 508 ms (RDQFFT = FFFFh) and PCLK of 25 MHz, the wait times range from 10.2 µs (RDQFFT = 0001h) to 671 ms (RDQFFT = FFFFh).

Register Name: **RDQFFT**
Register Description: **Receive Done Queue FIFO Flush Timer**
Register Address: **0744h**

7	6	5	4	3	2	1	0
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
15	14	13	12	11	10	9	8
TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 15 / Receive Done Queue FIFO Flush Timer Control Bits (TC0 to TC15). Please note that on system reset, the timer will be set to 0000h which is defined as an illegal setting. If the Receive Done Queue FIFO is to be activated (RDQFE = 1), then the Host must first configure the timer to a proper state and then set the RDQFE bit to one.

0000h = illegal setting
0001h = Timer Count Resets to 1
FFFFh = Timer Count Resets to 65536

Register Name: **RDMAQ**
Register Description: **Receive DMA Queues Control**
Register Address: **0780h**

7	6	5	4	3	2	1	0
n/a	n/a	RDQF	RDQFE	RFQSF	RFQLF	RFQFC	RFQFE
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	RDQT2	RDQT1	RDQT0

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Receive Free Queue FIFO Enable (RFQFE). See Section 8.1.3 for details.

Bit 1 / Receive Free Queue FIFO Control (RFQFC). See Section 8.1.3 for details.

Bit 2 / Receive Free Queue Large Buffer FIFO Flush (RFQLF). See Section 8.1.3 for details.

Bit 3 / Receive Free Queue Small Buffer FIFO Flush (RFQSF). See Section 8.1.3 for details.

Bit 4 / Receive Done Queue FIFO Enable (RDQFE). To enable the DMA to burst write descriptors to the Done Queue, this bit must be set to a one. If this bit is set to zero, messages will be written one at a time.

0 = Done Queue Burst Write Disabled
1 = Done Queue Burst Write Enabled

Bit 5 / Receive Done Queue FIFO Flush (RDQF). When this bit is set to one, the internal Done Queue FIFO will be flushed by sending all data into the Done Queue. This bit must be set to zero for proper operation.

0 = FIFO in normal operation
1 = FIFO is flushed

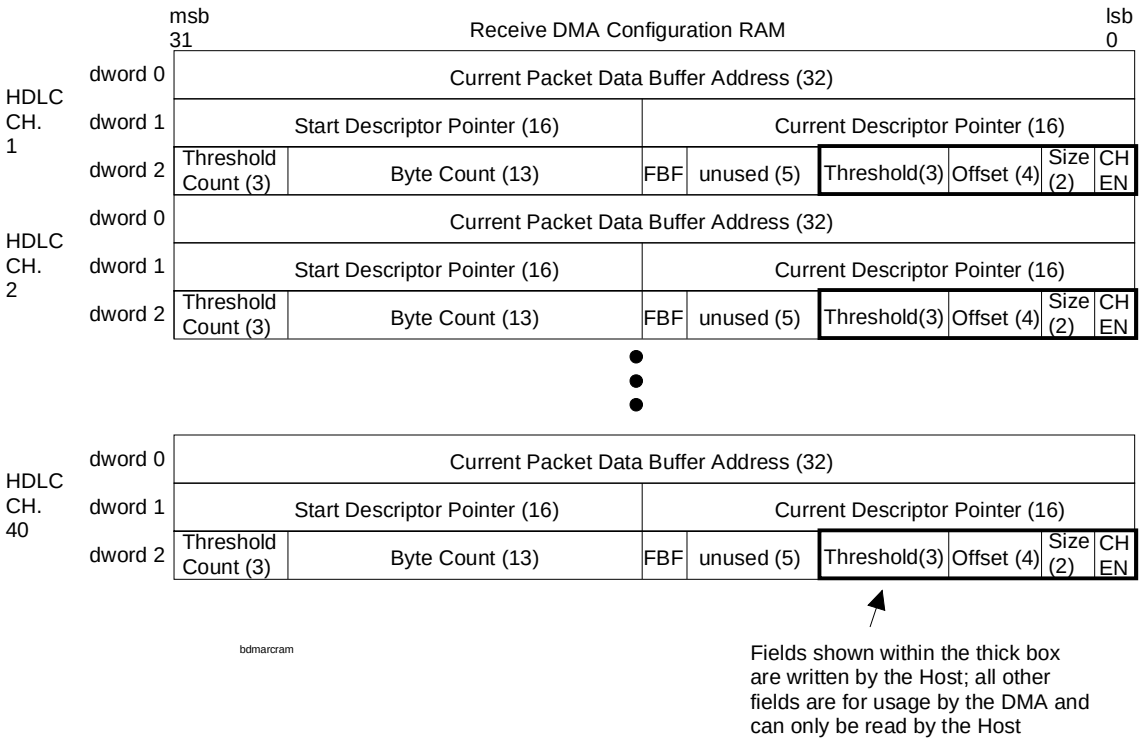
Bits 8 to 10 / Receive Done Queue Status Bit Threshold Setting (RDQT0 to RDQT2). These three bits determine when the DMA will set the Receive DMA Done Queue Write (RDQW) status bit in the Status Register for DMA (SDMA) register.

000 = set the RDQW status bit after each descriptor write to the Done Queue
001 = set the RDQW status bit after 2 or more descriptors are written to the Done Queue
010 = set the RDQW status bit after 4 or more descriptors are written to the Done Queue
011 = set the RDQW status bit after 8 or more descriptors are written to the Done Queue
100 = set the RDQW status bit after 16 or more descriptors are written to the Done Queue
101 = set the RDQW status bit after 32 or more descriptors are written to the Done Queue
110 = set the RDQW status bit after 64 or more descriptors are written to the Done Queue
111 = set the RDQW status bit after 128 or more descriptors are written to the Done Queue

8.1.5 DMA CONFIGURATION RAM

Onboard the device there is a set of 120 dwords (3 dwords per channel times 40 channels) that are used by the host to configure the DMA and by the DMA to store values locally when it is processing a packet. Most of the fields within the DMA Configuration RAM are for use by the DMA and the Host will never write to these fields. The Host is only allowed to write (i.e. configure) to the lower word of dword 2 for each HDLC channel. The Host configurable fields are denoted with a thick box as shown below.

RECEIVE DMA CONFIGURATION RAM Figure 8.1.5A



- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 0; Bits 0 to 31 / Current Data Buffer Address. The current 32-bit address of the data buffer that is being used. This address is used by the DMA to keep track of where data should be written to as it comes in from the Receive FIFO.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 1; Bits 0 to 15 / Current Descriptor Pointer. This 16-bit value is the offset from the Receive Descriptor Base Address of the current Receive Descriptor being used by the DMA to describe the specifics of the data being stored in the associated data buffer.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 1; Bits 16 to 31 / Starting Descriptor Pointer. This 16-bit value is the offset from the Receive Descriptor Base Address of the first Receive Descriptor in a link-list chain of descriptors. This pointer will be written into the Done Queue by the DMA after a specified number of data buffers (see the Threshold value below) have been filled.

- HOST MUST CONFIGURE -
dword 2; Bit 0 / Channel Enable (CHEN). This bit is controlled by the host to enable and disable a HDLC channel.
0 = HDLC Channel Disabled
1 = HDLC Channel Enabled

- HOST MUST CONFIGURE -
dword 2; Bits 1 & 2 / Buffer Size Select. These bits are controlled by the host to select the manner in which the Receive DMA will store incoming packet data.
00 = use large size data buffers only
01 = use small size data buffers only
10 = fill a small buffer first followed then by large buffers as needed
11 = illegal state and should not be selected

- HOST MUST CONFIGURE -
dword 2; Bits 3 to 6 / Buffer Offset. These four bits are controlled by the host to determine if the packet data written into the first data buffer should be offset by up to 15 bytes. This allows the host complete control over the manner in which data will be written into main memory.
0000 (0h) = 0 byte offset from the data buffer address of the first data buffer
0001 (1h) = 1 byte offset from the data buffer address of the first data buffer
1111 (Fh) = 15 byte offset from the data buffer address of the first data buffer

- HOST MUST CONFIGURE -
dword 2; Bits 7 to 9 / Threshold. These three bits are controlled by the host to determine when the DMA should write into the Done Queue that data is available for processing.
000 = DMA should write to the Done Queue only after packet reception is complete
001 = DMA should write to the Done Queue after 1 data buffer has been filled
010 = DMA should write to the Done Queue after 2 data buffers have been filled
011 = DMA should write to the Done Queue after 3 data buffers have been filled
100 = DMA should write to the Done Queue after 4 data buffers have been filled
101 = DMA should write to the Done Queue after 5 data buffers have been filled
110 = DMA should write to the Done Queue after 6 data buffers have been filled
111 = DMA should write to the Done Queue after 7 data buffers have been filled

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 2; Bits 10 to 14 / DMA Reserved. Could be any value when read. Should be set to zero when written to by the Host.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 2; Bit 15 / First Buffer Fill (FBF). This bit will be set to a one by the Receive DMA when it is in the process of filling the first buffer of a packet. This bit is used by the DMA to know when to switch to Large Buffers when the Buffer Size Select field is set to 10.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 2; Bits 16 to 28 / Byte Count. The DMA uses these 13 bits to keep track of the number of bytes stored in the data buffer. Maximum is 8191 bytes (0000h =0 bytes / 1FFFh = 8191 bytes).

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 2; Bits 29 to 31 / Threshold Count. These three bits keep track of the number of data buffers that have been filled so that the Receive DMA knows when to write to the Done Queue based on the Host controlled field called Threshold.

000 = threshold count is 0 data buffers
001 = threshold count is 1 data buffer
010 = threshold count is 2 data buffers
011 = threshold count is 3 data buffers
100 = threshold count is 4 data buffers
101 = threshold count is 5 data buffers
110 = threshold count is 6 data buffers
111 = threshold count is 7 data buffers

Register Name: **RDMACIS**
Register Description: **Receive DMA Channel Configuration Indirect Select**
Register Address: **0770h**

7	6	5	4	3	2	1	0
n/a	n/a	HCID5	HCID4	HCID3	HCID2	HCID1	HCID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	n/a	RDCW2	RDCW1	RDCW0

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 5 / HDLC Channel ID (HCID0 to HCID5).
000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

Bits 8 to 10 / Receive DMA Configuration RAM Word Select Bits 0 to 2 (RDCW0 to RDCW2).
000 = lower word of dword 0
001 = upper word of dword 0
010 = lower word of dword 1
011 = upper word of dword 1
100 = lower word of dword 2 (only word that the Host can write to)
101 = upper word of dword 2
110 = illegal state
111 = illegal state

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the internal Receive DMA Configuration RAM, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the channel location indicated by the HCID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the RDMAC register, the IAB bit will be set to zero. When the host wishes to write data to the internal Receive DMA Configuration RAM, this bit should be written to a zero by the host. This causes the device to take the data that is current present in the RDMAC register and write it to the channel location indicated by the HCID bits. When the device has completed the write, the IAB bit will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **RDMAC**
Register Description: **Receive DMA Channel Configuration**
Register Address: **0774h**

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8
D15	D14	D13	D12	D11	D10	D9	D8

Note: bits that are underlined are read only, all other bits are read-write.

Bits 0 to 15 / Receive DMA Configuration RAM Data (D0 to D15). Data that is written to or read from the Receive DMA Configuration RAM.

8.2 TRANSMIT SIDE

8.2.1 OVERVIEW

The Transmit DMA uses a scatter gather technique to read packet data from main memory. The Host will keep track of and decide where (and when) the DMA should grab the outgoing packet data from. There are a set of descriptors that get handed back and forth between the Host and the DMA. Via these descriptors the Host can inform the DMA where to obtain the packet data from and the DMA can tell the Host when the data has been transmitted.

The operation of the Transmit DMA has three main areas as shown in Figures 8.2.1A and 8.2.1B and Table 8.2.1A. The Host will write to the Pending Queue informing the DMA which channels have packet data that is ready to be transmitted. Associated with each Pending Queue Descriptor is a data buffer that contains the actual data payload of the HDLC packet. The data buffers can be between 1 and 8191 bytes in length (inclusive). If an outgoing packet requires more than memory than a data buffer contains, then the Host can link the data buffers to handle packets of any size.

The Done Queue Descriptors contain information that the DMA wishes to pass to the Host. The DMA will write to the Done Queue when it has completed transmitting either a complete packet or data buffer (see the discussion on DMA Update to the Done Queue below). Via the Done Queue Descriptors the DMA informs the Host about the status of the outgoing packet data. If an error occurs in the transmission, the Done Queue can be used by the Host to recover the packet data that did not get transmitted and the Host can then re-queue the packets for transmission.

If enabled, the DMA can burst read the Pending Queue Descriptors and burst write the Done Queue Descriptors. This helps minimize PCI Bus accesses, freeing the PCI Bus up to do more time critical functions. See Sections 8.2.3 and 8.2.4 for more details on this feature.

TRANSMIT DMA MAIN OPERATIONAL AREAS Table 8.2.1A

Name	Section	Description
Packet Descriptors	8.2.2	A dedicated area of memory that describes the location and attributes of the packet data.
Pending Queue Descriptors	8.2.3	A dedicated area of memory that the Host will write to inform the DMA that packet data is queued and ready for transmission
Done Queue Descriptors	8.2.4	A dedicated area of memory that the DMA will write to inform the Host that the packet data has been transmitted

HOST LINKING OF DATA BUFFERS

As mentioned earlier, the data buffers are limited to a length of 8191 bytes. If an outgoing packet requires more memory space than the available data buffer contains, then the Host can link multiple data buffers together to handle a packet length of any size. The Host does this via the End of Frame (EOF) bit in the Packet Descriptor. Each data buffer has a one-to-one association with a Packet Descriptor. If the Host wishes to link multiple data buffers together, then the EOF bit will be set to zero in all but the last data buffer. Figure 8.2.1A contains an example for HDLC channel number 5 where the Host has linked three data buffers together. The transmit DMA knows where to find the next data buffer when the EOF bit is set to zero via the Next Descriptor Pointer field.

HOST LINKING OF PACKETS (PACKET CHAINING)

The Host also has the option to link multiple packets together in a chain. Via the Chain Valid (CV) bit in the Packet Descriptor, the Host can inform the transmit DMA that the Next Descriptor Pointer field contains the descriptor of another HDLC packet that is ready for transmission. The transmit DMA ignores the CV bit until it sees EOF = 1 which indicates the end of a packet. If CV = 1 when EOF = 1, then this indicates to the transmit DMA that it should use the Next Descriptor Pointer field to find the next packet in the chain. Figure 8.2.1C provides an example of packet chaining. Each column in Figure 8.2.1C represents a separate packet chain. In column 1, three data buffers have been linked together by the Host for Packet #1 and then the Host has created a packet chain by setting CV = 1 in the last descriptor of Packet #1.

DMA LINKING OF PACKETS (HORIZONTAL LINK LISTING)

The transmit DMA also has the ability to link packets together. Internally, the transmit DMA can store up to two packets chains but if the Host places more packet chains into the Pending Queue, then the transmit DMA must begin linking these chains together externally. The transmit DMA does this by writing to Packet Descriptors. As an example, see Figure 8.2.1C. If columns 1 and 2 were the only two packet chains queued for transmission, then the transmit DMA would not need to begin linking packet chains together but as soon as column 3 was queued for transmission, the transmit DMA had to store the third chain externally because it had no more room internally. The transmit DMA links the packet chain in the third column to the one in the second column by writing the 1st descriptor of the third chain in the Next Pending Descriptor Pointer field of the 1st descriptor of the second column (it also sets the PV bit to one). As shown in Figure 8.2.1C, this chaining was carried one step farther to link the forth column to the third.

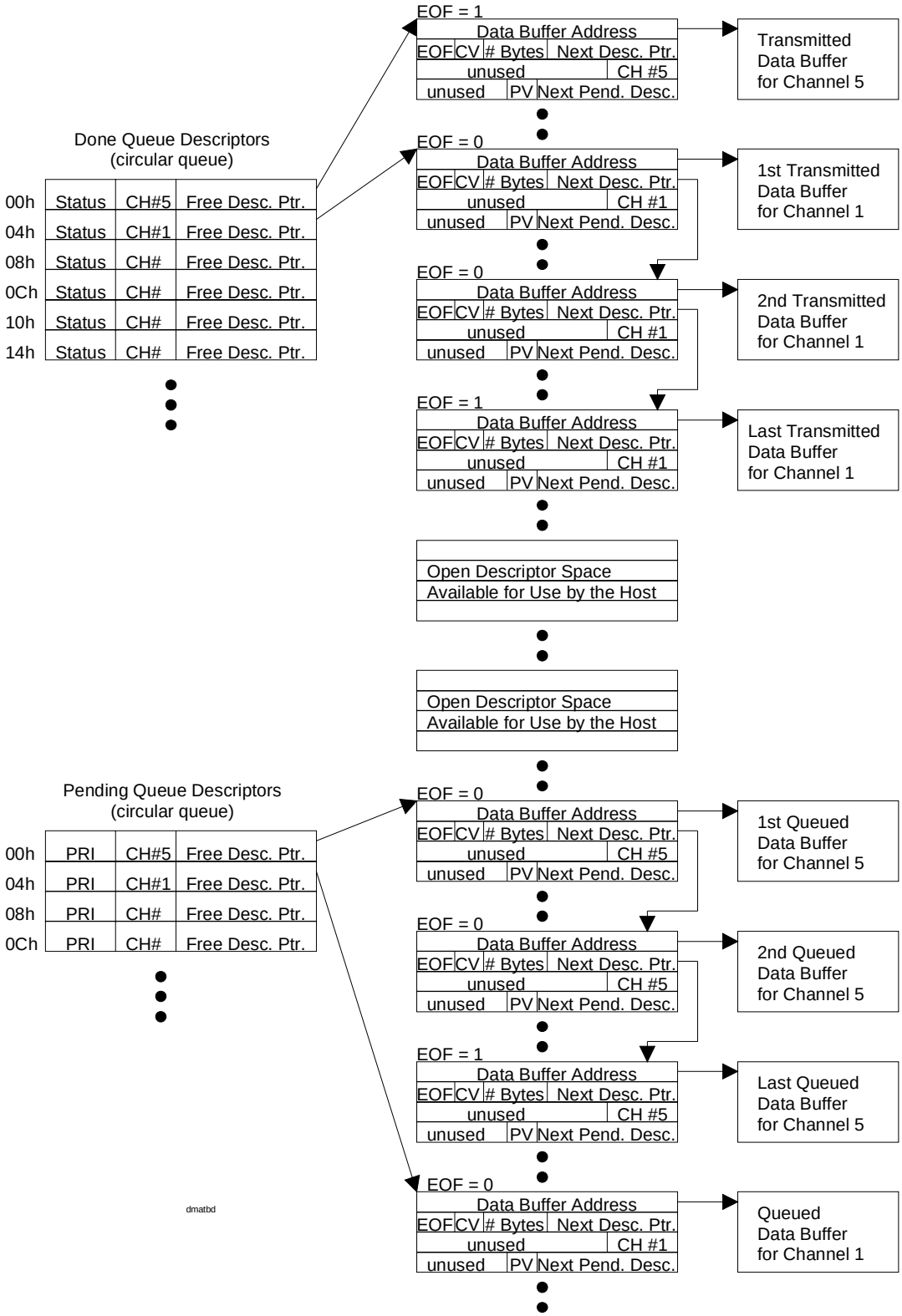
PRIORITY PACKETS

The Host has the option to change the order in which packets are transmitted by the DMA. If the Host sets the Priority Packet (PRI) bit in the Pending Queue Descriptor to a one, then the transmit DMA knows that this packet is a priority packet and should be transmitted ahead of all standard packets. The rules for packet transmission are:

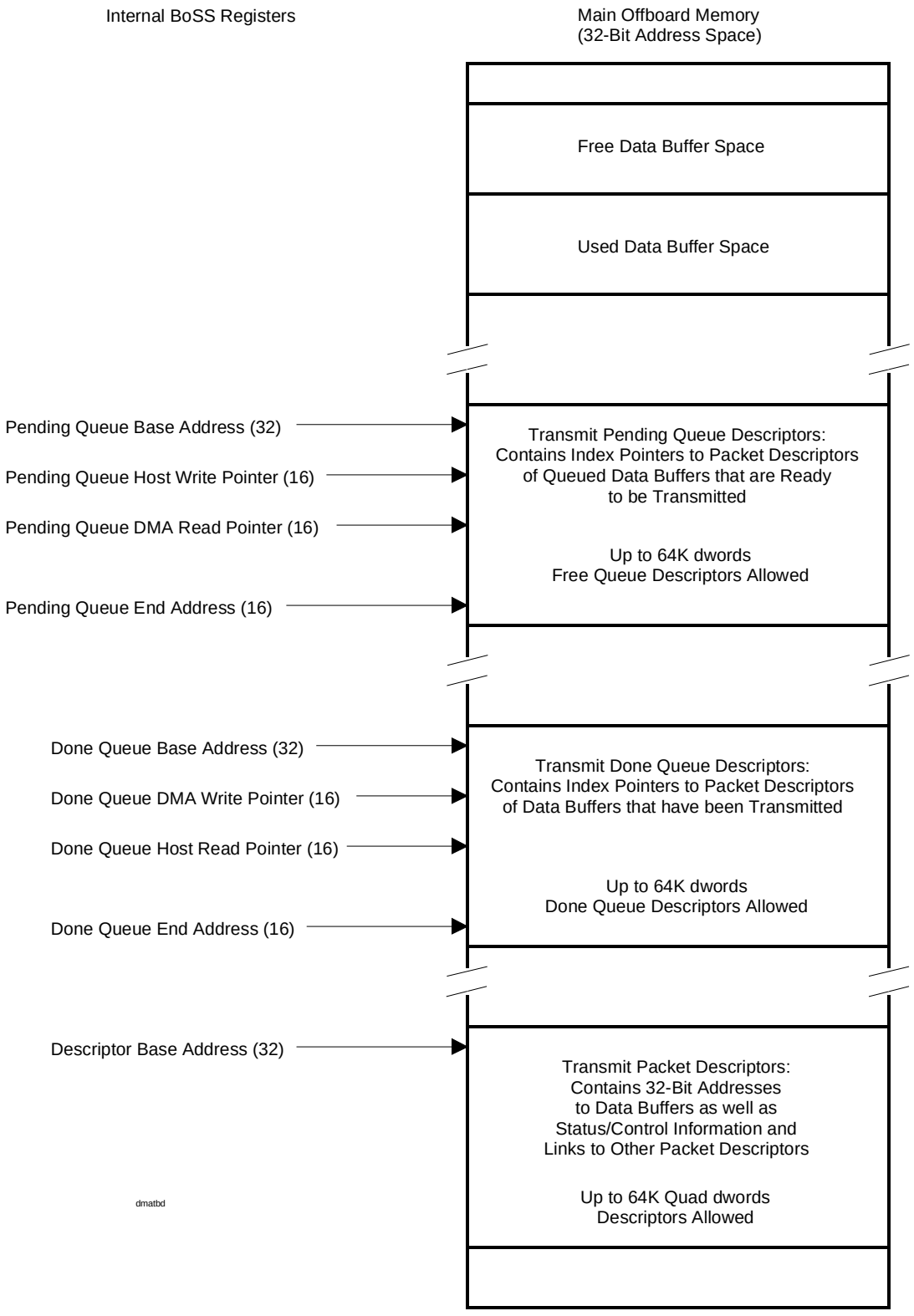
1. Priority packets will be transmitted as soon as the current standard packet (not packet chain) finishes transmission.
2. All priority packets will be transmitted before any more standard packets are transmitted.
3. Priority packets are ordered on a first come, first served basis.

Figure 8.2.1D provides an example of a set of priority packets interrupting a set of standard packets. In the example, the first priority packet chain (shown in column 2) was read by the transmit DMA from the Pending Queue while it was transmitting standard packet #1. It waited until standard packet #1 was complete and then begin sending the priority packets. While column 2 was being sent, the priority packet chains of columns 3 and 4 arrived in the Pending Queue so the transmit DMA linked column four to column three and then waited until all of the priority packets were transmitted before returning to the standard packet chain in column 1. Note that the packet chain in column 1 was interrupted to transmit the priority packets. In other words, the transmit DMA did not wait for the complete packet to finish transmitting, only the current packet.

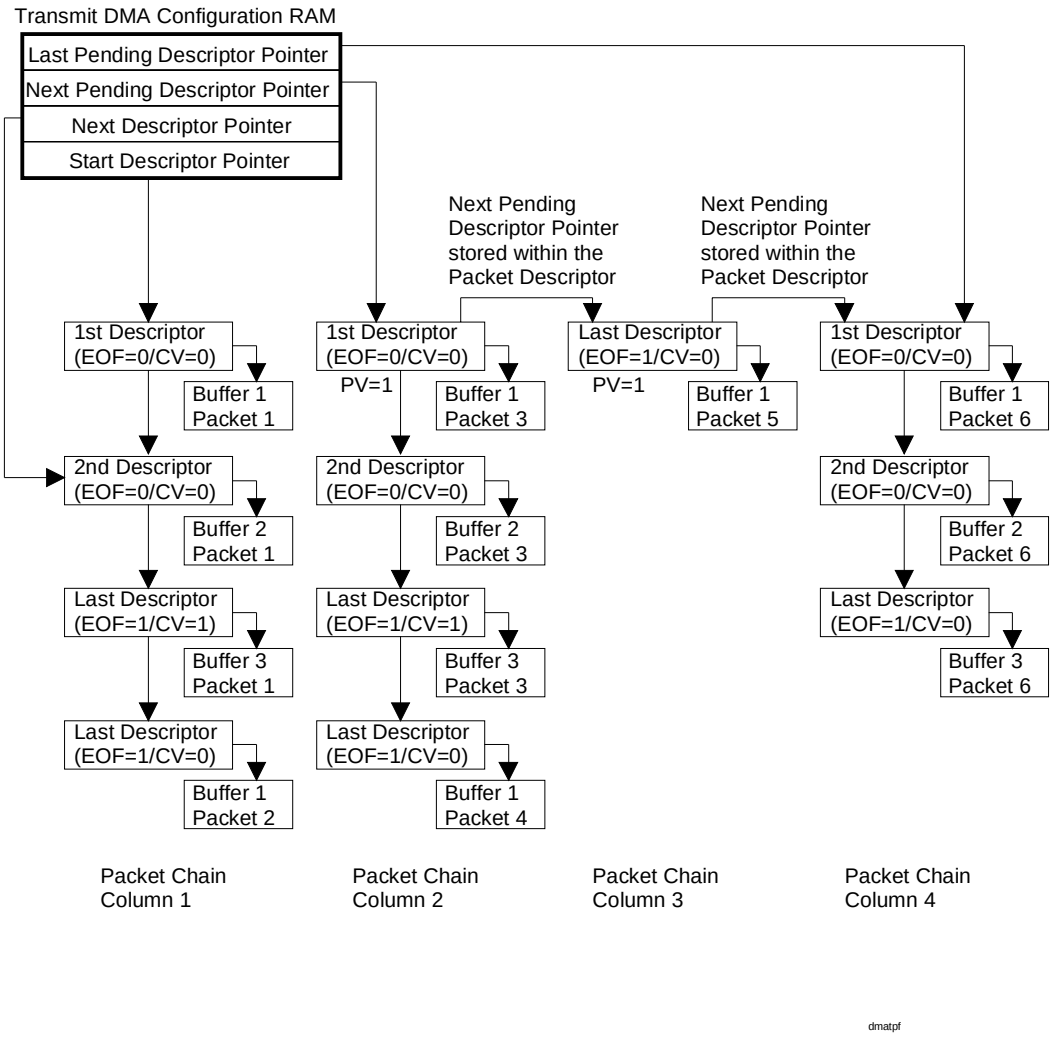
TRANSMIT DMA OPERATION Figure 8.2.1A



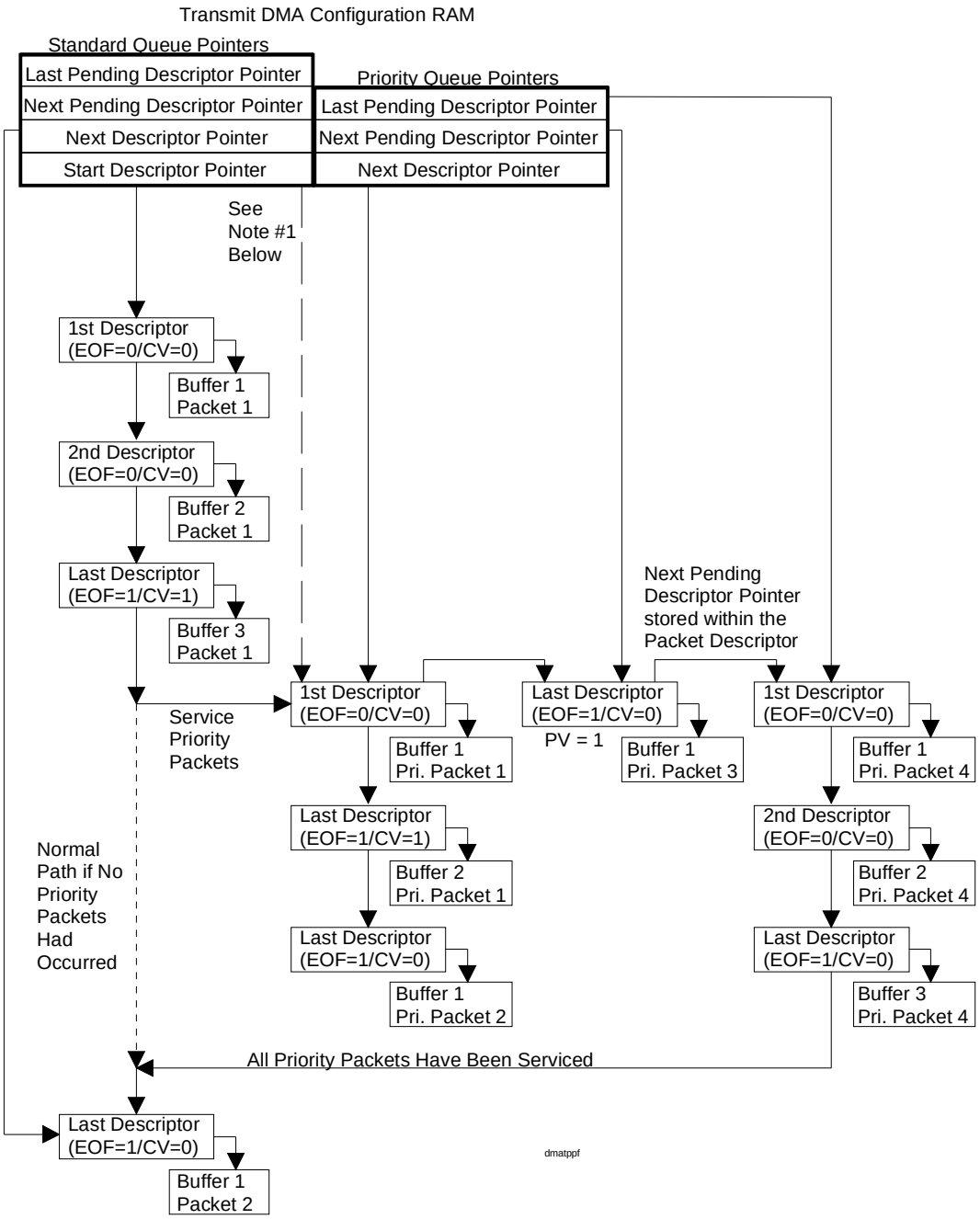
TRANSMIT DMA MEMORY ORGANIZATION Figure 8.2.1B



TRANSMIT DMA PACKET HANDLING Figure 8.2.1C



TRANSMIT DMA PRIORITY PACKET HANDLING Figure 8.2.1D



Note #1
The Start Descriptor Pointer field in the Transmit DMA Configuration RAM is used by both the nomal and priority pending queues.

DMA UPDATES TO THE DONE QUEUE

The Host has two options as to when the transmit DMA should write descriptors that have completed transmission to the Done Queue. On a channel-by-channel basis, via the Done Queue Select (DQS) bit in the Transmit DMA Configuration RAM, the Host can condition the DMA to:

1. write to the Done Queue only when the complete HDLC packet has been transmitted (DQS = 0)
2. write to the Done Queue when each data buffer has been transmitted (DQS = 1)

The Status field in the Done Queue Descriptor will be configured based on the setting of the DQS bit. If DQS = 0, then when a packet has successfully completed transmission the Status field will be set to 000. If DQS = 1, then when the first data buffer has successfully completed transmission the Status field will be set to 001. When each middle buffer (i.e. the second through the next to last) has successfully completed transmission the Status field will be set to 010. When the last data buffer of a packet has successfully completed transmission the Status field will be set to 011.

ERROR CONDITIONS

While processing packets for transmission, the DMA can encounter a number of error conditions, which include;

- PCI error (an abort error)
- transmit FIFO underflow
- channel is disabled (CHEN = 0) in the Transmit DMA Configuration RAM
- channel number discrepancy between the Pending Queue & the Packet Descriptor
- byte count of 0 bytes in the Packet Descriptor.

If any of these errors occur, the transmit DMA will automatically disable the affected channel by setting the Channel Enable (CHEN) bit in the Transmit DMA Configuration RAM to zero and then it will write the current descriptor into the Done Queue with the appropriate error status as shown in Table 8.2.1B below.

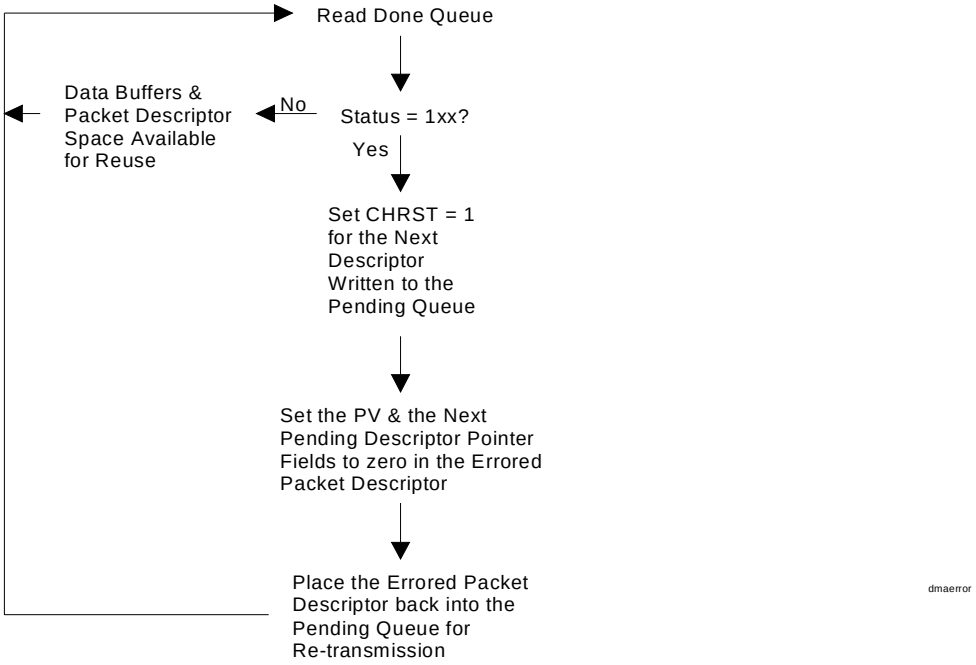
DONE QUEUE ERROR STATUS CONDITIONS Table 8.2.1B

Packet Status	Description of the Error
100	software provisioning error; this channel was not enabled
101	descriptor error; either byte count = 0 or channel code inconsistent with Pending Queue
110	PCI error; either parity or abort
111	transmit FIFO error; it has underflowed

Since the transmit DMA has disabled the channel, any remaining queued descriptors will not be transmitted and will be written to the Done Queue with a Packet Status of 100 (i.e. reporting that the channel was not enabled). At this point the Host has two options. Option 1, it can wait until all of the remaining queued descriptors are written to the Done Queue with an errored status and then manually re-enable the channel by setting the CHEN bit to one and then re-queue all of the affected packets. Option 2, as soon as it detects an errored status, it can force the channel active again by setting the Channel Reset (CHRST) bit to a one for the next descriptor that it writes to the Pending Queue for the affected channel. As soon as the transmit DMA detects that the CHRST is set to a one, it will re-enable the channel by forcing the CHEN bit to a one. The DMA will not re-enable the channel until it has finished writing all of the previously queued descriptors to the Done Queue. Then the Host can collect the errored descriptors as they arrive in the Done Queue and then re-queue them for transmission by writing descriptors to the Pending Queue so the transmit DMA knows where to find the packets that did not get transmitted (software housekeeping note: the Host must set the Next Pending Descriptor Pointer and PV

fields in the Packet Descriptor to zero to ready them for transmission). The second option allows the software a cleaner error recovery technique. See Figure 8.2.1E for more details.

TRANSMIT DMA ERROR RECOVERY ALGORITHM Figure 8.2.1E



HOST ACTIONS

The Host will typically handle the Transmit DMA as follows:

1. The Host will place readied packets into the Pending Queue.
2. The Host will either poll or be interrupted that some outgoing packets have completed transmission and that it should read the Done Queue.
3. If Done Queue reports that an error was incurred and that a packet was not transmitted, then the Host must re-queue the packet for transmission.

TRANSMIT DMA ACTIONS

A typical scenario for the Transmit DMA is as follows:

1. The transmit DMA constantly reads the Pending Queue looking for packets that are queued for transmission.
2. The transmit DMA will update the Done Queue as packets or data buffers complete transmission.
3. If an error occurs, then the transmit DMA will disable the channel and wait for the Host to request that the channel be enabled.

8.2.2 PACKET DESCRIPTORS

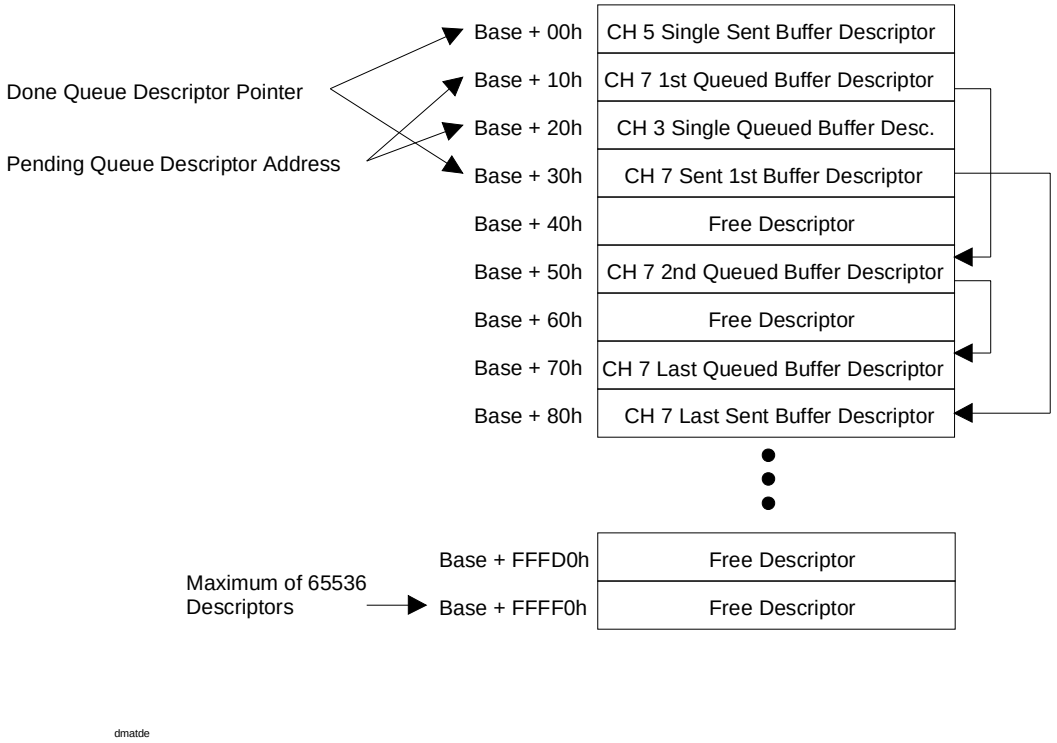
In main memory resides a contiguous section up to 65,536 quad dwords that make up the Transmit Packet Descriptors. The Transmit Packet Descriptors are aligned on a quad dword basis and can be placed anywhere in the 32-bit address space via the Transmit Descriptor Base Address (see Table 8.2.2A). Associated with each descriptor is a data buffer. The data buffer can be up to 8191 bytes long and must be a contiguous section of main memory. The host will inform the DMA of the actual size of the data buffer via the Byte Count field that resides in the Packet Descriptor. If an outgoing packet requires more space than the data buffer allows, then Packet Descriptors will be link-listed together by the Host to provide a chain of data buffers. Figure 8.2.2A is an example of how three descriptors were linked together for an incoming packet on HDLC Channel 7. Channel 3 only required a single data buffer and hence only one Packet Descriptor was used. Figure 8.2.1A shows a similar example for channels5 and 1.

Packet Descriptors can be either pending (i.e. queued up by the host and ready for transmission by the DMA) or completed (i.e. have been transmitted by the DMA and are available for processing by the host). Pending Packet Descriptors are pointed to by the Pending Queue Descriptors and completed Packet Descriptors are pointed to by the Done Queue Descriptors.

TRANSMIT DESCRIPTOR ADDRESS STORAGE Table 8.2.2A

Register Name	Acronym	Address
Transmit Descriptor Base Address 0 (lower word)	TDBA0	0850h
Transmit Descriptor Base Address 1 (upper word)	TDBA1	0854h

TRANSMIT DESCRIPTOR EXAMPLE Figure 8.2.2A



dmatde

TRANSMIT PACKET DESCRIPTORS Figure 8.2.2B

dword 0

Data Buffer Address (32)

dword 1

EOF	CV	unused	Byte Count (13)	Next Descriptor Pointer (16)

dword 2

unused (26)	HDLC CH# (6)

dword 3

unused (15)	PV	Next Pending Descriptor Pointer (16)

(Note 1: the organization of the Transmit Descriptor is not affected by the enabling of Big Endian)
(Note 2: the format of the Transmit Descriptor is almost identical to the Receive Descriptor; this lessens the burden of the Host in preparing descriptors in store-and-forward applications)

dword 0; Bits 0 to 31 / Data Buffer Address. Direct 32-bit starting address of the data buffer that is associated with this transmit descriptor.

dword 1; Bits 0 to 15 / Next Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address of the next descriptor in the chain. Only valid if EOF = 0 (next descriptor in the same packet chain) or if EOF = 1 and CV = 1 (first descriptor in the next packet).

dword 1; Bits 16 to 28 / Byte Count. Number of bytes stored in the data buffer. Maximum is 8191 bytes (0000h = 0 bytes / 1FFFh = 8191 bytes).

dword 1; Bit 29 / Unused. This bit is ignored by the transmit DMA and can be set to any value.

dword 1; Bit 30 / Chain Valid (CV). If CV is set to a one when EOF = 1, then this indicates that the Next Descriptor Pointer field is valid and corresponds to the first descriptor of the next packet that is queued up for transmission. The CV bit is ignored when EOF = 0.

dword 1; Bit 31 / End Of Frame (EOF). When set to a one, this bit indicates that the descriptor is the last buffer in the current packet. When set to a zero, this bit indicates that Next Descriptor Pointer field is valid and points to the next descriptor in the packet chain.

dword 2; Bits 0 to 5 / HDLC Channel Number. HDLC channel number, which can be from 1 to 40.
000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

dword 2; Bits 6 to 31 / Unused. These bits are ignored by the transmit DMA and can be set to any value.

dword 3; Bits 0 to 15 / Next Pending Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address to another the descriptor chain that is queued up for transmission. The transmit DMA can store up to 2 queued packet chains internally but additional packet chains must be stored as a link list by the transmit DMA using this field. This field is only valid if PV = 1 and it should be set to 0000h by the Host when the Host is preparing the descriptor.

dword 3; Bit 16 / Pending Descriptor Valid (PV). If set, this bit indicates that the Next Pending Descriptor Pointer field is valid and corresponds to the first descriptor of the next packet chain that is queued up for transmission. This field is written to by the transmit DMA to link descriptors together and should always be set to 0 by the Host.

dword 3; Bits 17 to 31 / Unused. These bits are ignored by the transmit DMA and can be set to any value.

8.2.3 PENDING QUEUE

The Host will write to the Transmit Pending Queue, the location of the readied descriptor, channel number and control information. The descriptor space is indicated via a 16-bit pointer which the DMA will use along with the Transmit Packet Descriptor Base Address to find the exact 32-bit address of the associated Transmit Packet Descriptor.

TRANSMIT PENDING QUEUE DESCRIPTOR Figure 8.2.3A

dword 0

unused	Status(3)	CH RST	PRI	unused(2)	HDLC CH#(6)	Descriptor Pointer (16)

(Note: the organization of the Pending Queue is not affected by the enabling of Big Endian)

dword 0; Bits 0 to 15 / Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address to the first descriptor in a packet chain (can be a single descriptor) that is queued up for transmission.

dword 0; Bits 16 to 21 / HDLC Channel Number. HDLC channel number, which can be from 1 to 40.
000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

dword 0; Bits 22 to 23 / Unused. Not used by the DMA. Can be set to any value by the Host and will be ignored by the transmit DMA.

dword 0; Bit 24 / Priority Packet (PRI). If this bit is set to a one, then this indicates to the transmit DMA that the packet or packet chain pointed to by the Descriptor Pointer field should be transmitted immediately after the current packet transmission (whether it be standard or priority) is complete.

dword 0; Bit 25 / Channel Reset (CHRST). Under normal operating conditions, this bit should always be set to zero. When an error condition occurs and the transmit DMA places the channel into an out-of-service state by setting the Channel Enable (CHEN) bit in the Transmit DMA Configuration Register to zero, the Host can force the channel active again by setting the CHRST bit to a one. Only the first descriptor loaded into the Pending Queue after an error condition should have CHRST set to a one, all subsequent descriptors (until another error condition occurs) should have CHRST set to zero. The transmit DMA examines this bit and will force channel active (CHEN = 1) if CHRST is set to one. If CHRST is set to zero, then the transmit DMA will not modify the state of the CHEN bit. See Section 8.2.1 for more details on how error conditions are handled.

dword 0; Bits 26 to 28 / Packet Status. Not used by the DMA. Can be set to any value by the Host and will be ignored by the transmit DMA. This field will be used when the transmit DMA when it writes to the Done Queue to inform the Host of the status of the outgoing packet data.

dword 0; Bits 29 to 31 / Unused. Not used by the DMA. Can be set to any value by the Host and will be ignored by the transmit DMA.

The Transmit DMA will read from the Transmit Pending Queue Descriptor circular queue which data buffers and their associated descriptors are ready for transmission. To keep track of the addresses of the circular queue in the Transmit Pending Queue, there are a set of internal addresses within the device that

are accessed by both the Host and the DMA. On initialization, the Host will configure all of the registers shown in Table 8.2.3A. After initialization, the DMA will only write to (i.e. change) the read pointers and the Host will only write to the write pointers.

EMPTY CASE

The Transmit Pending Queue is considered empty when the read and write pointers are identical.

TRANSMIT PENDING QUEUE EMPTY STATE

	empty descriptor	
	empty descriptor	
	empty descriptor	
read pointer >	empty descriptor	< write pointer
	empty descriptor	
	empty descriptor	
	empty descriptor	

FULL CASE

The Transmit Pending Queue is considered full when the read pointer is ahead of the write pointer by one descriptor. Hence, one descriptor must always remain empty.

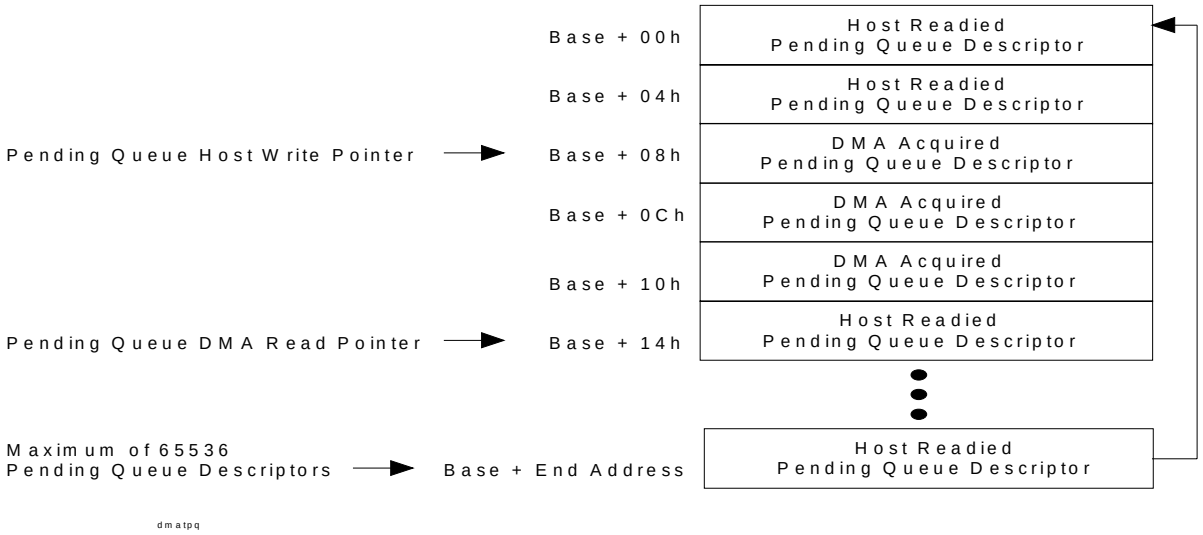
TRANSMIT PENDING QUEUE FULL STATE

	valid descriptor	
	valid descriptor	
	empty descriptor	< write pointer
read pointer >	valid descriptor	
	valid descriptor	
	valid descriptor	
	valid descriptor	

TRANSMIT PENDING QUEUE INTERNAL ADDRESS STORAGE Table 8.2.3A

Register Name	**Acronym**	**Address**
Transmit Pending Queue Base Address 0 (lower word)	TPQBA0	0800h
Transmit Pending Queue Base Address 1 (upper word)	TPQBA1	0804h
Transmit Pending Queue Host Write Pointer	TPQWP	080Ch
Transmit Pending Queue DMA Read Pointer	TPQRP	0810h
Transmit Pending Queue End Address	TPQEA	0808h

TRANSMIT PENDING QUEUE STRUCTURE Figure 8.2.3B



Once the Transmit DMA is activated (by setting the TDE control bit in the Master Configuration register; see Section 4), it can begin reading data out of the pending queue. It knows where to read data out of the pending queue by reading the Read Pointer and adding it to the Base Address to obtain the actual 32-bit address. Once the DMA has read the Pending Queue, it increments the Read Pointer by one dword. A check must be made to make sure the incremented address does not exceed the Transmit Pending Queue End Address. If the incremented address does exceed this address, then the incremented read pointer will be set equal to 0000h.

STATUS / INTERRUPTS

On each read of the Pending Queue by the DMA, the DMA will set the Status Bit for Transmit DMA Pending Queue Read (TPQR) in the Status Register for DMA (SDMA). The status bits can also (if enabled) cause an hardware interrupt to occur. See Section 4 for more details.

PENDING QUEUE BURST READING

The DMA has the ability to read the Pending Queue in bursts. This allows for a more efficient use of the PCI Bus. The DMA can grab descriptors from the Pending Queue in-groups rather than one at a time, freeing up the PCI Bus for more time critical functions.

Internal to the device there is a FIFO that can store up to 16 Pending Queue Descriptors (16 dwords since each descriptor occupies one dword). The Host must configure the Pending Queue FIFO for proper operation via the Transmit DMA Queues Control (TDMAQ) register (see below).

When enabled via the Transmit Pending Queue FIFO Enable (TPQFE) bit, the Pending Queue FIFO will not read the Pending Queue until it reaches the Low Water Mark. When the Pending Queue FIFO reaches the Low Water Mark (which is four descriptors) it will attempt to fill the FIFO with additional descriptors by burst reading the Pending Queue. Before it reads the Pending Queue, it checks (by examining the Transmit Pending Queue Host Write Pointer) to make sure that the Pending Queue contains enough descriptors to fill the Pending Queue FIFO. If the Pending Queue does not have enough descriptors to fill the FIFO, then it will only read enough to empty the Pending Queue. If the FIFO detects that there are no Pending Queue descriptors available for it to read, then it will wait and try again later. If the Pending Queue FIFO can read descriptors from the Pending Queue, then it will burst read them, increment the read pointer, and set the Status Bit for Transmit DMA Pending Queue Read (TPQR) in the Status Register for DMA (SDMA). See Section 4 for more details on Status Bits.

Register Name: **TDMAQ**
Register Description: **Transmit DMA Queues Control**
Register Address: **0880h**

7	6	5	4	3	2	1	0
n/a	n/a	n/a	n/a	TDQF	TDQFE	TPQF	TPQFE
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	TDQT2	TDQT1	TDQT0

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Transmit Pending Queue FIFO Enable (TPQFE). To enable the DMA to burst read descriptors from the Pending Queue, this bit must be set to a one. If this bit is set to zero, descriptors will be read one at a time.

0 = Pending Queue Burst Read Disabled
1 = Pending Queue Burst Read Enabled

Bit 1 / Transmit Pending Queue FIFO Flush (TPQF). When this bit is set to one, the internal Pending Queue FIFO will be flushed (currently loaded Pending Queue Descriptors are lost). This bit must be set to zero for proper operation.

0 = FIFO in normal operation
1 = FIFO is flushed

Bit 2 / Transmit Done Queue FIFO Enable (TDQFE). See Section 8.2.4 for details.

Bit 3 / Transmit Done Queue FIFO Flush (TDQF). See Section 8.2.4 for details.

Bits 8 to 10 / Transmit Done Queue Status Bit Threshold Setting (TDQT0 to TDQT2). See Section 8.2.4 for more details.

8.2.4 DONE QUEUE

The DMA will write to the Transmit Done Queue when it has finished either transmitting a complete packet chain or a complete data buffer. This option is selected by the Host when it configures the DQS field in the Transmit DMA Configuration RAM. See Section 8.2.5 for more details on the Transmit DMA Configuration RAM. The descriptor location is indicated in the Done Queue via a 16-bit pointer which the Host will use along with the Transmit Descriptor Base Address to find the exact 32-bit address of the associated Transmit Descriptor.

TRANSMIT DONE QUEUE DESCRIPTOR Figure 8.2.4A

dword 0

Status(3)	CH RST	PRI	00b	HDLC CH#(6)	Descriptor Pointer (16)

(Note: the organization of the Done Queue is not affected by the enabling of Big Endian)

dword 0; Bits 0 to 15 / Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address to either the first descriptor in a HDLC packet (can be a single descriptor) that has been transmitted (DQS = 0) or the descriptor that corresponds to a single data buffer that has been transmitted (DQS = 1).

dword 0; Bits 16 to 21 / HDLC Channel Number. HDLC channel number, which can be from 1 to 40.

000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

dword 0; Bits 22 to 23 / Unused. Set to 00b by the DMA.

dword 0; Bit 24 / Priority Packet (PRI). This field is meaningless in the Done Queue and could be set to any value. See the Pending Queue description in Section 8.2.3 for details.

dword 0; Bit 25 / Channel Reset (CH RST). This field is meaningless in the Done Queue and could be set to any value. See the Pending Queue description in Section 8.2.3 for details.

dword 0; Bits 26 to 28 / Packet Status. These three bits report the final status of an outgoing packet. All of the error states cause a HDLC abort sequence (8 ones in a row followed by continuous Interfill Bytes of either FFh or 7Eh) to be sent and the channel will be placed out of service by the transmit DMA setting the Channel Enable (CHEN) bit in the Transmit DMA Configuration RAM to zero. The status state of 000 will only be used when the channel has been configured by the Host to write to the Done Queue only after a complete HDLC packet (can be a single data buffer) has been transmitted (i.e. DQS = 0). The status states of 001, 010, and 011 will only be used when the channel has been configured by the Host to write to the Done Queue after each data buffer has been transmitted (i.e. DQS = 1).

000 = packet transmission complete and the Descriptor Pointer field corresponds to the first descriptor in a HDLC packet (can be a single descriptor) that has been transmitted (DQS = 0)
001 = first buffer transmission complete of a multi (or single) buffer packet (DQS = 1)
010 = middle buffer transmission complete of a multi-buffer packet (DQS = 1)
011 = last buffer transmission complete of a multi-buffer packet (DQS = 1)
100 = software provisioning error; this channel was not enabled
101 = descriptor error; either byte count = 0 or channel code inconsistent with Pending Queue
110 = PCI error; abort
111 = transmit FIFO error; it has underflowed

dword 0; Bits 29 to 31 / Unused. Not used by the DMA. Could be any value when read.

The Host will read from the Transmit Done Queue to find which data buffers and their associated descriptors have completed transmission. The Transmit Done Queue is circular queue. To keep track of the addresses of the circular queue in the Transmit Done Queue, there are a set of internal addresses within the device that accessed by both the Host and the DMA. On initialization, the Host will configure all of the registers shown in Table 8.2.4A. After initialization, the DMA will only write to (i.e. change) the write pointer and the Host will only write to the read pointer.

EMPTY CASE

The Transmit Done Queue is considered empty when the read and write pointers are identical.

TRANSMIT DONE QUEUE EMPTY STATE

	empty descriptor	
	empty descriptor	
	empty descriptor	
read pointer >	empty descriptor	< write pointer
	empty descriptor	
	empty descriptor	
	empty descriptor	

FULL CASE

The Transmit Done Queue is considered full when the read pointer is ahead of the write pointer by one descriptor. Hence, one descriptor must always remain empty.

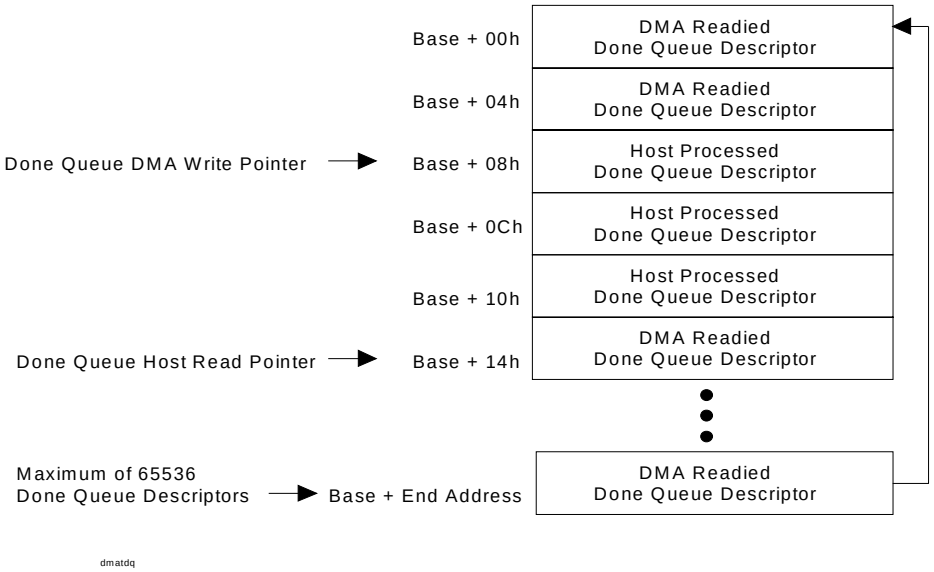
TRANSMIT DONE QUEUE FULL STATE

	valid descriptor	
	valid descriptor	
	empty descriptor	< write pointer
read pointer >	valid descriptor	
	valid descriptor	
	valid descriptor	
	valid descriptor	

TRANSMIT DONE QUEUE INTERNAL ADDRESS STORAGE Table 8.2.4A

Register Name	**Acronym**	**Address**
Transmit Done Queue Base Address 0 (lower word)	TDQBA0	0830h
Transmit Done Queue Base Address 1 (upper word)	TDQBA1	0834h
Transmit Done Queue DMA Write Pointer	TDQWP	0840h
Transmit Done Queue Host Read Pointer	TDQRP	083Ch
Transmit Done Queue End Address	TDQEA	0838h
Transmit Done Queue FIFO Flush Timer	TDQFFT	0844h

TRANSMIT DONE QUEUE STRUCTURE Figure 8.2.4B



Once the Transmit DMA is activated (via the TDE control bit in the Master Configuration register; see Section 4 for more details), it can begin writing data to the Done Queue. It knows where to write data into the Done Queue by reading the Write Pointer and adding it to the Base Address to obtain the actual 32-bit address. Once the DMA has written to the Done Queue, it increments the Write Pointer by one dword. A check must be made to make sure the incremented address does not exceed the Transmit Done Queue End Address. If the incremented address does exceed this address, then the incremented write pointer will be set equal to 0000h (i.e. the Base Address).

STATUS BITS / INTERRUPTS

On writes to the Done Queue by the DMA, the DMA will set the Status Bit for Transmit DMA Done Queue Write (TDQW) in the Status Register for DMA (SDMA). The Host can configure the DMA to either set this status bit on each write to the Done Queue or only after multiple (from 2 to 128) writes. The Host controls this by setting the TDQT0 to TDQT2 bits in the Transmit DMA Queues Control (TDMAQ) register. See the description of the TDMAQ register at the end of this section for more details. The DMA also checks the Transmit Done Queue Host Read Pointer to make sure that an overflow does not occur. If this does occur, then the DMA will set the Status Bit for Transmit DMA Done Queue Write Error (TDQWE) in the Status Register for DMA (SDMA) and it will not write to the Done Queue nor will it increment the Write Pointer. In such a scenario, information on transmitted packets will be lost and unrecoverable. Each of the status bits can also (if enabled) cause an hardware interrupt to occur. See Section 4 for more details.

DONE QUEUE BURST WRITING

one Queue FIFO can write descriptors to the Done Queue, then it will burst write them, increment the write pointer, and set the Status Bit for Transmit DMA Done Queue Write (TDQW) in the Status Register for DMA (SDMA). See Section 4 for more details on Status bits.

DONE QUEUE FIFO FLUSH TIMER

To make sure that the Done Queue FIFO does get flushed to the Done Queue on a regular basis, the Transmit Done Queue FIFO Flush Timer (TDQFFT) is used by the DMA to determine the maximum wait time in between writes. The TDQFFT is a 16-bit programmable counter that is decremented every PCLK divided by 256. It is only monitored by the DMA when the Transmit Done Queue FIFO is enabled (TDQFE = 1). For a 33 MHz PCLK, the timer is decremented every 7.76 µs and for a 25 MHz clock it is decremented every 10.24 µs. Each time the DMA writes to the Done Queue it resets the timer to the count placed into it by the Host. On initialization, the Host will set a value into the TDQFFT that indicates the maximum time the DMA should wait in between writes to the Done Queue. For example, with a PCLK of 33 MHz, the range of wait times are from 7.8 µs (RDQFFT = 0001h) to 508 ms (RDQFFT = FFFFh) and PCLK of 25 MHz, the wait times range from 10.2 µs (RDQFFT = 0001h) to 671 ms (RDQFFT = FFFFh).

Register Name: **TDQFFT**
Register Description: **Transmit Done Queue FIFO Flush Timer**
Register Address: **0844h**

7	6	5	4	3	2	1	0
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
15	14	13	12	11	10	9	8
TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 15 / Transmit Done Queue FIFO Flush Timer Control Bits (TC0 to TC15). Please note that on system reset, the timer will be set to 0000h which is defined as an illegal setting. If the Receive Done Queue FIFO is to be activated (TDQFE = 1), then the Host must first configure the timer to a proper state and then set the TDQFE bit to one.

0000h = illegal setting
0001h = Timer Count Resets to 1
FFFFh = Timer Count Resets to 65536

Register Name: **TDMAQ**
Register Description: **Transmit DMA Queues Control**
Register Address: **0880h**

7	6	5	4	3	2	1	0
n/a	n/a	n/a	n/a	TDQF	TDQFE	TPQF	TPQFE
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	n/a	TDQT2	TDQT1	TDQT0

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Transmit Pending Queue FIFO Enable (TPQFE). See Section 8.2.3 for details
Bit 1 / Transmit Pending Queue FIFO Flush (TPQLF). See Section 8.2.3 for details.

Bit 3 / Transmit Done Queue FIFO Enable (TDQFE). To enable the DMA to burst write descriptors to the Done Queue, this bit must be set to a one. If this bit is set to zero, descriptors will be written one at a time.

0 = Done Queue Burst Write Disabled
1 = Done Queue Burst Write Enabled

Bit 4 / Transmit Done Queue FIFO Flush (TDQF). When this bit is set to one, the internal Done Queue FIFO will be flushed by sending all data into the Done Queue. This bit must be set to zero for proper operation.

0 = FIFO in normal operation
1 = FIFO is flushed

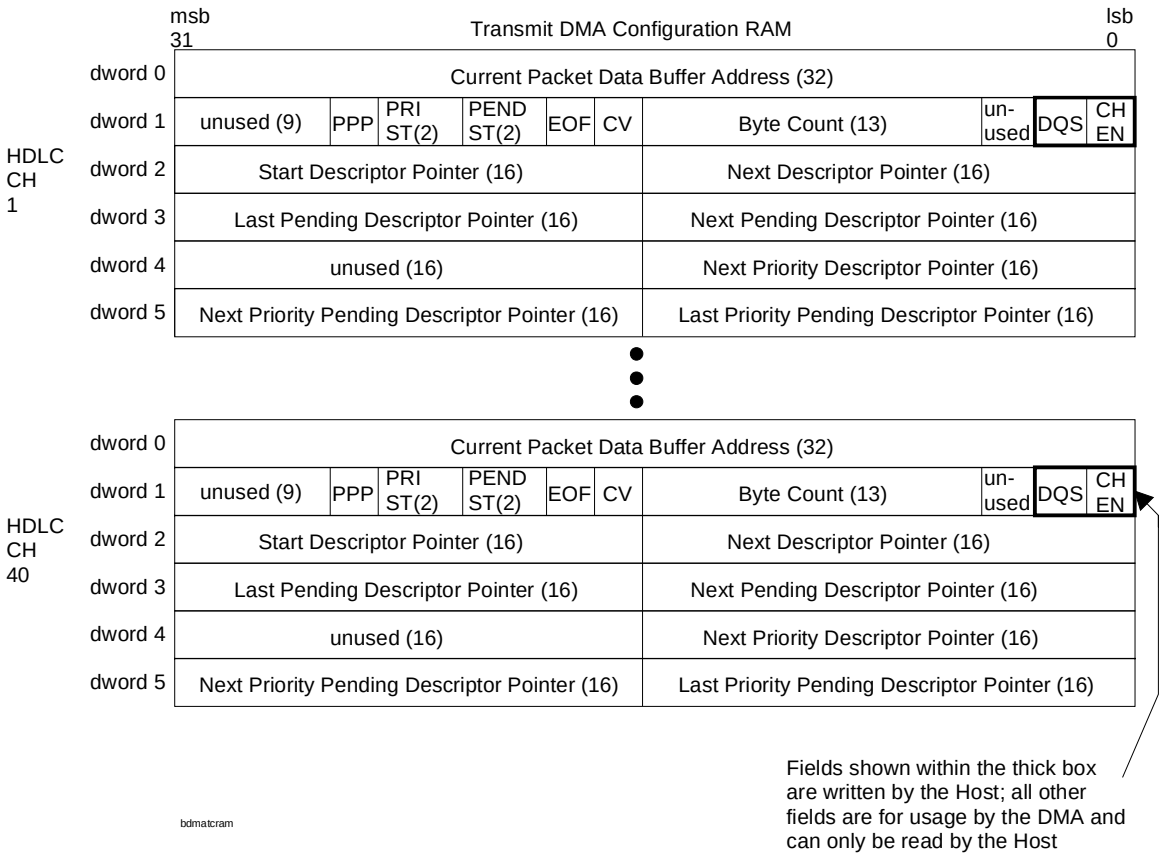
Bits 8 to 10 / Transmit Done Queue Status Bit Threshold Setting (TDQT0 to TDQT2). These three bits determine when the DMA will set the Transmit DMA Done Queue Write (TDQW) status bit in the Status Register for DMA (SDMA) register.

000 = set the TDQW status bit after each descriptor write to the Done Queue
001 = set the TDQW status bit after 2 or more descriptors are written to the Done Queue
010 = set the TDQW status bit after 4 or more descriptors are written to the Done Queue
011 = set the TDQW status bit after 8 or more descriptors are written to the Done Queue
100 = set the TDQW status bit after 16 or more descriptors are written to the Done Queue
101 = set the TDQW status bit after 32 or more descriptors are written to the Done Queue
110 = set the TDQW status bit after 64 or more descriptors are written to the Done Queue
111 = set the TDQW status bit after 128 or more descriptors are written to the Done Queue

8.2.5 DMA CONFIGURATION RAM

Onboard the device there is a set of 240 dwords (6 dwords per channel times 40 channels) that are used by the Host to configure the DMA and by the DMA to store values locally when it is processing a packet. Most of the fields within the DMA Configuration RAM are for use by the DMA and the Host will never write to these fields. The Host is only allowed to write (i.e. configure) to the lower word of dword 1 for each HDLC channel. The Host configurable fields are denoted with a thick box as shown below.

TRANSMIT DMA CONFIGURATION RAM Figure 8.2.5A



- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -

dword 0; Bits 0 to 31 / Current Data Buffer Address. The current 32-bit address of the data buffer that is being used. This address is used by the DMA to keep track of where data should be read from as it is passed to the transmit FIFO.

- HOST MUST CONFIGURE -

dword 1; Bit 0 / Channel Enable (CHEN). This bit is controlled by both the Host and the transmit DMA to enable and disable a HDLC channel. The DMA will automatically disable a channel when an error condition occurs (see Section 8.2.1 for a discussion on error conditions). The DMA will automatically enable a channel when it detects that the Channel Reset (CHRST) bit in the Pending Queue descriptor is set to a one.

0 = HDLC Channel Disabled
1 = HDLC Channel Enabled

- HOST MUST CONFIGURE -

dword 1; Bit 1 / Done Queue Select (DQS). This bit determines whether the transmit DMA will write to the Done Queue only after a complete HDLC packet (which may be only a single buffer) has been transmitted (in which case the Descriptor Pointer in the Done Queue will correspond to the first descriptor of the packet) or whether it should write to the Done Queue after each data buffer has been transmitted (in which case the Descriptor Pointer in the Done Queue will correspond to a single data buffer). The setting of this bit also affects the reporting of the Status field in the Transmit Done Queue. When DQS = 0, the only non-errored status possible is a setting of 000. When DQS = 1, then the non-errored settings of 001, 010, and 011 are possible.

0 = write to the Done Queue only after a complete HDLC packet has been transmitted
1 = write to the Done Queue after each data buffer is transmitted

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 1; Bit 2/ Unused. This field is not used by the DMA and could be any value when read.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 1; Bits 3 to 15 / Byte Count. The DMA uses these 13 bits to keep track of the number of bytes stored in the data buffer. Maximum is 8191 bytes (0000h =0 bytes / 1FFFh = 8191 bytes).

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 1; Bit 16 / Chain Valid (CV). This is an internal copy of the CV field that resides in the current Packet Descriptor that the DMA is operating on. See Section 8.2.2 for more details on the CV bit.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 1; Bit 17 / End Of Frame (EOF). This is an internal copy of the EOF field that resides in the current Packet Descriptor that the DMA is operating on. See Section 8.2.2 for more details on the EOF bit.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 1; Bits 18 to 19 / Pending State (PENDST). This field is used by the transmit DMA to keep track of queued descriptors as they arrive from the Pending Queue and for the DMA to know when it should create a horizontal linked list of transmit descriptors and where it can find the next valid descriptor. This field handles standard packets and the PRIST field handles priority packets.

State	Next Descriptor Pointer field	Next Pending Descriptor Pointer field
00	not valid	not valid
01	valid	not valid
10	not valid	valid
11	valid	valid

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 1; Bits 20 to 21 / Priority State (PRIST). This field is used by the transmit DMA to keep track of queued priority descriptors as they arrive from the Pending Queue and for the DMA to know when it should create a horizontal linked list of transmit priority descriptors and where it can find the next valid priority descriptor. This field handles priority packets and the PENDST field handles standard packets.

State	Next Priority Descriptor Pointer field	Next Priority Pending Descriptor Pointer field
00	not valid	not valid
01	valid	not valid
10	not valid	valid
11	valid	valid

-FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD –
dword 1; Bit 22/ Processing Priority Packet (PPP). This bit will be set to a one when the DMA is currently processing a priority packet.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -
dword 1; Bits 23 to 31/ Unused. This field is not used by the DMA and could be any value when read

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD –

dword 2; Bits 0 to 15 / Next Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address of the next Transmit Packet Descriptor for the packet that is currently being transmitted. Only valid if EOF = 0 or if EOF = 1 and CV = 1.**- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -**

dword 2; Bits 16 to 31 / Start Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address of the 1st Transmit Packet Descriptor for the packet that is currently being transmitted. If DQS = 0, then this pointer is written back to the Done Queue when the packet has completed transmission. This field is used by the DMA for processing standard as well as priority packets.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -

dword 3; Bits 0 to 15 / Next Pending Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address of the 1st Transmit Packet Descriptor for the packet that is queued up next for transmission.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -

dword 3; Bits 16 to 31 / Last Pending Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address of the 1st Transmit Packet Descriptor for the packet that is queued up last for transmission.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -

dword 4; Bits 0 to 15 / Next Priority Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address of the next Transmit Priority Packet Descriptor for the priority packet that is currently being transmitted. Only valid if EOF = 0 or if EOF = 1 and CV = 1.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -

dword 4; Bits 16 to 31/ Unused. This field is not used by the DMA and could be any value when read.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -

dword 5; Bits 0 to 15 / Last Priority Pending Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address of the 1st Transmit Priority Packet Descriptor for the priority packet that is queued up last for transmission.

- FOR DMA USAGE ONLY / HOST CAN ONLY READ THIS FIELD -

dword 5; Bits 16 to 31 / Next Priority Pending Descriptor Pointer. This 16-bit value is the offset from the Transmit Descriptor Base Address of the 1st Transmit Priority Packet Descriptor for the packet priority that is queued up next for transmission.

Register Name: **TDMACIS**
Register Description: **Transmit DMA Configuration Indirect Select**
Register Address: **0870h**

7	6	5	4	3	2	1	0
n/a	n/a	HCID5	HCID4	HCID3	HCID2	HCID1	HCID0
15	14	13	12	11	10	9	8
IAB	IARW	n/a	n/a	TDCW3	TDCW2	TDCW1	TDCW0

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bits 0 to 5 / HDLC Channel ID (HCID0 to HCID5).
000000 (00h) = HDLC Channel Number 1
100111 (27h) = HDLC Channel Number 40

Bits 8 to 11 / Transmit DMA Configuration RAM Word Select Bits 0 to 3 (TDCW0 to TDCW3).
0000 = lower word of dword 0
0001 = upper word of dword 0
0010 = lower word of dword 1 (only word that the Host can write to)
0011 = upper word of dword 1
0100 = lower word of dword 2
0101 = upper word of dword 2
0110 = lower word of dword 3
0111 = upper word of dword 3
1000 = lower word of dword 4
1001 = upper word of dword 4
1010 = lower word of dword 5
1011 = upper word of dword 5

Bit 14 / Indirect Access Read/Write (IARW). When the host wishes to read data from the internal Transmit DMA Configuration RAM, this bit should be written to a one by the host. This causes the device to begin obtaining the data from the channel location indicated by the HCID bits. During the read access, the IAB bit will be set to one. Once the data is ready to be read from the TDMAC register, the IAB bit will be set to zero. When the host wishes to write data to the internal Transmit DMA Configuration RAM, this bit should be written to a zero by the host. This causes the device to take the data that is current present in the TDMAC register and write it to the channel location indicated by the HCID bits. When the device has completed the write, the IAB bit will be set to zero.

Bit 15 / Indirect Access Busy (IAB). When an indirect read or write access is in progress, this read only bit will be set to a one. During a read operation, this bit will be set to a one until the data is ready to be read. It will be set to zero when the data is ready to be read. During a write operation, this bit will be set to a one while the write is taking place. It will be set to zero once the write operation has completed.

Register Name: **TDMAC**
Register Description: **Transmit DMA Configuration**
Register Address: **0874h**

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8
D15	D14	D13	D12	D11	D10	D9	D8

Note: bits that are underlined are read only, all other bits are read-write.

Bits 0 to 15 / Transmit DMA Configuration RAM Data (D0 to D15). Data that is written to or read from the Transmit DMA Configuration RAM.

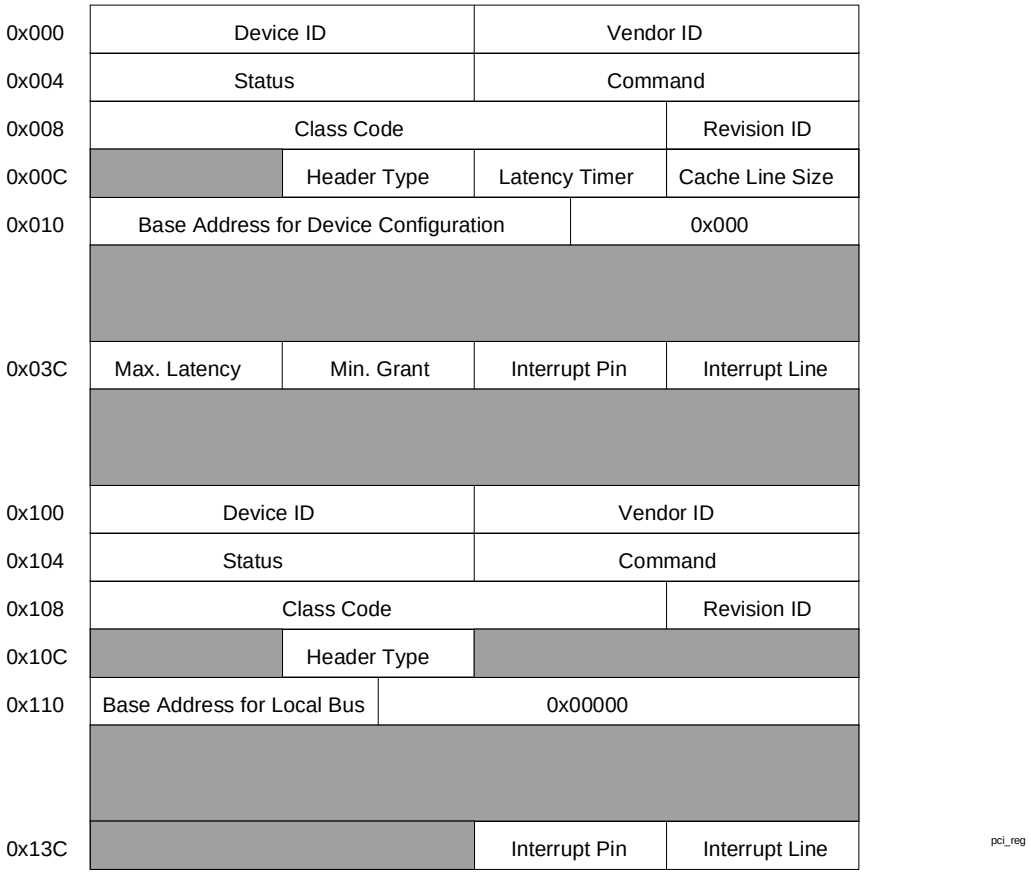
SECTION 9: PCI BUS

9.1 PCI GENERAL DESCRIPTION OF OPERATION

The PCI Block interfaces the DMA Block to an external high-speed bus. The PCI Block complies with Revision 2.1 (June 1, 1995) of the PCI Local Bus Specification. HDLC packet data will always pass to and from BoSS via the PCI bus. The user has the option to configure and monitor the internal device registers either via the PCI bus (Local Bus Bridge mode) or via the Local Bus (Local Bus Configuration mode). When the Local Bus Bridge mode is used, the Host on the PCI bus can also bridge to the Local Bus and will set/monitor the PCI Configuration registers. When the Local Bus Configuration mode is used, the CPU on the Local Bus will set/monitor the PCI Configuration registers.
The PCI Configuration registers (see Figure 9.1A) are described in detail in Section 9.2. The following is a set of notes that apply to the PCI Configuration registers:

1. All unused locations (the shaded areas of Figure 9.1A) will return zeros when read
2. Read only locations can be written with either a one or zero with no affect
3. All bits are read/write unless otherwise noted.

PCI CONFIGURATION MEMORY MAP Figure 9.1A

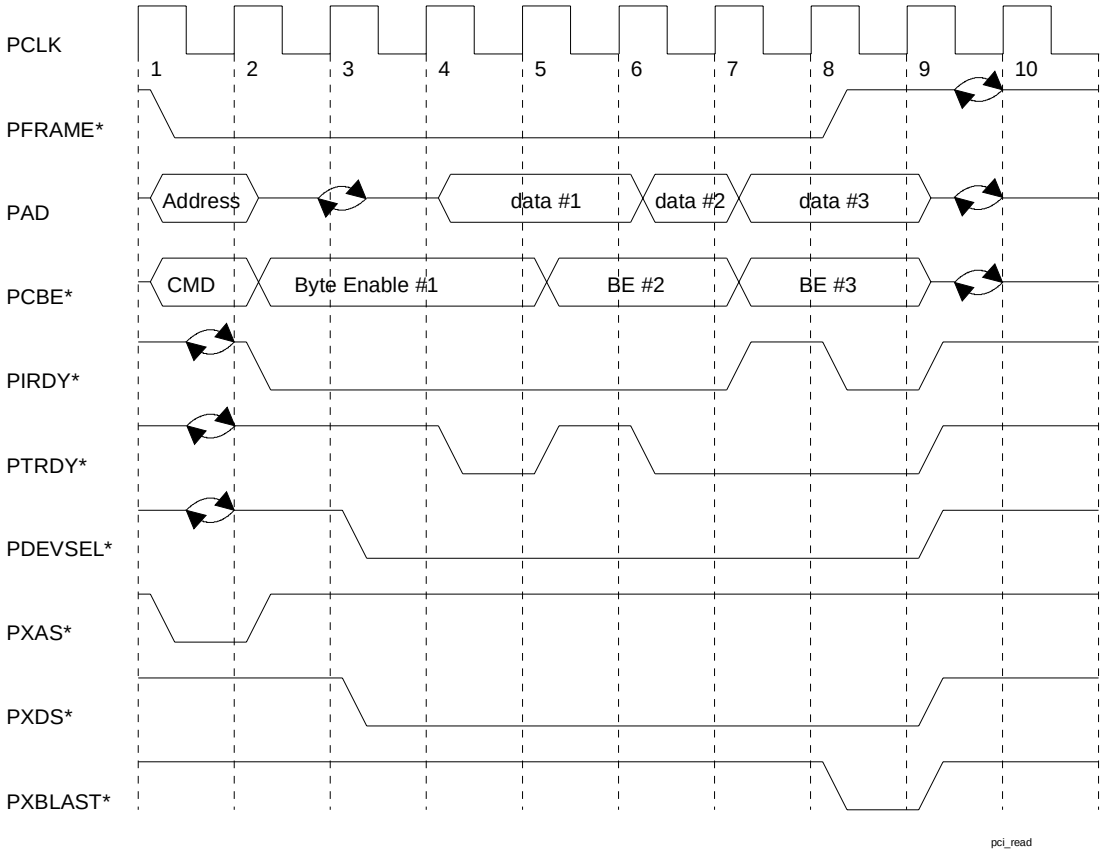


PCI READ CYCLE

A read cycle on the PCI bus is shown in Figure 9.1B. During clock cycle #1, the initiator asserts the PFRAME* signal and drives the address onto the PAD signal lines and the bus command (which would be a read) onto the PCBE* signal lines. The target reads the address and bus command and if the address matches it's own, then it will assert the PDEVSEL* signal and begin the bus transaction. During clock cycle #2, the initiator stops driving the address onto the PAD signal lines and switches the PCBE* signal lines to now indicate byte enables. It also asserts the PIRDY* signal and begins monitoring the PDEVSEL* and PTRDY* signals. During clock cycle #4, the target asserts PTRDY* indicating to the initiator that valid data is available to be read on the PAD signal lines by the initiator. During clock cycle #5, the target is not ready to provide data #2 because PTRDY* is deasserted. During clock cycle #6, the target again asserts PTRDY* informing the initiator to read data #2. During clock cycle #7, the initiator deasserts PIRDY* indicating to the target that it is not ready to accept data. During clock cycle #8, the initiator asserts PIRDY* and acquires data #3. Also during clock cycle #8, the initiator deasserts PFRAME* indicating to the target that the bus transaction is complete and no more data needs to be read. During clock cycle #9, the target deasserts PTRDY* and PDEVSEL* and the initiator deasserts PIRDY*.

The PXAS*, PXDS*, and PXBLAST* signals are not part of a standard PCI bus. These PCI extension signals that are unique to the device. They are useful in adapting the PCI bus to a proprietary bus scheme. They are only asserted when the device is a bus master.

PCI BUS READ Figure 9.1B

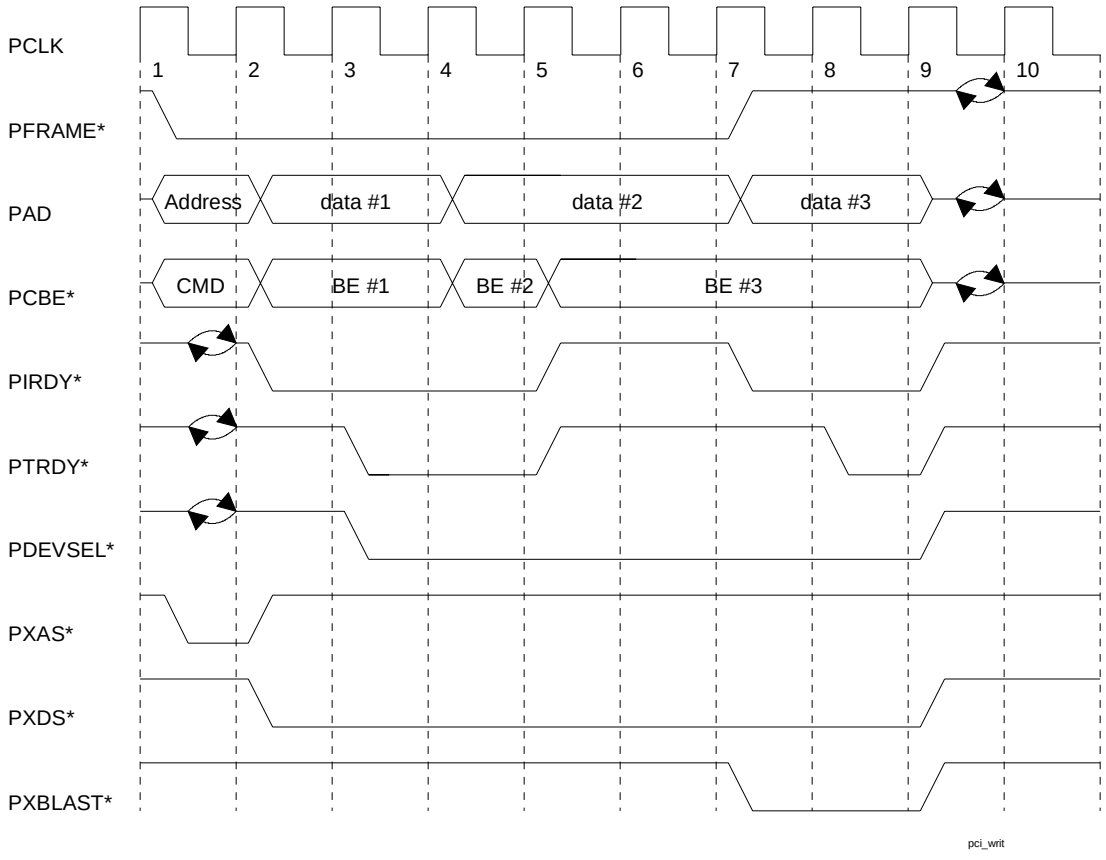


PCI WRITE CYCLE

A write cycle on the PCI bus is shown in Figure 9.1C. During clock cycle #1, the initiator asserts the PFRAME* signal and drives the address onto the PAD signal lines and the bus command (which would be a write) onto the PCBE* signal lines. The target reads the address and bus command and if the address matches it's own, then it will assert the PDEVSEL* signal and begin the bus transaction. During clock cycle #2, the initiator stops driving the address onto the PAD signal lines and begins driving data #1. It also switches the PCBE* signal lines to now indicate the byte enable for data #1. The initiator asserts the PIRDY* signal and begins monitoring the PDEVSEL* and PTRDY* signals. During clock cycle #3, the initiator detects that PDEVSEL* and PTRDY* are asserted which indicates that the target has accepted data #1 and the initiator begins driving the data and byte enable for data #2. During clock cycle #4, since PDEVSEL* and PTRDY* are asserted, data #2 is written by the initiator to the target. During clock cycle #5, both PIRDY* and PTRDY* are deasserted indicating that neither the initiator nor the target are ready for data #3 to be passed. During clock cycle #6, the initiator is now ready so it asserts PIRDY* and deasserts PFRAME* which indicates that data #3 will be the last one passed. During clock cycle #8, the target asserts PTRDY* which indicates to the initiator that data #3 is ready to be accepted by the target. During clock cycle #9, the initiator deasserts PIRDY* and stops driving the PAD and PCBE* signal lines. The target deasserts PDEVSEL* and PTRDY*.

The PXAS*, PXDS*, and PXBLAST* signals are not part of a standard PCI bus. These PCI extension signals that are unique to the device. They are useful in adapting the PCI bus to a proprietary bus scheme. They are only asserted when the device is a bus master.

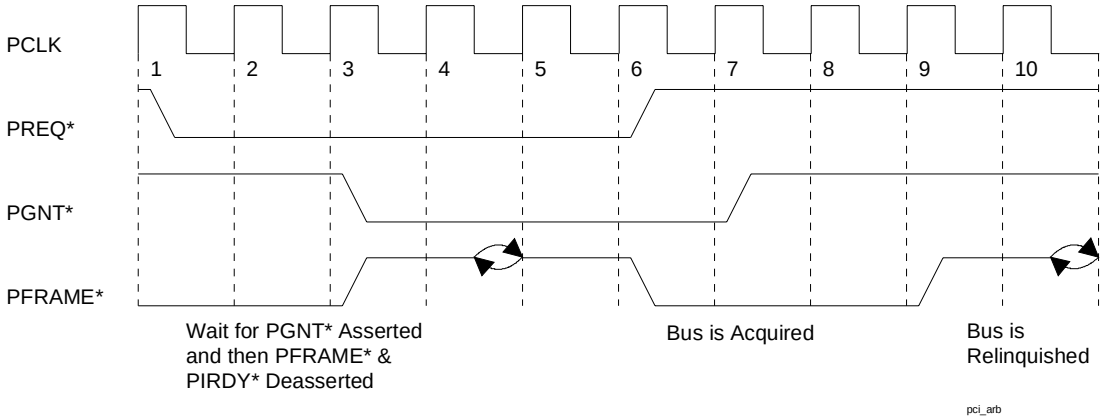
PCI BUS WRITE Figure 9.1C



PCI BUS ARBITRATION

The PCI bus can be arbitrated as shown in Figure 9.1D. The initiator will request bus access by asserting PREQ*. A central arbiter will grant the access some time later by asserting PGNT*. Once the bus has been granted, the initiator will wait until both PIRDY* and PFRAME* are deasserted (i.e. an idle cycle) before acquiring the bus and beginning the transaction. As shown in Figure 9.1C, the bus was still being used when it was granted and the device had to wait until clock cycle #6 before it acquired the bus and begin the transaction. The arbiter can deassert PGNT* at any time and the initiator must relinquish the bus after the current transfer is complete (which can be limited by the latency timer).

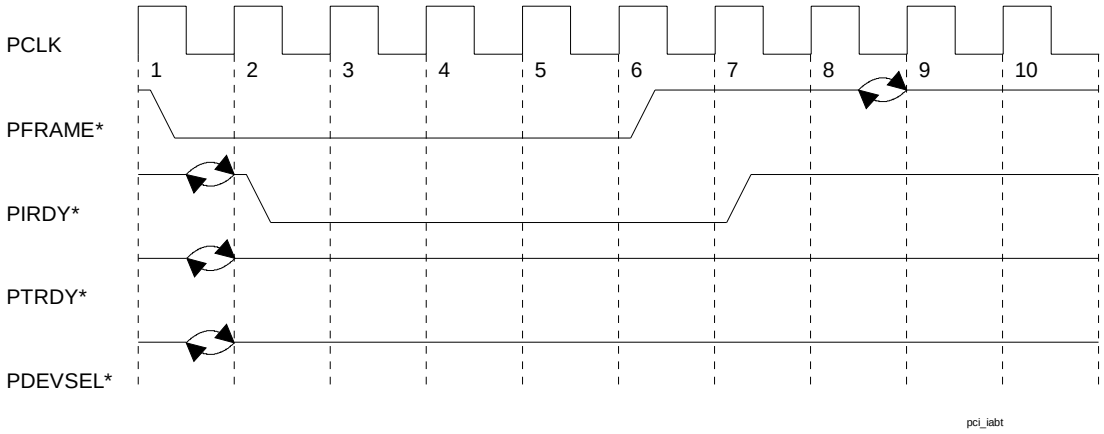
PCI BUS ARBITRATION SIGNALING PROTOCOL Figure 9.1D



PCI INITIATOR ABORT

If a target fails to respond to an initiator by asserting PDEVSEL* and PTRDY* within 5 clock cycles, then the initiator will abort the transaction by deasserting PFRAME* and then one clock later deasserting PIDRY* (see Figure 9.1E). If such a scenario occurs, it will be reported via the Master Abort status bit in the PCI Command/Status configuration register (see Section 9.2).

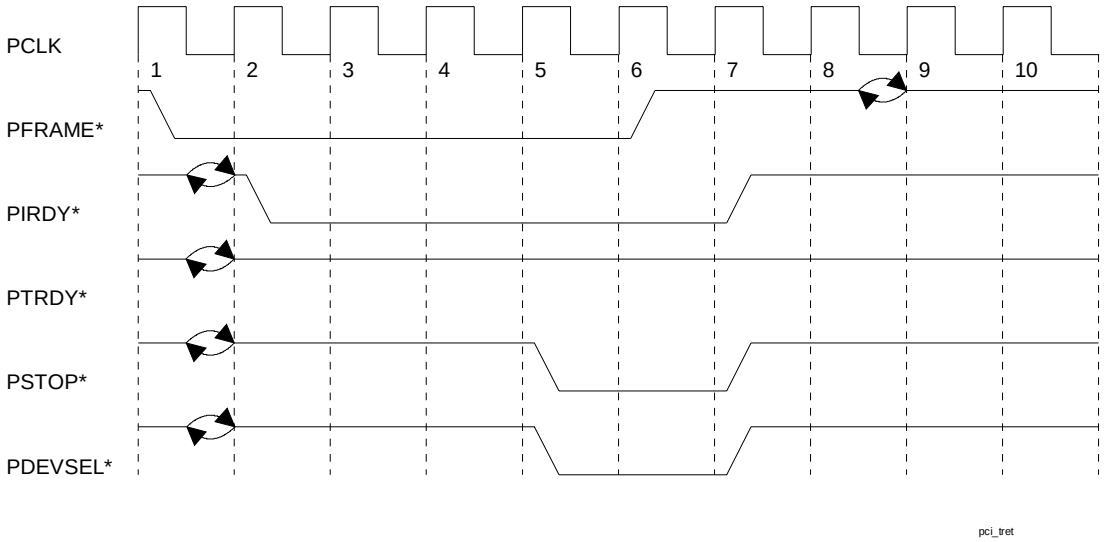
PCI INITIATOR ABORT Figure 9.1E



PCI TARGET RETRY

Targets can terminate the requested bus transaction before any data is transferred because the target is busy and temporarily unable to process the transaction. Such a termination is called a target retry and no data is transferred. A target retry is signaled to the initiator by the assertion of PSTOP* and not asserting PTRDY* on the initial data phase (see Figure 9.1F). When BoSS is a target, it will only issue a target retry when the Host is accessing the Local Bus. This will occur when the Local Bus is being operated in the arbitration mode and at the instant the Host requests access to the Local Bus, it is busy. See Section 10.1 for more details on the operation of the Local Bus.

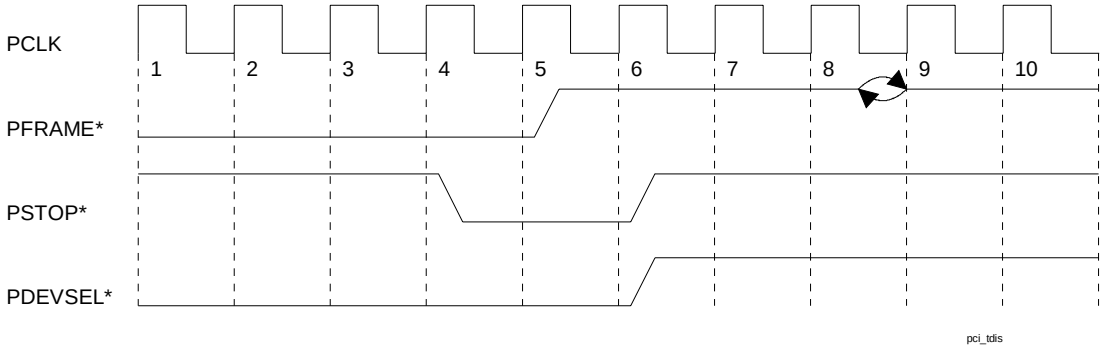
PCI TARGET RETRY Figure 9.1F



PCI TARGET DISCONNECT

A target can terminate a transaction prematurely by asserting PSTOP* (see Figure 9.1G). Depending on the current state of the ready signals when PSTOP* is asserted, data may or may not be transferred. The target will always deassert PSTOP* when it detects that the initiator has deasserted PFRAME*. When BoSS is a target it will disconnect with data after the first data phase is complete if the master attempts a burst transaction. This is because the device does not support burst transactions when it is a target. When it is an initiator and experiences a disconnect from the target, it will attempt another bus transaction (if it still has the bus granted) after waiting either one (disconnect without data) or two clock cycles (disconnect with data).

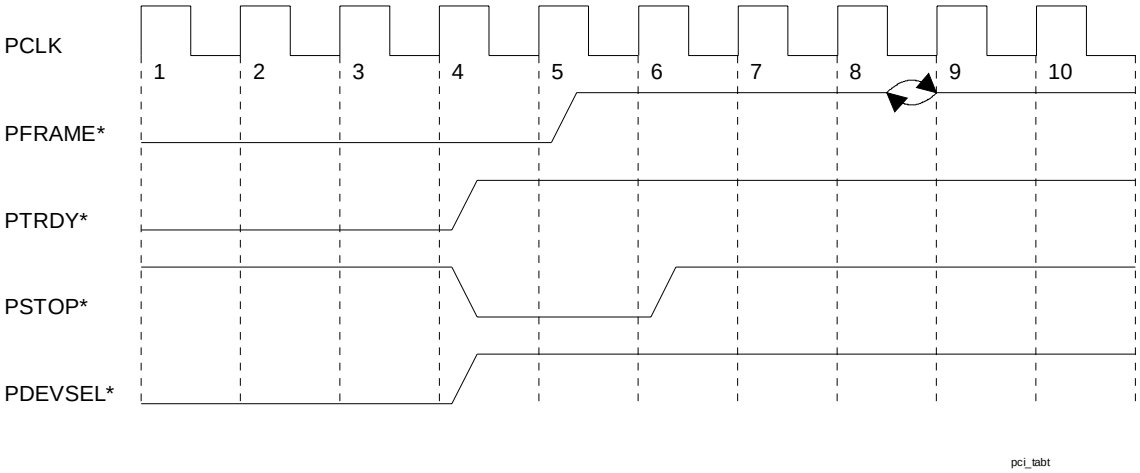
PCI TARGET DISCONNECT Figure 9.1G



PCI TARGET ABORT

Targets can also abort the current transaction, which means that it does not wish for the initiator to attempt the request again. No data is transferred in a target abort scenario. A target abort is signaled to the initiator by the simultaneous assertion of PSTOP* and deassertion of PDEVSEL* (see Figure 9.1H). When BoSS is a target, it will only issue a target abort when the Host is accessing the Local Bus. This will occur when the Host attempts a bus transaction with a combination of bytes enables (PCBE*) that is not supported by the Local Bus. If such a scenario occurs, it will be reported via the Target Abort Initiated status bit in the PCI Command/Status configuration register (see Section 9.2). See Section 10.1 for details on Local Bus operation. When BoSS is a bus master, if it detects a target abort, then it will be reported via the Target Abort Detected by Master status bit in the PCI Command/Status configuration register (see Section 9.2).

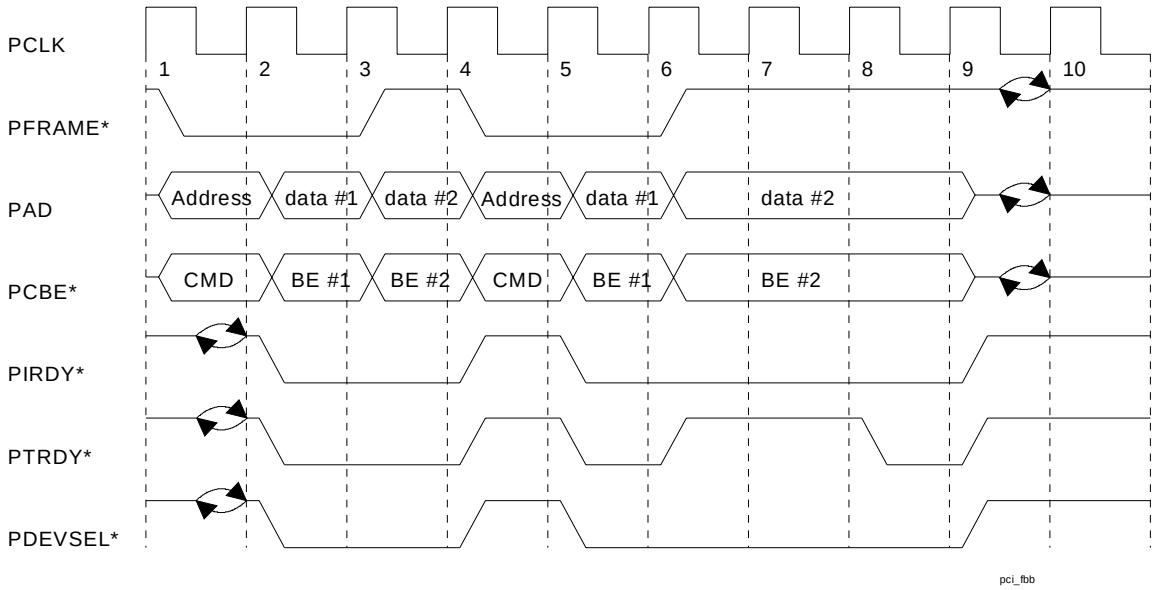
PCI TARGET ABORT Figure 9.1H



PCI FAST BACK-TO-BACK

Fast back-to-back transactions are two consecutive bus transactions without the usually required idle cycle (PFRAME* and PIRDY* deasserted) between them. This can only occur when there is a guarantee that there will not be any contention on the signal lines. The PCI specification allows two types of fast back-to-back transactions, those that access the same agent (Type 1) and those that do not (Type 2). Figure 9.1J shows an example of a fast back-to-back transaction where no idle cycle exists. As a bus master, BoSS is no capable of performing a Type 2 access. As a target, it can accept both types of fast back-to-back transactions.

PCI FAST BACK-TO-BACK Figure 9.1J



9.2 PCI CONFIGURATION REGISTER DESCRIPTION

Register Name: **PVID0**
Register Description: **PCI Vendor ID / Device ID Register 0**
Register Address: **0x000h**

lsb

Vendor ID (Read Only / set to EAh)

Vendor ID (Read Only / set to 13h)

Device ID (Read Only / set to 31h)

msb

Device ID (Read Only / set to 31h)

Bits 0 to 15 / Vendor ID. These read only bits identify Dallas Semiconductor as the manufacturer of the device. The Vendor ID was assigned by the PCI Special Interest Group and is fixed at 13EAh.

Bits 16 to 31 / Device ID. These read only bits identify the DS3131 as the device being used. The Device ID was assigned by Dallas Semiconductor and is fixed at 3131h.

Register Name: **PCMD0**
Register Description: **PCI Command / Status Register 0**
Register Address: **0x004h**

lsb

STEPC	PARC	VGA	MWEN	SCC	MASC	MSC	IOC

Reserved (Read Only / set to all zeros)	FBBEN	PSEC

FBBCT	UDF	66 MHz	Reserved (Read Only / set to all zeros)

msb

PPE	PSE	MABT	TABTM	TABT	DTS1	DTS0	PARR

Read only bits in the PCMD0 register are indicated above by being underlined. All other bits are read-write.

The lower word (bits 0 to 15) of the PCMD0 register is the Command portion and is used for control of the PCI bus. When all bits in the lower word are set to zero, then the device is logically disconnected from the bus for all accesses except for accesses to the configuration registers. The upper word (bits 16 to 31) is the Status portion and it is used for status information. Reads to the Status portion behave normally but writes are unique in that bits can be reset (i.e. forced to zero) but not set (i.e. forced to one). A bit in the Status portion will be reset when a one is written to that bit position. Bit positions that have a zero written to them will not be reset.

COMMAND BITS

Bit 0 / I/O Space Control (IOC). This read only bit is forced to zero by the device to indicate that it does not respond to I/O Space accesses.

Bit 1 / Memory Space Control (MSC). This read/write bit controls whether or not the device will respond to accesses by the PCI bus to the memory space (which is the internal device configuration registers). When this bit is set to zero, the device will ignore accesses attempted to the internal configuration registers and when set to one, the device will allow accesses to the internal configuration registers. This bit should be set to zero when the Local Bus is operated in the Configuration Mode. This bit is force to zero when a hardware reset is initiated via the PRST* pin.

0 = ignore accesses to the internal device configuration registers
1 = allow accesses to the internal device configuration registers

Bit 2 / Master Control (MASC). This read/write bit controls whether or not the device can act as a master on the PCI bus. When this bit is set to zero, the device cannot act as a master and when it is set to one, the device can act as a bus master. This bit is forced to zero when a hardware reset is initiated via the PRST* pin.

0 = deny the device from operating as a bus master
1 = allow the device to operate as a bus master

Bit 3 / Special Cycle Control (SCC). This read only bit is forced to zero by the device to indicate that it cannot decode Special Cycle operations.

Bit 4 / Memory Write & Invalidate Command Enable (MWEN). This read only bit is forced to zero by the device to indicate that it cannot generate the Memory Write and Invalidate command.

Bit 5 / VGA Control (VGA). This read only bit is forced to zero by the device to indicate that it is not a VGA compatible device.

Bit 6 / Parity Error Response Control (PARC). This read/write bit controls whether or not the device should ignore parity errors. When this bit is set to zero, the device will ignore any parity errors that it detects and continue to operate normally. When this bit is set to one, the device must act on parity errors. This bit is forced to zero when a hardware reset is initiated via the PRST* pin.

0 = ignore parity errors
1 = act on parity errors

Bit 7 / Address Stepping Control (STEPC). This read only bit is forced to zero by the device to indicate that it is not capable of address/data stepping.

Bit 8 / PCI System Error Control (PSEC). This read/write bit controls whether or not the device should enable the PSERR* output pin. When this bit is set to zero, the device will disable the PSERR* pin and when this bit is set to one, the device will enable the PSERR* pin. This bit is forced to zero when a hardware reset is initiated via the PRST* pin.

0 = disable the PSERR* pin
1 = enable the PSERR* pin

Bit 9 / Fast Back-to-Back Master Enable (FBBEN). This read only bit is forced to zero by the device to indicate that it is not capable of generating fast back-to-back transactions to different agents.

Bits 10 to 15 / Reserved. These read only bits are forced to zero by the device.

STATUS BITS

The upper words in the PCMD0 register are the Status portion, which report events as they occur. As mentioned earlier, reads of the Status portion occur normally but writes are unique in that bits can only be reset (i.e. forced to zero). This occurs when a one is written to a bit position. Writes with a zero to a bit position have no affect. This allows individual bits to be reset.

Bits 16 to 20 / Reserved. These read only bits are forced to zero by the device.

Bit 21 / 66 MHz Capable (66 MHz). This read only bit is forced to zero by the device to indicate that it is not capable of running at 66 MHz.

Bit 22 / User Definable Features Capable (UDF). This read only bit is forced to zero by the device to indicate that it does not support User Definable Features.

Bit 23 / Fast Back-to-Back Capable Target (FBBCT). This read only bit is forced to one by the device to indicate that it is capable of accepting fast back-to-back transactions when the transactions are not from the same agent.

Bit 24 / PCI Parity Error Reported (PARR). This read/write bit will be set to a one when the device is a bus master and detects or asserts the PPERR* signal when the PARC command bit is enabled. This bit can be reset (set to zero) by the Host by writing a one to this bit.

0 = no parity errors have been detected
1 = parity errors detected

Bit 25 & 26 / Device Timing Select Bits 0 & 1 (DTS0 & DTS1). These two read only bits are forced to 01b by the device to indicate that it is capable of the medium timing requirements for the PDEVSEL* signal.

Bit 27 / Target Abort Initiated (TABT). This read only bit is forced to zero by the device since it will not terminate a bus transaction with a target abort when the device is a target.

Bit 28 / Target Abort Detected by Master (TABTM). This read/write bit will be set to a one when the device is a bus master and it detects that a bus transaction has been aborted by the target with a target abort. This bit can be reset (set to zero) by the Host by writing a one to this bit.

Bit 29 / Master Abort (MABT). This read/write bit will be set to a one when the device is a bus master and the bus transaction is terminated with a master abort (except for Special Cycle). This bit can be reset (set to zero) by the Host by writing a one to this bit.

Bit 30 / PCI System Error Reported (PSE). This read/write bit will be set to a one when the device asserts the PSERR* signal (even if it is disabled via the PSEC Command bit). This bit can be reset (set to zero) by the Host by writing a one to this bit.

Bit 31 / PCI Parity Error Reported (PPE). This read/write bit will be set to a one when the device detects a parity error (even if parity is disabled via the PARC Command bit). This bit can be reset (set to zero) by the Host by writing a one to this bit.

Register Name: **PRCC0**
Register Description: **PCI Revision ID / Class Code Register 0**
Register Address: **0x008h**

lsb

Revision ID (Read Only / set to 00h)

Class Code (Read Only / set to 00h)

Class Code (Read Only / set to 80h)

msb

Class Code (Read Only / set to 02h)

Bits 0 to 7 / Revision ID. These read only bits identify the specific device revision and are selected by Dallas Semiconductor.

Bits 8 to 15 / Class Code Interface. These read only bits identify the sub-class interface value for the device and are fixed at 00h. See Appendix D of PCI Local Bus Specification Revision 2.1 for details.

Bits 16 to 23 / Class Code Sub-Class. These read only bits identify the sub-class value for the device and are fixed at 80h, which indicate "Other Network Controller". See Appendix D of PCI Local Bus Specification Revision 2.1 for details.

Bits 24 to 31 / Class Code Base Class. These read only bits identify the base class value for the device and are fixed at 02h, which indicate "Network Controllers". See Appendix D of PCI Local Bus Specification Revision 2.1 for details.

Register Name: **PLTH0**
Register Description: **PCI Latency Timer / Header Type Register 0**
Register Address: **0x00Ch**

lsb

Cache Line Size

Latency Timer

Header Type (Read Only / set to 80h)

msb

BIST (Read Only / set to 00h)

Bits 0 to 7 / Cache Line Size. These read/write bits indicates the cache line size in terms of dwords. If the burst size of a data read transaction exceeds this value, then the PCI Block will use the memory read multiple command. Valid settings are 04h (4 dwords), 08h, 10h, 20h, and 40h (64 dwords). Other settings are interpreted as 00h. These bits are forced to zero when a hardware reset is initiated via the PRST* pin.

Bits 8 to 15 / Latency Timer. These read/write bits indicate the value of the Latency Timer (in terms of the number of PCI clocks) for use when the device is a bus master. These bits are forced to zero when a hardware reset is initiated via the PRST* pin.

Bits 16 to 23 / Header Type. These read only bits are forced to 80h, which indicate a multifunction device.

Bits 24 to 31 / Built-In Self-Test (BIST). These read only bits are forced to zero.

Register Name: **PDCM**
Register Description: **PCI Device Configuration Memory Base Address Register**
Register Address: **0x010h**

lsb

Base Address (Read Only / set to 0h)	PF	TYPE1	TYPE0	MSI

Base Address	Base Address (Read Only / set to 0h)

Base Address

msb

Base Address

Read only bits in the PDCM register are indicated above by being underlined. All other bits are read-write.

Bit 0 / Memory Space Indicator (MSI). This read only bit is forced to zero to indicate that the internal device configuration registers are mapped to memory space.

Bits 1 & 2 / Type 0 & Type 1. These read only bits are forced to 00b to indicate that the internal device configuration registers can be mapped anywhere in the 32-bit address space.

Bit 3 / Prefetchable (PF). This read only bit is forced to zero to indicate that prefetching is not supported by the device for the internal device configuration registers.
Bits 4 to 11 / Base Address. These read only bits are forced to zero to indicate that the internal device configuration registers require 4k bytes of memory space.

Bits 12 to 31 / Base Address. These read/write bits define the location of the 4k memory space that is mapped to the internal configuration registers. These bits correspond to the most significant bits of the PCI address space.

Register Name: **PINTL0**
Register Description: **PCI Interrupt Line & Pin / Minimum Grant / Maximum Latency Register 0**
Register Address: **0x03Ch**

lsb

Interrupt Line
Interrupt Pin (Read Only / set to 01h)
Maximum Grant (Read Only / set to 05h)

msb

Maximum Latency (Read Only / set to 0 Fh)

Bits 0 to 7 / Interrupt Line. These read/write bits indicate and store interrupt line routing information. The device does not use this information, it is only posted here for use by the Host.

Bits 8 to 15 / Interrupt Pin. These read only bits are forced to 01h to indicate that the uses PINTA* as an interrupt.

Bits 16 to 23 / Minimum Grant. These read only bits are used to indicate to the Host, how long of a burst period the device needs assuming a clock rate of 33 MHz. The values placed in these bits specify a period of time in 0.25 µs increments. These bits are forced to 05h.

Bits 24 to 31 / Maximum Latency. These read only bits are used to indicate to the Host, how often the device needs to gain access to the PCI bus. The values placed in these bits specify a period of time in 0.25 µs increments. These bits are forced to 0 Fh.

Register Name: **PVID1**
Register Description: **PCI Vendor ID / Device ID Register 1**
Register Address: **0x100h**

lsb

Vendor ID (Read Only / set to EAh)
Vendor ID (Read Only / set to 13h)
Device ID (Read Only / set to 31h)

msb

Device ID (Read Only / set to 31h)

Bits 0 to 15 / Vendor ID. These read only bits identify Dallas Semiconductor as the manufacturer of the device. The Vendor ID was assigned by the PCI Special Interest Group and is fixed at 13EAh.

Bits 16 to 31 / Device ID. These read only bits identify the DS3131 as the device being used. The Device ID was assigned by Dallas Semiconductor and is fixed at 3131h.

Register Name: **PCMD1**
Register Description: **PCI Command / Status Register 1**
Register Address: **0x104h**

lsb

STEPC	PARC	VGA	MWEN	SCC	MASC	MSC	IOC

Reserved (Read Only / set to all zeros)	FBBEN	PSEC

FBBCT	UDF	66 MHz	Reserved (Read Only / set to all zeros)

msb

PPE	PSE	MABT	TABTM	TABT	DTS1	DTS0	PARR

Read only bits in the PCMD1 register are indicated above by being underlined. All other bits are read-write.

The lower word (bits 0 to 15) of the PCMD1 register is the Command portion and is used for control of the PCI bus. When all bits in the lower word are set to zero, then the device is logically disconnected from the bus for all accesses except for accesses to the configuration registers. The upper word (bits 16 to 31) is the Status portion and it is used for status information. Reads to the Status portion behave normally but writes are unique in that bits can be reset (i.e. forced to zero) but not set (i.e. forced to one). A bit in the Status portion will be reset when a one is written to that bit position. Bit positions that have a zero written to them will not be reset.

COMMAND BITS

Bit 0 / I/O Space Control (IOC). This read only bit is forced to zero by the device to indicate that it does not respond to I/O Space accesses.

Bit 1 / Memory Space Control (MSC). This read/write bit controls whether or not the device will respond to accesses by the PCI bus to the memory space (which is the Local Bus). When this bit is set to zero, the device will ignore accesses attempted to the Local Bus and when set to one, the device will allow accesses to the Local Bus. This bit should be set to zero when the Local Bus is operated in the Configuration Mode. This bit is force to zero when a hardware reset is initiated via the PRST* pin.

0 = ignore accesses to the Local Bus
1 = allow accesses to the Bus

Bit 2 / Master Control (MASC). This read only bit is forced to zero by the device since it cannot act as a bus master.

Bit 3 / Special Cycle Control (SCC). This read only bit is forced to zero by the device to indicate that it cannot decode Special Cycle operations.

Bit 4 / Memory Write & Invalidate Command Enable (MWEN). This read only bit is forced to zero by the device to indicate that it cannot generate the Memory Write and Invalidate command.

Bit 5 / VGA Control (VGA). This read only bit is forced to zero by the device to indicate that it is not a VGA compatible device.

Bit 6 / Parity Error Response Control (PARC). This read/write bit controls whether or not the device should ignore parity errors. When this bit is set to zero, the device will ignore any parity errors that it detects and continue to operate normally. When this bit is set to one, the device must act on parity errors. This bit is forced to zero when a hardware reset is initiated via the PRST* pin.

0 = ignore parity errors
1 = act on parity errors

Bit 7 / Address Stepping Control (STEPC). This read only bit is forced to zero by the device to indicate that it is not capable of address/data stepping.

Bit 8 / PCI System Error Control (PSEC). This read/write bit controls whether or not the device should enable the PSERR* output pin. When this bit is set to zero, the device will disable the PSERR* pin and when this bit is set to one, the device will enable the PSERR* pin. This bit is forced to zero when a hardware reset is initiated via the PRST* pin.

0 = disable the PSERR* pin
1 = enable the PSERR* pin

Bit 9 / Fast Back-to-Back Master Enable (FBBEN). This read only bit is forced to zero by the device to indicate that it is not capable of generating fast back-to-back transactions to different agents.

Bits 10 to 15 / Reserved. These read only bits are forced to zero by the device.

STATUS BITS

The upper words in the PCMD1 register are the Status portion, which report events as they occur. As mentioned earlier, reads of the Status portion occur normally but writes are unique in that bits can only be reset (i.e. forced to zero). This occurs when a one is written to a bit position. Writes with a zero to a bit position have no affect. This allows individual bits to be reset.

Bits 16 to 20 / Reserved. These read only bits are forced to zero by the device.

Bit 21 / 66 MHz Capable (66 MHz). This read only bit is forced to zero by the device to indicate that it is not capable of running at 66 MHz.

Bit 22 / User Definable Features Capable (UDF). This read only bit is forced to zero by the device to indicate that it does not support User Definable Features.

Bit 23 / Fast Back-to-Back Capable Target (FBBCT). This read only bit is forced to one by the device to indicate that it is capable of accepting fast back-to-back transactions when the transactions are not from the same agent.

Bit 24 / PCI Parity Error Reported (PARR). This read only bit is forced to a zero by the device since the device cannot act as a bus master.

Bit 25 & 26 / Device Timing Select Bits 0 & 1 (DTS0 & DTS1). These two read only bits are forced to 01b by the device to indicate that it is capable of the medium timing requirements for the PDEVSEL* signal.

Bit 27 / Target Abort Initiated (TABT). This read/write bit will be set to a one when the device terminates a bus transaction with a target abort. This will only occur when the Local Bus is being

operated in the bus arbitration mode and the Local Bus does not have bus control when the Host requests access. This bit can be reset (set to zero) by the Host by writing a one to this bit.

Bit 28 / Target Abort Detected by Master (TABTM). This read only bit is forced to a zero by the device since the device cannot act as a bus master.

Bit 29 / Master Abort (MABT). This read only bit is forced to a zero by the device since the device cannot act as a bus master.

Bit 30 / PCI System Error Reported (PSE). This read/write bit will be set to a one when the device asserts the PSERR* signal (even if it is disabled via the PSEC Command bit). This bit can be reset (set to zero) by the Host by writing a one to this bit.

Bit 31 / PCI Parity Error Reported (PPE). This read/write bit will be set to a one when the device detects a parity error (even if parity is disabled via the PARC Command bit). This bit can be reset (set to zero) by the Host by writing a one to this bit.

Register Name: **PRCC1**
Register Description: **PCI Revision ID / Class Code Register 1**
Register Address: **0x108h**

lsb

Revision ID (Read Only / set to 00h)

Class Code (Read Only / set to 00h)

Class Code (Read Only / set to 80h)

msb

Class Code (Read Only / set to 06h)

Bits 0 to 7 / Revision ID. These read only bits identify the specific device revision and is selected by Dallas Semiconductor.

Bits 8 to 15 / Class Code Interface. These read only bits identify the sub-class interface value for the device and is fixed at 00h. See Appendix D of PCI Local Bus Specification Revision 2.1 for details.

Bits 16 to 23 / Class Code Sub-Class. These read only bits identify the sub-class value for the device and is fixed at 80h which indicates "Other Bridge Device". See Appendix D of PCI Local Bus Specification Revision 2.1 for details.

Bits 24 to 31 / Class Code Base Class. These read only bits identify the base class value for the device and are fixed at 06h, which indicate "Bridge Devices". See Appendix D of PCI Local Bus Specification Revision 2.1 for details.

Register Name: **PLTH1**
Register Description: **PCI Latency Timer / Header Type Register 1**
Register Address: **0x10Ch**

lsb

Cache Line Size (Read Only / set to 00h)

Latency Timer (Read Only / set to 00h)

Header Type (Read Only / set to 80h)

msb

BIST (Read Only / set to 00h)

Bits 0 to 7 / Cache Line Size. These read only bits are forced to zero.

Bits 8 to 15 / Latency Timer. These read only bits are forced to a zero by the device since the device cannot act as a bus master.

Bits 16 to 23 / Header Type. These read only bits are forced to 80h, which indicate a multifunction device.

Bits 24 to 31 / Built-In Self-Test (BIST). These read only bits are forced to zero.

Register Name: **PLBM**
Register Description: **PCI Local Bus Memory Base Address Register**
Register Address: **0x110h**

lsb

Base Address (Read Only / set to 0h)	PF	TYPE1	TYPE0	MSI

Base Address	Base Address (Read Only / set to 0h)

Base Address

msb

Base Address

Read only bits in the PLBM register are indicated above by being underlined. All other bits are read-write.

Bit 0 / Memory Space Indicator (MSI). This read only bit is forced to zero to indicate that the Local Bus is mapped to memory space.

Bits 1 & 2 / Type 0 & Type 1. These read only bits are forced to 00b to indicate that the Local Bus can be mapped anywhere in the 32 bit address space.

Bit 3 / Prefetchable (PF). This read only bit is forced to zero to indicate that prefetching is not supported by the device for the Local Bus.

Bits 4 to 19 / Base Address. These read only bits are forced to zero to indicate that the Local Bus requires 1m byte of memory space.

Bits 20 to 31 / Base Address. These read/write bits define the location of the 1m-byte memory space that is mapped to the Local Bus. These bits correspond to the most significant bits of the PCI address space.

Register Name: **PINTL1**
Register Description: **PCI Interrupt Line & Pin / Minimum Grant / Maximum Latency Register 1**
Register Address: **0x13Ch**

lsb

Interrupt Line

Interrupt Pin (Read Only / set to 01h)

Maximum Grant (Read Only / set to 00h)

msb

Maximum Latency (Read Only / set to 00h)

Bits 0 to 7 / Interrupt Line. These read/write bits indicate and store interrupt line routing information. The device does not use this information, it is only posted here for use by the Host.

Bits 8 to 15 / Interrupt Pin. These read only bits are forced to 01h to indicate that the uses PINTA* as an interrupt.

Bits 16 to 23 / Minimum Grant. These read only bits are forced to zero.

Bits 24 to 31 / Maximum Latency. These read only bits are forced to zero.

SECTION 10: LOCAL BUS

10.1 LOCAL BUS GENERAL DESCRIPTION

The Local Bus on the DS3131 can be either enabled or disabled. When it is disabled, the device uses the Local Bus signals to connect to Bit Synchronous HDLC Controllers on ports 28 to 39 (HDLC channels 29 to 40). The Local Bus is enabled and disabled via a hardware control signal called LBPXS. When LBPXS is left open circuited (or tied high), the Local Bus is enabled. When LBPXS is tied low, the Local Bus is disabled. See Section 1 for some diagrams detailing the possible configurations.

The Local Bus can operate in two modes, as a PCI Bridge (master mode) and as a Configuration Bus (slave mode). This selection is made in hardware by tying the LMS pin high or low. Figures 10.1A through 10.1C describe the two modes. Figure 10.1A shows an example of the Local Bus being operated in the PCI Bridge Mode. In this example the Host can access the control ports on the xDSL devices via the Local Bus.

Figure 10.1B also shows an example of the PCI Bridge Mode but in this example, the Local Bus Arbitration is enabled which allows a Local CPU to control when the Host can have access to the Local Bus. To access the Local Bus, the Host must first request the bus and then wait until it is granted.

Figure 10.1C displays an example of the Configuration Mode. In this mode, the CPU on the Local Bus will configure and monitor the DS3131. In this mode, the Host on the PCI/Custom Bus cannot access the DS3131 and the PCI/Custom Bus is only used to transfer HDLC packet data to and from the Host.

Table 10.1A lists all of the Local Bus pins and their application in both operating modes. The Local Bus operates only in a non-multiplexed fashion, it is not capable of operating as a multiplexed bus. For both operating modes, the Local Bus can be set up for either Intel or Motorola type busses. This selection is made in hardware by tying the LIM pin high or low.

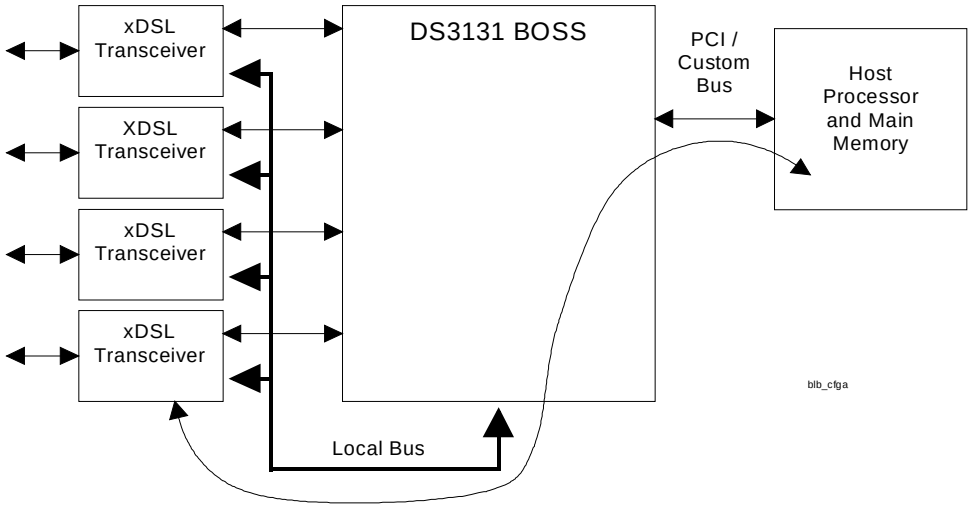
LOCAL BUS SIGNALS Table 10.1A (LBPXS Floating Or Tied High)

Signal Name	Signal Description	PCI Bridge Mode (LMS = 0)	Configuration Mode (LMS = 1)
LD[0:15]	Data Bus	Input on Read /Output on Write	Input on Write /Output on Read
LA[0:19]	Address Bus	Output	Input
LWR*(LR/W*)	Bus Write (Read/Write Select)	Output	Input
LRD*(LDS*)	Bus Read (Data Strobe)	Output	Input
LBHE*	Byte High Enable	Output	Tri-Stated
LIM	Intel/Motorola Select	Input	Input
LINT*	Interrupt	Input	Output
LMS	Mode Select	Input	Input
LCLK	Bus Clock	Output	Tri-Stated
LRDY*	Bus Ready	Input	Ignored
LCS*	Chip Select	Ignored	Input
LHOLD(LBR*)	Hold Request (Bus Request)	Output	Tri-Stated
LHLDA(LBG*)	Hold Acknowledge (Bus Grant)	Input	Ignored
LBGACK*	Bus Acknowledge	Output	Tri-Stated

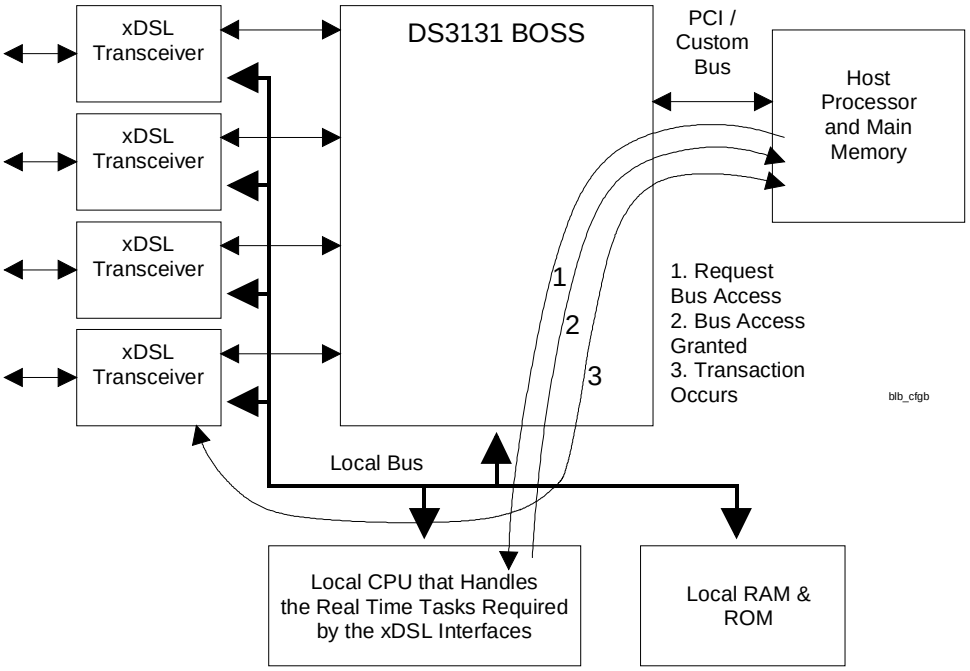
NOTES:

1. Signals shown in parenthesis ( ) are active when Motorola Mode (LIM = 1) is selected.
2. Signals suffixed with an asterisk (*) are active low signals.

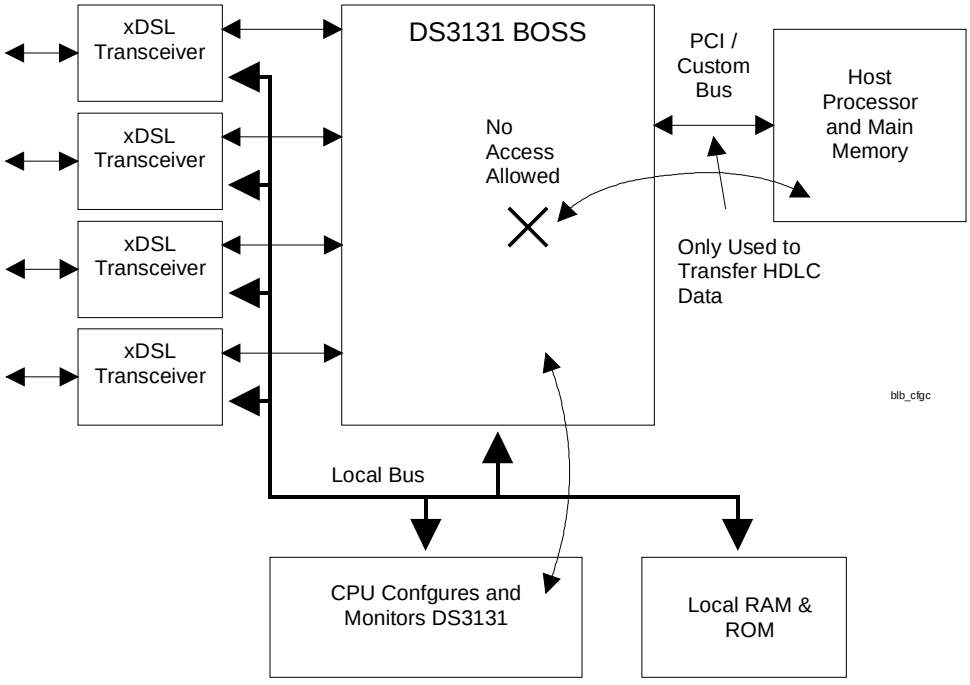
BRIDGE MODE Figure 10.1A



BRIDGE MODE WITH ARBITRATION ENABLED Figure 10.1B



CONFIGURATION MODE Figure 10.1C



PCI BRIDGE MODE

In the PCI Bridge Mode, data from the PCI bus can be transferred to the Local Bus. In this mode, the Local Bus acts as a "master" and can create all the needed signals to control the bus. In the PCI Bridge Mode, the user must configure the Local Bus Bridge Mode Control Register (LBBMC) which is described in Section 10.2.

With 20 address lines, the Local Bus can address a 1m-byte address space. The Host on the PCI bus will determine where to map this 1M byte address space within the 32-bit address space of the PCI bus by configuring the Base Address in the PCI Configuration Registers (see Section 9).

BRIDGE MODE 8 & 16 BIT ACCESS

During a bus access by the Host, the Local Bus can determine how to map the four possible byte positions from/to the PCI bus to/from the Local Bus data bus (LD) pins by examining the PCBE* signals and the Local Bus Width (LBW) control bit which resides in the Local Bus Bridge Mode Control (LBBMC) register. If the Local Bus is to be used as an 8-bit bus (LBW = 1), then the Host must only assert one of the PCBE* signals. The PCI data will be mapped to/from the LD[7:0] signal lines, the LD[15:0] signal lines remain inactive. The Local Bus Block will drive the A0 and A1 address lines according to the assertion of the PCBE* signals by the Host. See Table 10.1B for details. If the Host asserts more than one of the PCBE* signals when the Local Bus is configured as an 8-bit bus, then the Local Bus will reject the access and the PCI Block will return a Target Abort to the Host. See Section 9 for details on a Target Abort.

LOCAL BUS 8-BIT WIDTH ADDRESS / LBHE* SETTING Table 10.1B

PCBE* [3:0]	A1	A0	LBHE*
1110	0	0	1
1101	0	1	1
1011	1	0	1
0111	1	1	1

NOTE:

1. All other possible states for PCBE* will cause the device to return a Target Abort to the Host.
2. The 8-bit data picked from the PCI bus will be routed/sample to/from the LD[7:0] signal lines.
3. If no PCBE* signals are asserted during an access, a Target Abort is not return and no transaction occurs on the Local Bus.

If the Local Bus is to be used as a 16-bit bus, then the LBW control bit must be set to zero. In 16-bit accesses, by asserting the appropriate PCBE* signals (see Table 10.1C) the Host can either perform a 16-bit access or an 8-bit access. For a 16-bit access, the Host will enable the combination of either PCBE0*/PCBE1* or PCBE2*/PCBE3* and the Local Bus block will map the word from/to the PCI bus to/from the LD[15:0] signals. For an 8-bit access in the 16-bit bus mode, the Host must assert just one of the PCBE0* to PCBE3* signals. If the Host asserts a combination of PCBE* signals not supported by the Local Bus, then the Local Bus will reject the access and the PCI Block will return a Target Abort to the Host. See Section 9 for details on a Target Abort. Section 10.3 contains a number of timing examples for the Local Bus.

LOCAL BUS 16-BIT WIDTH ADDRESS / LD / LBHE* SETTING Table 10.1C

PCBE* [3:0]	8/16	A1	A0	LD[15:8]	LD[7:0]	LBHE*
1110	8	0	0		active	1
1101	8	0	1	active		0
1100	16	0	0	active	active	0
1011	8	1	0		active	1
0111	8	1	1	active		0
0011	16	1	0	active	active	0

NOTE:

1. All other possible states for PCBE* will cause the device to return a Target Abort to the Host.
2. The 16-bit data picked from the PCI bus will be routed/sample to/from the LD[7:0] & LD[15:8] signal lines as shown.
3. If no PCBE* signals are asserted during an access, a Target Abort is not return and no transaction occurs on the Local Bus.

BRIDGE MODE BUS ARBITRATION

In the Bridge Mode, the Local Bus has the ability to arbitrate for bus access. In order for this feature to operate, the Host must access the PCI Bridge Mode Control Register (LBBMC) and enable it via the LARBE control bit (the default is bus arbitration disabled). If bus arbitration is enabled, then before a bus transaction can occur, the Local Bus will first request bus access by asserting the LHOLD (LBR*) signal and then wait for the bus to be granted from the Local Bus arbiter by sensing that the LHLDA (LBG*) has been asserted. If the Host on the PCI Bus attempts a Local Bus access when the Local Bus is not granted by the Local Bus master (LBGACK* is deasserted), then the Local Bus block will immediately inform the Host that the Local Bus is busy and cannot be accessed at this time (in other words, come back later) by issuing a PCI Target Retry. See Section 9 for details on the PCI Target Retry. When this happens, the Local Bus block will not attempt the bus access and will keep the LA, LD, LBHE*, LWR*(LR/W*), and LRD*(LDS*) signals tri-stated.

If the Host attempts a Local Bus access when the bus is busy, the Local Bus block will go ahead and request bus access and after it has been granted, it will seize the bus for the time programmed into the Local Bus Arbitration Timer (LAT0 to LAT3 in the LBBMC register) which can be from 32 to 1048576 clocks. As long as the local bus has been granted and the arbitration timer has at least 16 clocks left, then the Host is allowed to access the Local Bus. See Figure 10.1D and the timing examples in Section 10.3 for more details.

BRIDGE MODE BUS TRANSACTION TIMING

When the Local Bus is operated in PCI Bridge Mode, the bus transaction time can be determined either from an external ready signal (LRDY*) or from the PCI Bridge Mode Control Register (LBBMC) which will allow a bus transaction time of 1 to 11 LCLK cycles. If the total access time to the Local Bus exceeds 16 PCLK cycles, the PCI access will time out and a PCI Target Retry will be sent to the Host. This will only occur when LRDY* has not been detected within 9 clocks. If this happens, the Local Bus Error (LBE) status bit in the Status Master (SM) register will be set. Additional details on the LBE status bit can be found in Section 4 and more details on transaction timing can be found in Figure 10.1D and the timing examples in Section 10.3.

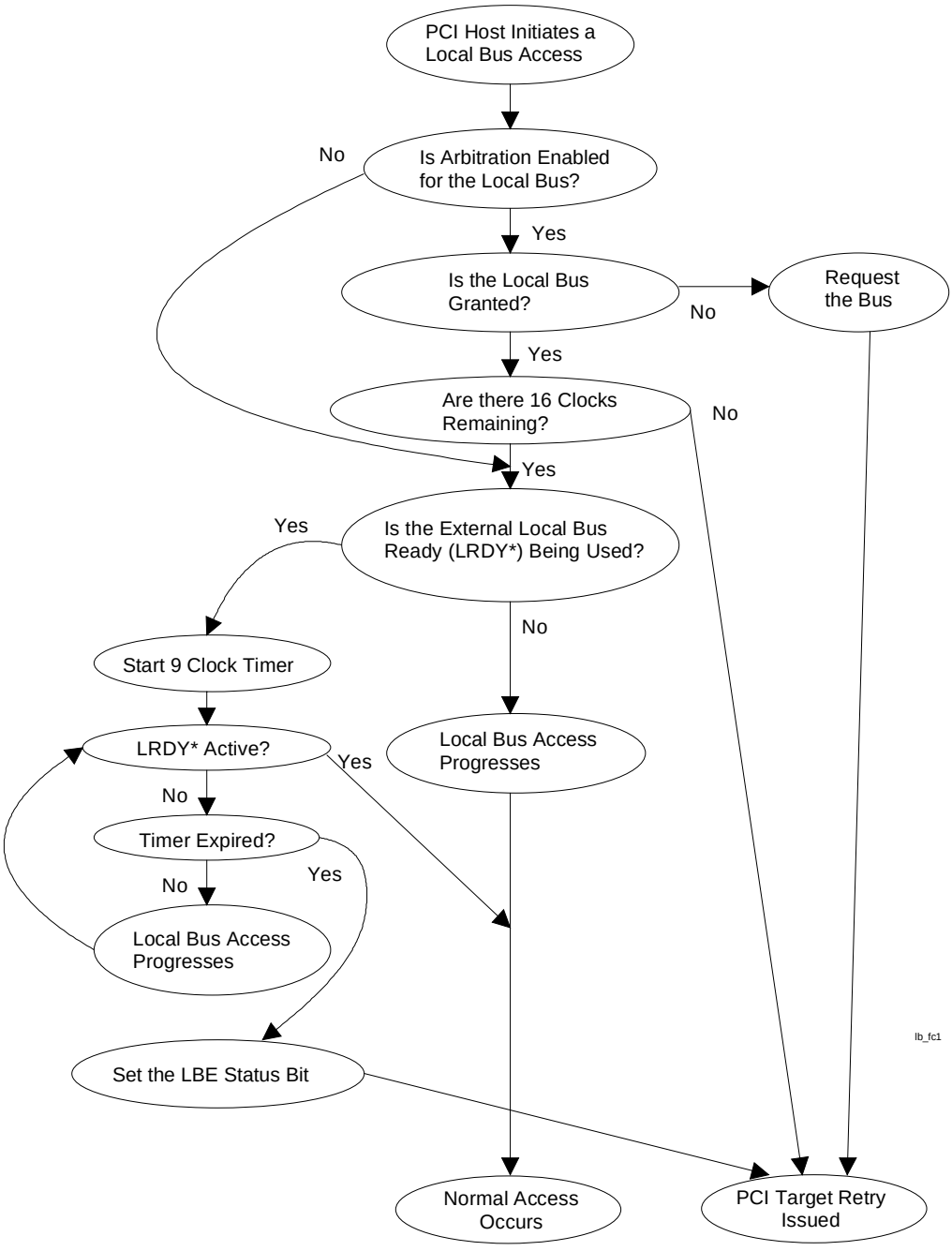
BRIDGE MODE INTERRUPT

In the PCI Bridge mode, the Local Bus can detect an external interrupt via the LINT* signal. If the Local Bus detects that the LINTA* signal has been asserted, then it will set the LBINT status bit in the Status Master (SM) register. The setting of this status bit can cause a hardware interrupt to occur at the PCI bus via the PINTA* signal. This interrupt can be masked via the ISM register. See Section 4 for more details.

CONFIGURATION MODE

In the Configuration Mode, the Local Bus is used only to configure the device and obtain status information from the device. It is also used to configure the PCI Configuration Registers and hence the PCI Bus signal PIDSEL is disabled when the Local Bus is in the Configuration Mode. Data cannot be passed from the Local Bus to the PCI bus in this mode. The PCI bus will only be used as a high speed I/O bus for the HDLC packet data. In this mode, bus arbitration, bus format, and the user settable bus transaction time features are disabled. In the Configuration Mode, all bus accesses are based on 16-bit addresses and 16-bit data. The upper four addresses (LA[19:16]) are ignored and 8-bit data accesses are not allowed. See Section 12 for details on the AC timing requirements.

LOCAL BUS ACCESS FLOWCHART Figure 10.1D



10.2 LOCAL BUS BRIDGE MODE CONTROL REGISTER DESCRIPTION

Register Name: **LBBMC**
Register Description: **Local Bus Bridge Mode Control Register**
Register Address: **0040h**

Note: This register can only be accessed via the PCI Bus and hence only in the PCI Bridge Mode. In the Configuration Mode, this register cannot be accessed. It will be set to all zeros upon a hardware reset issued via the PRST* pin. It will not be affected by a software reset issued via the RST control bit in the Master Reset and ID (MRID) register.

7	6	5	4	3	2	1	0
n/a	LBW	LRDY3	LRDY2	LRDY1	LRDY0	LARBE	LCLKE
15	14	13	12	11	10	9	8
n/a	n/a	n/a	n/a	LAT3	LAT2	LAT1	LAT0

Note: bits that are underlined are read only, all other bits are read-write; default value for all bits is 0.

Bit 0 / Local Bus Clock Enable (LCLKE).
0 = tri-state the LCLK output signal pin
1 = allow LCLK to appear at the pin

Bit 1 / Local Bus Arbitration Enable (LARBE). When enabled, the LHOLD (LBR*), LBGACK*, and LHLDA (LBG*) signal pins are active and the proper arbitration handshake sequence must occur for a proper bus transaction. When disabled, the LHOLD (LBR*), LBGACK* and LHLDA (LBG*) signal pins are deactivated and bus arbitration on the Local Bus is not invoked. Also the Arbitration Timer is enabled (see the description of the LAT0 to LAT3 bits) when LARBE is set to a one.
0 = Local Bus Arbitration is disabled
1 = Local Bus Arbitration is enabled

Bit 2 / Local Bus Ready Control Bit 0 (LRDY0). Lsb

Bit 3 / Local Bus Ready Control Bit 1 (LRDY1).

Bit 4 / Local Bus Ready Control Bit 2 (LRDY2).

Bit 5 / Local Bus Ready Control Bit 3 (LRDY3). msb
These control bits determine the duration of the Local Bus transaction in the PCI Bridge Mode. The bus transaction can either be control via the external LRDY* input signal or via a predetermined period of 1 to 11 LCLK periods.

0000 = use the LRDY* signal input pin to control the bus transaction
0001 = bus transaction is defined as **1** LCLK period
0010 = bus transaction is defined as **2** LCLK periods
0011 = bus transaction is defined as **3** LCLK periods
0100 = bus transaction is defined as **4** LCLK periods
0101 = bus transaction is defined as **5** LCLK periods
0110 = bus transaction is defined as **6** LCLK periods
0111 = bus transaction is defined as **7** LCLK periods
1000 = bus transaction is defined as **8** LCLK periods
1001 = bus transaction is defined as **9** LCLK periods
1010 = bus transaction is defined as **10** LCLK periods
1011 = bus transaction is defined as **11** LCLK periods
1100 = illegal state
1101 = illegal state
1110 = illegal state
1111 = illegal state

Bit 6 / Local Bus Width (LBW).
0 = 16 bits
1 = 8 bits

Bits 8 to 11 / Local Bus Arbitration Timer Setting (LAT0 to LAT3). These four bits determine the total time the Local Bus will seize the bus when it has been granted in the Arbitration Mode (LARBE = 1). This period is measured from LHLDA (LBG*) being detected to LBGACK* inactive.

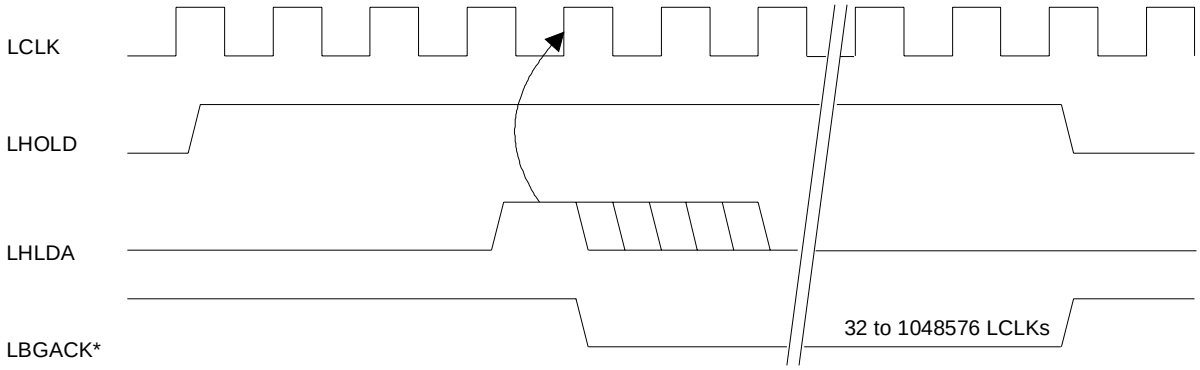
	33 MHz	**25 MHz**
0000 = when granted, hold the bus for 32 LCLKs	0.97 µs	1.3 µs
0001 = when granted, hold the bus for 64 LCLKs	1.9 µs	2.6 µs
0010 = when granted, hold the bus for 128 LCLKs	3.9 µs	5.1 µs
0011 = when granted, hold the bus for 256 LCLKs	7.8 µs	10.2 µs
0100 = when granted, hold the bus for 512 LCLKs	15.5 µs	20.5 µs
1101 = when granted, hold the bus for 262144 LCLKs	7.9 ms	10.5 ms
1110 = when granted, hold the bus for 524288 LCLKs	15.9 ms	21.0 ms
1111 = when granted, hold the bus for 1048576 LCLKs	31.8 ms	41.9 ms

10.3 EXAMPLES OF BUS TIMING FOR LOCAL BUS PCI BRIDGE MODE OPERATION

Figure 10.3A

8-Bit Read Cycle
Intel Mode (LIM = 0)
Arbitration Enabled (LARBE = 1)
Bus Transaction Time = 4 LCLK (LRDY = 0100)

An attempted access by the Host causes the Local Bus to request the bus. If bus access has not been granted (LBGACK* deasserted), then the timing shown at the top of the page will occur with LHOLD being asserted and then once LHLDA is detected, the Local Bus will grab the bus for 32 to 1048576 clocks and then release it. If the bus has already been granted (LBGACK* asserted), then the timing shown at the bottom of the page will occur.



Note: LA / LD / LBHE* / LWR* / LRD* are tri-stated.

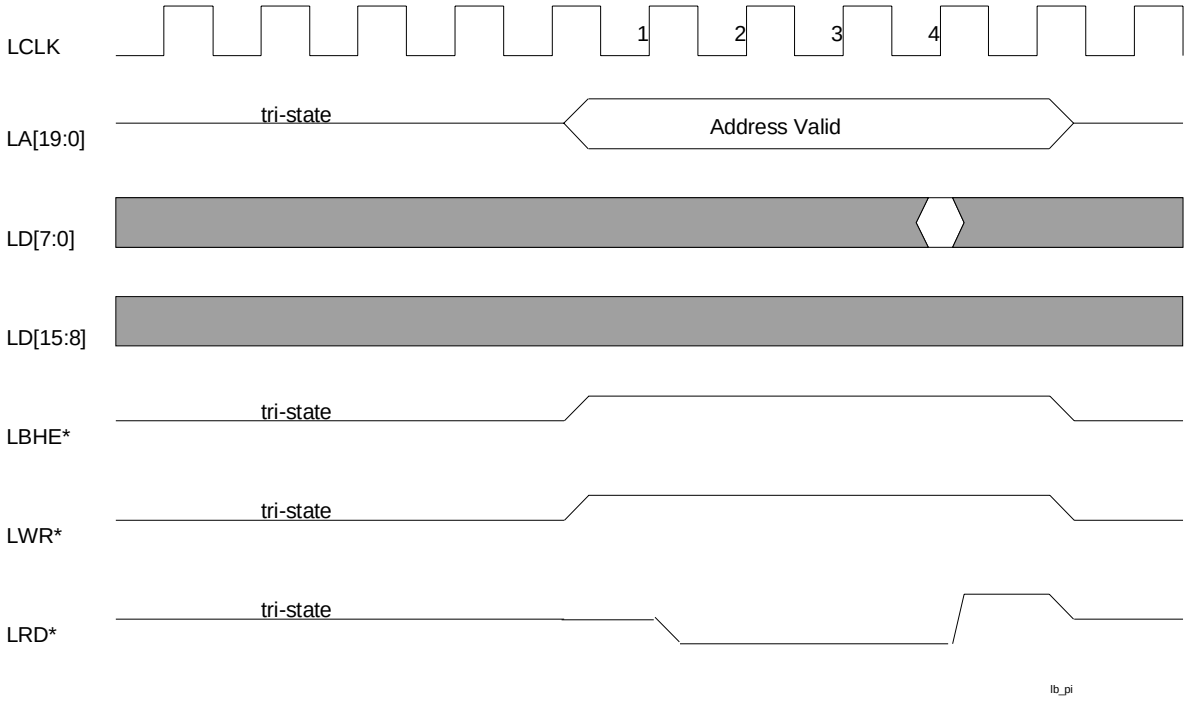
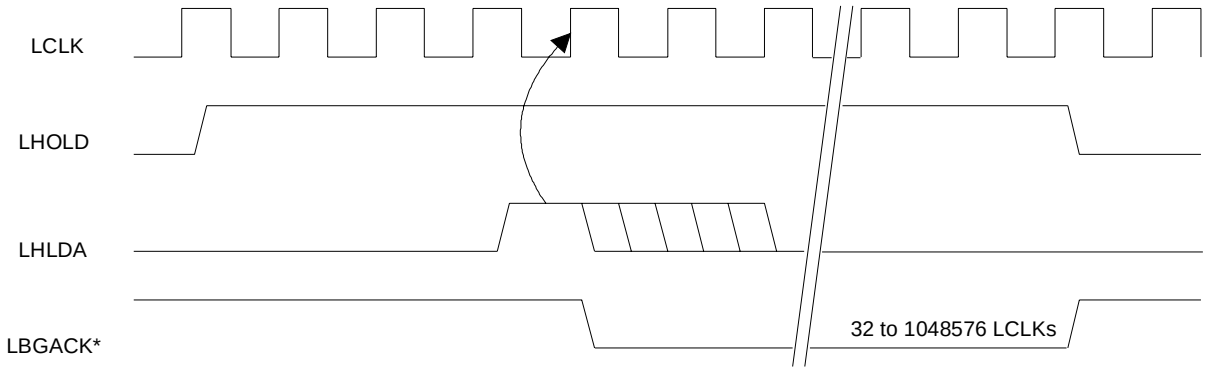


Figure 10.3B

16-Bit Write Cycle
Intel Mode (LIM = 0)
Arbitration Enabled (LARBE = 1)
Bus Transaction Time = 4 LCLK (LRDY = 0100)

An attempted access by the Host causes the Local Bus to request the bus. If bus access has not been granted (LBGACK* deasserted), then the timing shown at the top of the page will occur with LHOLD being asserted and then once LHLDA is detected, the Local Bus will grab the bus for 32 to 1048576 clocks and then release it. If the bus has already been granted (LBGACK* asserted), then the timing shown at the bottom of the page will occur.



Note: LA / LD / LBHE* / LWR* / LRD* are tri-stated.

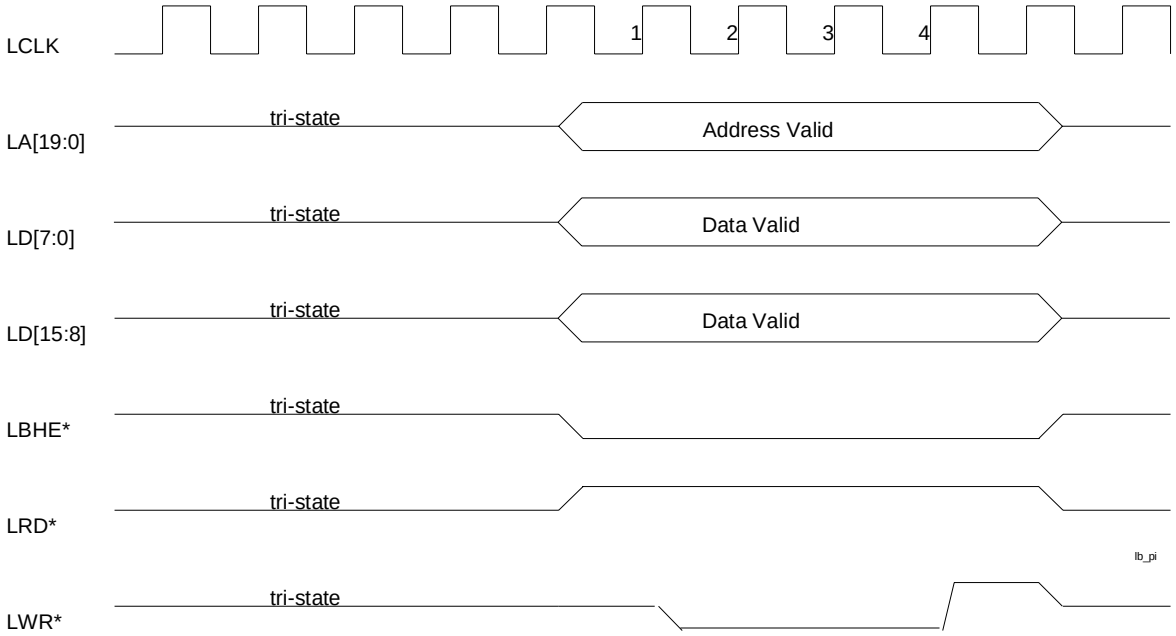
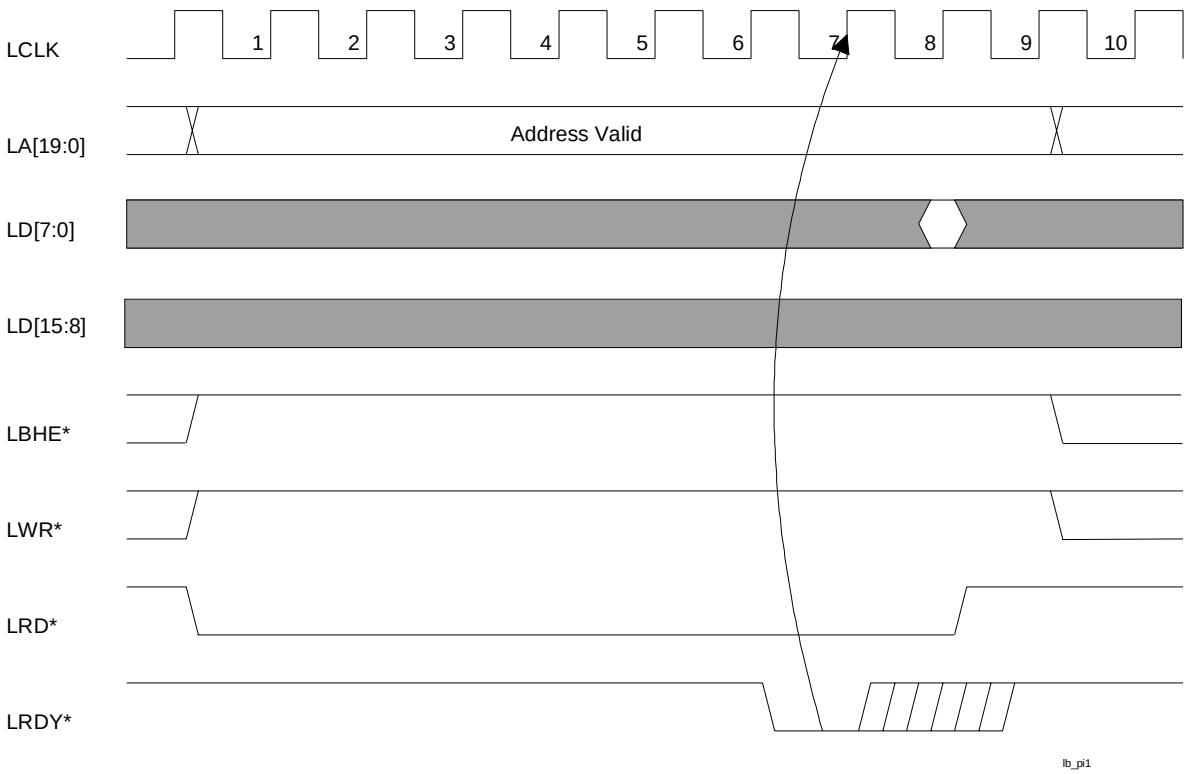


Figure 10.3C

8-Bit Read Cycle
Intel Mode (LIM = 0)
Arbitration Disabled (LARBE = 0)
Bus Transaction Time = Timed from LRDY* (LRDY = 0000)

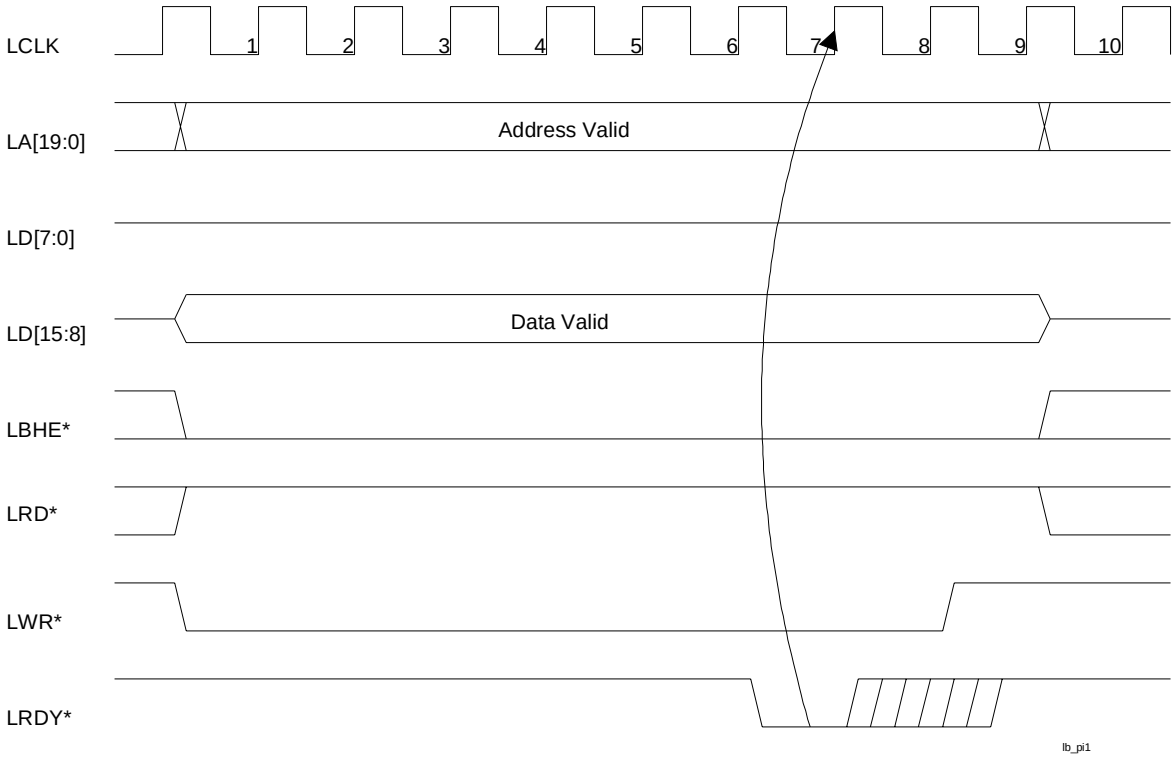


NOTE:

The LRDY* signal must be detected by the 9th LCLK or the bus access attempted by the Host will be unsuccessful and the LBE status bit will be set.

Figure 10.3D

16-Bit Write (only upper 8-bits active) Cycle
Intel Mode (LIM = 0)
Arbitration Disabled (LARBE = 0)
Bus Transaction Time = Timed from LRDY* (LRDY = 0000)



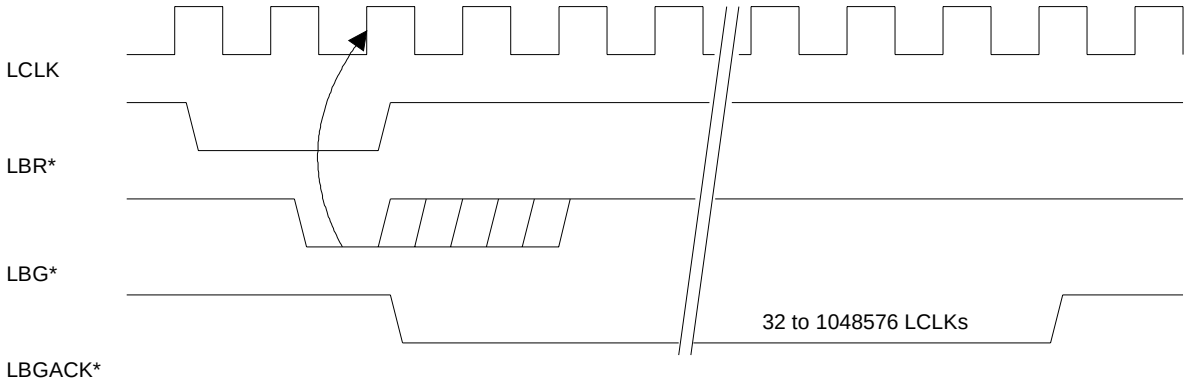
NOTE:

The LRDY* signal must be detected by the 9th LCLK or the bus access attempted by the Host will be unsuccessful and the LBE status bit will be set.

Figure 10.3E

8-Bit Read Cycle
Motorola Mode (LIM = 1)
Arbitration Enabled (LARBE = 1)
Bus Transaction Time = 6 LCLK (LRDY = 0110)

An attempted access by the Host causes the Local Bus to request the bus. If bus access has not been granted (LBGACK* deasserted), then the timing shown at the top of the page will occur with LBR* being asserted and then once LBG* is detected, the Local Bus will grab the bus for 32 to 1048576 clocks and then release it. If the bus has already been granted (LBGACK* asserted), then the timing shown at the bottom of the page will occur.



Note: LA / LD / LBHE* / LDS* / LR/W* are tri-stated.

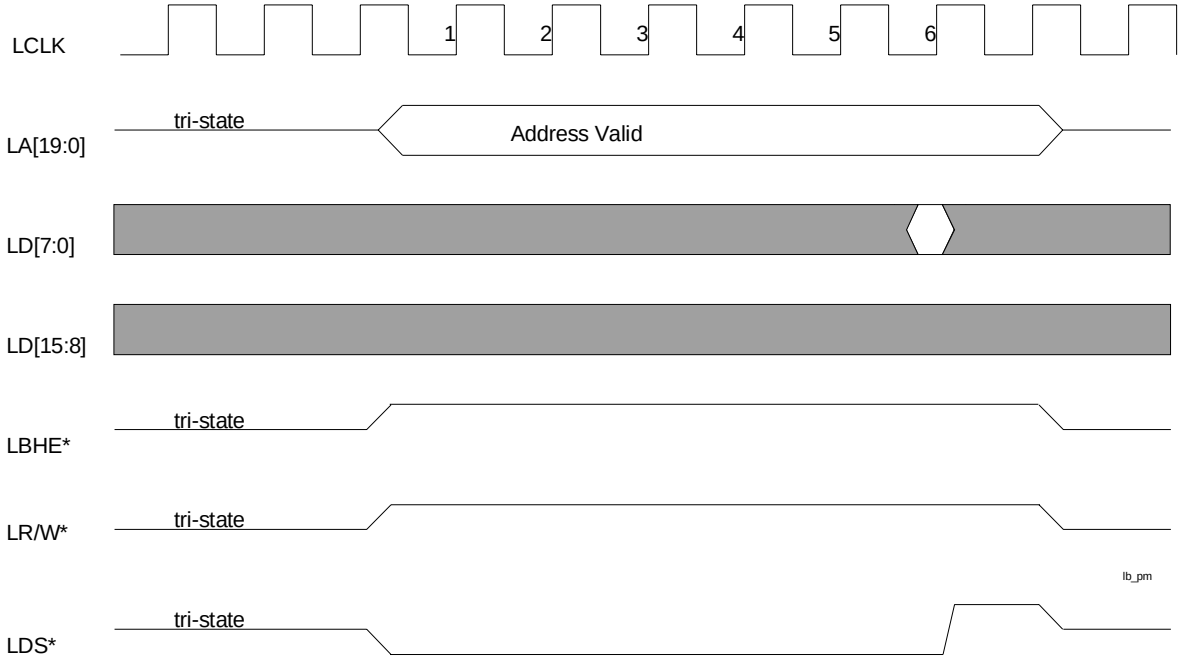
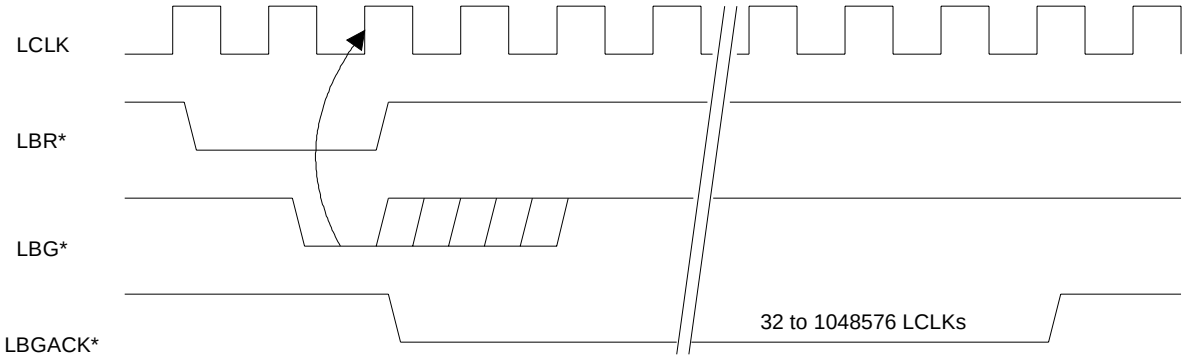


Figure 10.3F

8-Bit Write Cycle
Motorola Mode (LIM = 1)
Arbitration Enabled (LARBE = 1)
Bus Transaction Time = 6 LCLK (LRDY = 0110)

An attempted access by the Host causes the Local Bus to request the bus. If bus access has not been granted (LBGACK* deasserted), then the timing shown at the top of the page will occur with LBR* being asserted and then once LBG* is detected, the Local Bus will grab the bus for 32 to 1048576 clocks and then release it. If the bus has already been granted (LBGACK* asserted), then the timing shown at the bottom of the page will occur.



Note: LA / LD / LBHE* / LDS* / LR/W* are tri-stated.

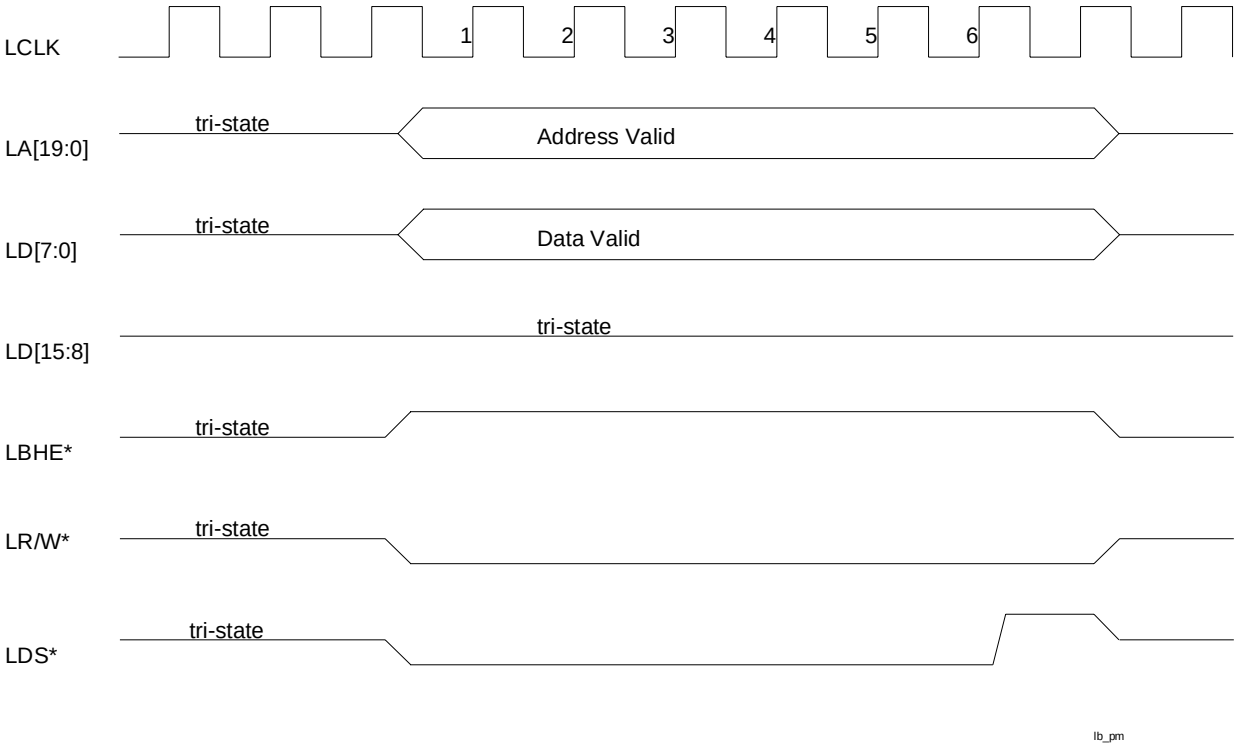
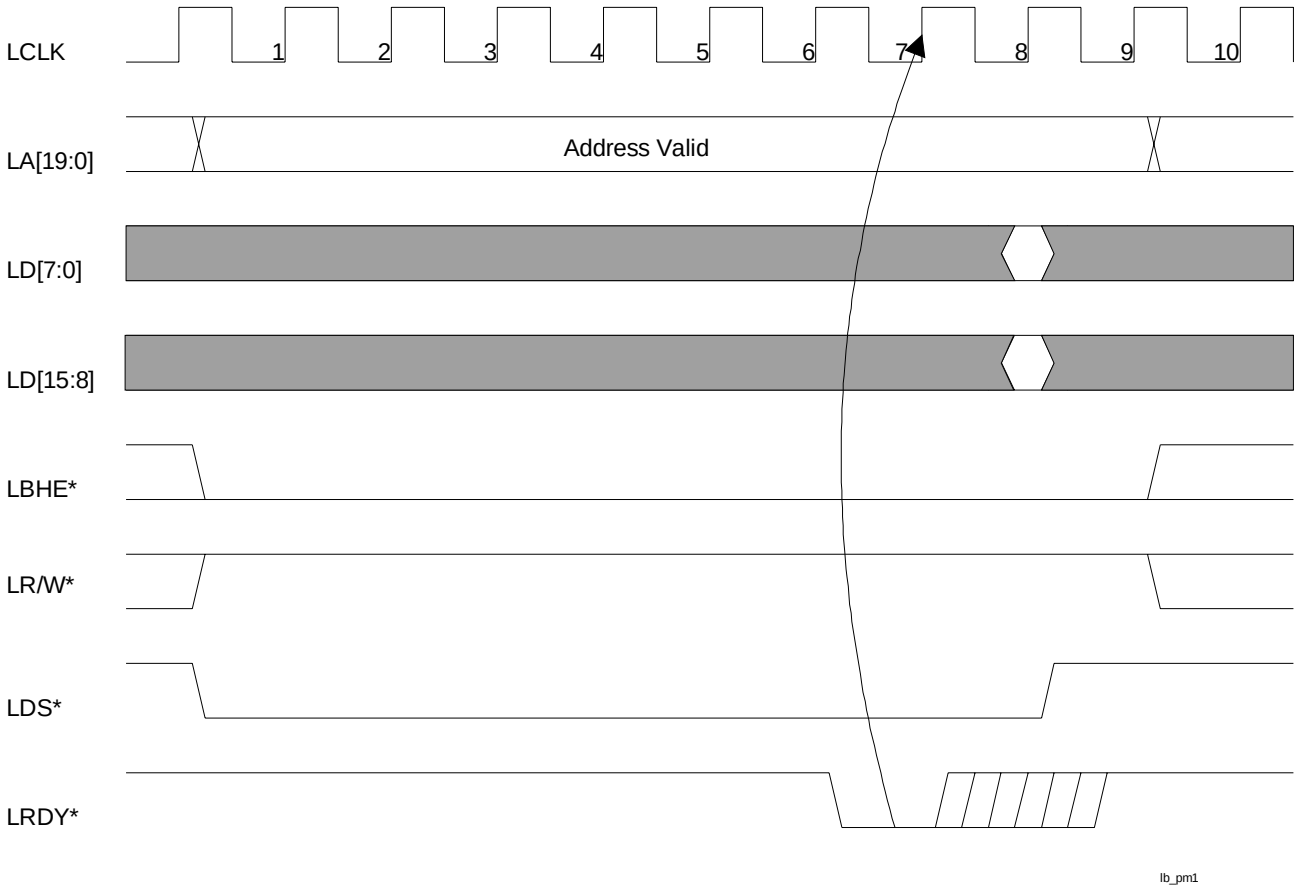


Figure 10.3G

16-Bit Read Cycle
Motorola Mode (LIM = 1)
Arbitration Disabled (LARBE = 0)
Bus Transaction Time = Timed from LRDY* (LRDY = 0000)

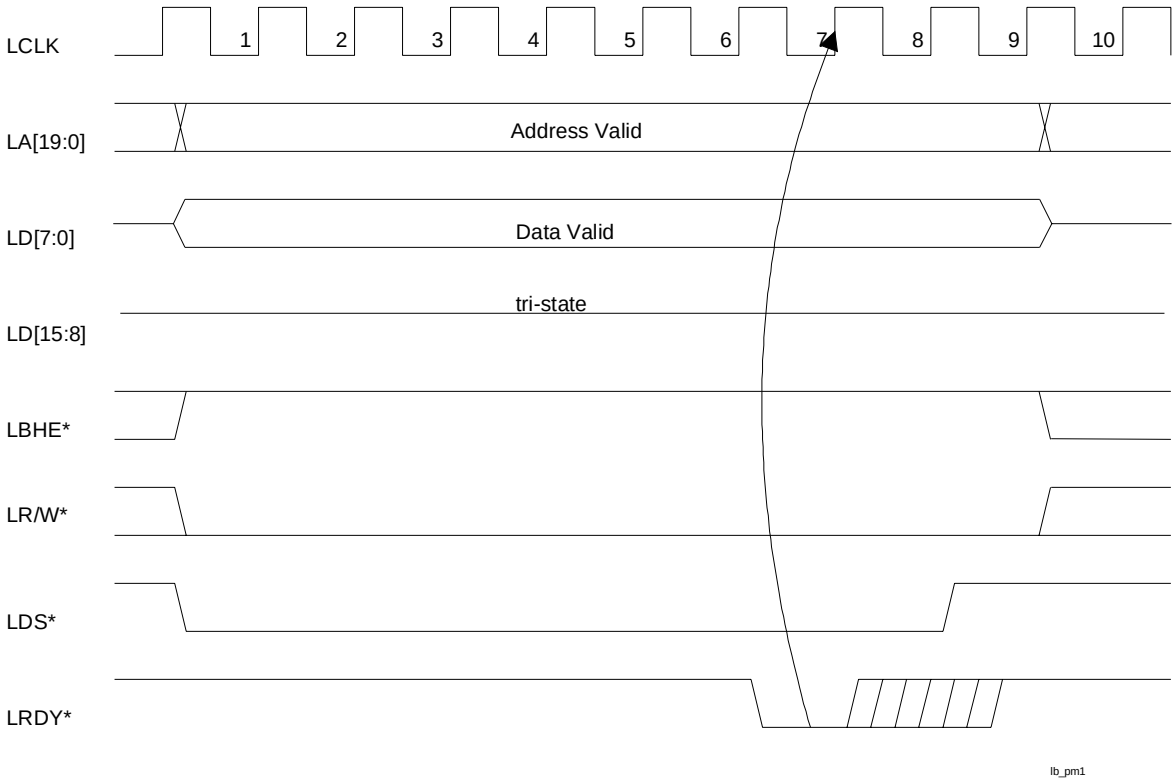


NOTE:

The LRDY* signal must be detected by the 9th LCLK or the bus access attempted by the Host will be unsuccessful and the LBE status bit will be set.

Figure 10.3H

8-Bit Write Cycle
Motorola Mode (LIM = 1)
Arbitration Disabled (LARBE = 0)
Bus Transaction Time = Timed from LRDY* (LRDY = 0000)



NOTE:

The LRDY* signal must be detected by the 9th LCLK or the bus access attempted by the Host will be unsuccessful and the LBE status bit will be set.

SECTION 11: JTAG

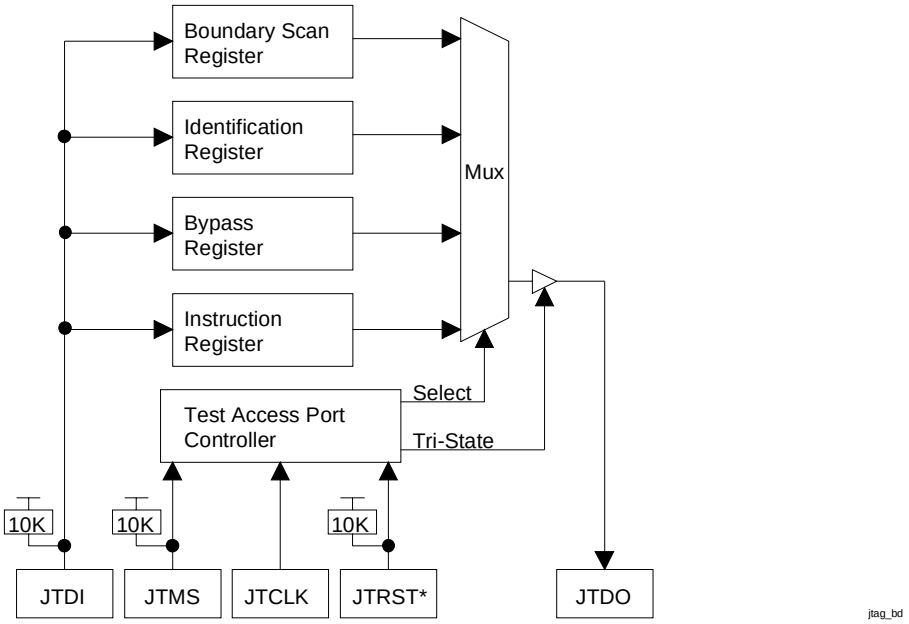
11.1 JTAG DESCRIPTION

The DS3131 device supports the standard instruction codes SAMPLE/PRELOAD, BYPASS, and EXTEST. Optional public instructions included are HIGHZ, CLAMP, and IDCODE. See Figure 11.1A for a Block Diagram. The DS3131 contains the following items, which meet the requirements, set by the IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture:

Test Access Port (TAP)
TAP Controller
Instruction Register
Bypass Register
Boundary Scan Register
Device Identification Register.

The Test Access Port has the necessary interface pins, namely JTCLK, JTRST*, JTDI, JTDO, and JTMS. Details on these pins can be found in Section 2.4. Details on the Boundary Scan Architecture and the Test Access Port can be found in IEEE 1149.1-1990, IEEE 1149.1a-1993, and IEEE 1149.1b-1994.

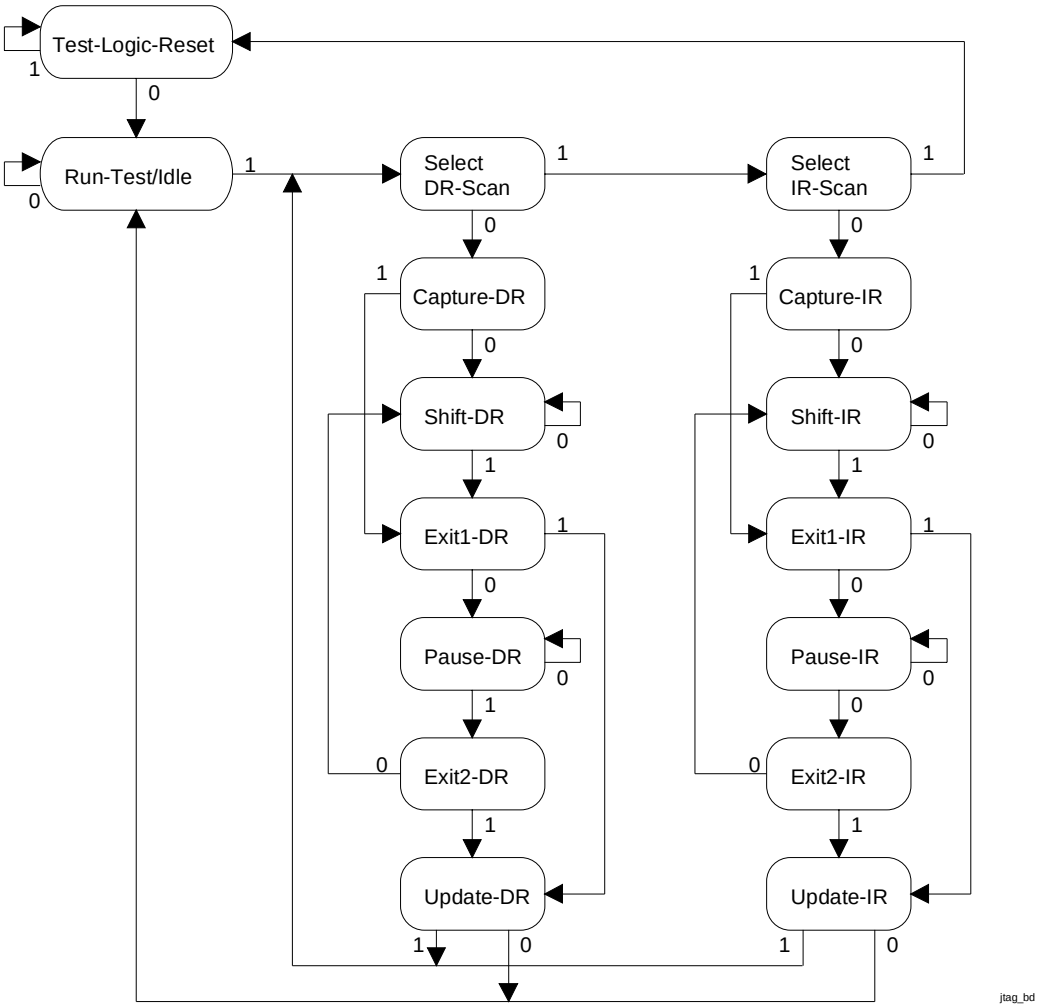
JTAG BLOCK DIAGRAM Figure 11.1A



11.2 TAP CONTROLLER STATE MACHINE DESCRIPTION

This section covers the details on the operation of the Test Access Port (TAP) Controller State Machine. Please see Figure 11.2A for details on each of the states described below. The TAP controller is a finite state machine, which responds to the logic level at JTMS on the rising edge of JTCLK.

TAP CONTROLLER STATE MACHINE Figure 11.2A



TEST-LOGIC-RESET

Upon power-up of the DS3131, the TAP controller will be in the Test-Logic-Reset state. The Instruction register will contain the IDCODE instruction. All system logic on the DS3131 will operate normally.

RUN-TEST-IDLE

Run-Test-Idle is used between scan operations or during specific tests. The Instruction register and Test register will remain idle.

SELECT-DR-SCAN

All test registers retain their previous state. With JTMS low, a rising edge of JTCLK moves the controller into the Capture-DR state and will initiate a scan sequence. JTMS high moves the controller to the Select-IR-SCAN state.

CAPTURE-DR

Data may be parallel loaded into the Test Data registers selected by the current instruction. If the instruction does not call for a parallel load or the selected register does not allow parallel loads, the Test register will remain at its current value. On the rising edge of JTCLK, the controller will go to the Shift-DR state if JTMS is low or it will go to the Exit1-DR state if JTMS is high.

SHIFT-DR

The Test Data register selected by the current instruction will be connected between JTDI and JTDO and will shift data one stage towards its serial output on each rising edge of JTCLK. If a Test register selected by the current instruction is not placed in the serial path, it will maintain its previous state.

EXIT1-DR

While in this state, a rising edge on JTCLK with JTMS high will put the controller in the Update-DR state which terminates the scanning process. A rising edge on JTCLK with JTMS low will put the controller in the Pause-DR state.

PAUSE-DR

Shifting of the Test registers is halted while in this state. All Test registers selected by the current instruction will retain their previous state. The controller will remain in this state while JTMS is low. A rising edge on JTCLK with JTMS high will put the controller in the Exit2-DR state.

EXIT2-DR

While in this state, a rising edge on JTCLK with JTMS high will put the controller in the Update-DR state and terminate the scanning process. A rising edge on JTCLK with JTMS low will enter the Shift-DR state.

UPDATE-DR

A falling edge on JTCLK while in the Update-DR state will latch the data from the shift register path of the Test registers into the data output latches. This prevents changes at the parallel output due to changes in the shift register. A rising edge on JTCLK with JTMS low, will put the controller in the Run-Test-Idle state. With JTMS high, the controller will enter the Select-DR-Scan state.

SELECT-IR-SCAN

All Test registers retain their previous state. The Instruction register will remain unchanged during this state. With JTMS low, a rising edge on JTCLK moves the controller into the Capture-IR state and will initiate a scan sequence for the Instruction register. JTMS high during a rising edge on JTCLK puts the controller back into the Test-Logic-Reset state.

CAPTURE-IR

The Capture-IR state is used to load the shift register in the Instruction register with a fixed value. This value is loaded on the rising edge of JTCLK. If JTMS is high on the rising edge of JTCLK, the controller will enter the Exit1-IR state. If JTMS is low on the rising edge of JTCLK, the controller will enter the Shift-IR state.

SHIFT-IR

In this state, the shift register in the Instruction register is connected between JTDI and JTDO and shifts data one stage for every rising edge of JTCLK towards the serial output. The parallel registers, as well as all Tests registers remain at their previous states. A rising edge on JTCLK with JTMS high will move the controller to the Exit1-IR state. A rising edge on JTCLK with JTMS low will keep the controller in the Shift-IR state while moving data one stage through the Instruction shift register.

EXIT1-IR
A rising edge on JTCLK with JTMS low will put the controller in the Pause-IR state. If JTMS is high on the rising edge of JTCLK, the controller will enter the Update-IR state and terminate the scanning process.

PAUSE-IR
Shifting of the Instruction register is halted temporarily. With JTMS high, a rising edge on JTCLK will put the controller in the Exit2-IR state. The controller will remain in the Pause-IR state if JTMS is low during a rising edge on JTCLK.

EXIT2-IR
A rising edge on JTCLK with JTMS low will put the controller in the Update-IR state. The controller will loop back to the Shift-IR state if JTMS is high during a rising edge of JTCLK in this state.

UPDATE-IR
The instruction shifted into the Instruction shift register is latched into the parallel output on the falling edge of JTCLK as the controller enters this state. Once latched, this instruction becomes the current instruction. A rising edge on JTCLK with JTMS low will put the controller in the Run-Test-Idle state. With JTMS high, the controller will enter the Select-DR-Scan state.

11.3 INSTRUCTION REGISTER AND INSTRUCTIONS
The Instruction register contains a shift register as well as a latched parallel output and is 3 bits in length. When the TAP controller enters the Shift-IR state, the instruction shift register will be connected between JTDI and JTDO. While in the Shift-IR state, a rising edge on JTCLK with JTMS low will shift data one stage towards the serial output at JTDO. A rising edge on JTCLK in the Exit1-IR state or the Exit2-IR state with JTMS high will move the controller to the Update-IR state. The falling edge of that same JTCLK will latch the data in the instruction shift register to the instruction parallel output. Instructions supported by the DS3131 and their respective operational binary codes are shown in Table 11.3A.

INSTRUCTION CODES Table 11.3A

Instructions	Selected Register	Instruction Codes
SAMPLE/PRELOAD	Boundary Scan	010
BYPASS	Bypass	111
EXTEST	Boundary Scan	000
CLAMP	Boundary Scan	011
HIGHZ	Boundary Scan	100
IDCODE	Device Identification	001

SAMPLE/PRELOAD
A mandatory instruction for the IEEE 1149.1 specification. This instruction supports two functions. The digital I/Os of the DS3131 can be sampled at the Boundary Scan register without interfering with the normal operation of the device by using the Capture-DR state. SAMPLE/PRELOAD also allows the DS3131 to shift data into the Boundary Scan register via JTDI using the Shift-DR state.

EXTEST

EXTEST allows testing of all interconnections to the DS3131. When the EXTEST instruction is latched in the instruction register, the following actions occur. Once enabled via the Update-IR state, the parallel outputs of all digital output pins will be driven. The Boundary Scan register will be connected between JTDI and JTDO. The Capture-DR will sample all digital inputs into the Boundary Scan register.

BYPASS

When the BYPASS instruction is latched into the parallel Instruction register, JTDI connects to JTDO through the one-bit Bypass Test register. This allows data to pass from JTDI to JTDO not affecting the device's normal operation.

IDCODE

When the IDCODE instruction is latched into the parallel Instruction register, the Identification Test register is selected. The device identification code will be loaded into the Identification register on the rising edge of JTCLK following entry into the Capture-DR state. Shift-DR can be used to shift the identification code out serially via JTDO. During Test-Logic-Reset, the identification code is forced into the instruction register's parallel output. The device ID code will always have a one in the LSB position. The next 11 bits identify the manufacturer's JEDEC number and number of continuation bytes followed by 16 bits for the device and 4 bits for the version. The device ID code for the DS3131 is **00008143h**.

11.4 TEST REGISTERS

IEEE 1149.1 requires a minimum of two Test registers; the Bypass register and the Boundary Scan register. An optional Test register has been included in the DS3131 design. This Test register is the Identification register and is used in conjunction with the IDCODE instruction and the Test-Logic-Reset state of the TAP controller.

BYPASS REGISTER

This is a single one-bit shift register used in conjunction with the BYPASS, CLAMP, and HIGHZ instructions, which provides a short path between JTDI and JTDO.

IDENTIFICATION REGISTER

The Identification register contains a 32-bit shift register and a 32-bit latched parallel output. This register is selected during the IDCODE instruction and when the TAP controller is in the Test-Logic-Reset state.

BOUNDARY SCAN REGISTER

This register contains both a shift register path and a latched parallel output for all control cells and digital I/O cells and is TBD bits in length. Table 11.4A shows all of the cell bit locations and definitions.

BOUNDARY SCAN CONTROL BITS Table 11.4A

To be completed.

SECTION 12: AC CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Lead with Respect to VSS (except VDD)	-0.3V to 5.5V
Supply Voltage (VDD) with Respect to VSS	-0.3V to 3.63V
Operating Temperature	0C to +70C
Storage Temperature	-55C to +125C
Soldering Temperature	See J-STD-020A Specification

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

Note: The typical values listed below are not production tested.

RECOMMEND DC OPERATING CONDITIONS (0ºC TO +70ºC)

Parameter	Symbol	Min	Typ	Max	Units	Notes
Logic 1	VIH	2.0		5.5	V	
Logic 1 (Schmitt Input for PCLK)	VIHS	1.7		5.5	V	
Logic 0	VIL	-0.3		0.8	V	
Logic 0 (Schmitt Input for PCLK)	VILS	-0.3		0.7	V	
Supply	VDD	3.0		3.6	V	

DC CHARACTERISTICS (0ºC to +70ºC; VDD = 3.0V to 3.6V)

Parameter	Symbol	Min	Typ	Max	Units	Notes
Supply Current @ VDD = 3.6V	IDD			TBD	mA	1
Lead Capacitance	CIO		7		pF	
Schmitt Hysteresis	VTH		0.6		V	
Input Leakage	IIL	-10		+10	µA	2
Input Leakage (w/ pull-ups)	IILP	-500		+500	µA	2
Output Leakage	ILO	-10		+10	µA	3
Output Current (2.4V)	IOH	-4.0			mA	
Output Current (0.4V)	IOL	+4.0			mA	

NOTES:

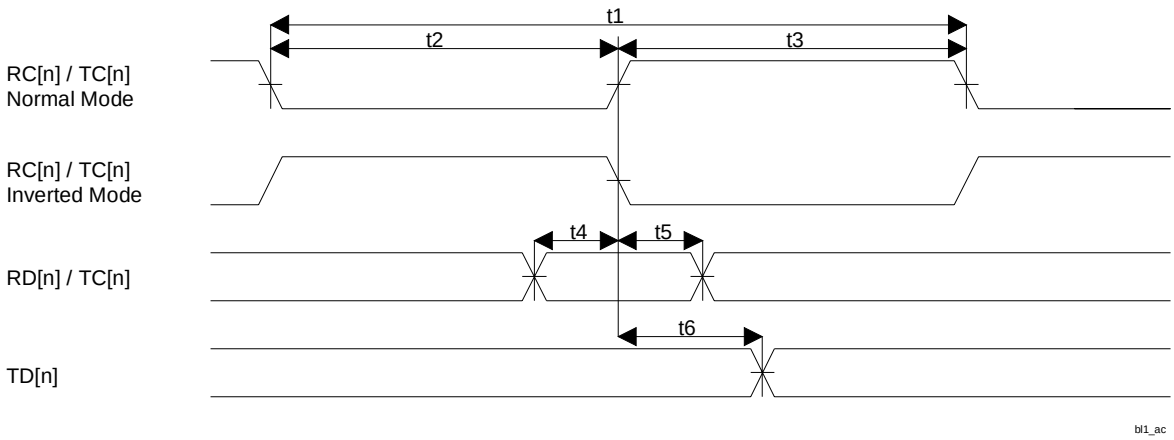
1. RC0 to RC39 and TC0 to TC39 = 2.048 MHz / PCLK = 33 MHz / other inputs at VDD or grounded / other outputs left open circuited.
2. 0V < Vin < VDD.
3. Outputs in Tri-State.

AC CHARACTERISTICS - LAYER ONE PORTS

(0ºC TO +70ºC; VDD = 3.0V TO 3.6V)

Parameter	Symbol	Min	Typ	Max	Units	Notes
RC / TC Clock Period	t1	19			ns	
RC / TC Clock Low Time	t2	8			ns	
RC / TC Clock High Time	t3	8			ns	
RD Set Up Time to the Falling Edge or Rising Edge of RC	t4	2			ns	
RD Hold Time from the Falling Edge or Rising Edge of RC	t5	1			ns	
Delay from the Rising Edge or Falling Edge of TC to Data Valid on TD	t6	3		10	ns	

LAYER ONE PORT AC TIMING DIAGRAM Figure 12A

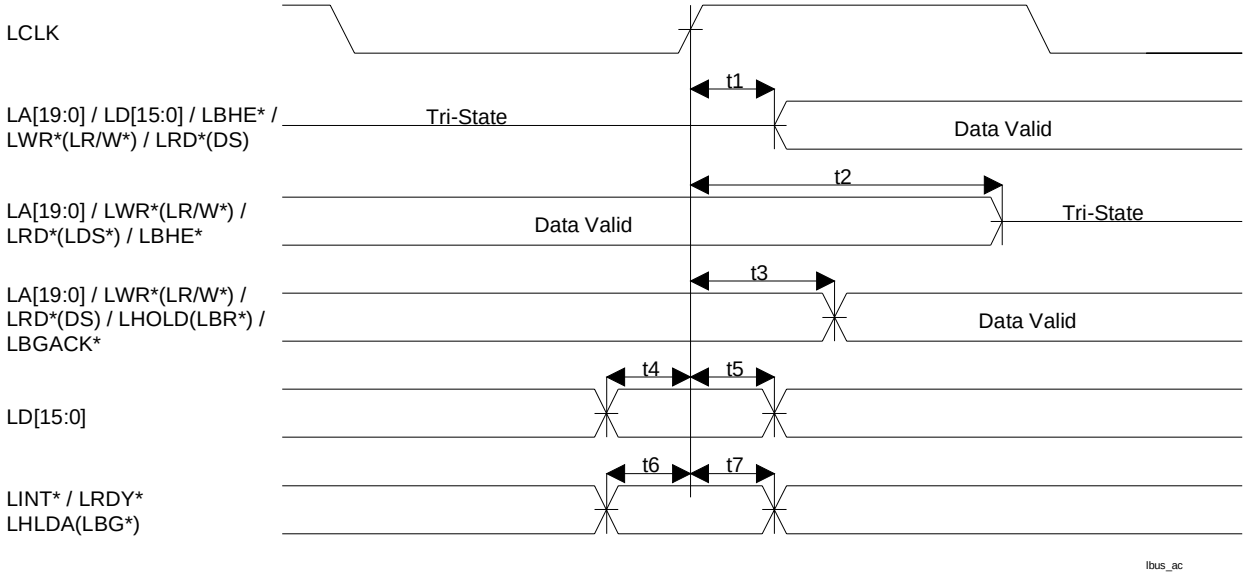


AC CHARACTERISTICS - LOCAL BUS IN BRIDGE MODE (LMS = 0)

(0ºC to +70ºC; VDD = 3.0V to 3.6V)

Parameter	Symbol	Min	Typ	Max	Units	Notes
Delay Time from the Rising Edge of LCLK to Output Valid from Tri-State	t1	2		10	ns	
Delay Time from the Rising Edge of LCLK to Tri-State from Output Valid	t2	2		15	ns	
Delay Time from the Rising Edge of LCLK to Output Valid from an Already ActiveDrive State	t3	2		10	ns	
LD[15:0] Set Up Time to the Rising Edge of LCLK	t4	5			ns	
LD[15:0] Hold Time from the Rising Edge of LCLK	t5	2			ns	
Input Set Up Time to the Rising Edge of LCLK	t6	10			ns	
Input Hold Time from the Rising Edge of LCLK	t7	5			ns	

LOCAL BUS BRIDGE MODE (LMS = 0) AC TIMING DIAGRAM Figure 12B



AC CHARACTERISTICS - LOCAL BUS IN CONFIGURATION MODE (LMS = 1)

(0ºC to +70ºC; VDD = 3.0V to 3.6V)

Parameter	Symbol	Min	Typ	Max	Units	Notes
Set Up Time for LA[15:0] Valid to LCS* Active	t1	0			ns	
Set Up Time for LCS* Active to Either LRD*, LWR*, or LDS* Active	t2	0			ns	
Delay Time from Either LRD* or LDS* Active to LD[15:0] Valid	t3			120	ns	1
Hold Time from Either LRD*, LWR*, or LDS* Inactive to LCS* Inactive	t4	0			ns	
Hold Time from LCS* Inactive to LD[15:0] Tri-State	t5	5		20	ns	
Wait Time from Either LWR* or LDS* Active to Latch LD[15:0]	t6	75			ns	1
LD[15:0] Set Up Time to Either LWR* or LDS* Inactive	t7	40			ns	
LD[15:0] Hold Time from Either LWR* or LDS* Inactive	t8	2			ns	
LA[15:0] Hold from Either LWR* or LDS* Inactive	t9	5			ns	

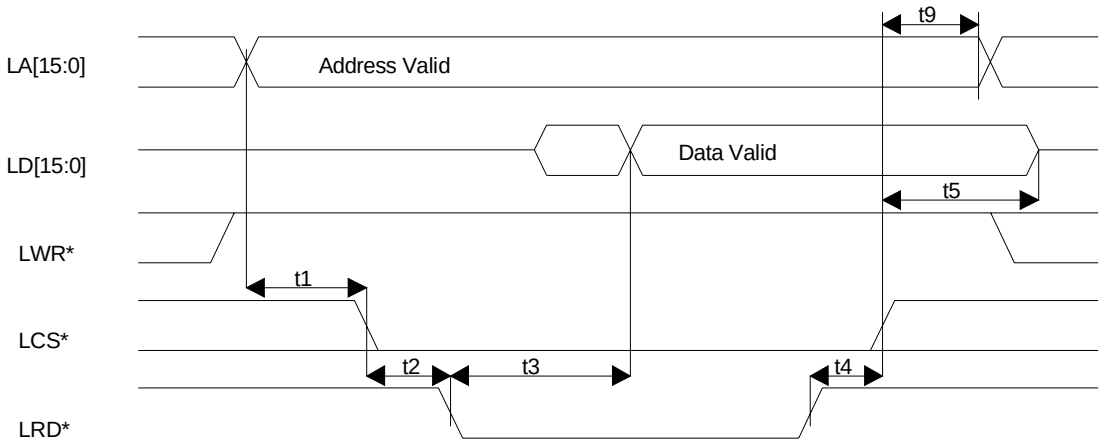
NOTES:

1. The 65 ns number is based on a PCLK of 33 MHz. The formula is [(2 x PCLK Period) + 5] ns.

LOCAL BUS CONFIGURATION MODE (LMS = 1) AC TIMING DIAGRAM

Figure 12C

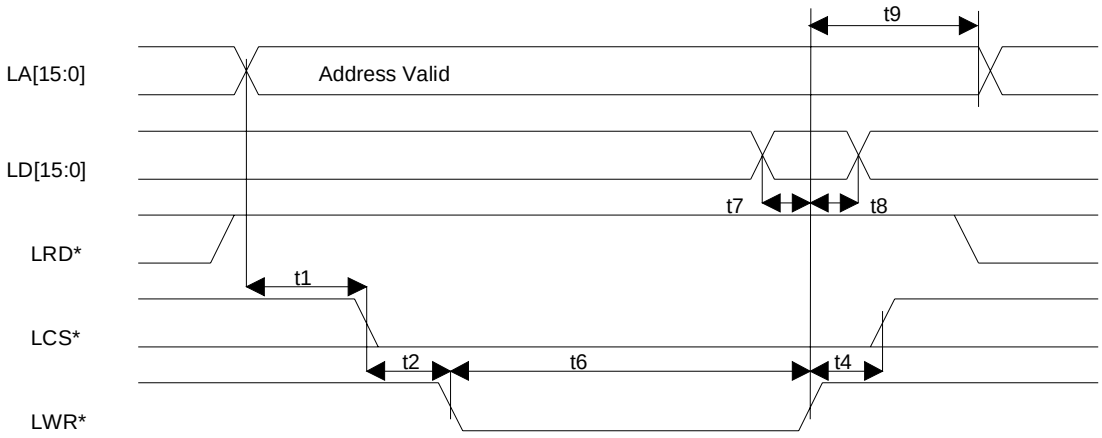
INTEL READ CYCLE



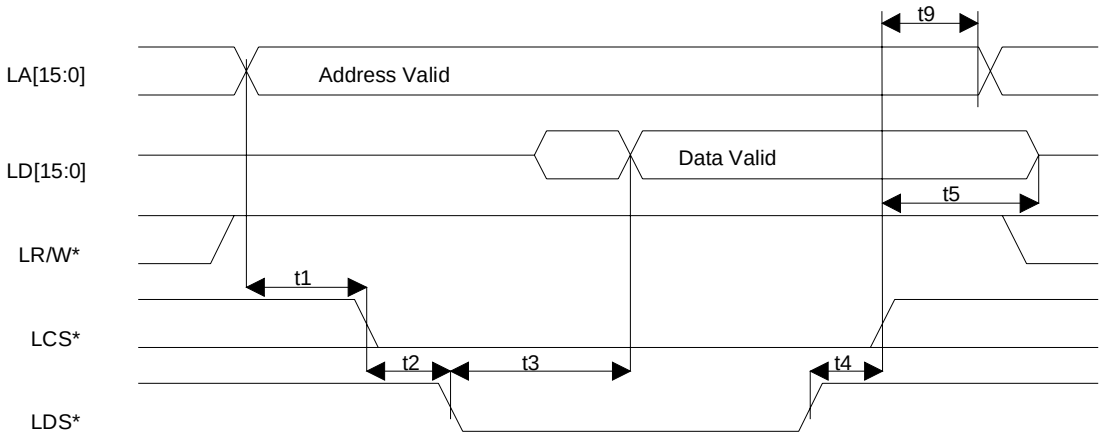
LOCAL BUS CONFIGURATION MODE (LMS = 1) AC TIMING DIAGRAM

Figure 12C Continued

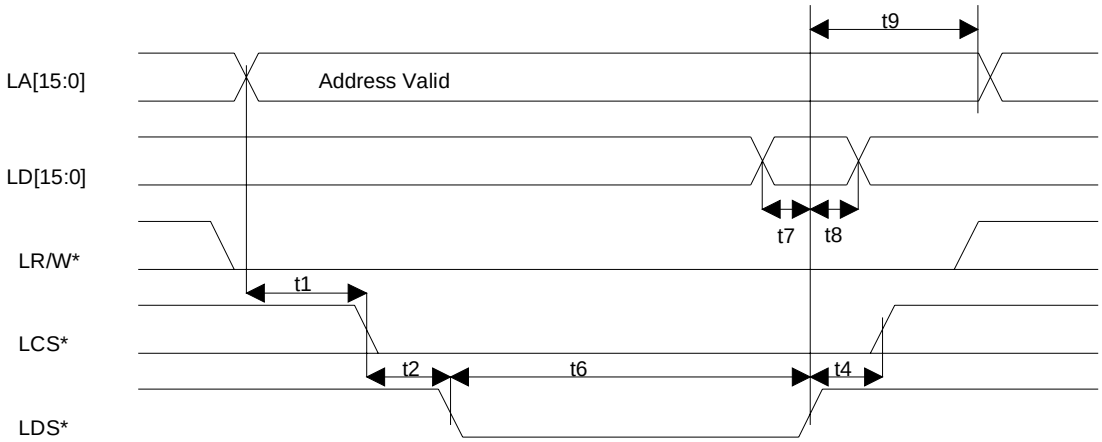
INTEL WRITE CYCLE



MOTOROLA READ CYCLE



MOTOROLA WRITE CYCLE



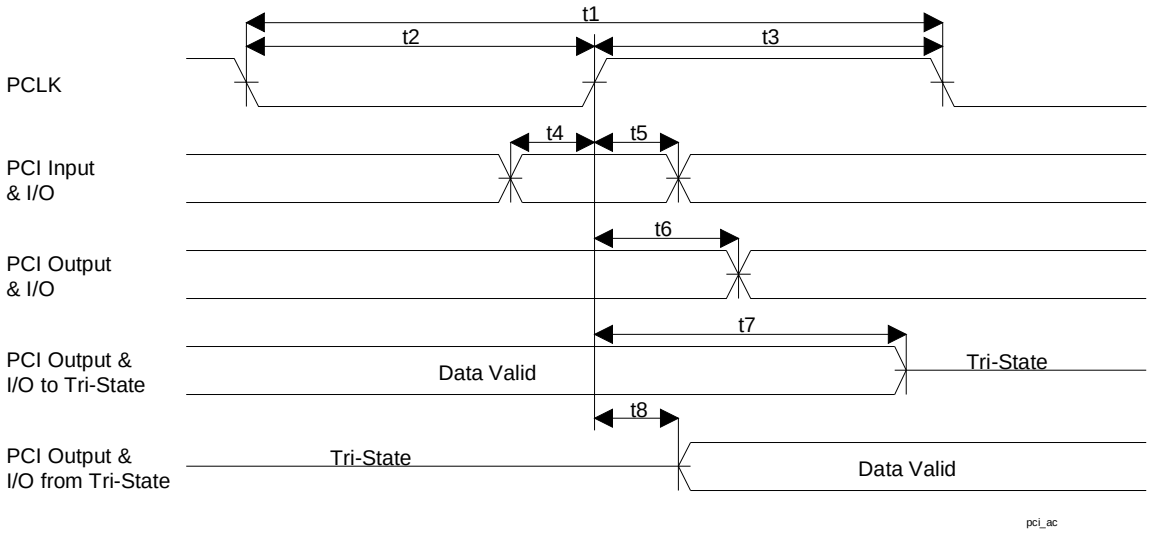
lb_ac1

AC CHARACTERISTICS - PCI BUS INTERFACE

(0ºC to +70ºC; VDD = 3.0V to 3.6V)

Parameter	Symbol	Min	Typ	Max	Units	Notes
PCLK Period	t1	30		40	ns	
PCLK Low Time	t2	12			ns	
PCLK High Time	t3	12			ns	
All PCI Inputs & I/O Set Up Time to the Rising Edge of PCLK	t4	7			ns	
All PCI Inputs & I/O Hold Time from the Rising Edge of PCLK	t5	0			ns	
Delay from the Rising Edge of PCLK to Data Valid on all PCI Outputs & I/O	t6	2		11	ns	
Delay from the Rising Edge of PCLK to Tri-State on all PCI Outputs & I/O	t7			28	ns	
Delay from the Rising Edge of PCLK to Data Valid from Tri-State on all PCI Outputs & I/O	t8	2			ns	

PCI BUS INTERFACE AC TIMING DIAGRAM Figure 12D

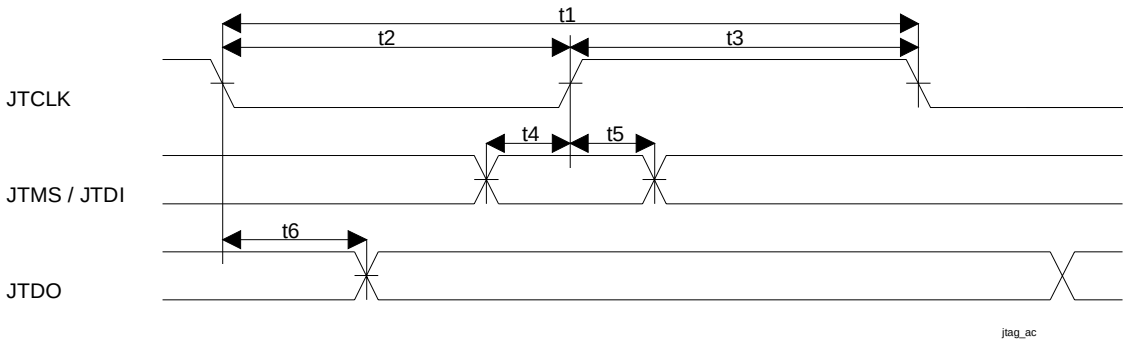


AC CHARACTERISTICS - JTAG TEST PORT INTERFACE

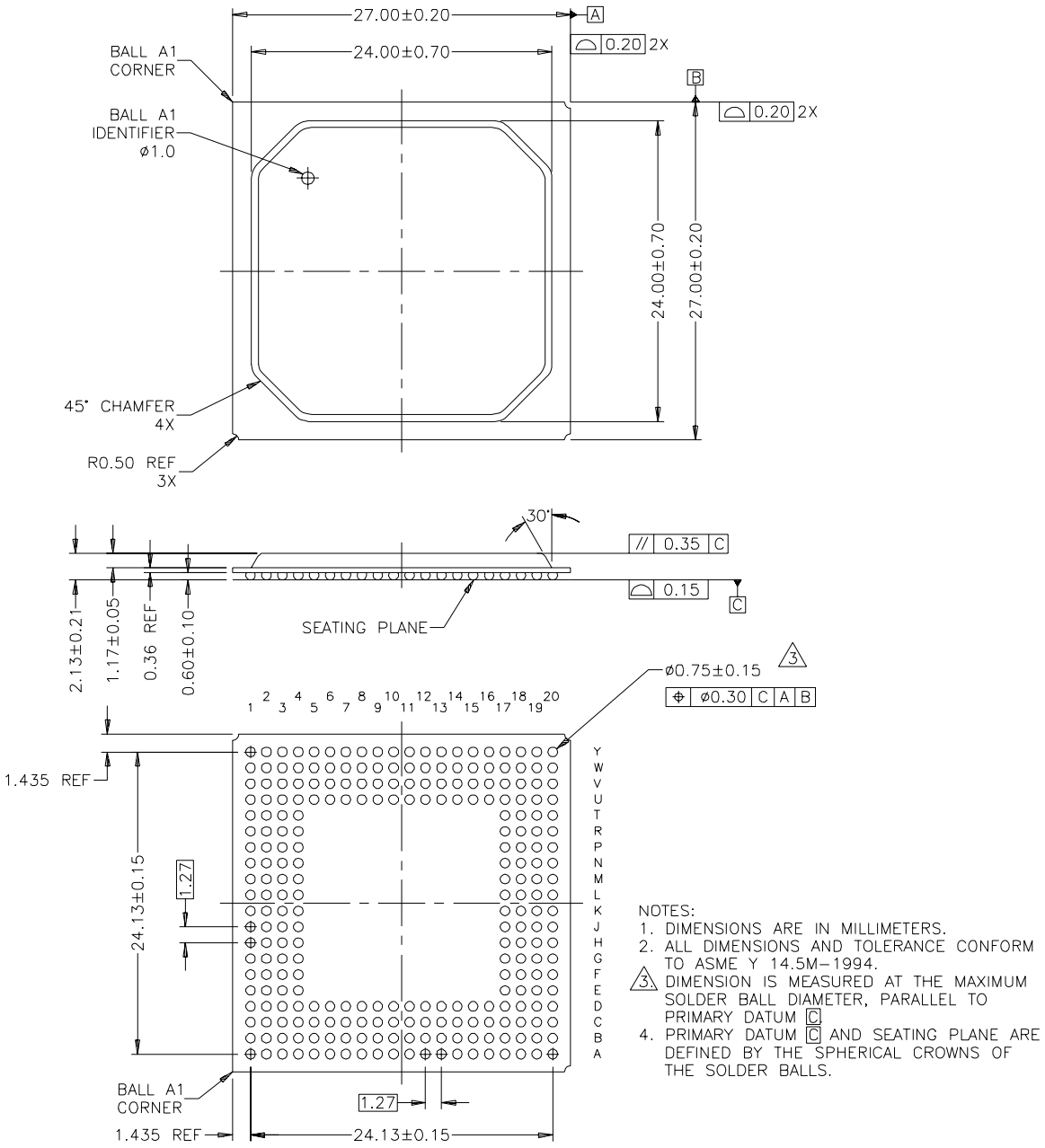
(0ºC to +70ºC; VDD = 3.0V to 3.6V)

Parameter	Symbol	Min	Typ	Max	Units	Notes
JTCLK Clock Period	t1	1000			ns	
JTCLK Clock Low Time	t2	400			ns	
JTCLK Clock High Time	t3	400			ns	
JTMS / JTDI Set Up Time to the Rising Edge of JTCLK	t4	50			ns	
JTMS / JTDI Hold Time from the Rising Edge of JTCLK	t5	50			ns	
Delay Time from the Falling Edge of JTCLK to Data Valid on JTDO	t6	2		50	ns	

JTAG TEST PORT INTERFACE AC TIMING DIAGRAM Figure 12E



SECTION 13: MECHANICAL DIMENSIONS



NOTES:
1. DIMENSIONS ARE IN MILLIMETERS.
2. ALL DIMENSIONS AND TOLERANCE CONFORM TO ASME Y 14.5M–1994.
3. DIMENSION IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.

SECTION 14: APPLICATIONS

Section 14 describes some possible applications for the DS3131. The numbers of potential configurations are numerous and only a few are shown. Users are encouraged to contact the factory for support of their particular application. Contact information is shown in Table 14A.

TELECOM APPLICATIONS SUPPORT CONTACT INFORMATION Table 14A

Email	telecom.support@dalsemi.com
Web	www.dalsemi.com

14.1 T1/E1 AND T3/E3 APPLICATIONS

Figure 14A shows an application where 28 T1 lines are being terminated into the DS3131. In this application, the 28 T1 lines have been demultiplexed down from a T3 link and the DS21FF42 and DS21FT42 framers are used to locate the frame boundaries of the incoming T1 links. The Local Bus (if enabled) can be used to configure and monitor the T1 framers.

28 T1 LINES DEMUXED FROM A T3 LINE Figure 14A

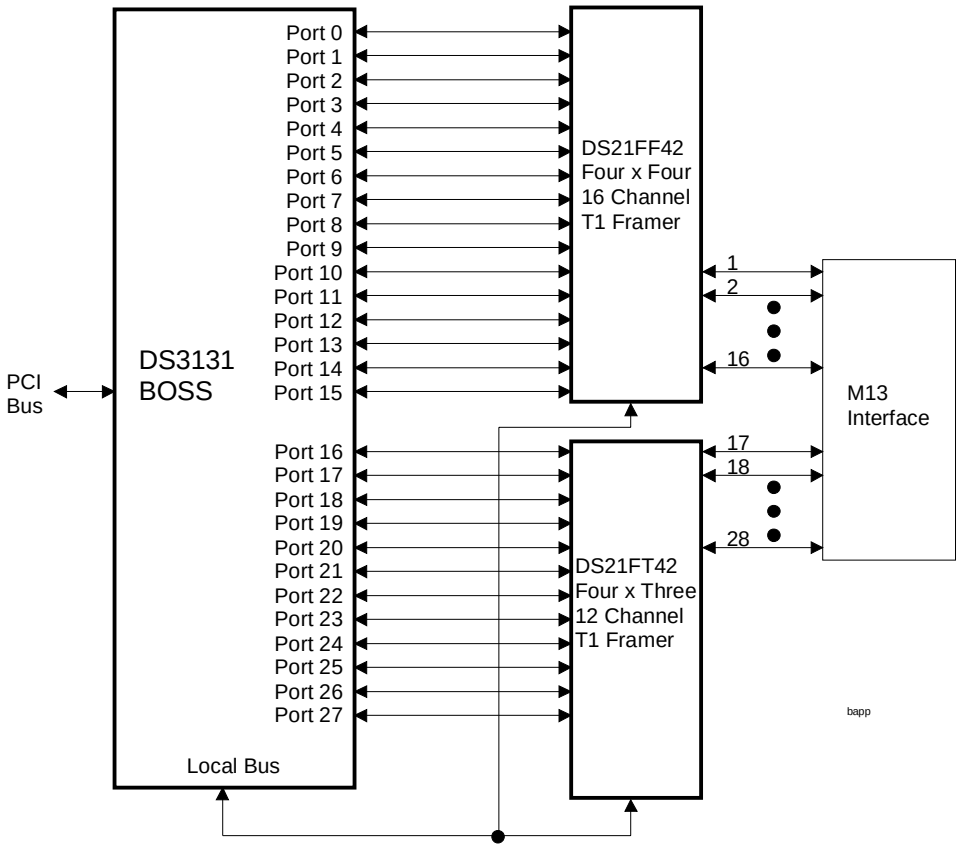


Figure 14B shows an application where up to 40 T1 or E1 ports are interfaced to a single DS3131. In this application, The Dallas Semiconductor Quad T1 and E1 Transceivers (DS21Q352 / DS21Q552 / DS21Q354 / DS21Q554) perform the line interface function and frames to the T1 or E1 line. The Local Bus (if enabled) can be used to configure and monitor the T1/E1 transceivers.

MULTI-PORT T1 OR E1 APPLICATION Figure 14B

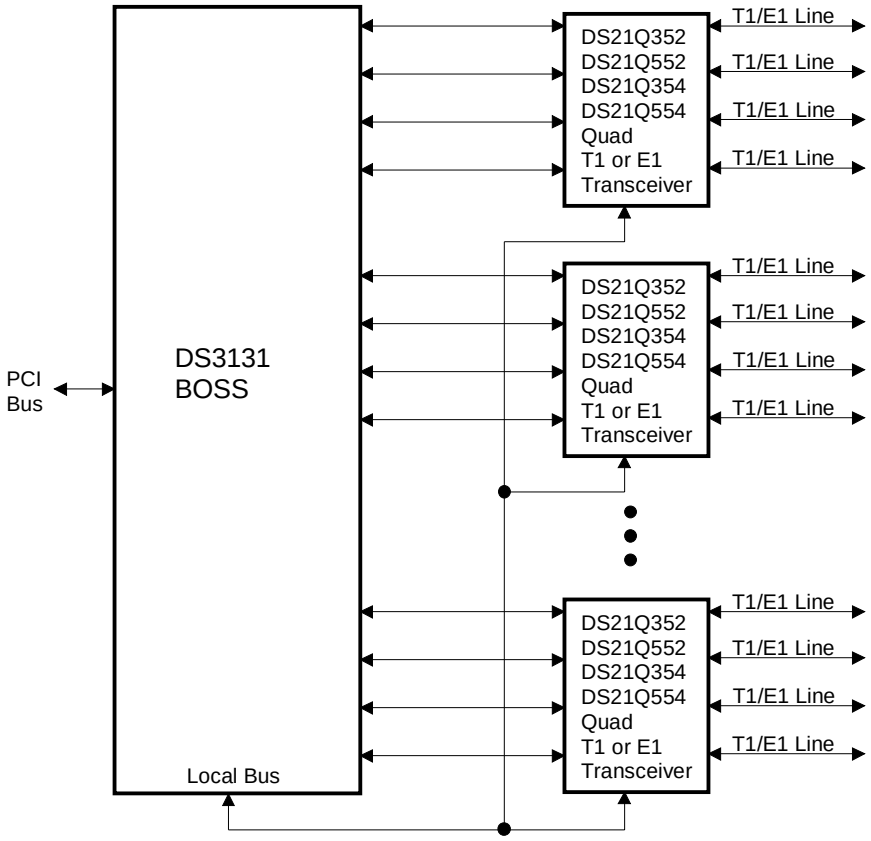
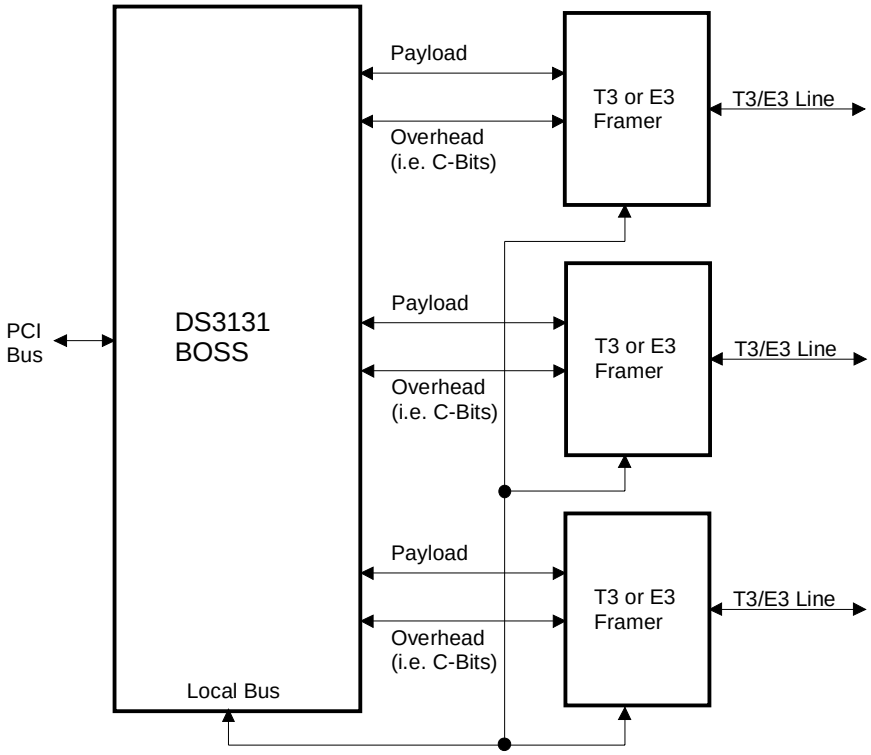


Figure 14C shows an application where three T3 or E3 framers are interfaced to a single DS3131. The DS3131 is used to terminate both the payload of the T3 or E3 link as well as the overhead channels (like the C-Bits in a T3 application). The Local Bus (if enabled) can be used to configure and monitor the T3/E3 framers.

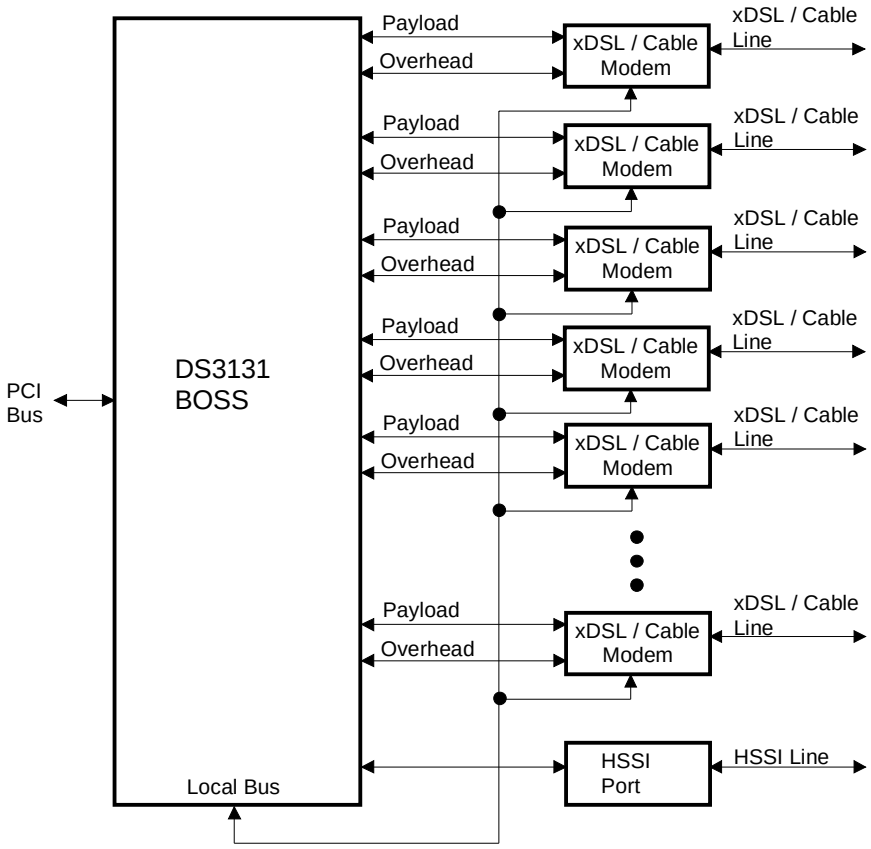
UNCHANNELIZED T3 OR E3 APPLICATION Figure 14C



14.2 DSL & CABLE MODEM APPLICATIONS

Figure 14D shows an application where multiple xDSL or Cable Modems are interfaced to a single DS3131. Such an application would exist in a DSLAM (DSL Access Multiplexer). The DS3131 is used to terminate both the payload of the xDSL/Cable Modem link as well as the overhead channels (like the links used for signaling and provisioning). In the application shown in Figure 14D, one of the Ports on the DS3131 has been dedicated to a High Speed Serial Interface (HSSI) to allow the concentrated packet traffic to be linked to another entity. The Local Bus (if enabled) can be used to configure and monitor the xDSL/Cable Modems.

DSLAM / CABLE MODEM APPLICATION Figure 14D



14.3 SONET/SDH APPLICATIONS

Figure 14E shows an application where the overhead links on multiple SONET or SDH lines is being terminated into a single DS3131. The Local Bus (if enabled) can be used to configure and monitor the SONET/SDH interfaces.

SONET/SDH OVERHEAD TERMINATION APPLICATION Figure 14E