1.0 Product Description

1.1 Introduction

The highly integrated CMOS Bt8230 ATM Segmentation and Reassembly Controller Chip (SRC) combines unique features making it a state-of-the-art ATM controller. At introduction, the Bt8230 will support all requirements contained in ATM Forum UNI 3.1 and the related ANSI and ITU standards. The architecture of the device is such that the new traffic management algorithms for Available Bit Rate (ABR) service to be defined in UNI 4.0 can be supported in a future version.

The Bt8230 combines a Peripheral Component Interconnect (PCI) bus interface, segmentation and reassembly controllers, local memory controller, DMA coprocessor, and a proprietary scheduling algorithm that enables each segmentation channel to achieve an independent bit rate of up to 200 Mbit/s per channel.

The Bt8230 is especially suited for today's terminal adapters, routers/hubs, and other Wide Area Network (WAN) applications. The device supports thousands of open connections simultaneously with robust Operation and Maintenance (OAM), signaling, and Interim Local Management Interface (ILMI) features. Other key features include support for random Virtual Path Indicator/ Virtual Channel Indicator (VPI/VCI) assignment, interleaved AAL5 and AAL3/4 support, and termination of signaling and ILMI into local memory to maintain management connections independent of the host.

Incorporated into the Bt8230 is a simplified traffic priority management scheme that allows each connection to be individually monitored. Both segmentation and reassembly processes deal with host buffers as pipelines of information. Partial packets may be added to the transmit stream or released as they are received for additional processing by the host. The host buffer structures resemble those found in common operating systems so the data can be processed by the host without the need to physically copy the information to another memory location. A send immediate mode supports low-latency connections. The Bt8230 transmit scheduling algorithm supports the currently defined “leaky buckets” and will be upgraded to support the ABR algorithms as defined in UNI 4.0.

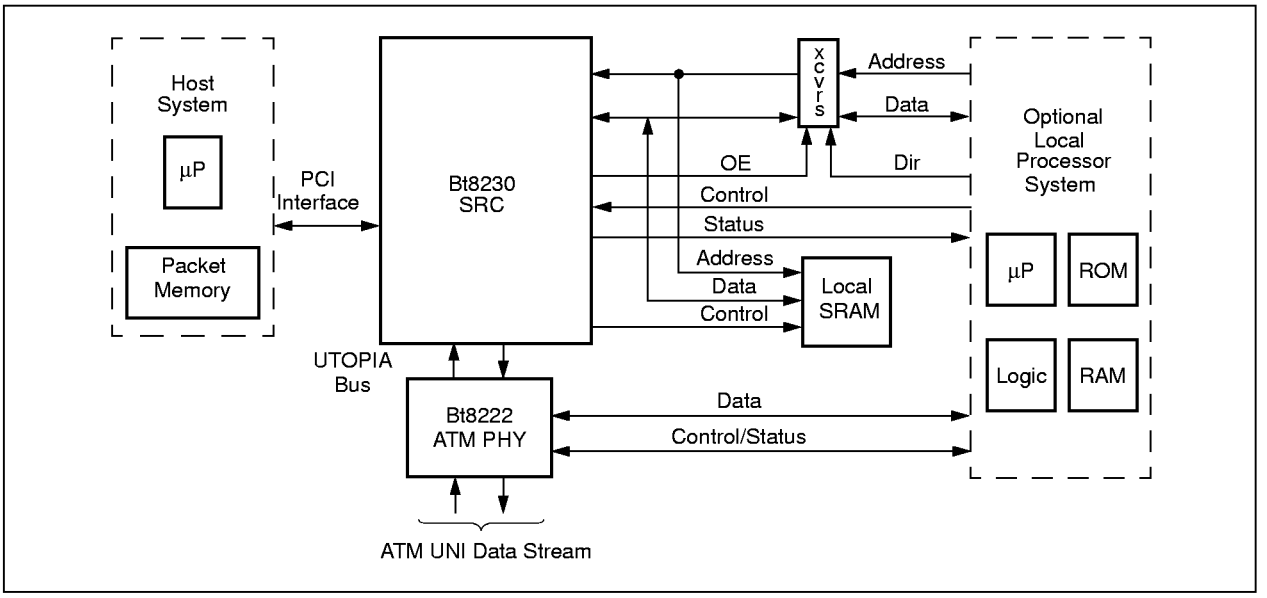
The optional local processor architecture provides the flexibility for system designers and programmers to track the rapidly changing ATM standards. OAM, ILMI, and signaling can all be ported to this processor to off-load the host system. At the same time, a processor-less system is a cost-effective option with the Bt8230 architecture. Therefore, this architecture is especially suited to ATM network interfaces for file servers, routers/hubs, Digital Service Units (DSUs), and

WAN Data Terminal Equipment (DTE), as well as cost-sensitive PCI ATM cards. Combined with the Bt8222 ATM Receiver/Transmitter, the Bt8230 provides an ATM solution capable of full-duplex operations at bit rates up to 155.52 Mbps ATM speed without the cost constraints of requiring a large amount of local memory to buffer incoming ATM cells.

The Bt8230 is implemented as a 33 MHz CMOS device in a 208-pin Plastic Quad Flat Pack (PQFP) with direct TTL-level interfaces to the PCI bus. The device allows incoming ATM cells to be transferred directly into the main system CPU without local buffering. The Bt8230 architecture uniquely balances the strengths of the PCI bus with the need to achieve sustained 155.52 Mbps ATM speeds.

With the Bt8230 and Bt8222, a highly integrated and economical PCI ATM adapter can be built. The interface card could, therefore, consist of the Bt8230, an ATM link interface or framer chip, the Bt8222, an optional external control processor, a suitable physical transceiver device capable of driving optical fibers or twisted pair cable, interface connectors, and SRAM for control memory. The SRAM size is driven primarily by the number of connections to be supported. Approximately 250 kbytes are needed per 1000 connectors. Figure 1-1 shows the Bt8230 connected to an ATM physical device and an optional local processor.

Figure 1-1. Bt8230 System Block Diagram



1.1.1 DMA Coprocessor

The Bt8230 incorporates an on-chip DMA coprocessor that works in close conjunction with the segmentation and reassembly processors. Once a transfer has been initiated by either the reassembly or segmentation coprocessor, the DMA coprocessor is responsible for gaining access to the PCI bus, transferring the requested data, and notifying the segmentation or reassembly coprocessor that the transfer is complete. The DMA coprocessor transfers all data using the read and write burst buffers in the PCI bus interface, and depends on the PCI bus master logic to execute the PCI bus burst protocol.

1.1.2 Segmentation Coprocessor

The segmentation coprocessor is responsible for transmitting 52-octet data to the ATM physical interface block. This coprocessor segments large (up to 64 kbyte) ATM Adaptation Layer (AAL) Service Data Units (SDUs) located in host or local RAM into a controlled stream of 44- or 48-byte ATM cell payloads, which are then formatted into 52-byte ATM cells, not including the Header Error Control (HEC) byte. (A blank HEC byte is added by the ATM physical interface block.) This task is performed with the help of state tables in local RAM memory. Rate entries for each segmentation connection allow the segmentation coprocessor to automatically multiplex cells from a number of outgoing channels onto a single outgoing link while preserving the bandwidth relations between the channels. The segmentation coprocessor can also supply the CPCS-PDU header, trailer, and pad fields as part of the segmentation process.

1.1.3 Reassembly Coprocessor

The reassembly coprocessor processes the 52-octet cells from the ATM physical interface block. The ATM physical block strips the HEC byte before passing the cell to the reassembly coprocessor. It controls the writing of the Common Part Conversion Sublayer (CPCS) payload to host memory and performs all necessary SAR and CPCS checks.

A scatter method is used by the reassembly coprocessor to write the payload portion of the ATM cell to host memory. It maintains a free buffer queue and status queue in local memory to control the scatter operation. The free buffer queue is updated by the host processor to point to available cell buffers in host memory. The status queue is updated by the reassembly coprocessor with information about how the cell buffers are used. A hash table and a reassembly VCC table located in local memory operate the various CPCS connections and are maintained by the reassembly coprocessor. The hash table allows the reassembly coprocessor to quickly locate the appropriate reassembly VCC table based upon the received VPI/VCI value in the ATM cell header, and permits the assignment of any VPI/VCI value to a particular connection.

1.1.4 PCI Bus Interface

The PCI bus interface responds to read and write requests by the host CPU as a PCI slave, allowing access to the chip resources by software on the host. The Bt8230 is capable of acting as a DMA bus master on the PCI bus. As a result, the PCI bus interface implements the full set of address, data, and control signals required to drive the bus as a master, and contains the logic required to support arbitration for the PCI bus. This interface is PCI Version 2.0 compliant.

1.1.5 Local Memory Interface

The Bt8230 contains a memory controller that allows the device to access up to 8 Mbytes of static RAM memory. The memory controller also coordinates access to the internal control and status registers by the host and local processors. Both zero and single wait state access are supported to SRAM, which enables various price and performance trade-offs to be made. In addition, SRAM devices organized by_4, by_8, and by_16 may be used. Access to internal control and status registers follows the timing requirements of SRAM access, which simplifies system implementations.

1.1.6 Local Processor Interface

The local processor interface in the Bt8230 allows an external Reduced Instruction Set Computer (RISC) CPU to be directly connected to the device to serve as a local controlling intelligence that can handle initialization, connection management, overall data management, error recovery, and OAM functions. The interface is designed to connect directly to Intel i960CA/CF/Jx processors, but is generic enough to connect to other popular processors with little additional logic. The processor interface is "loosely coupled," meaning that the processor connects to the Bt8230 through bidirectional transceivers and buffers for the address and data buses. This allows the processor fast access to Bt8230 memory and registers, but insulates the Bt8230 from processor instruction and data cache fills. It also allows the processor to control multiple Bt8230 or physical devices if desired.

1.1.7 ATM Physical Interface

The ATM physical interface communicates with and controls the ATM link interface chip, which carries out all the transmission convergence and physical media dependent functions defined by the ATM protocol. This block strips the HEC byte from the received cell and adds a blank HEC byte to the transmit cell. No HEC processing is performed in the Bt8230. Three modes of operation are provided: standard UTOPIA, slave UTOPIA, and Bt8222. Standard UTOPIA mode conforms to the UTOPIA standard. Slave UTOPIA mode reverses the control direction to allow the ATM physical chip to drive the interface. Bt8222 mode allows the Brooktree physical framer part to be connected directly to the Bt8230.

1.2 Logic Diagram and Pin Descriptions

The Bt8230 is packaged in a 208–pin Plastic Quad Flat (PQFP). A functionally partitioned logic diagram of the Bt8230 is shown in Figure 1-2. Pin descriptions, names, input/output assignments are detailed in Table 1-1.

Figure 1-2. Bt8230 Logic Diagram

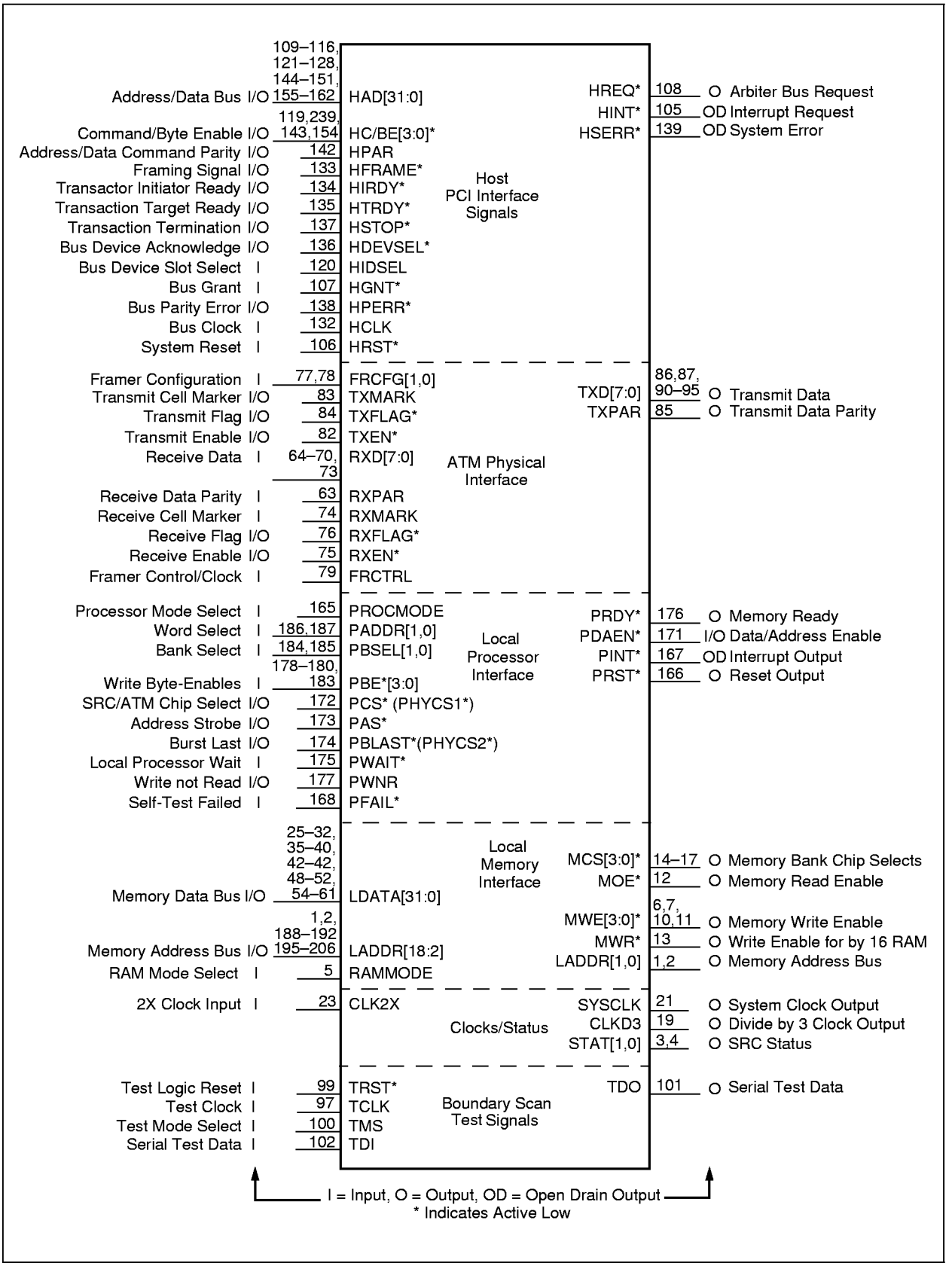


Table 1-1. Hardware Signal Definitions *(1 of 5)*

	Pin Label	Signal Name	I/O	Definition
Host PCI Interface Signals	HAD[31:0]	Multiplexed Address/Data Bus	I/O	Used by the PCI host or the Bt8230 to transfer addresses or data over the PCI bus.
	HC/BE[3:0]*	Command/Byte Enable	I/O	Outputs a command (during PCI address phases) or byte enables (during data phases) for each bus transaction.
	HPAR	Address/Data Command Parity	I/O	Supplies the even parity computed over the HAD[31:0] and HC/BE[3:0]* lines during valid data phases; it is sampled (when the Bt8230 is acting as a target) or driven (when the Bt8230 acts as an initiator) one clock edge after the respective data phase.
	HFRAME*	Framing Signal	I/O	A high-to-low HFRAME* transition indicates that a new transaction is beginning (with an address phase). A low-to-high transition indicates that the next valid data phase will end the current transaction.
	HIRDY*	Transaction Initiator Ready	I/O	Used by the transaction initiator or bus master (either the Bt8230 or the PCI host) to indicate ready for data transfer. A valid data phase ends when both HIRDY* and HTRDY* are sampled asserted on the same clock edge.
	HTRDY*	Transaction Target Ready	I/O	Used by the transaction target or bus slave (either the Bt8230 or the PCI bus memory) to indicate that it is ready for a data transfer. A valid data phase ends when both HIRDY* and HTRDY* are sampled asserted on the same clock edge.
	HSTOP*	Transaction Termination	I/O	Driven by the current target or slave (either the Bt8230 or the PCI bus memory) to abort, disconnect, or retry the current transfer. The HSTOP* line is used by the PCI master in conjunction with the HTRDY* and HDEVSEL* lines to determine the type of transaction termination.
	HDEVSEL*	Bus Device Acknowledge	I/O	Driven by a target to indicate to the initiator that the address placed on the HAD[31:0] lines (together with the command on the HC/BE[3:0]* lines) has been decoded and accepted as a valid reference to the target's address space. Once asserted, it is held by the Bt8230 (when acting as a slave) until HFRAME* is deasserted; otherwise, it indicates (in conjunction with HSTOP* and HTRDY*) a target abort.
	HIDSEL	Bus Device Slot Select	I	Signals the Bt8230 that it is being selected for a configuration space access.
	HREQ*	Arbiter Bus Request	O	Asserted by the Bt8230 to request control of the PCI bus.
	HGNT*	Bus Grant	I	Asserted to indicate to the Bt8230 that it has been granted control of the PCI bus, and may begin driving the address/data and control lines after the current transaction has ended (indicated by HFRAME*, HIRDY* and HTRDY* all deasserted simultaneously).

Table 1-1. Hardware Signal Definitions (2 of 5)

	Pin Label	Signal Name	I/O	Definition
PCI Interface Signals	HINT*	Interrupt Request	OD	Signals an interrupt request to the PCI host, and is tied to the INTA* line on the PCI bus.
	HPERR*	Bus Parity Error	I/O	Driven asserted by the Bt8230 (as a bus slave) or by a target addressed by the Bt8230 when it acts as a bus master to indicate a parity error on the HAD[31:0] and HC/BE[3:0]* lines. It is asserted when the Bt8230 is a bus slave or sampled when the Bt8230 is a bus master on the second clock edge after a valid data phase. The Bt8230 drives the HPERR* line only when acting as a slave.
	HSERR*	System Error	OD	Indicates a system error or a parity error on the HAD[31:0] and HC/BE[3:0]* lines during an address phase. This pin is handled in the same way as HPERR*, and is only driven by the Bt8230 when it acts as a bus slave.
	HCLK	Bus Clock	I	Supplies the PCI bus clock signal.
	HRST*	System Reset	I	Performs a hardware reset of the Bt8230 and associated peripherals when asserted. Must be asserted for 16 cycles of HCLK.
ATM Physical Interface	FRCFG[1,0]	Framer Configuration	I	Configuration pins FRCFG[1,0] determine what framer interface Bt8230 supports. 00 = Bt8222 interface 01 = UTOPIA interface 10 = Slave UTOPIA interface 11 = Reserved, do not use
	TXD[7:0]	Transmit Data	O	Carries outgoing data bytes to the framer chip in all framer modes.
	TXPAR	Transmit Data Parity	O	Outputs the 8-bit odd parity computed over the TXD[7:0] lines in all framer modes.
	TXMARK	Transmit Cell Marker	I/O	In both UTOPIA and slave UTOPIA modes, the TXMARK line is asserted by the Bt8230 when the starting byte of a 53-byte cell is being output. In Bt8222 mode, this pin is asserted by the framer to indicate that it is expecting the starting byte of a 53-byte ATM cell to be transferred on the TXD[7:0] lines as the next valid data byte.
	TXFLAG*	Transmit Flag	I/O	In UTOPIA mode, TXFLAG* indicates that the transmit buffer in the downstream link interface chip is full, and no more data can be accepted. In slave UTOPIA mode, this pin indicates to the link interface chip that the Bt8230 transmit buffer is empty. In Bt8222 mode, TXFLAG* indicates that no more data is available in the transmit buffer of the Bt8230.
	TXEN*	Transmit Enable	I/O	Indicates that valid data has been placed on the TXD[7:0], TXPAR, and TXMARK lines in the current clock cycle when the Bt8230 is in UTOPIA or slave UTOPIA mode. This pin is an output in UTOPIA mode and an input in slave UTOPIA mode. In Bt8222 mode, the TXEN* input indicates that the downstream framer device is requesting the Bt8230 to transfer a byte of data on the TXD[7:0] lines.
	RXD[7:0]	Receive Data	I	Transfers incoming data bytes from the link interface or framer chip to the Bt8230 in all framer modes.

Table 1-1. Hardware Signal Definitions (3 of 5)

	Pin Label	Signal Name	I/O	Definition
ATM Physical Interface	RXPAR	Receive Data Parity	I	Should be driven with the 8-bit odd parity computed over the RXD[7:0] lines by the link interface or framer chip in all framer modes.
	RXMARK	Receive Cell Marker	I	Indicates that the current byte being transferred on the RXD[7:0] lines is the starting byte of a 53-byte cell.
	RXFLAG*	Receive Flag	I/O	In UTOPIA mode, RXFLAG* indicates that the receive buffer in the downstream link interface chip is empty, no more data can be transferred, and the RXD[7:0], RXPAR, and RXMARK lines are invalid. In slave UTOPIA mode, this pin indicates to the framer chip that the receive FIFO in the Bt8230 is full. In Bt8222 mode, the RXFLAG* output indicates that the Bt8230 internal FIFO buffer is full and any subsequent attempts to transfer data will be ignored.
	RXEN*	Receive Enable	I/O	In UTOPIA mode, RXEN* indicates that the Bt8230 is ready to receive data on the RXD[7:0], RXPAR, and RXMARK lines in the next clock cycle. This pin is an output in UTOPIA mode, and an input in slave UTOPIA mode. In Bt8222 mode, the RXEN* input is driven active by the framer to signify that valid data has been presented on the RXD[7:0], RXPAR, FRCTRL, and RXMARK lines.
	FRCTRL	Framer Control/Clock	I	In UTOPIA and slave UTOPIA mode, the FRCTRL line should be driven with a clock that is synchronous to that used by the framer device for interfacing to the Bt8230. The TXD[7:0], TXPAR, TXMARK, TXFLAG* TXEN*, RXD[7:0], RXPAR, RXMARK, RXFLAG*, and RXEN* lines must be synchronous to this clock in UTOPIA mode, and maintain the specified setup and hold times with reference to its rising edge. In Bt8222 mode, this pin marks a cell as invalid and the Bt8230 discards the cell.
Local Processor Interface	PROCMODE	Processor Mode Select	I	When grounded, this input selects the local processor mode. When pulled to a logic high, the stand-alone mode is selected.
	PADDR[1,0]	Word Select Inputs	I	The PADDR[1,0] inputs are connected to the word select field of the CPU address bus (address bits [3, 2] for the Intel i80960CA processor, which can perform 4-word burst transactions). These inputs are used by the Bt8230 to allow single-cycle bursts to be performed without requiring very short memory access times.
	PBSEL[1,0]	Bank Select Inputs	I	Select one of four banks of memory to be accessed. They are decoded by the memory controller to generate the appropriate chip/bank selects to the external memory.

Table 1-1. Hardware Signal Definitions (4 of 5)

	Pin Label	Signal Name	I/O	Definition
Local Processor Interface	PBE[3:0]*	Write Byte Enable	I	Supply byte enables for each local processor memory access. These pins are only relevant during writes by the local processor to local memory. Each byte enable line corresponds to a specific byte lane in the LDATA[31:0] data bus: PBE[0]* corresponds to LDATA[7:0], PBE[1]* to LDATA[15:8], PBE[2]* to LDATA[23:16], and PBE[3]* to LDATA[31:24].
	PCS* (PHYCS1*)	SRC Chip Select ATM PHY Chip Select (in stand-alone mode)	I/O	In local processor mode with PROCMODE tied low, PCS* is the SRC chip select input. In stand-alone mode, this pin is PHYCS1*, which may be connected to the chip select input of the Bt8222 PHY device.
	PAS*	Address Strobe	I/O	Indicates a local processor address cycle. In stand-alone mode, PAS* is used to drive the AS* pin of the Bt8222 device.
	PWNR	Write/not Read	I/O	The PWNR input indicates the direction of a local processor transfer. A logic one indicates a write while a logic zero indicates a read. During stand-alone mode, this output provides the same function for the Bt8222.
	PWAIT*	Processor Wait	I	Used by the local processor or external logic to insert wait states for read or write transactions.
	PBLAST* (PHYCS2*)	Burst Last ATM PHY Chip Select (in stand-alone mode)	I/O	In local processor mode, this input is used by the processor to indicate the end of a transaction. If burst transfers are not required, tie this input low. During stand-alone mode, this output is a second chip select, PHYCS2*.
	PRDY*	Memory Ready	O	Signals that the memory or control register has accepted the data on a write, or that data is available to latch by the local processor on a read cycle.
	PDAEN*	Data/Address Enable	I/O	Connected to the output enable input of the bidirectional transceivers and buffers used to isolate the Bt8230 data and address bus from the local processor. In stand-alone mode, this input is connected to the physical device's interrupt output(s).
	PFAIL*	Self-Test Failed	I	The local processor can indicate a failure of its internal self-test or initialization processes by asserting the PFAIL* input to the Bt8230.
	PINT*	Interrupt Output	OD	Asserted by the Bt8230 to the local processor to signal an interrupt request in local processor mode.
	PRST*	Reset Output	O	Asserted by the Bt8230 to the local processor whenever the HRST* input is asserted, or when the LP_ENABLE bit in the CONFIG0 register is a logic low.

Table 1-1. Hardware Signal Definitions (5 of 5)

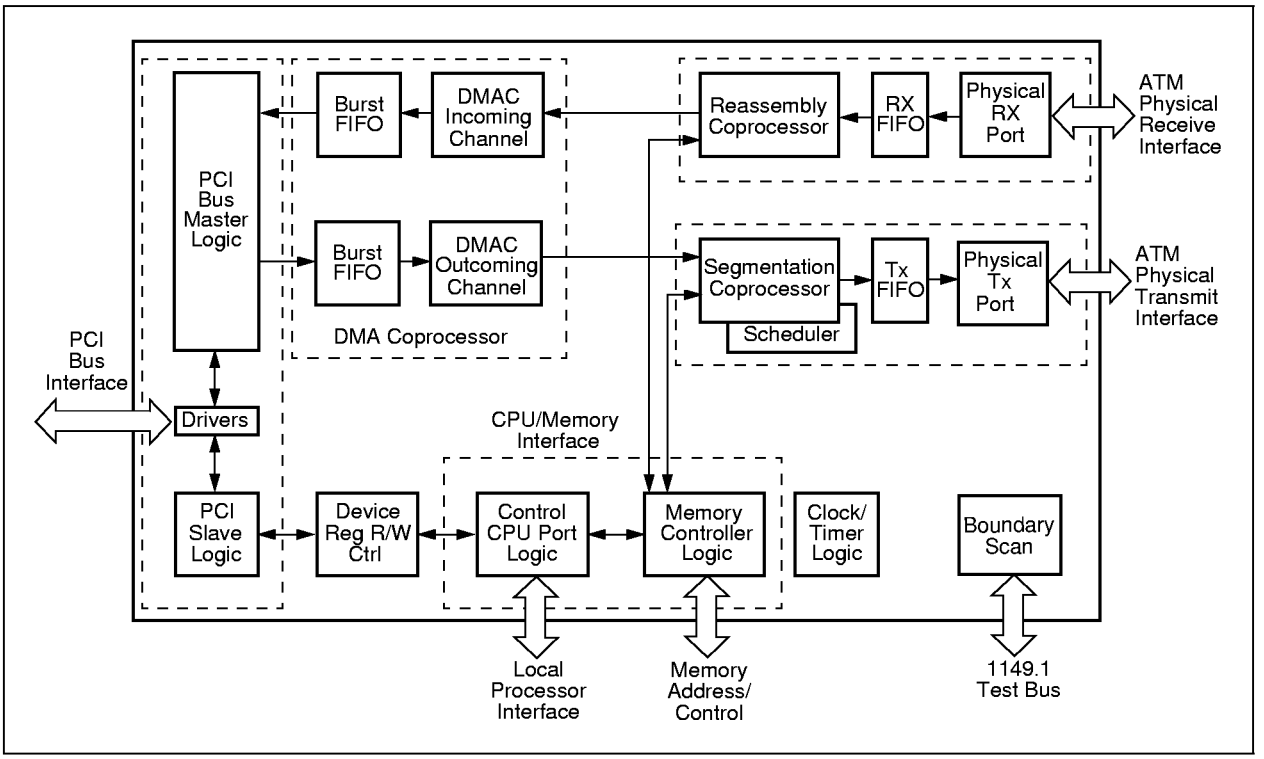
	Pin Label	Signal Name	I/O	Definition
Local Memory Interface	LDATA[31:0]	Memory Data Bus	I/O	Data I/O bus. Used for memory reads and writes as well as Control and Status Register access by the local processor.
	LADDR[18:2]	Memory Address Bus	I/O	Address I/O bus. Used for memory reads and writes as well as Control and Status Register access by the local processor.
	LADDR[1,0]	Memory Address Bus	O	The two least significant bits of address I/O bus. Used for memory reads and writes as well as CSR access by the local processor.
	MCS[3:0]*	Memory Bank Chip Selects	O	Selects one of four addressable banks of SRAM memory.
	MOE*	Memory Read Enable	O	Indicates that a read cycle is proceeding and the memory device output buffers should be enabled, driving data onto the LDATA[31:0] lines.
	MWE[3:0]*	Memory Byte Write Enables	O	Memory byte write enables for by_4 or by_8 SRAMs. For by_16 devices, these outputs are byte enables that are active on writes and reads.
	MWR*	Memory Write	O	Memory write enable for by_16 SRAMs.
	RAMMODE	RAM Mode Select	I	Selects RAM chips supported. 1 = by_16 memory devices. 0 = by_4 or by_8 memory devices.
Clocks and Status	CLK2X	2x Clock Input	I	Double frequency (from SYSCLK) TTL clock input.
	SYSCLK	System Clock	O	This divide by 2 of CLK2X is the internal system clock as well as the external system clock.
	CLKD3	Divide by 3 Clock	O	This output clock is a 50% duty cycle one-third divide of CLK2X; it may be used for the UTOPIA interface clock.
	STAT[1,0]	SRC Status	O	Bt8230 internal status outputs. Internal status controlled by the STAT_MODE[3:0] field in the CONFIG0 Register.
Boundary Scan Test Signals	TRST*	Test Logic Reset	I	When a logic low, this signal asynchronously resets the boundary scan test circuitry and puts the test controller into the reset state. This state allows normal system operation. Tie to ground when boundary scan is not implemented
	TCLK	Test Clock	I	Generated externally by the system board or by the tester. Tie to ground when boundary scan is not implemented.
	TMS	Test Mode Select	I	Decoded to control test operations.
	TDO	Serial Test Data	O	Outputs serial test pattern data.
	TDI	Serial Test Data	I	Input for serial test pattern data.
Supply Voltage	PWR	Power	–	Sixteen pins are provided for supply voltage.
	GND	Ground	–	Twenty-two pins are provided for ground.

2.0 Functional Description

2.1 Overview

The functional description section assumes that the reader is familiar with and has access to the standards listed in Appendix A. The Bt8230 consists of three coupled coprocessors: a DMA coprocessor that performs data transfers to and from the memory of the host system; a segmentation coprocessor, and a reassembly coprocessor; both of which handle high-speed I/O between the Bt8230 and an external ATM Physical Device (PHY) chip. In addition, the Bt8230 contains local processor support logic that allows a local processor to be connected to the Bt8230; a PCI bus interface responsible for accepting read and write commands from the DMA coprocessor (passed through the burst FIFO buffers), and which performs actual data transfers on the PCI bus; and a boundary scan testing facility. Figure 2-1 depicts the overall Bt8230 functional blocks and organization.

Figure 2-1. Bt8230 Functional Block Diagram



2.2 DMA Coprocessor

The Direct Memory Access (DMA) coprocessor is intended to perform high-speed sustained data transfers to and from the host memory space. It is controlled by the segmentation and reassembly coprocessors.

The major functions of the DMA coprocessor are to transfer data from the host memory (through the PCI bus) to the segmentation coprocessor, and to transfer data from the reassembly coprocessor to the host memory space (through the PCI bus).

In all modes of operation, the DMA coprocessor maintains a high level of performance. It uses burst transfers when possible to maximize utilization of the host bus bandwidth, performs byte switching to accommodate misaligned transfers, and carries out concurrent input and output transfers (alternating burst reads and burst writes) to support simultaneous input and output data streams.

2.2.1 DMA Read

For outgoing messages, DMA read cycles move data from host memory to the segmentation coprocessor using a gather DMA method. The maximum burst size is thirteen 32-bit words which correspond to one cell. The burst size can be reduced by setting the MAX_BURST_LEN field in the PCI Configuration Register at a value less than 13 words.

2.2.2 DMA Write

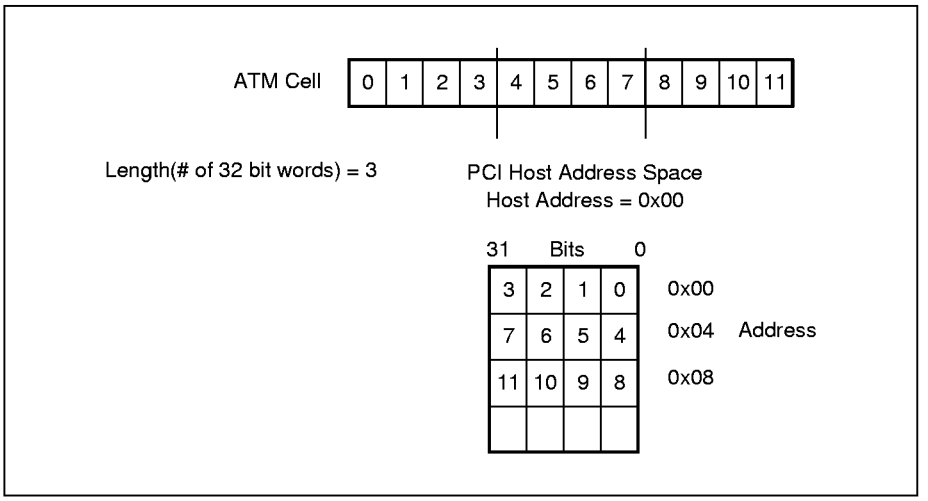
For incoming messages, DMA write cycles move data from the reassembly coprocessor to host memory using a scatter DMA method. The maximum burst size is fourteen 32-bit words which correspond to one ATM cell and a status word appended to performance monitoring cells. The burst size can be reduced by setting the MAX_BURST_LEN field in the PCI Configuration Register at a value less than 13 words.

2.2.3 Misaligned Transfers

The reassembly and segmentation processors handle data internally on word addresses. The DMA coprocessor must be capable of handling transfers from the PCI bus without the same constraint (i.e., with data that is not aligned on word boundaries). In addition, the length of the transfer is specified in bytes, not 32-bit words, even though the data bus widths are all 32 bits.

To facilitate this, byte-switching logic is used within the Bt8230. When the Bt8230 specifies a host address with the LSBs = 00, it is implied that the data is byte aligned. Figure 2-2 shows how a byte-aligned address would map into the PCI host address space for a little endian system. Selecting between big and little endian systems is done through ENDIAN [bit 12] in the Configuration Register 0 [CONFIG0;0x14].

Figure 2-2. Little Endian Aligned Transfer



When the Bt8230 specifies a host address with the LSBs ≠ 00, it is implied that the data is misaligned. Figure 2-3 shows how a misaligned address would map into the PCI host address space for a little endian system.

Figure 2-3. Little Endian Misaligned Transfer

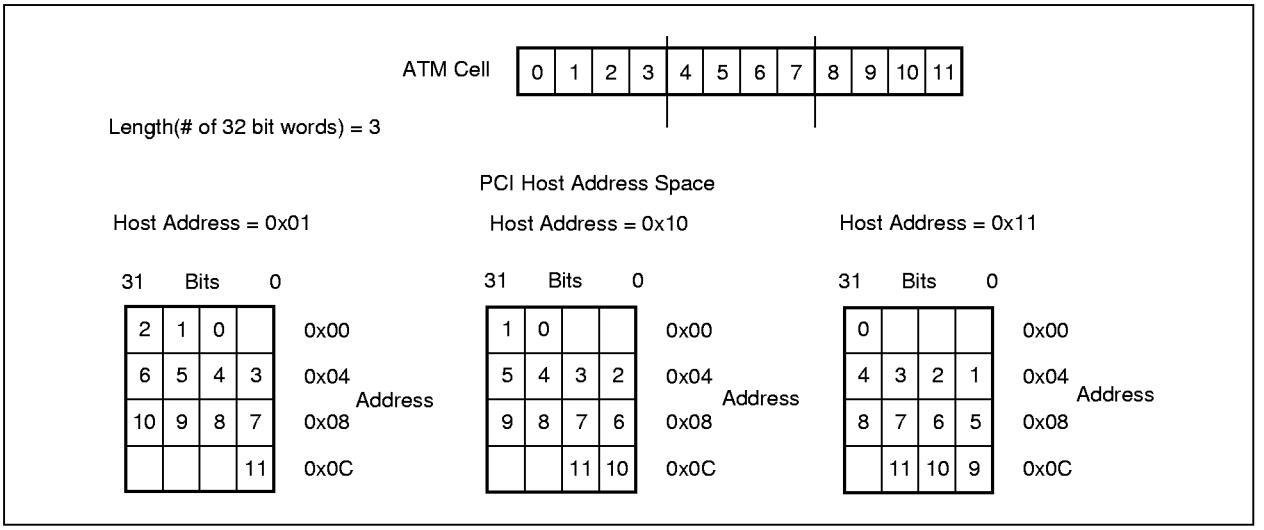
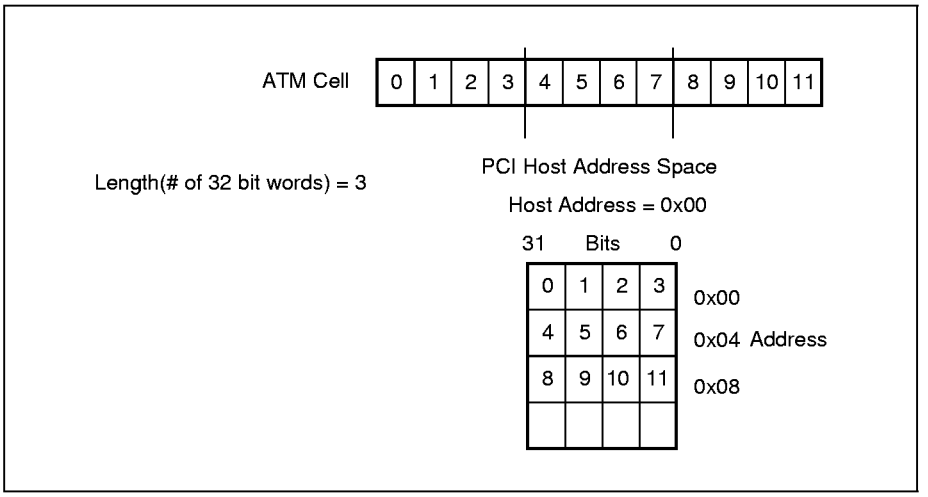


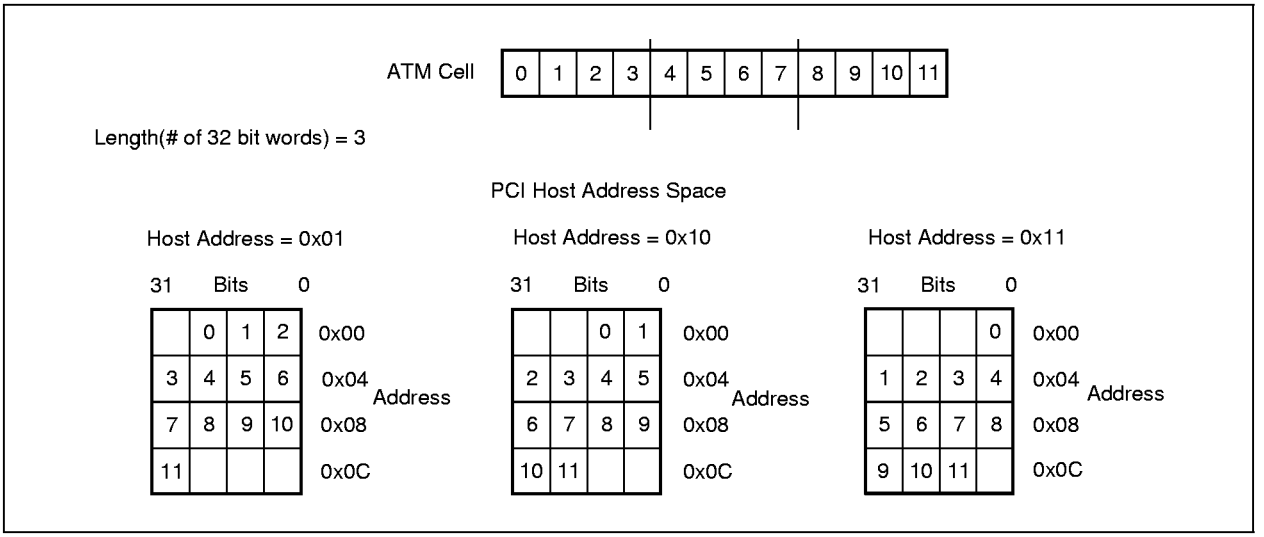
Figure 2-4 shows how a byte-aligned address would map into the PCI host address space for a big endian system.

Figure 2-4. Big Endian Aligned Transfer



When the Bt8230 specifies a host address with the LSBs ≠ 00, it is implied that the data is not aligned. Figure 2-5 shows how an unaligned address would map into the PCI host address space for a big endian system.

Figure 2-5. Big Endian Misaligned Transfer

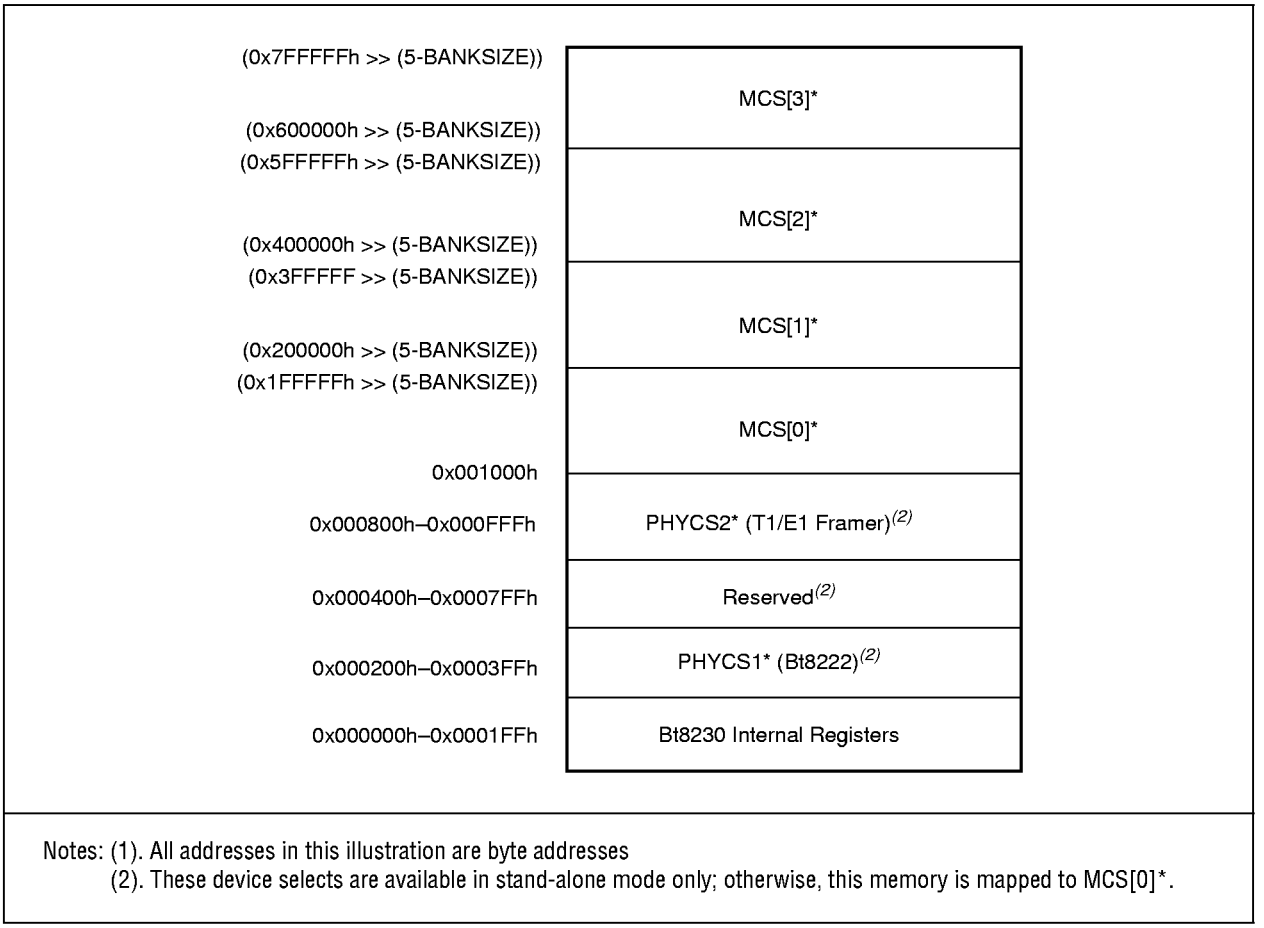


2.3 Memory Interface

To simplify system implementations, the Bt8230 integrates a complete memory controller designed for direct interface to common Static RAMs (SRAMs). The control and status registers, as well as the physical interface devices in the stand-alone mode of operation, are mapped into the bottom of the memory map. Consequently, accesses to these resources are also controlled by the memory controller. Figure 2-6 shows the Bt8230 address map.

Up to 8 MByte of external memory using SRAM devices can be accessed by the Bt8230. The amount of memory required is heavily dependent on the number of Virtual Channel Connections (VCCs) implemented, as well as the number of VCCs that are currently active. Memory requirements are discussed in detail in subsection 2.5.1.

Figure 2-6. Bt8230 Memory Map(1)



Notes: (1). All addresses in this illustration are byte addresses
(2). These device selects are available in stand-alone mode only; otherwise, this memory is mapped to MCS[0]*.

2.3.1 Memory Bank Characteristics

The external memory is organized in 1 to 4 banks of up to 2 MByte each. The system may use any number of banks to fulfill the memory requirements, the only requirement is that the banks must be of the same size and organization. The local processor selects between the banks via the PBSEL[1,0] inputs. BANKSIZE[2:0] [bits 9–7] in the CONFIG0 register denote the size of the memory banks and allow the Bt8230 to incorporate the various bank sizes into contiguous memory. Table 2-1 gives the coding of the BANKSIZE[2:0] control bits.

Table 2-1. Memory Bank Size

BANKSIZE	Bank Memory Organization	Total Bank Size (Bytes)	PBSEL[1,0] Connection	Typical Implementation
111	Reserved			
110	Reserved			
101	512 K x 32	2 M	A[22:21]	Four 512 K x 8
100	256 K x 32	1 M	A[21:20]	Two 256 K x 16, Eight 256 K x 4
011	128 K x 32	512 K	A[20:19]	Four 128 K x 8
010	64 K x 32	256 K	A[19:18]	Two 64 K x 32, Eight 64 K x 4
001	32 K x 32	128 K	A[18:17]	Four 32 K x 8
000	16 K x 32	64 K	A[17:16]	Two 16 K x 16, Eight 16 K x 4

The memory controller is designed to work with standard by_8 and by_4 SRAM devices as well as with by_16 devices. Grounding the RAMMODE input selects the by_4 or by_8 mode of operation, pulling RAMMODE to a logic one selects by_16 operation. When by_16 operation is selected, the MWE[3:0]* outputs become byte enables for both reads and writes. Figure 2-7 shows a typical half MB bank implementation using by_8 SRAM devices. Figure 2-8 shows a typical 1 MB bank using by_16 RAM. To connect different sized RAM banks, simply use more or less address bits, all other control remains the same.

NOTE: The number and type of SRAM chips used affect the address and data bus capacitance and, therefore, the AC timing specifications and the required SRAM speed. Also note that the use of by_4 devices causes more address bus loading than the use of by_8 or by_16 devices. See Chapter 4.0 for detailed timing information.

Figure 2-7. 0.5 MB SRAM Bank Utilizing by_8 Devices

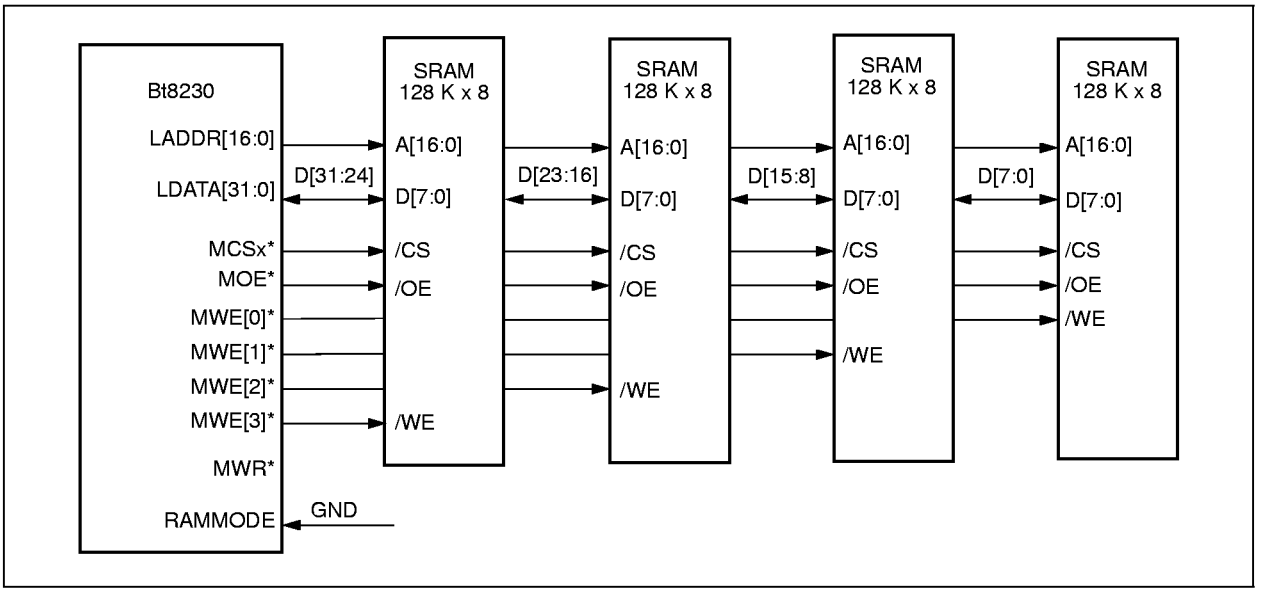
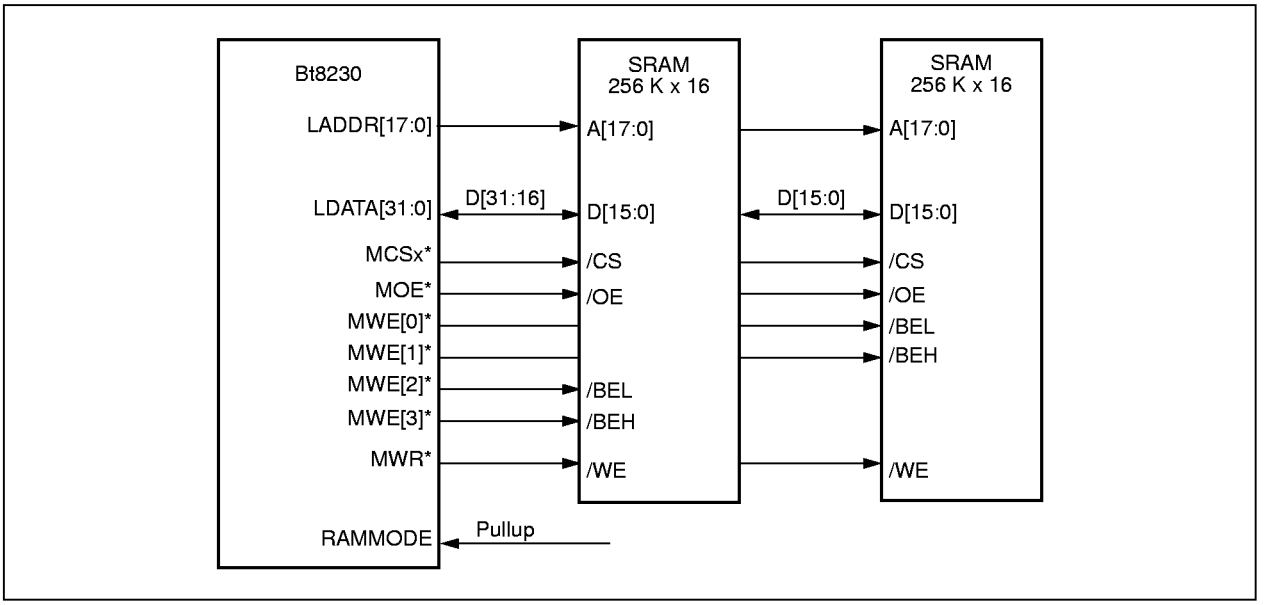


Figure 2-8. 1 MB SRAM Bank Utilizing by_16 Devices



The memory map contains space allocated to the Bt8222 physical interface IC, and to a future Brooktree T1/E1 framer. This mapping is only valid when the PROCMODE input pin is pulled high indicating stand-alone operation with no local processor present. Stand-alone operation is detailed in a subsection 2.4.6. When PROCMODE is logic low and the local processor is present, addresses 0x100h through 0xFFFh are available for general use and are mapped to MCS[0]*.

The MEMCTRL bit in the CONFIG0 Register selects the number of wait states that the memory controller uses to access the SRAM. A logic zero indicates zero wait state or single-cycle memory, while a logic one indicates one wait state or two-cycle memory. The power-on default is MEMCTRL = 1, selecting one wait state or two-cycle memory accesses.

Accesses made to the control registers by the local processor follow the convention for SRAM accesses, that is, either zero or one wait state depending on MEMCTRL programming. Subsequently, the local processor sees no functional timing differences between accesses to registers or SRAM. The internal register accesses from the PCI slave interface are always zero wait state.

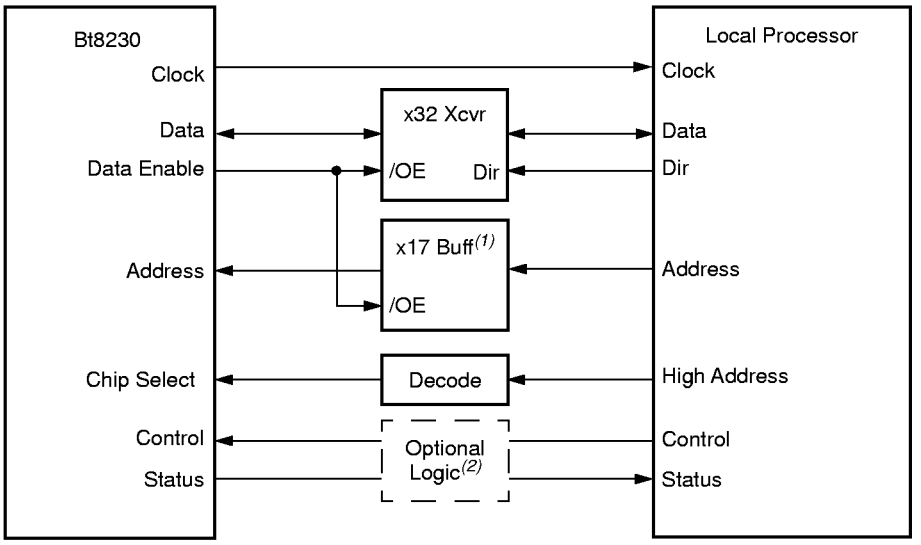
SRAM access time requirements are directly proportional to the system clock speed, as well as the amount and organization of the memory. The required system clock speed for a given application is dependent on the physical line rate, number of VCCs, and the percentage on idle cells versus assigned cells. Refer to subsection 2.4.7 for more information. Memory access times and other requirements are specified at three typical implementations of 1, 2, and 4 banks of by_8 SRAM. In terms of address bus loading, one bank of by_8 SRAM equals one-half bank of by_16 or two banks of by_4. In this way, the system designer may choose the appropriate SRAM characteristics to suit the amount of memory and organization required for the application. See Chapter 4.0 for timing information.

2.4 Local Processor Interface

2.4.1 Overview

The Bt8230 integrated circuit may be used in conjunction with an external processor that performs initialization, link management, monitoring, and control functions. The local processor interface consists of a "loosely coupled" architecture where it interfaces to the Bt8230 through bidirectional transceivers and buffers that are controlled by the local processor and the Bt8230, as shown in Figure 2-9. This architecture allows the processor access to all of the Bt8230 local SRAM memory and control registers, while insulating the Bt8230 from processor instruction and data cache fills. This also allows the local processor the option to control multiple Bt8230 and/or physical interface devices.

Figure 2-9. Bt8230—Local Processor Interface



Notes: (1). Required to address full 8 MByte range, typically fewer address lines are used.
(2). Required for non-i960 processors.

The processor interface is a generic synchronous interface based on the Intel i960CA 32-bit architecture, and is completely compatible with the i960CA/CF and the new i960Jx processors. Other synchronous and asynchronous processors (e.g., from Motorola, AMD, IDT) can be interfaced using external circuitry. The only requirement is that the processor have a 32-bit bus and that the control signals be synchronized to SYSCLK.

To access the Bt8230 local memory or control registers, the processor must arbitrate with the Bt8230 for access to the memory controller. Due to the requirements of reassembly and segmentation access to SRAM, and the implications of PCI bus utilization, the local processor has the lowest priority in the memory arbitration scheme. Since the local processor is typically used for low bandwidth supervision and maintenance functions, this should be acceptable.

When the local processor accesses the Bt8230's control registers or local memory, a local processor memory request is generated internal to the Bt8230. The memory arbiter then coordinates this request with requests from other memory consumers and grants the memory bus to the local processor at the appropriate time. The local processor is held off during this process by the insertion of a variable number of wait states; accomplished by the i960 withholding READY* or RDYRCV*. Once the local processor is granted the memory system, the transceivers are enabled to allow the local processor's address and data to access the SRAM or control registers. The conclusion of the data transaction is signaled by the assertion of PRDY*. Wait states may be inserted by the processor at any time by asserting PWAIT*. The last data cycle in a burst is indicated by the PBLAST* signal. In this manner, non-i960 processor half-speed buses or slow transceivers can be accounted for.

The LP_BWAIT bit in the CONFIG0 Register will automatically add a single wait state between the first access in a burst and subsequent accesses. This can be used to simplify the design of memory controllers for processors that do not produce a wait output and which require more time between data cycles in a burst.

2.4.2 Interface Pin Description

The local processor bus interface consists of the control, address, and status signals described in Table 2-2. Reference Table 1-1 and Figure 2-15, i80960CA/CF interface description for the following discussion

Table 2-2. Processor Interface Pins *(1 of 2)*

Signal	Dir(1)	Description
PROCMODE	I	Processor interface mode select input—A logic low on this input enables the local processor mode of operation.
PCS*	I	Processor interface chip select—A logic low on this signal in conjunction with a logic low on PAS* at the rising edge of SYSCLK initiates a memory request to the memory controller.
PAS*	I	Processor address strobe— A logic low on this signal in conjunction with a logic low on PCS* latches the value of PWNR, PBSEL[1,0], PADDR[1,0], and PBE[3:0]* at the rising edge of SYSCLK.
PWNR	I	Processor write/read select—A logic one on this input indicates a write cycle, a logic zero indicates a read cycle. Latched at rising edge of SYSCLK when PAS* and PCS* are active.
PADDR[1,0]	I	Word select address inputs—Indicates the word address for a single cycle access, or the first word for a multi-cycle burst access. Latched at rising edge of SYSCLK when PAS* and PCS* are active.
PBSEL[1,0]	I	Bank select inputs—Decode to select MCS[3:0]*; see Figure 2-6 for details. Latched at rising edge of SYSCLK when PAS* and PCS* are active.

Table 2-2. Processor Interface Pins (2 of 2)

Signal	Dir(1)	Description
PBE[3:0]*	I	Byte select inputs—Active low. Allows individual bytes of selected word to be written. Not active on reads. Latched at rising edge of SYSCLK when PAS* and PCS* active. PBE[3]* controls writes to LDATA[31:24], PBE[2]* controls LDATA[23:16], etc.
PWAIT*	I	Processor wait input—Allows processor to insert variable number of wait states to extend memory transaction. Must be active on rising edge of SYSCLK with PRDY* active to insert wait cycle. May be used to interface to half speed or slow processor bus or to allow the use of slow transceivers. If the insertion of wait states is not required, set this input to a logic high.
PBLAST*	I	Processor burst last input—Indicates the last word of a cycle. Must be active on rising edge of SYSCLK with PRDY* active to indicate last cycle. If burst accesses are not required, this signal should be set to a logic low.
PRDY*	O	Processor interface ready signal—A logic low on this signal at rising edge of SYSCLK indicates that the present cycle has been completed. If a read cycle, the data is valid to latch by the processor; if a write cycle, the data has been written and may be removed from the bus. When PRDY* is active, wait states may be inserted with PWAIT*, or a single or burst cycle may be terminated by PBLAST*(2).
PDAEN*	O	Processor data and address bus enable—Connects to output enable input of bidirectional transceiver and buffer to enable data and address for processor cycles and disable otherwise. May also be used to indicate that the processor has successfully arbitrated for access to the memory controller(3).

Notes: (1). Direction given with respect to the Bt8230.
(2). This output corresponds to the READY* or RDYRCV* input in the i960 architecture.
(3). The processor system is responsible for controlling the direction of the bidirectional data bus transceiver. In the i960 architecture, this may be controlled by the DT/R* signal.

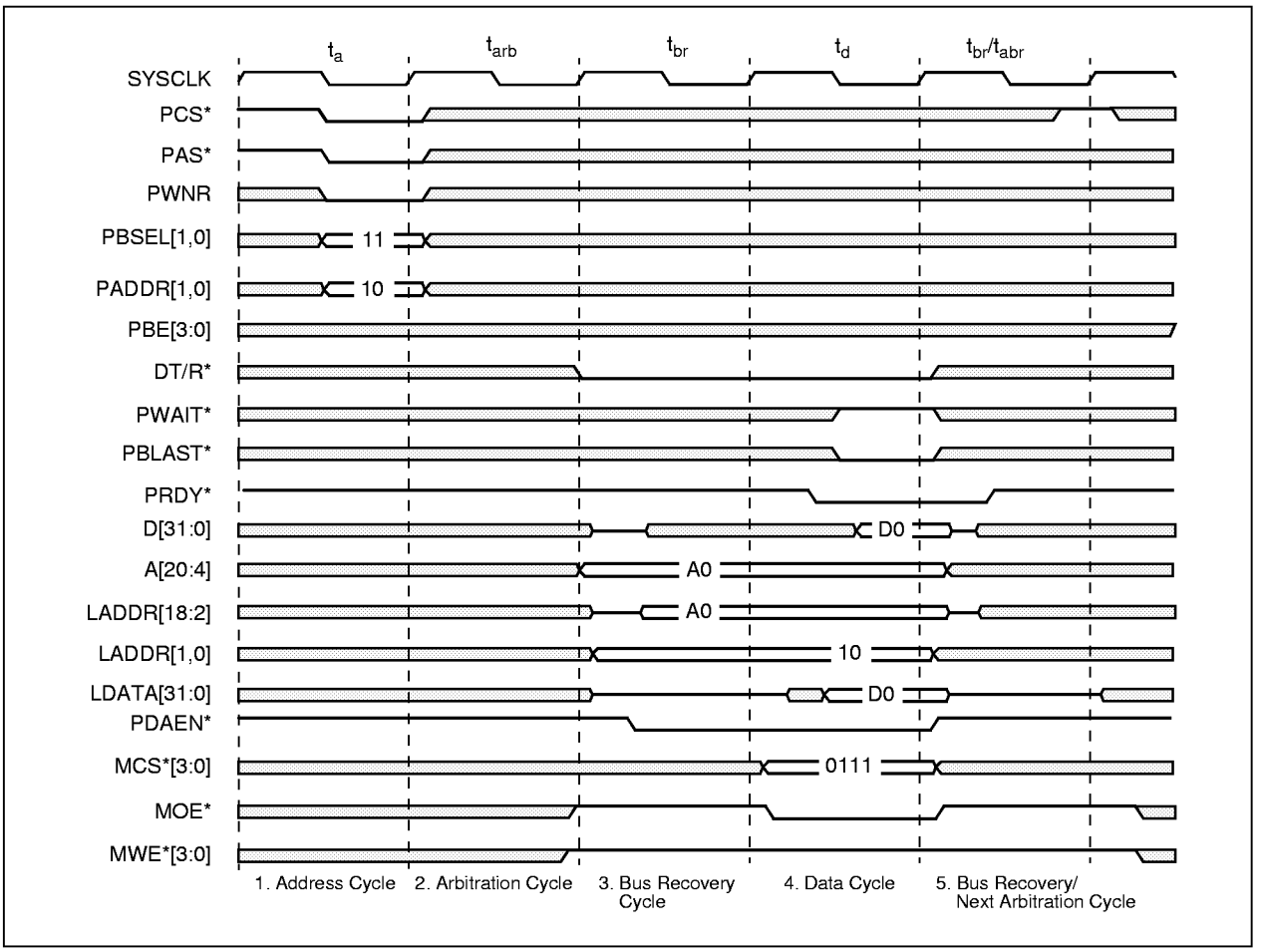
2.4.3 Bus Cycle Descriptions

Throughout the bus cycle descriptions, cycle refers to a single SYSCLK cycle ending with a rising edge. An arbitration cycle is one in which the memory requests from the local processor and internal memory consumers are compared and the one with the highest priority is granted the memory access on the next cycle. A memory access that was previously arbitrated may occur on an arbitration cycle. Once the local processor has successfully acquired the memory controller, it holds the bus until it is relinquished by the assertion of PBLAST* on the last data cycle. Therefore, local processor burst transfers will always be completed and may theoretically be of arbitrary length. However, in practice, burst transfers should be limited to four or less. The maximum arbitration delay for a local processor access is on the order of 20 cycles; however, it will typically be from 1 to 4 cycles. This parameter is heavily influenced by the SYSCLK frequency, line rate, number of VCCs, idle cell ratio, and SRAM access speed. Therefore, a system design in which local processor accesses must occur within a fixed time period is not recommended.

2.4.3.1 Single Read Cycle, Zero Wait State Example

Figure 2-10 illustrates a single read cycle with zero wait states. During the address cycle (cycle 1) at the rising edge of SYSCLK with PCS* and PAS* active, a memory request is generated by the processor interface circuitry. Also at this time, the read/write select, bank select, and word select inputs (PWNR, PBSEL[1,0], and PADDR[1,0]) are internally latched. Note that the byte enables (PBE[3:0]*) are don't cares during reads. During cycle 2, this local processor memory request is processed by the memory arbitration circuitry. If no other memory consumers request an access on the same cycle, the local processor is granted access on cycle 3. However, to take into account bus transceiver turn-around, cycle 3 is always a wait or bus recovery state which gives sufficient time for the address from the processor to access the SRAM. For zero wait state SRAM, unless a wait state is inserted by the processor, the data is available to be latched into the processor on cycle 4, which is indicated by the assertion of PRDY*. Cycle 5 is an arbitration cycle for the internal memory consumers which may have requested access during the processor access. It also serves as a bus recovery cycle for the processor. Note that once the PCS*, PAS*, PWNR, PBSEL[1,0], and PADDR[1,0] are sampled at cycle 1, they are don't cares for the remainder of the access. Also note that DT/R* is an output supplied by the local processor to indicate the direction of the data transceivers, and that the Bt8230 PDAEN* signal is active to enable data and address.

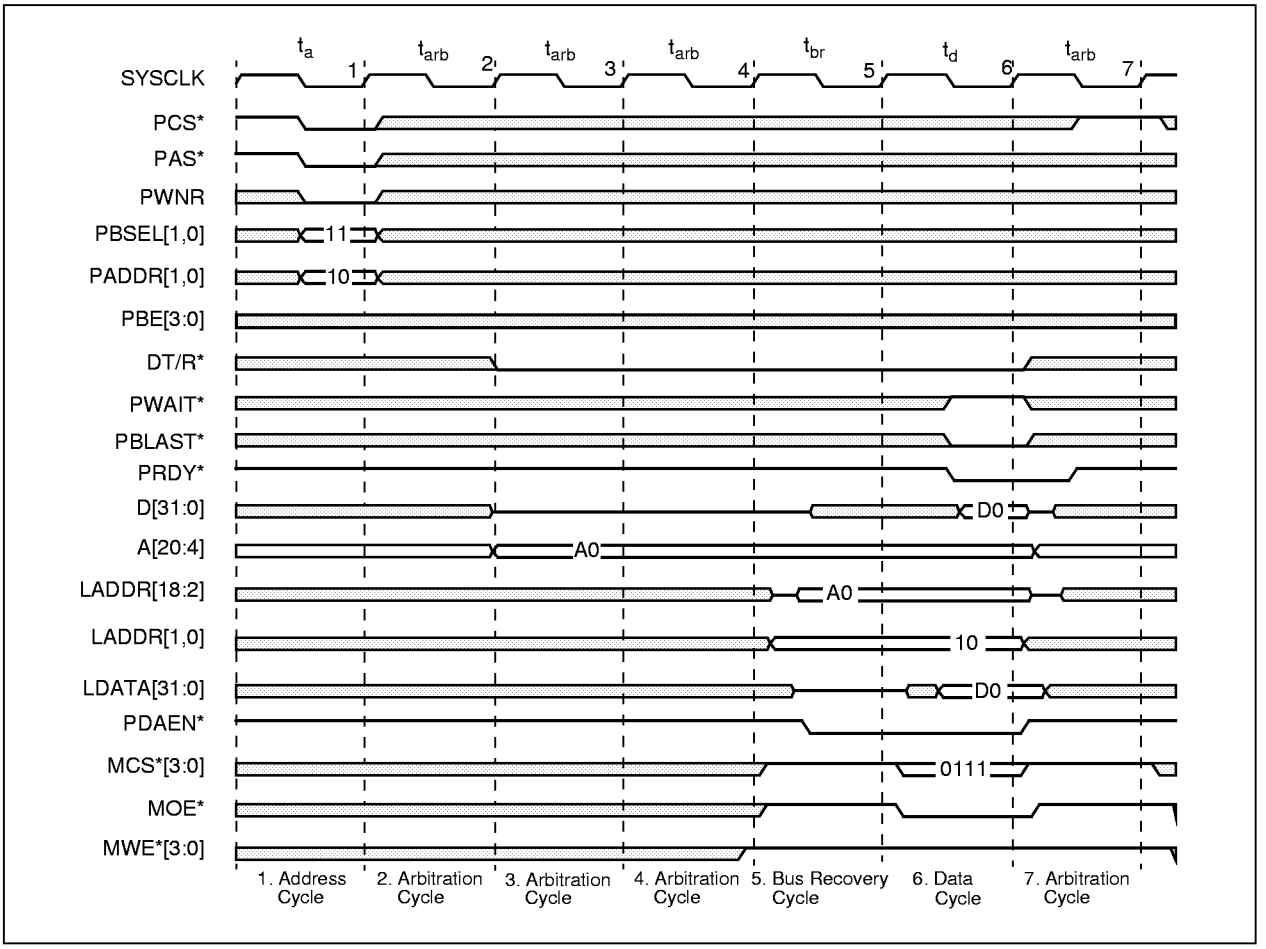
Figure 2-10. Local Processor Single Read Cycle



2.4.3.2 Single Read Cycle, Wait States Inserted By Memory Arbitration

Figure 2-11 illustrates a local processor single read cycle with arbitration wait states. This example is similar to the preceding one, except that here the local processor is not able to access the RAM immediately because of higher priority memory requests on cycles 2 and 3. On cycle 4, the memory controller allows the local processor access to the address and data bus and the transaction takes place at the end of cycle 6.

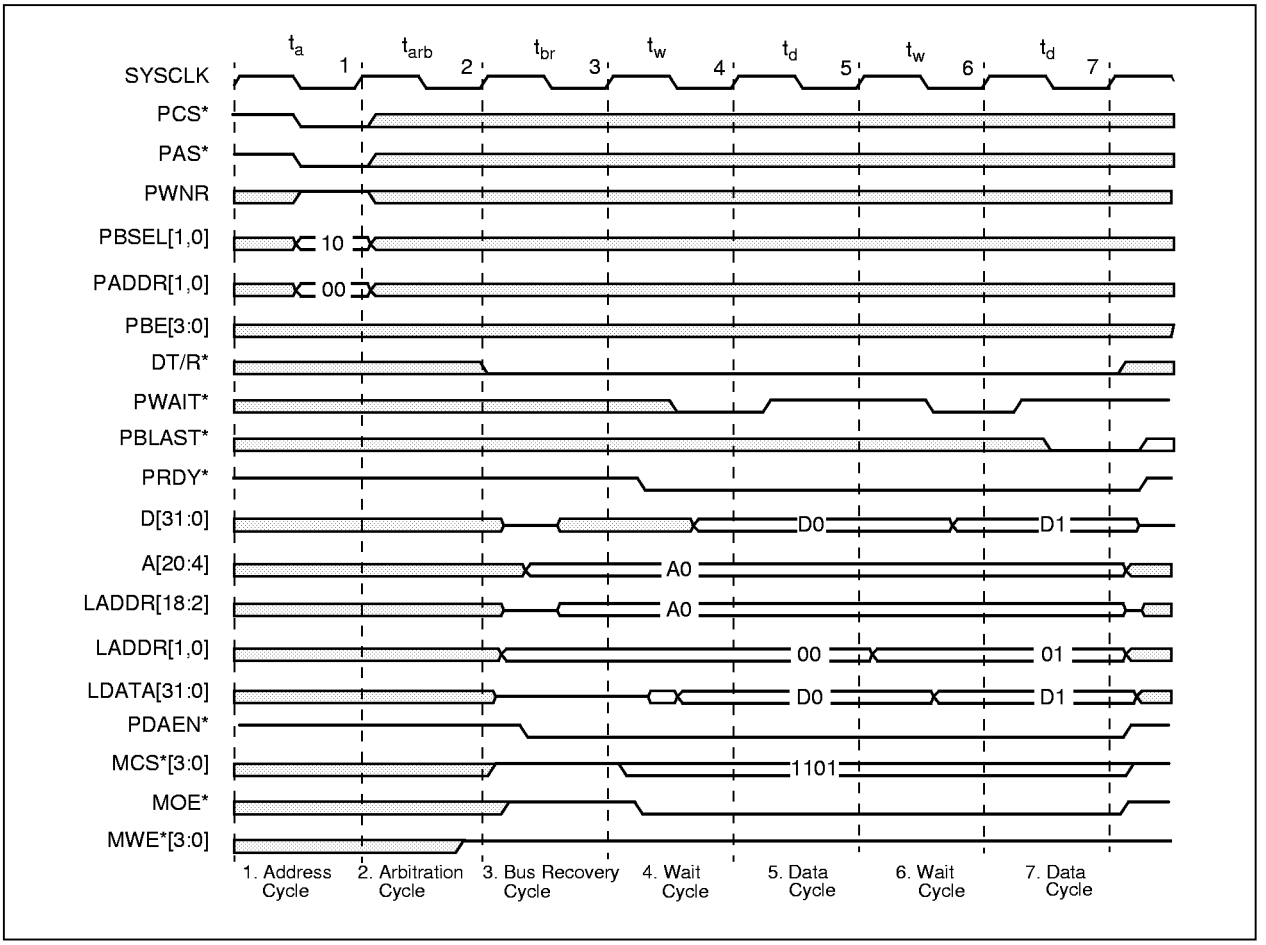
Figure 2-11. Local Processor Single Read Cycle with Arbitration Wait States



2.4.3.3 Double Read Burst With Processor Wait States

In Figure 2-12 the processor inserts wait states on cycle 4 and cycle 6 to allow additional time for the reads to occur. At the rising edge of SYSCLK on cycle 4 and cycle 6, the combination of PWAIT* low and PRDY* low extends the read by one more cycle. Note that the local processor word select inputs (PADDR[1,0]) are latched at cycle 1, and the Bt8230 word select address lines, LADDR[1,0], are incremented automatically at the beginning of cycle 6.

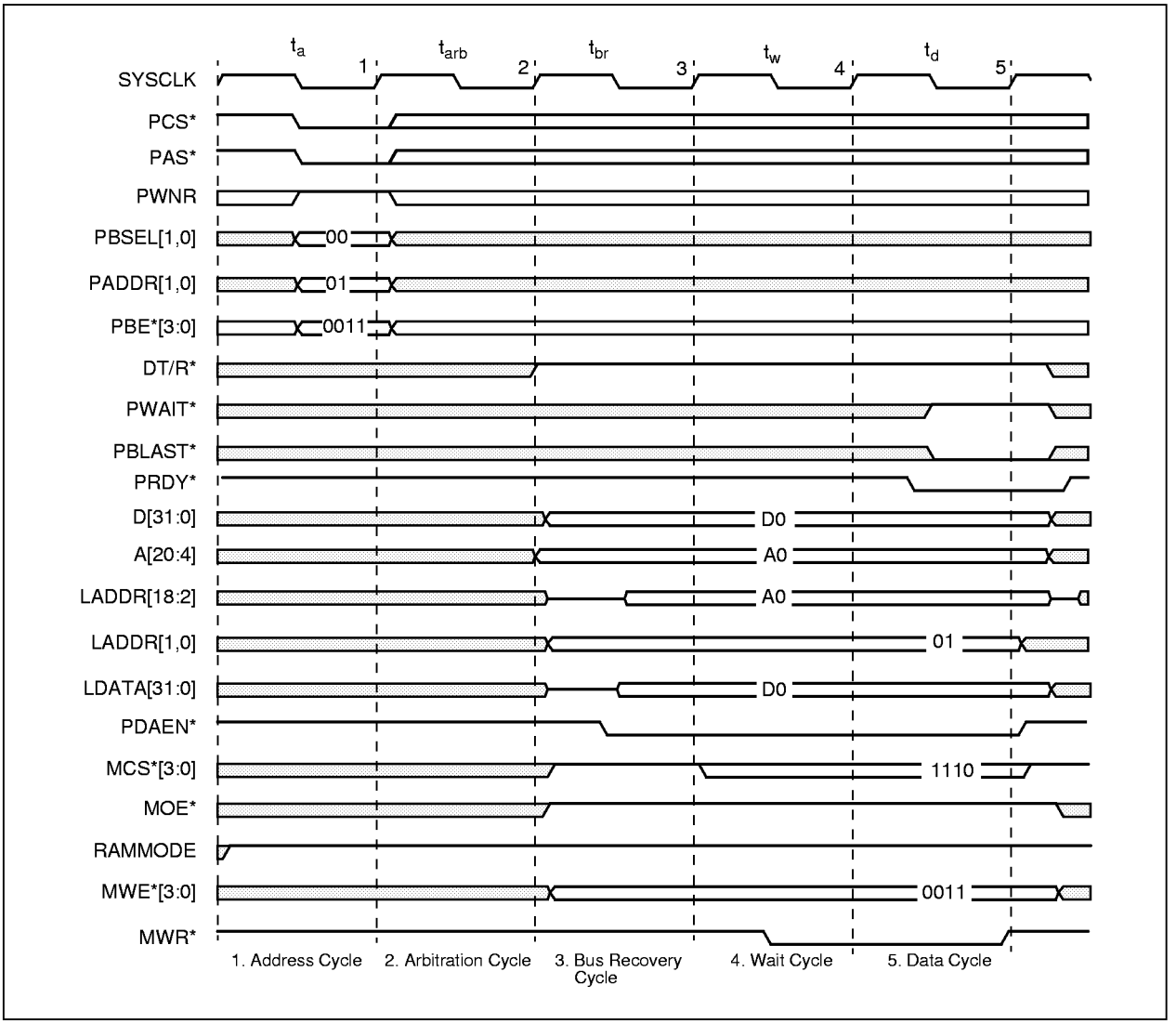
Figure 2-12. Local Processor Double Read with Wait States Inserted



2.4.3.4 Single Write With One-Wait-State Memory

In Figure 2-13, the local processor performs a single write into one-wait-state memory. There is no arbitration delay. Note that RAMMODE is a logic high, indicating that by_16 RAM is used. Here the PBE[3:0]* inputs are latched at cycle 1 and are used to select the byte enables that are active during the cycle when MWR* is active. In this case, the two most significant bits are active, indicating a 16-bit write to the two most significant bytes.

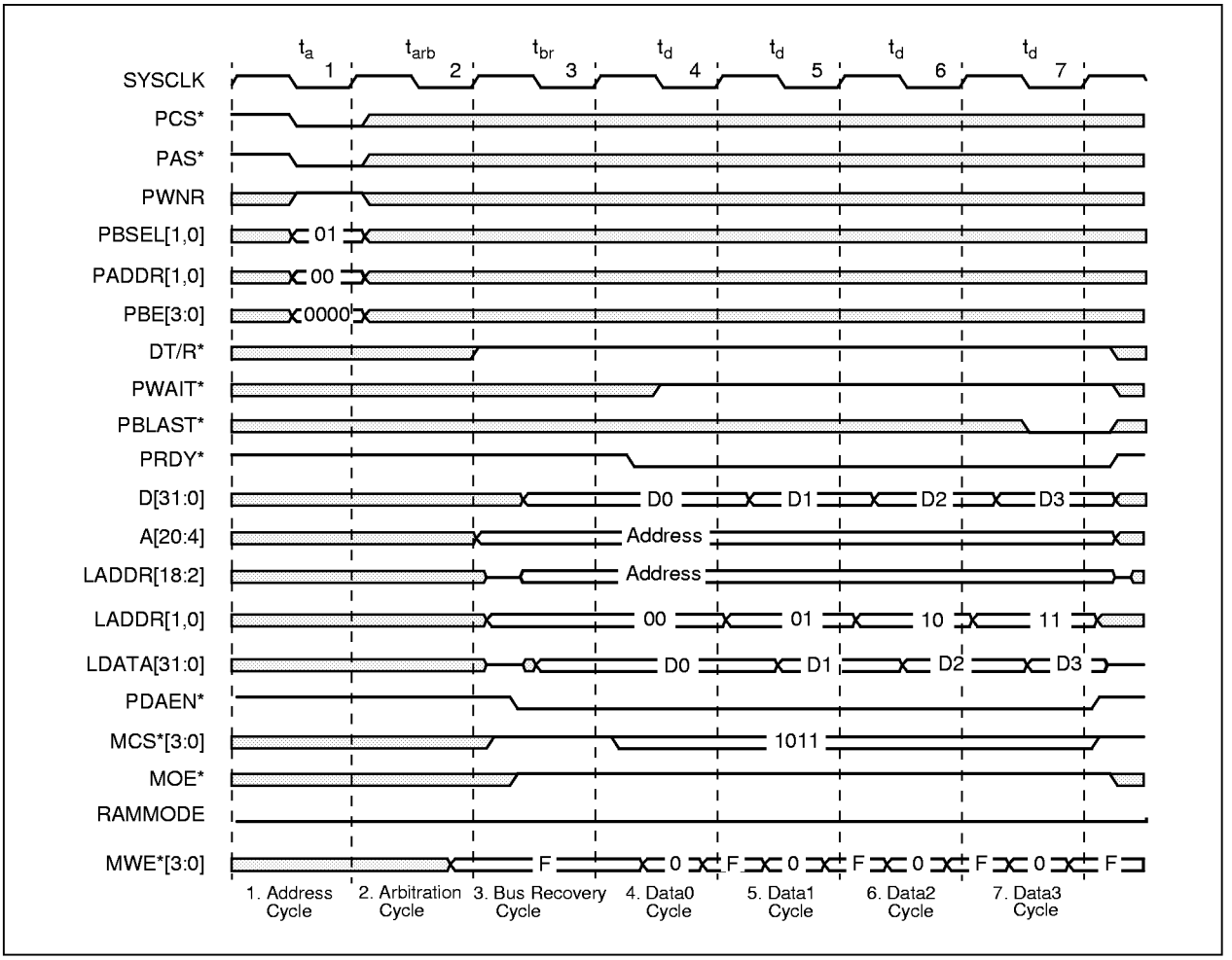
Figure 2-13. Local Processor Single Write with One Wait State by_16 SRAM



2.4.3.5 Quad Write Burst, No Wait States

In Figure 2-14, a quad burst write access to zero-wait-state memory is shown. RAMMODE is logic low, selecting by_8 or by_4 RAM mode. Here PBE[3:0]* is latched on cycle 1, indicating that the write is active on all bytes and the MWE*[3:0] outputs are active as write strobes while MWR* is not used. Note that the local memory word select addresses, LADDR[1,0], are incremented automatically by the Bt8230 on each successive write cycle. Although the i960 architecture has the limitation that a quad word transfer must start on a quad word boundary, the Bt8230 does not have that limitation, the PADDR[1,0] bits may be any value and are incremented as long as the burst transfer proceeds.

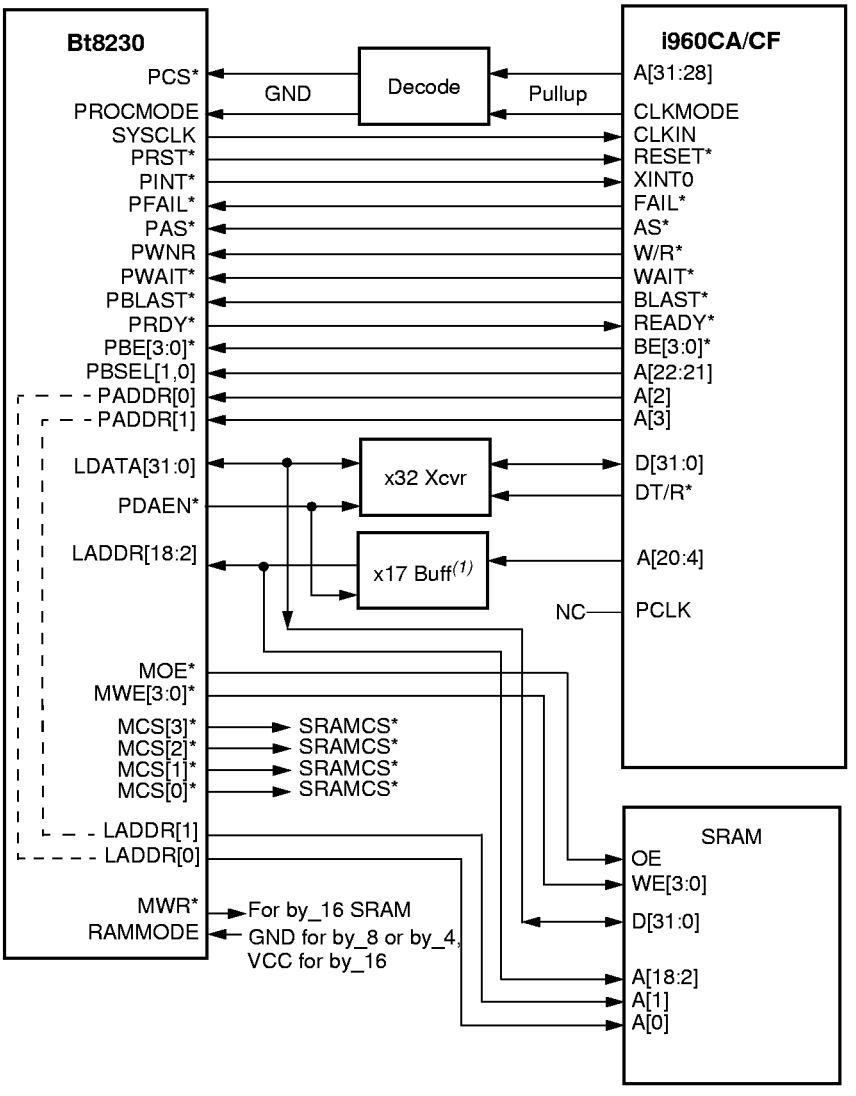
Figure 2-14. Local Processor Quad Write, No Wait States



2.4.4 i80960CA/CF Processor Interface

Figure 2-15 illustrates the signal interface between the Bt8230 device and the i960CA/CF processor. The memory region decoded for PCS* should be set for N_{RAD} and N_{WAD} = 2, N_{RDD} and N_{WDD} = 0 or 1 depending on the use of zero or one wait state SRAM, and N_{XDA} = 1. In addition, external ready control must be enabled and burst may be enabled or disabled at the system designers option. Pulling up the i960 CLKMODE input to a logic one selects the divide-by-one clock mode, making i960 PCLK synchronous to SYSCLK.

Figure 2-15. i960CA/CF to the Bt8230 Interface



Note (1): Required for full 8 MByte address range

This configuration is for addressing the entire 8 MB of SRAM. In the majority of systems, the SRAM requirements will be considerably less. The implications of this are that the PBSEL[1,0] inputs may be driven by lower order address lines and there will be less than 17 address lines to buffer. Therefore, in most applications, the data transceivers may utilize two x16 parts, such as a 74ABT16245, and the address buffer may utilize a single x16 74ABT16244.

NOTE: The i960CA/CF signals a failure of its internal self-test upon reset or power-up by asserting its FAIL* output. This line is connected to the PFAIL* pin of the Bt8230, and the status of this pin is reflected in the Host Interrupt Status Register [HOST_ISTAT0;0xC0].

2.4.5 i80960Jx Operating Mode

The major difference between the i80960Jx processor and the i80960CA is that the Jx utilizes a multiplexed address/data bus structure while the CA/CF is non-multiplexed. However, in the Bt8230 system. The demultiplexing of address/data takes place on the processor side of the address buffers and, therefore, does not affect the Bt8230. Otherwise, the Jx has the same bus control signals as the CA/CF with the exception of the WAIT* signal, which the Jx does not possess. The insertion of wait states, if required, must be accomplished by an external memory controller which in any case, is required for a Jx implementation.

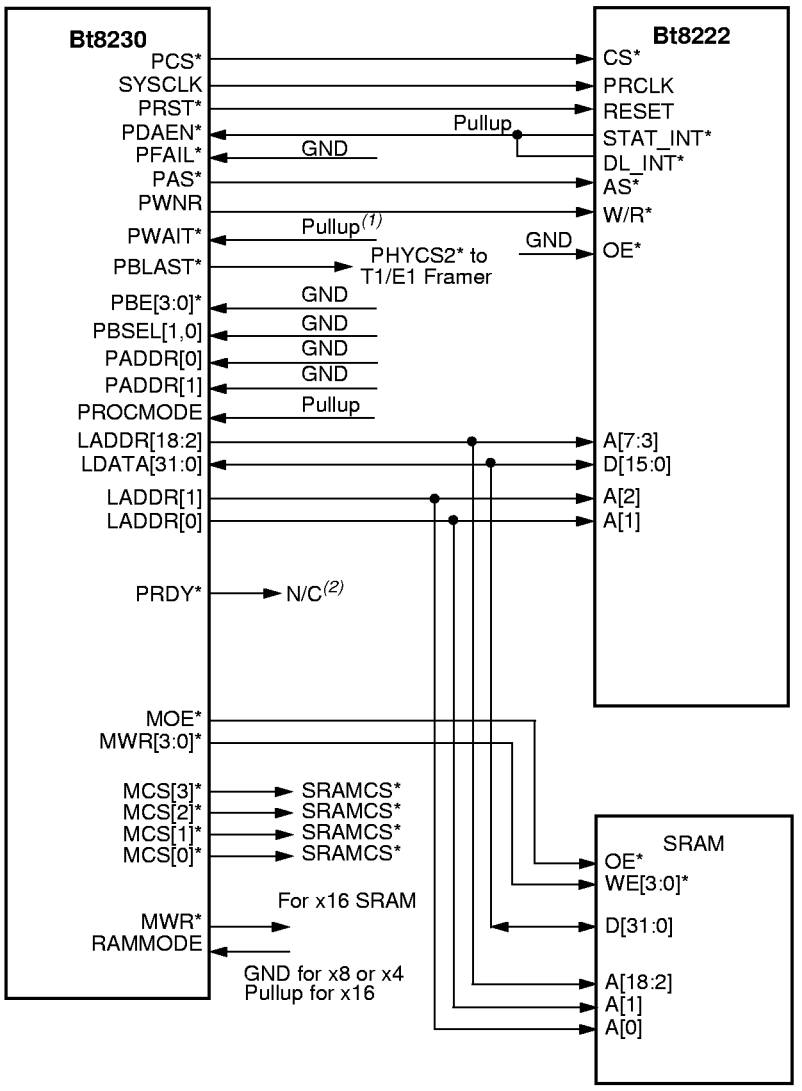
2.4.6 Stand-Alone Operation

Stand-alone interface pins and descriptions are given in Table 2-3. Figure 2-16 shows the signal interface between the Bt8230 and the Bt8222 ATM receiver/transmitter device with no local processor. The PCS*, PAS*, and PWNR pins are now outputs providing chip select, address strobe, and write/read control to the Bt8222. PDAEN* is now an input connected to the interrupt sources of the Bt8222. PBLAST* is a second chip select which may be used to connect a future Brooktree T1/E1 framer since the Bt8222 does not contain this function. The PRDY* output is active and indicates the cycles in which the data transaction occurs. The PWAIT* input is active and may be used to prolong the cycle as shown in Figure 2-17. Physical interface devices other than the Bt8222 may be connected by using PWAIT* to extend the read or write cycle and by using external logic to translate the Bt8230 control signals.

Table 2-3. Stand-Alone Interface Pins

Signal	Dir(1)	Description
PROCMODE	I	Processor interface mode select input. A logic one enables stand-alone operation without a local processor.
PCS*	O	Chip select output for PHY device number 1. Synchronous to SYSCLK. See Figure 2-6.
PBLAST*	O	Chip select output for PHY device number 2. Synchronous to SYSCLK. See Figure 2-6.
PAS*	O	PHY address strobe. Synchronous to SYSCLK.
PWNR	O	PHY write/read select. A logic one on this output indicates a write cycle, a logic zero indicates a read cycle. Synchronous to SYSCLK.
PRDY*	O	PHY interface ready signal. A logic low on this signal at rising edge of SYSCLK indicates that the data cycle has been completed
PWAIT*	I	PHY wait input. Allows external logic to insert wait states to extend data cycles. Only active when PRDY* is active.
PDAEN*	I	PHY interrupt input, active low, level sensitive(2).
PADDR[1,0]	I	Not used, pull to logic zero.
PBSEL[1,0]	I	Not used, pull to logic zero.
PBE[3:0]*	I	Not used, pull to logic zero.
Notes: (1). Direction given with respect to the Bt8230. (2). See the HOST_ISTAT0 Register for details.		

Figure 2-16. Bt8230 to the 8222 Microprocessor Interface (Stand-Alone Operation)



Notes: (1). May be driven by external circuitry to extend cycles.
(2). May be used by external circuitry.

Figure 2-17. Bt8230/PHY Functional Timing with Inserted Wait States

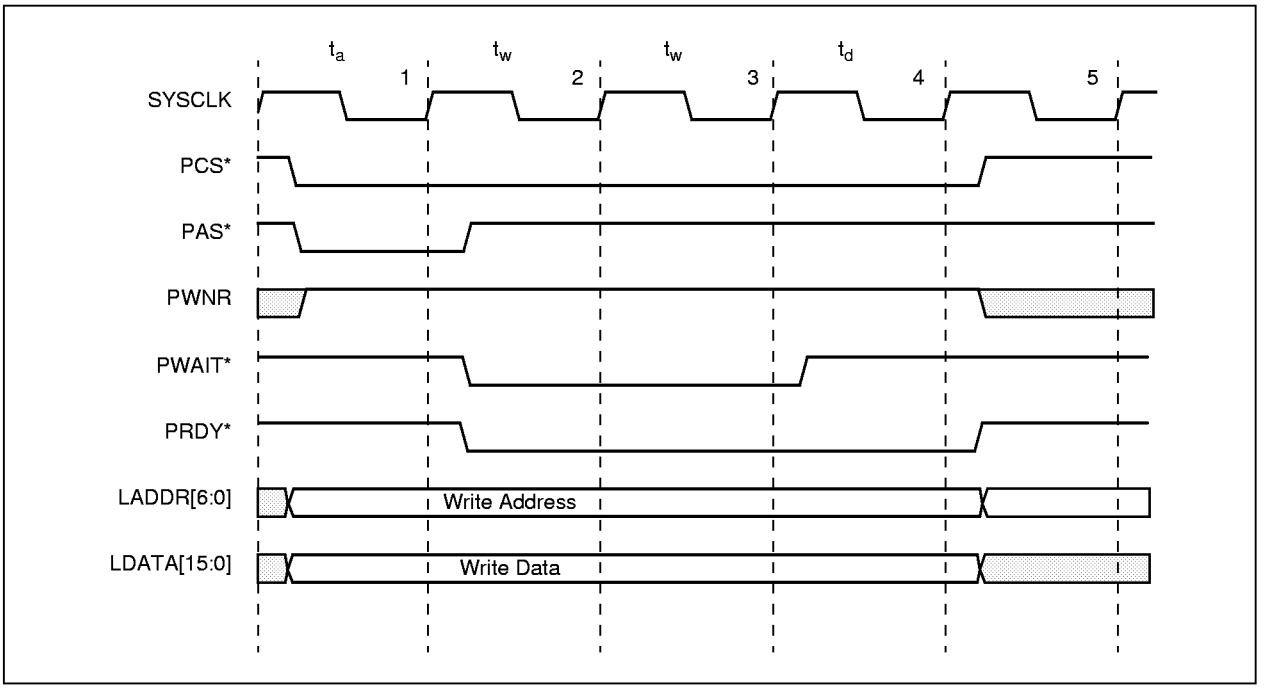
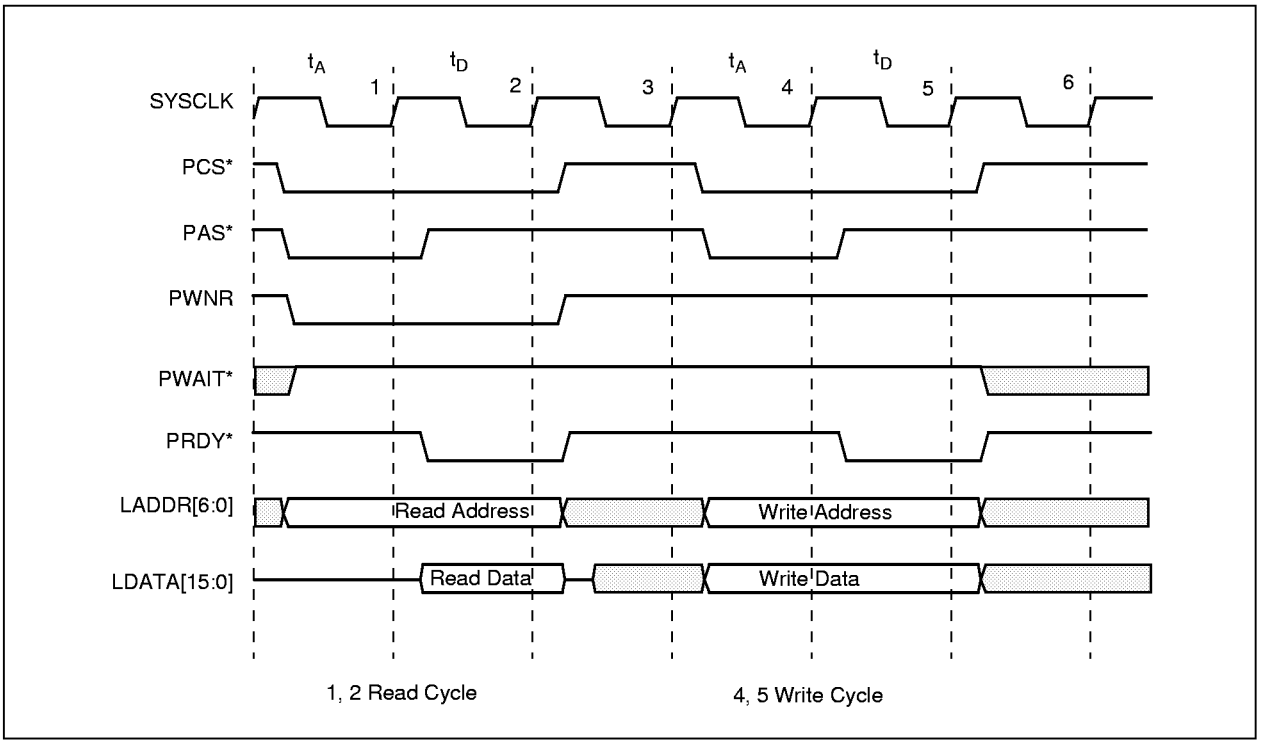


Figure 2-18 shows a read and write Bt8222 access. At cycle 1 (the rising edge of SYSCLK) the Bt8222 samples PCS*, PAS*, and PWNR low, indicating a read cycle. By the next rising edge of SYSCLK at cycle 2, the data is output by the Bt8222 to be latched by the Bt8230. The same procedure occurs for a write except that at cycle 4, PWNR is sampled high. The Bt8222 then latches the data to the appropriate internal register on the next SYSCLK rising edge at cycle 5.

Figure 2-18. Bt8230/Bt8222 Read/Write Functional Timing



2.4.7 System Clocking

The Bt8230 derives all of its timing from a 2x clock input, CLK2X. This clock is internally divided by 2 to create the system clock, SYSCLK. This system clock is used internal to the device and is output to the system to provide the clock to an external processor or PHY device. All processor interface signals are synchronous to SYSCLK. In addition, a CLK2X asymmetrically divided by 3, CLKD3, output is provided and may be used as the clock for the UTOPIA ATM physical interface. Alternatively, SYSCLK may be used as the clock source for the UTOPIA ATM physical interface if the frequency is 25 MHz or less. In either case, the clock signal would be looped externally to the FRCTRL input. For example, if CLK2X is 66 MHz, then CLKD3 is 22 MHz and is suitable for the UTOPIA interface. If CLK2X is 50 MHz, then SYSCLK is 25 MHz and is suitable for the UTOPIA interface. The CLK2X frequency required for a given application is a function of the physical line rate, number of VCCs, active concurrent VCCs, and the SRAM cycle time.

2.4.8 Real-Time Clock Alarm

A real-time clock counter and alarm registers are built into the Bt8230. This real-time clock consists simply of a 7-bit prescaler (configured via the DIVIDER field in the CONFIG0 Register) that accepts the SYSCLK input and outputs a constant (nominally 1 MHz) pulse train, and a 32-bit read/write counter (the Real Time Clock Register [CLOCK;0x00]) that counts the number of pulses output by the prescaler since the system was initialized. When the prescaler is set to generate a 1 MHz pulse train, the CLOCK counter counts in 1 µs intervals. An interrupt is generated when the CLOCK counter overflows, i.e., more than 2^{32} pulses have occurred since it was cleared to zero. If this happens, the CLOCK counter simply wraps around to zero and starts counting over. The control processor or host software is responsible for noting the overflow.

One simple real-time alarm is implemented in the Bt8230. This consists of the Alarm Register 1 [ALARM1;0x04] which is continuously compared to the Clock Register [CLOCK;0x00]. When a match is detected, the corresponding interrupt is generated to the local or host processors. Either processor may then respond to this interrupt and reload a new value into the ALARM1 Register.

2.4.9 Bt8230 Reset

The Bt8230 must be reset by the host processor prior to system initialization for proper operation. This can be done in one of two ways: by asserting the external HRST* pin (which is normally connected to the system power-up reset circuitry), or by setting the GLOBAL_RESET bit in the Configuration Register 0 [CONFIG0;0x14]. The HRST* pin must be deasserted and GLOBAL_RESET must be cleared before beginning the Bt8230 initialization process.

Asserting the HRST* input pin automatically causes the local processor reset pin to be driven active, resetting the local processor. The reset to the local processor will stay active until the LP_ENABLE bit in the CONFIG0 Register is set to a logic high. When using the GLOBAL_RESET bit, the processor must manually set or clear the appropriate bits in the CONFIG0 Register.

2.5 Segmentation Coprocessor

The segmentation coprocessor is responsible for segmenting host or local data buffers into ATM cells for transmission. For each ATM cell time slot, an internal scheduler determines if there is a connection that should send a cell or insert an idle cell. The payload for a cell may be read from host memory across the PCI bus or from local Bt8230 memory. The segmentation coprocessor adds the ATM header and any programmed adaptation layer formatting. The cell is then added to the transmit FIFO for eventual transmission. The segmentation coprocessor is capable of generating all CPCS-PDU overhead for both AAL3/4 and AAL5.

Most segmentation traffic should originate in host memory buffers. The option to segment from a Bt8230 local memory buffer is intended to allow a local processor to generate maintenance and signaling messages in local memory.

The major components of the segmentation coprocessor are:

- A segmentation controller that performs the segmentation and cell multiplexing processing required by the AAL3/4 and AAL5 and ATM protocol layers, creating interleaved streams of Segmentation and Reassembly (SAR) PDUs by segmenting multiple CPCS Protocol Data Units (PDUs).
- A 128-word transmit FIFO that accepts and buffers segmented cells to the framer prior to transmission.
- A CRC-10 generator responsible for computing a 10-bit Cycle Redundancy Check (CRC) over the 48-byte payload field of each transmitted cell. This CRC is used for the AAL3/4 and Operation and Maintenance (OAM) SAR-PDU CRC.
- A CRC-32 generator responsible for computing a 32-bit CRC over the 48-byte payload field of each transmitted cell. This CRC is used for the AAL5 CPCS-PDU CRC.

A flowchart of the internal hardware process is shown in Figure 2-19. A flow chart of the host interaction with the segmentation coprocessor is shown in Figure 2-20.

Figure 2-19. Bt8230 Transmit Hardware Flow Chart

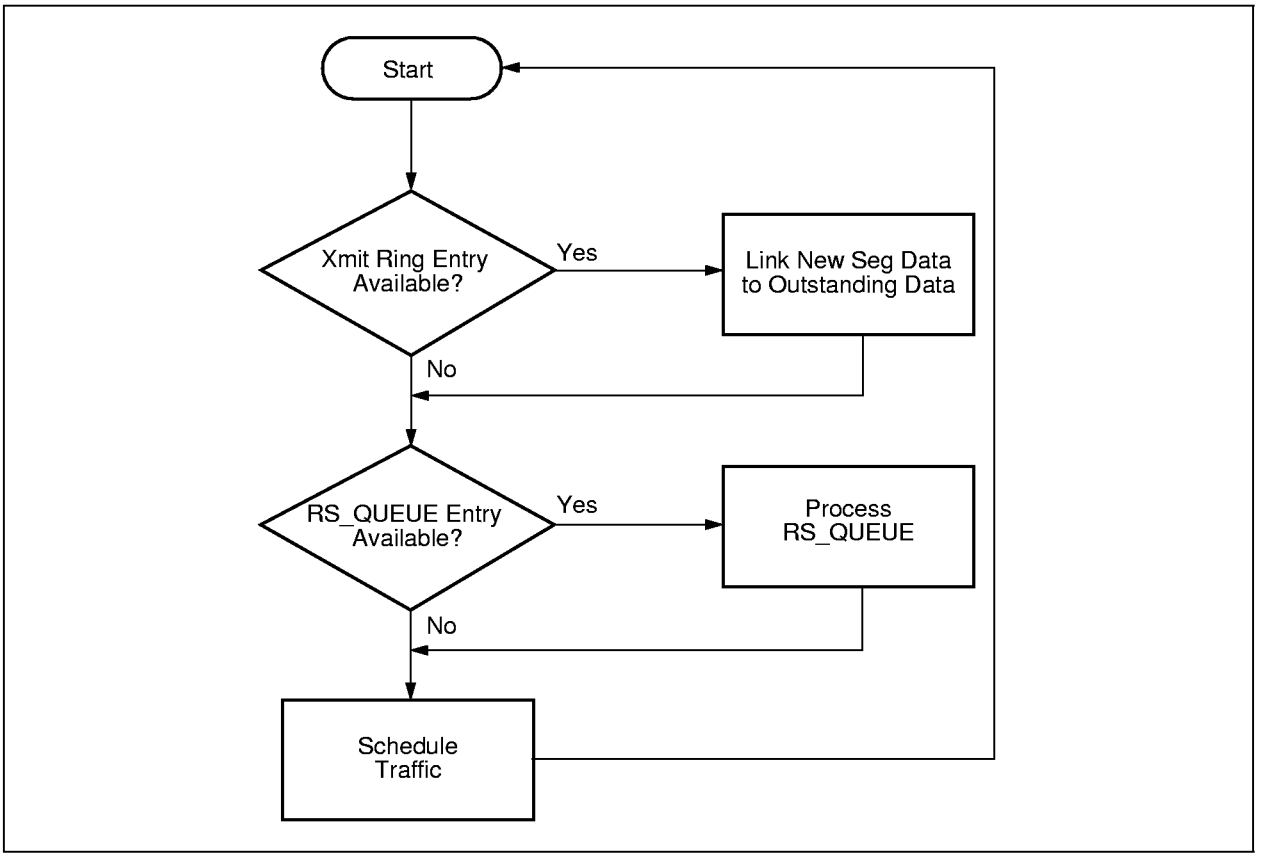
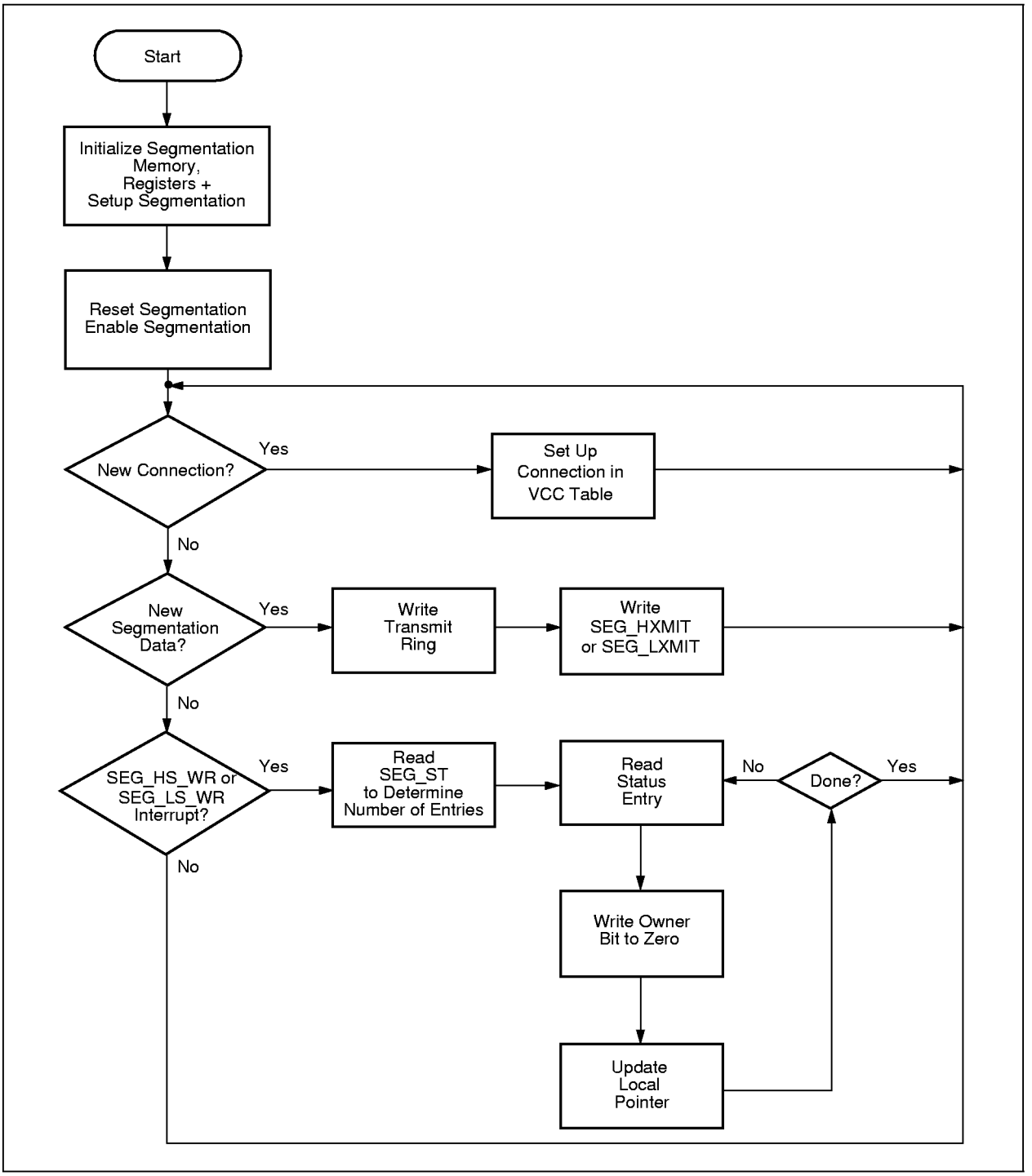


Figure 2-20. Bt8230 Transmit Software Flow Chart



Each VCC can send cells at a fixed-rate or in an Available Bit Rate (ABR) mode. The parameters for each fixed-rate VCC are individually specified in terms of the ATM Forum UNI 3.1 Specification, Generic Cell Rate Algorithm (GCRA). This algorithm specifies a rate in terms of two parameters: *I* and *L*. *I* is the average intercell interval for the VCC. *L* specifies how early a VCC may send a cell relative to the intercell interval *I*. Higher values of *L* allow greater intercell variation and increased burstiness of the traffic. The traffic management method used in the segmentation coprocessor will use the allowed intercell variation in *L* only as required to multiplex multiple VCCs. Setting a high value for *L* will not increase the burstiness of a VCC unless required to multiplex the VCC with other VCCs. The values of *I* and *L* for each VCC may be changed during segmentation, if required.

All VCCs that are specified as ABR connections share the bandwidth that is not allocated by the fixed-rate VCCs. The ABR VCCs are maintained on a circular queue. The VCC at the top of the queue will transmit an ATM cell during each cell interval that is not used by any fixed rate VCC. The VCC at the top of the queue is then placed at the bottom of the queue.

The segmentation coprocessor reads the payload data for each VCC using a gather DMA. This allows the data for each VCC to be located in multiple noncontiguous buffers, either in host or local memory. Each buffer can be any size up to 65535 bytes and a cell payload can cross a buffer boundary.

2.5.1 Memory Set Up

The segmentation coprocessor uses seven areas of local memory to store segmentation state information:

1. host status queue area
2. pm state area
3. local status queue
4. reserved area
5. VCC table
6. buffer descriptor area
7. local transmit ring

Each area can be located anywhere in the memory space. The base address for each area except the buffer descriptor area is set by the user in a register. Each buffer descriptor is located individually with a separate pointer in another buffer descriptor or in the VCC table. The size of each memory area is determined by the maximum limits set in the SEG_CTRL Register. The base pointer, size, initialization required, and purpose of each memory area is shown in Table 2-4.

The user should set the segmentation size limits in the SEG_CTRL Register. This fixes the size of the host status queue, the local status queue, a VCC table, and reserved area. Bt8230 local memory for each of these areas should be allocated and the Segmentation Status Queue Base [SEG_SBASE;0x24] and Segmentation Virtual Connection Base Register [SEG_VBASE;0x28] registers set with the corresponding base addresses. Each segmentation descriptor can be allocated anywhere in unused Bt8230 local memory. Buffer descriptors are discussed further in subsection 2.5.5.

Table 2-4. Bt8230 Segmentation Local Memory Configuration

Base Pointer	Description
Host Status Queue	
SEG_HSBASE[15:0] field in SEG_SBASE Register	Size: (MAXHS * 8 bytes) + <Max number of PM channels> * 32 bytes Initialization: Set to all zero Purpose: Indicates to the host that a host buffer is completely segmented. PM words are maintained at the top of the queue.
Local Status Queue	
SEG_LSBASE[15:0] field in SEG_SBASE Register	Size: MAXLS * 8 bytes Initialization: Set to all zero Purpose: Indicates to the local processor that a local buffer is completely segmented.
Reserved Segmentation Area	
SEG_RSVDB[15:0] field in SEG_VBASE Register	Size: (MAXI + MAXPND) * 4 bytes Initialization: Set to 0xFFFFFFFF Purpose: Internal state information for segmentation.
VCC Table	
SEG_VCCB[15:0] field in SEG_VBASE Register	Size: <Number of VCCs> * 32 bytes Maximum size is 65535 * 32 bytes Size can change as VCCs are added and removed. Initialization: Each VCC as described in subsection 2.5.3. Purpose: Stores state and rate information for each VCC.
Buffer Descriptors	
NXT_FREE[20:0] field in SEG_FR_BD Register	Size: <Number of active descriptors> * 16 bytes Size can change as buffers are added and removed. Initialization: Each descriptor as described in subsection 2.5.5. Purpose: Stores size, location, and segmentation options for each active segmentation buffer.
Local Transmit Ring	
SEG_LRBASE[15:0] field in SEG_LRBASE Register	Size: MAXLR * 16 bytes Size can change as buffers are added and removed. Initialization: Each descriptor as described in subsection 2.5.5. Purpose: Stores size, location, and segmentation options for each active segmentation buffer.

2.5.2 Initialization of Buffer Descriptors

As part of the Bt8230 initialization, local memory for buffer descriptors must be designated. Each active transmit buffer requires a buffer descriptor stored in local memory. Therefore, the Segmentation Descriptors space must be large enough to hold the maximum expected number of active buffers. At initialization, all buffer descriptors must be linked together into a single chain by using the NEXT_PTR field in the buffer descriptors. The NEXT_PTR field of the last buffer descriptor must be set to zero. The NXT_FREE[20:0] field in the Segmentation Free Buffer Descriptor Register [SEG_FR_BD;0x34] is set to the address of the first buffer descriptor in the chain. The format of each buffer descriptor is shown in Table 2-5. Descriptions of the individual fields are given in Table 2-6.

Table 2-5. Buffer Descriptor Format

<table>
<tr><td colspan="4">BUFF_ADDR (32)</td></tr>
<tr><td colspan="3">BASIZE(16)</td><td>VCC_INDEX (16)</td></tr>
<tr><td>Rsvd (5)</td><td>VCI_DATA (4)</td><td>PTI_DATA (3)
GFC_DATA (4)</td><td></td></tr>
<tr><td colspan="3">VCC_CTRL (16)</td><td>LENGTH (16)</td></tr>
<tr><td>Rsvd (9)</td><td colspan="2">NEXT_PTR (21)</td><td>Rsvd (2)</td></tr>
<tr><td colspan="4">Note: The dashed line is a word selector. The word above is used when the buffer contains an AAL3/4 BOM with GEN_PDU bit active; otherwise, the word below the line is used. Both words are never used at the same time.</td></tr>
</table>

Table 2-6. Buffer Descriptor Format Field Descriptions *(1 of 2)*

Name	Description
BUFF_ADDR	Address in the host or local memory space of beginning of segmentation buffer. Host or local source is determined by the LOCAL option (VCC_CTRL[14]) in the VCC_CTRL field.
BASIZE	Used for the BAsize field in the AAL3/4 header when the GEN_PDU option is selected (VCC_CTRL[5]).
VCC_INDEX	Identifies the VCC in the VCC table.
VCI_DATA	Data for WR_VCI option (VCC_CTRL[10]). Not used for AAL3/4 BOM buffer when GEN_PDU bit is active. Normally used to generated OAM cells.
PTI_DATA	Data for WR_PTI option (VCC_CTRL[11]). Not used for AAL3/4 BOM buffer when GEN_PDU bit is active. Normally used to generate OAM cells.
GFC_DATA	Data for WR_GFC option (VCC_CTRL[12]). Not used for AAL3/4 BOM buffer when GEN_PDU bit is active.

Table 2-6. Buffer Descriptor Format Field Descriptions (2 of 2)

Name	Description
VCC_CTRL	Segmentation options for the buffer. The VCC_CTRL field controls segmentation options on a per-buffer basis. These options should only be changed at a cell or CPCS-PDU boundary. The options are: VCC_CTRL[15] FAST_START—When data is available for VCC, send cell as soon as possible. This is useful for transmission of latency sensitive traffic such as AAL0. GCRA mode VCCs will send cell immediately instead of waiting for rate parameter/time to elapse before sending first cell. ABR VCCs will be placed to the top instead of the bottom of the ABR queue only if VCC is not currently in queue. VCC_CTRL[14] LOCAL—Buffer address is a buffer in Bt8230 local memory instead of host memory. Local buffers must begin on word-aligned addresses. VCC_CTRL[13] SET_CI—Sets the middle bit of the ATM header PTI field for all cells. VCC_CTRL[12] WR_GFC—Overwrites the ATM header GFC field for all cells with GFC_DATA. Global GFC changes (active for all buffers of VCC) can be set in VCC structure ATM header. VCC_CTRL[11] WR_PTI—Overwrites the ATM header PTI field for all cells with PTI_DATA. Used to generate F5 and RM OAM cells. OAM cells are not included in PM TUC or BIP16 calculations. VCC_CTRL[10] WR_VCI—Sets the ATM header VCI value for all cells to VCI_DATA (MSBs of VCI are set to zero). Used to generate F4 OAM cells. OAM cells are not included in PM TUC or BIP16 calculations. VCC_CTRL[9] SET_CLP—Sets the ATM header CLP bit for all cells to 1. VCC_CTRL[8] CELL—Read entire 52-octet ATM cell from segmentation buffer. The ATM_HEADER stored in the VCC structure (Table 2-7) is not used in this mode VCC_CTRL[7] ABORT—Generates an abort cell and end processing of descriptor. VCC_CTRL[6] STM_MODE—When set high, a SEG_xS_WRITE interrupt occurs on each buffer. When low, the interrupt occurs on PDU boundaries. VCC_CTRL[5] GEN_PDU—Generates AAL3/4 header and trailer and AAL5 trailer. Setting this bit on an AAL5 cell will also enable the CRC32 calculation. VCC_CTRL[4] CRC_10—Overwrites last 10 bits of a cell with CRC10 calculation. VCC_CTRL[3] AAL3/4—Selects AAL3/4 operation. AAL5 operation is used if not set. This option also allows OAM cells by selecting AAL5 operation. VCC_CTRL[2] Reserved—Set to zero. VCC_CTRL[1] BOM—Buffer contains beginning of message (AAL3/4 only). VCC_CTRL[0] EOM—Buffer contains end of message.
LENGTH	Number of bytes of data contained in the buffer. If the EOM bit (VCC_CTRL[0]) is not set, this length must be set to an integral multiple of 4 bytes. This location is not changed by the Bt8230.
NEXT_PTR	Pointer to next descriptor for the VCC. The two least significant bits of the pointer are assumed to be zero (word-aligned).

2.5.3 Initializing Segmentation VCCs

State information for each segmentation VCC is kept in a VCC structure entry in the segmentation VCC table. Each connection is assigned a 16-bit VCC index by the user when the connection is set up. The VCC index is used to locate the VCC structure in the VCC table for a particular connection. The VCC structure for a connection is located at address (SEG_VCCB * 128) + (VCC index * 32).

To set up a segmentation VCC, choose an unused VCC index and write the appropriate fields in the VCC structure. The 32-byte VCC structure is presented in Table 2-7. VCC structure field descriptions are given in Table 2-8.

Table 2-7. Segmentation VCC Structure

RUN (1)	Rsvd(1)	RATE_MODE (1)	Rsvd(1)	PM (7)	CURR_DESCR (21)
UU (8)			Rsvd(3)		LAST_DESCR (21)
ATM_HEADER (32)					
CRC (32)					
CPI (8)		BETAG (8)	ST (2)	SN (4)	MID(10)
PDU_LEN (16)				BUFFER_LEN (16)	
L_HIGH (8)			I (24)		
Rsvd (16)				L_LOW (16)	
Rsvd (32)					

Note: The dashed line is a word selector. in AAL5, the word above the dotted is used. In AAL3/4, the word below the dashed line is used. Both words are never used at the same time.

Table 2-8. Segmentation VCC Structure Field Descriptions (1 of 2)

Field Name	Description
RUN	A flag indicating that segmentation is proceeding. This flag is set to one by the Bt8230 when the host supplies data for the VCC; it is set to zero by the Bt8230 when there is no data available for the VCC.
RATE_MODE	Controls transmit rate-shaping mode 0 GCRA—Transmitted traffic conforms to GCRA(I,L) 1 ABR—VCC sends cells during available slots.
PM	Performance Monitoring index. 127 VCCs may be selected for automatic OAM PM generation. Each monitored VCC has a unique performance monitoring index. An index of 0x7F indicates that the performance monitoring is not enabled for the VCC.
CURR_DESCR	Pointer to the current buffer descriptor for the VCC. This field is automatically updated by the Bt8230. The two least significant bits of the pointer are assumed to be zero (word-aligned).
UU	AAL5 User-to-user indication.
LAST_DESCR	Pointer to the last buffer descriptor for the VCC. This field is automatically updated by the Bt8230. The two least significant bits of the pointer are assumed to be zero (word-aligned).
ATM_HEADER	Used for each ATM cell for the VCC. The transmitted header may be modified by option bits in the current buffer descriptor.
CRC	Cyclic Redundancy Check. For AAL5 VCCs, this field holds the accumulated value of the 32-bit CRC. For AAL3/4 this word contains the CPI, BETAG, ST, SN and MID fields as described below.

Table 2-8. Segmentation VCC Structure Field Descriptions (2 of 2)

Field Name	Description
CPI	AAL3/4 CPI field.
BETAG	AAL3/4 BTag and ETag field
ST	AAL3/4 segment type field
SN	AAL3/4 Sequence Number field
MID	AAL3/4 Message ID field
PDU_LEN	CPCS-PDU Trailer Length field. This field is generated by the Bt8230.
BUFFER_LEN	Buffer Length. Number of bytes of data read from the host segmentation buffer.
L_HIGH	*L* Field High. Eight most significant bits of the rate control *L* field.
I	Rate Control *I* Field. This field is the corresponding parameter in the ATM UNI 3.0 GCRA specification (Section 3.6.2.4.1). This parameter is a fixed point number with 10 fractional bits and is in units of cell slots.
L_LOW	*L* Field Low. The 16 least significant bits of the rate control *L* field. This field is the corresponding parameter in the ATM UNI 3.0 GCRA specification (Section 3.6.2.4.1). This parameter is a fixed-point number with 10 fractional bits and is in units of cell slots.

To set up a connection, write the following fields:
For all VCCs:
- ATM_HEADER
- Write all Rsvd fields to zero
- Set RATE_MODE for desired transmit rate shaping.
- Set RUN to zero
- Set PM field to pm index or 0x7F to disable
- Set CURR_DESCR to zero
- Set Word 7 Rsvd field (next to L_LOW) to 0xFFFF.
- Set BUFFER_LEN to zero
- Set PDU_LEN to zero

For rate-controlled VCCs:
- I
- L_HIGH, L_LOW

For AAL3/4 VCCs
- CPI
- MID
- SN (if specific beginning value desired)

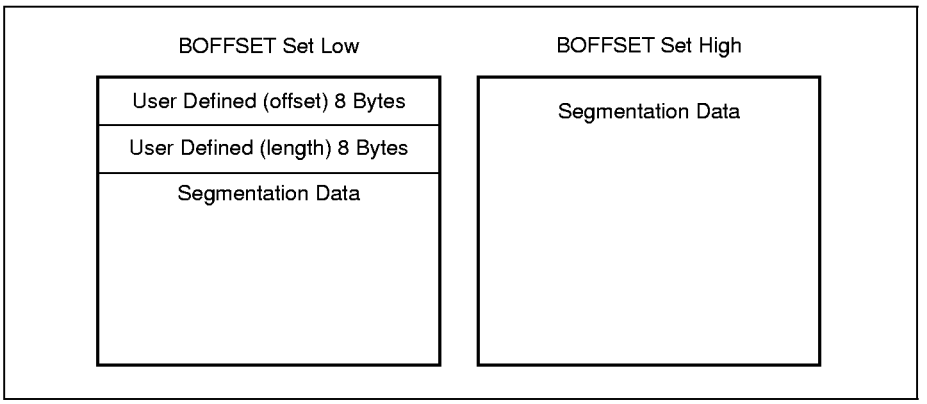
For AAL5 VCCs
- UU
- CRC to 0xFFFF_FFFF

All other fields in the VCC structure are generated by the Bt8230.

2.5.4 Segmentation Buffer Format

The segmentation buffers consist of two user-defined words followed by the segmentation data as shown in Figure 2-21. It is intended that the two user-defined words be used for buffer management and contain the offset to the first word of data and the total length of the buffer. The actual data contained in the buffer might be less than the total size of the buffer and the number of bytes of data is given by the length field in the buffer descriptor. The BOFFSET bit in the Segmentation Control Register [SEG_CTRL;0x20] may be set to eliminate the two user-defined words and set the offset from the start of the buffer to segmentation data to zero.

Figure 2-21. Segmentation Buffer Format



2.5.5 Adding Segmentation Data

Once a VCC structure has been initialized, segmentation data can be supplied to the VCC. The Bt8230 is able to perform an intelligent DMA operation to gather the data for a single connection from noncontiguous areas in host or local memory. This is accomplished by the use of segmentation transmit rings. There are two transmit rings: a host transmit ring located in host memory and a local transmit ring located in local memory. Each entry on a transmit ring is 16 bytes and has the same format as a segmentation buffer descriptor. The format of each transmit ring descriptor is given in Table 2-9.

Table 2-9. Transmit Ring Descriptor Format

BUFF_ADDR (32)				
BASIZE(16)				VCC_INDEX (16)
Rsvd (5)	VCI_DATA (4)	PTI_DATA (3)	GFC_DATA (4)	
VCC_CTRL (16)				LENGTH (16)
User Defined (32)				

Note: The dashed line is a word selector. The word above is used when the buffer contains an AAL3/4 BOM with GEN_PDU bit active; otherwise, the word below the line is used. Both words are never used at the same time.

Bit mapping for host transmit descriptors is the same in big and little endian modes. In addition, the entries must be word (32 bits) aligned. To queue one or more buffers, the host writes the first three words of a transmit ring entry with the buffer address and control information for a segmentation buffer. The last word of each transmit ring entry is not used by the Bt8230. The host writes the ring entries in ascending order and restarts at the beginning of the ring after the end of the ring is reached. To initiate processing of one or more queued entries, the host writes the SEG_HNEW[7:0] field in the Host Transmit Ring Register [SEG_HXMIT;0x40] with the number of queued entries. When the Bt8230 processes a transmit entry, it copies the information from the transmit ring entry into a buffer descriptor stored in Bt8230 local memory. Once processed, the transmit ring entry can be reused by the host, even before the corresponding segmentation buffer has been processed by the Bt8230. The number of transmit ring entries that are pending processing is given by the SEG_HPND[13:0] field in the SEG_HXMIT Register.

The operation of the local transmit ring is analogous to the host transmit ring except that the ring is located in Bt8230 local memory and the Local Transmit Ring Register [SEG_LXMIT;0x44] is used instead of the SEG_HXMIT Register. The local transmit ring is provided to allow easy message insertion by the local processor. Typically, these messages would be local (segmented from local memory) signaling and OAM messages. The WR_PTI (VCC_CTRL[11]) and WR_VCI (VCC_CTRL[10]) control bits in the transmit ring VCC_CTRL field can be used to multiplex OAM cells with data traffic on a single segmentation VCC structure. If these options are used, the OAM buffers must be inserted at CPCS-PDU boundaries in the data traffic. This can be accomplished by either having each CPCS-PDU contained within a single buffer, or by placing OAM buffers only after CPCS-PDU buffers which contain the end of a PDU. When the CPCS-PDU buffers reside on the host and use the host transmit ring, and the OAM buffers reside in the local memory and use the local transmit ring, SEG_XMT_ALT in the SEG_CTRL Register can be used to ensure that the OAM buffers are inserted only at the end of a CPCS-PDU. The host should write (and notify the Bt8230 via the SEG_HNEW[7:0] field) the host transmit ring only with complete CPCS-PDUs. Each PDU may comprise more than one buffer, but all buffers for the PDU should be added at once. With the SEG_XMT_ALT bit set low, the Bt8230 will process the local transmit ring with the OAM buffers only when the host transmit ring is empty (at the end of a PDU). Note that processing a transmit ring entry does not mean that the buffer has completed segmentation, only that the buffer information has been added to the local control information.

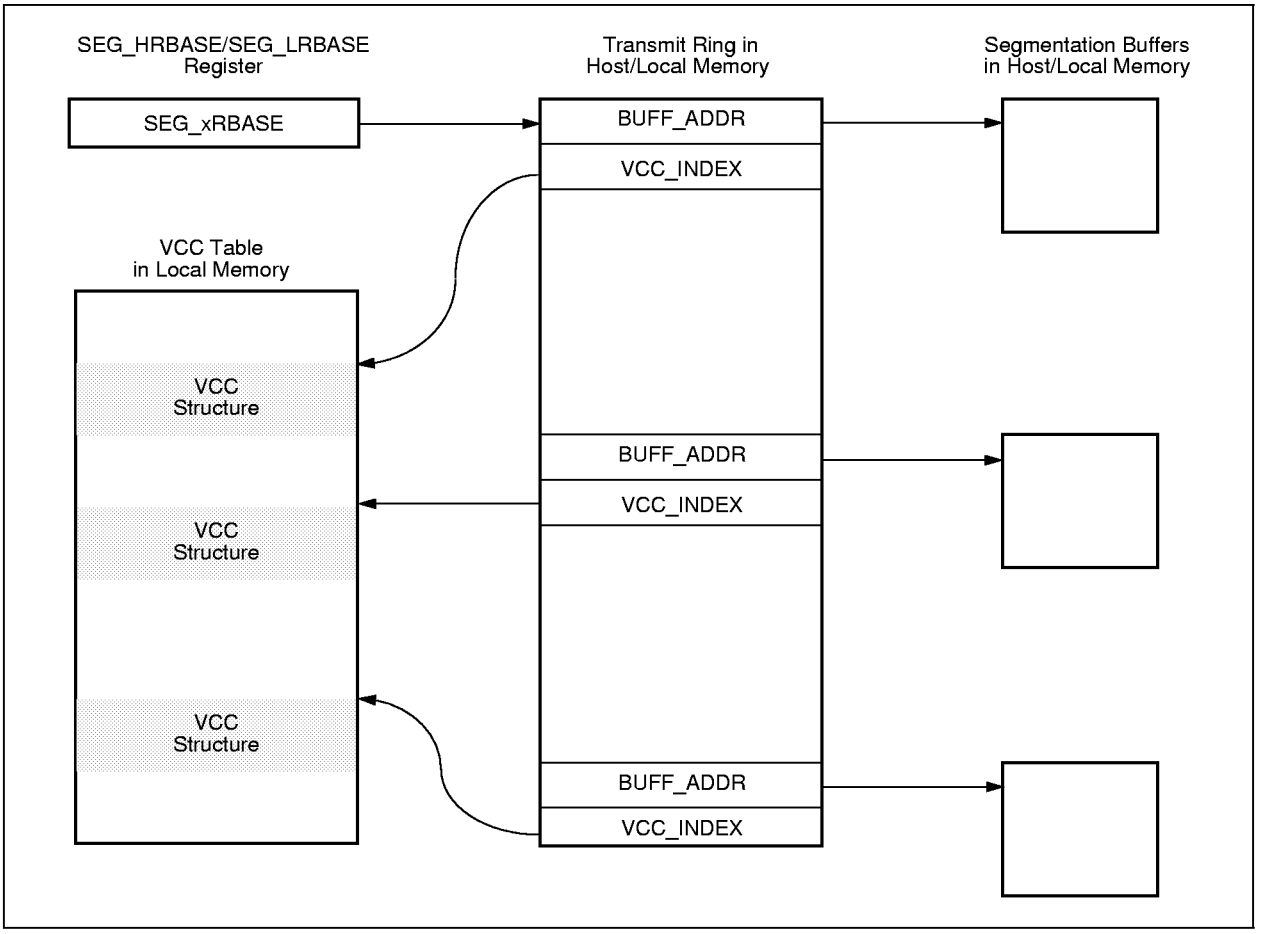
OAM buffers may also be added on a separate VCC structure from the data traffic. This alleviates any need to place the OAM buffer on CPCS-PDU boundaries at the cost of an additional VCC structure in memory. A single VCC structure may be used for OAM cells on all VCCs by using the CELL segmentation mode where the ATM header is included in the buffer data. If OAM cells use a separate VCC structure from data traffic, the SEG_XMT_ALT bit may be set high. This causes the Bt8230 to service the host and local transmit rings alternately and can provide lower latency service for the local transmit ring.

The Bt8230 cannot prevent the host or local processor from overwriting transmit ring entries that have not been serviced by the Bt8230. If the SEG_HNEW[7:0] or SEG_LNEW[7:0] fields are written with a value that would cause the corresponding transmit ring to overflow, the new data is ignored and HRING_OVFL (for host transmit ring) or LRING_OVFL (for local transmit

ring) is set in the Host Interrupt Status Register [HOST_ISTAT0;0xC0] or Local Processor Interrupt Status Register [LP_STAT0;0xE0], respectively. These status bits may optionally cause interrupts.

The Bt8230 will not service either transmit ring if there are no free segmentation buffer descriptors. This condition will clear automatically when a buffer completes segmentation and releases the corresponding buffer descriptor. This condition sets the BD_FULL bit in the HOST_ISTAT0 and LP_STAT0 registers and optionally causes an interrupt. Figure 2-22 illustrates how the data structures are linked.

Figure 2-22. Bt8230 Segmentation Linked Buffer Structure



2.5.6 Descriptor Status

When the segmentation coprocessor completes segmentation of a buffer, it writes an entry onto a status queue. If the buffer descriptor LOCAL option (VCC_CTRL[14]) is not set, the status entry is written on the host status queue; otherwise, the status entry is written on the local status queue. A buffer is complete when the data is all segmented, an abort cell is sent (due to an ABORT option), or a length error occurs. A length error indicates that the maximum CPCS-PDU length was exceeded and any additional data in the buffer was discarded. The format of both the host and local status queue entries is given in Table 2-10. Field descriptions of these entries are given in Table 2-11.

Table 2-10. Status Queue Entry Format

OWNER (1)	LEN_ERR (1)	STOP (1)	DONE (1)	PDU_DONE (1)	Rsvd (11)	VCC_INDEX (16)
BUFF_ADDR (32)						

Table 2-11. Status Queue Entry Format Field Descriptions

Field Name	Description
OWNER	Indicates that the host or local processor owns the descriptor.
LEN_ERR	Maximum PDU length exceeded and descriptor aborted.
STOP	VCC has stopped because no more segmentation data is available.
DONE	Set when finished with buffer.
PDU_DONE	Set when finished with CPCS-PDU.
VCC_INDEX	VCC index value of connection.
BUFF_ADDR	Beginning address of completed buffer.

The Bt8230 will read the OWNER bit before writing a status entry. If the OWNER bit is zero, the entry is written with the OWNER bit set to one and the SEG_HS_WRITE (for a host status write) or SEG_LS_WRITE (for a local status write) status bit in the HOST_ISTAT0 or LP_ISTAT0 Register is set to optionally cause an interrupt. If the OWNER bit is a one, a status overflow has occurred and Bt8230 segmentation halts until the condition is cleared. This sets the SEG_HS_FULL (for a host status write) or SEG_LS_FULL (for a local status write) status bit in the HOST_ISTAT0 or LP_ISTAT0 Register. This overflow condition clears SEG_ENABLE in the SEG_CTRL Register and SEG_RUN in the HOST_ISTAT0 and LP_ISTAT0 registers. After the overflow condition has been cleared, the segmentation coprocessor is restarted by setting the SEG_ENABLE bit. The host or local processor should write the OWNER bit to zero after it has read each status entry. The Bt8230 can be set to always write status regardless of the setting of the OWNER bit by setting the SEG_HS_DIS (for the host status queue) or the SEG_LS_DIS (for the local status queue) in the SEG_CTRL Register.

2.5.7 Rate-Controlled Segmentation

All VCCs that have the RATE_MODE set to GCRA are transmitted according to the *I* and *L* parameters in the VCC structure. The segmentation coprocessor multiplexes multiple rate-controlled VCCs by scheduling future transmissions for each VCC. The scheduling algorithm accommodates multiple VCCs contending for a single cell transmit opportunity and rates that are not integral number of cell slots. For each VCC, the segmentation coprocessor will attempt to send a cell every *I* cell slots. If several VCCs attempt to transmit during the same cell slot, they are sent in the same order as the last transmission for each VCC. The segmentation coprocessor will send cells for a VCC at intervals longer than the *I* parameter for the VCC either because multiple VCCs attempted to send a cell during the same cell slot or because the *I* parameter is set to a fractional number of cells and the

segmentation coprocessor transmits the VCC on the next cell slot boundary. In both cases, the segmentation coprocessor will decrease the cell interval for subsequent cells to maintain an average cell interval of *I*. The amount that a cell interval may be decreased is determined by the *L* parameter.

The segmentation coprocessor will never send traffic that would violate GCRA (*I, L*). The slowest rate that can be configured is 1/(MAXI –ACT_VBR) where ACT_VBR is the number of actively segmenting VBR channels. The MAXI[1,0] field is in the Segmentation Control Register [SEG_CTRL;0x20]. For GCRA only traffic, the MAXPND[1,0] field in the SEG_CTRL Register should be set to 00. See section 3.6.2.4.1 of the ATM Forum UNI Specification, Version 3.1 for further details on GCRA. An example of the VBR setup follows:

Desired Configuration

line rate = STS3 transport
channel throughput = 10 Mbps
burst size = 3 cell maximum
MAXI_schedule_size = 16384 since SEG_CTRL(MAXI) = 11
ACT_VBR = 1000—NOTE: This is the number of actively segmenting VBR channels, not necessarily the total number of VBR channels configured.

I Parameter Calculation

STS3 payload throughput = 260/270 + 155.52Mbps = 149.76 Mbps
I = 149.76 Mbps / 10 Mbps = 14.976—NOTE: This means that on average, every 14.976 cells will contain payload information for this channel.
SEG_VCC_TABLE(I) = 14.976 * 1024 = 15335.424 => 15335 = 0x3BE7
NOTE: The actual average channel throughput rate will be 149.76 Mbps * (1024/15335) = 10.0003 Mbps

L Parameter Calculation

L = (MBS–1) * (I–1) where MBS is the maximum burst size in cells.
NOTE: Reference ATM Forum Specification Version 3.1, Section 3.6.2.4.3.3
L = (3–1) (149.76–1) = 27.952
SEG_VCC_TABLE(L) = 27.952 * 1024 = 28622.848 => 28622 = 0x6FCE
SEG_VCC_TABLE (L_HIGH) = 0x00
SEG_VCC_TABLE (L_LOW) = 0x6FCE

Period Setting Resolution

resolution = 1/(payload throughput) * (1/1024)
= 1/149.76 Mbps * (1/1024)
= 6.677 nsec * (1/1024)
= 6.52 psec

Minimum Configurable Throughput

Min_Rate = payload rate / (MAXI_schedule_size – ACT_VBR = 149.76 Mbps / (16384 – 1000) = 9.73 kbps

2.5.8 Available Bit-Rate Segmentation

Setting the RATE_MODE to ABR in the VCC structure causes a VCC to be segmented from the available bit-rate queue maintained internally by the segmentation coprocessor. This option allows "best effort" service. During each cell slot that does not have any rate-controlled VCCs that attempt to transmit, an ABR cell is sent. If there is not any ABR data to send, an idle cell is transmitted. The available bit-rate queue contains all ABR VCCs that have data ready to segment. The VCC at the top of the queue transmits during the first available cell slot. The VCC is then removed from the top of the queue and placed at the bottom of the queue. The MAXPND field in the SEG_CTRL Register needs to be set such that the size is greater than the number of simultaneous active ABR connections.

The aggregate ABR transmission rate may be controlled with the Segmentation ABR Parameter Register [SEG_ABR;0x30]. If the ABREN bit of this register is set, the state of the ABR GCRA is checked before each potential ABR transmission. If the state of the ABR GCRA does not allow a cell to be sent, an idle cell is sent instead. The *I* and *L* parameters for the ABR GCRA are given by the SEG_ABR Register. The I and L parameters are limited to the following I + L is ≤ 0x3FFFF.

2.5.9 Segmentation Restart

A segmentation VCC may halt because the segmentation data is exhausted. The VCC will automatically resume segmentation when more data is available from the Host or Local Transmit Rings.

2.5.10 OAM, ILMI, and Signaling

Operation and Maintenance (OAM), Interim Local Management Interface (ILMI), and signaling messages are management messages generated by the host. These messages may be generated from Bt8230 memory instead of host memory by setting the LOCAL option (VCC_CTRL[14]) in the buffer descriptor. This may be done on a buffer-by-buffer basis. OAM F5 flows have the same Virtual Channel Indicator/Virtual Path Indicator (VCI/VPI) as normal traffic but a different PTI field. This is accomplished by using the WR_PTI field (VCC_CTRL[11]) in the buffer descriptor. OAM F4 cells occur on a specific VCI by using the WR_VCI option (VCC_CTRL[10]) in the buffer descriptor. The CRC_10 option (VCC_CTRL[4]) in the buffer descriptor can be used for both F4 and F5 cells to generate the required 10-bit CRC. Signaling and ILMI messages occur on a specific VCI. A separate VCC data structure can be set up to handle these messages.

OAM Performance Monitoring (PM) can be enabled for up to 127 VCCs by setting the PM field in the VCC structure. For each monitored VCC, a block count and BIP-16 is maintained. A forward monitoring OAM PM cell is automatically inserted at the end of each PM block. A backward reporting cell is generated whenever the reassembly coprocessor writes a PM entry in the RSM/SEG Queue. This cell will not include any forward monitoring information. For backward reporting without forward monitoring, set BLOCK_SIZE = 0. A PM table is maintained at the top of the Segmentation Host Status Queue. The format of each entry is shown in Table 2-12.

Table 2-12. PM Table Format

ATM_HEADER (32)	
TUC (16)	BIP (16)
TARGET (16)	BLOCK_SIZE (16)
Rsvd (24)	MSN (8)

The ATM_HEADER is used for all generated PM cells. The TUC and BIP fields are used for forward monitoring and are updated by the Bt8230. Target is the TUC at the end of the block. A forward monitoring cell is generated when the TUC reached the target. BLOCK_SIZE is the size of each block and is used to update the TARGET field after a forward cell is sent. MSN is the forward monitoring sequence number.

To set up performance monitoring on a VCC, the host selects an unused PM value (0–126) and initializes the corresponding PM table entry as follows:

- Set ATM_HEADER to the appropriate value.
- Set BIP to all zero.
- Set TUC to desired value (can use zero).
- Set BLOCK_SIZE to appropriate value.
- Set MSN to desired value (can use zero).
- Set TARGET = TUC + BLOCK_SIZE.
- The reassembly VCC_INDEX and segmentation VCC_INDEX are equal on the PM channel. For AAL3/4 channels, the SEG_VCC_INDEX must equal the RSM_VCC_INDEX value in the RSM_VCC_TABLE pointed to by VCC_PNTR #1 in the respective AAL3/4 reassembly Hash Bucket (see Table 2-16).

For F4 PM flows, the RSM_VCC_INDEX must equal one of the SEG_VCC_INDEX values. The RSM_VCC_INDEX is determined by the special F4 OAM reassembly hash bucket that needs to be configured. For all VCs within the VP group, the PM value in the respective VCC table entries need to be assigned the same value.

Once the table entry has been initialized, performance monitoring processing is started by writing the PM index into the VCC structure. PM processing operates automatically until stopped by writing the PM field in the VCC structure to all-ones.

2.5.11 Idle Cell Generation

The segmentation coprocessor will generate an idle cell if there are no rate-controlled VCCs that attempt to transmit during a given cell slot and the ABR queue is empty. Every byte in an idle cell payload is generated as zero. If the SEG_432 bit in the SEG_CTRL Register is set to a logic high, the idle cell header equals 0x0000_0001 and the payload is 0x6A.

2.5.12 ATM Header Generation

The ATM header for each non-idle ATM cell is constructed by the segmentation coprocessor according to the VCC_CTRL options in the current buffer descriptor and the ATM_HEADER field in the VCC structure. Table 2-13 gives the source of each field in the header. For a given field, later entries in this table have priority over earlier entries.

Table 2-13. Bt8230 Header Source Fields

Field	VCC_CTRL Options	Source
GFC	WR_GFC	ATM_HEADER GFC_DATA
VPI		ATM_HEADER
VCI[15:4]	WR_VCI	ATM_HEADER All zero
VCI[3:0]	WR_VCI	ATM_HEADER VCI_DATA
PT[2]	WR_PTI	ATM_HEADER PTI_DATA
PT[1]	WR_PTI SET_CI	ATM_HEADER PTI_DATA 1
PT[0]	WR_PTI EOM set, AAL3/4 not set	ATM HEADER PTI_DATA 1 on last cell of buffer
CLP	SET_CLP	ATM_HEADER 1
HEC		Generated as all zero in ATM physical interface.

2.5.13 AAL3/4 SAR-PDU Generation

Setting the AAL3/4 option in a buffer descriptor causes the Bt8230 to generate the AAL3/4 ST, SN, MID, LI, and CRC fields for each ATM cell. Each field is generated as shown in Table 2-14.

On the first cell for a buffer with BOM set in the buffer descriptor, the PDU_LEN field in the VCC structure is reset to 0. The PDU_LEN field is updated for every byte read from the segmentation buffer. On the last generated cell for a buffer with EOM set in the buffer descriptor, the segmentation coprocessor will pad the SAR-PDU payload with zeros to 44 bytes.

Table 2-14. AAL3/4 SAR-PDU Generation

Field	Function
ST	BOM for first cell generated from buffer with BOM option set. EOM for last cell generated from buffer with EOM option set. SSM for cell generated from buffer with BOM and EOM set and a descriptor length field ≤ 44 COM for all other generated cells
SN	Read from the SN field in the VCC structure. The SN field is incremented modulo 16 after each use.
MID	Read from the MID field in the VCC structure.
LI	Generated by the segmentation coprocessor.
CRC	Generated as all zero. The CRC10 field may be overwritten with the CRC-10 generator before transmission by setting the CRC_10 option in the buffer descriptor.

Setting the GEN_PDU and AAL3/4 options causes the Bt8230 to generate the AAL3/4 CPCS-PDU header and trailer. On the first cell of a buffer with BOM set, the CPCS-PDU header is generated as the first 4 bytes of the SAR-PDU payload. On the last cell of a buffer with EOM set, the segmentation processor writes an all zero PAD field after the end of the segmentation buffer data to complement the CPCS-PDU payload to an integral number of 4-byte words. The segmentation coprocessor then adds the 4-byte CPCS-PDU trailer, and zero-fill octets for the remainder of the SAR-PDU payload. The CPCS-PDU trailer is generated in a separate cell if the CPCS-PDU payload plus PAD is an integral multiple of 44 bytes. The CPI is read from the CPI field in the VCC structure. The AL fields is set to all zero. The Btag and Etag fields are read from the BETAG field in the VCC structure. The BETAG field is incremented after it is used as an Etag field. The BAsize field is read from the BASIZE field in the buffer descriptor. The length field is read from the PDU_LEN field in the VCC structure.

If the PDU length in the VCC structure exceeds the maximum PDU length, the buffer is aborted and a length error is placed on the status queue. The maximum PDU length is:

```
EOM set and GEN_PDU not set 65544
all other cases 65535
```

2.5.14 AAL5 SAR-PDU Generation

If the AAL3/4 option is not set for a buffer descriptor, the SAR-PDU is formatted for AAL5. After the last cell of a buffer with EOM set, the PDU_LEN and CRC in the VCC structure are reset. The CRC field is reset to all ones. The PDU length and CRC fields in the VCC structure are updated for each transmitted cell. If EOM option is set in the buffer descriptor, the SAR-PDU for the last cell is padded with zeros to fill 48 bytes. If the GEN_PDU and EOM option bits are set in the buffer descriptor, an all-zero PAD field and the AAL5 CPCS-PDU trailer are inserted at the end of the data from the buffer. The PAD field is sized so that CPCS-PDU payload plus PAD plus CPCS-PDU trailer is an integral multiple of 48 bytes. The CPCS-PDU trailer will be generated in a separate cell if the length of the CPCS-PDU payload modulo 48 is zero or in the range of 41–47 bytes.

The CPCS-UU field is read from the VCC structure, the CPI is encoded with all zeros, and the length and CRC fields are read from the VCC structure. The CRC field from the VCC structure is ones complemented before it is used in the CPCS-PDU trailer.

If the PDU length in the VCC structure exceeds the maximum PDU length, the buffer is aborted and a length error is placed on the status queue. The maximum PDU length is:

```
EOM set and GEN_PDU not set 65568
all other cases 65535
```

2.5.15 AAL0 SAR-PDU Generation

AAL0 cells may be sent as AAL5 cells by setting EOM and not setting the GEN_PDU in each buffer descriptor. This prevents length errors from occurring and allows the data from the host buffer to be segmented unchanged. If a VCC only occasionally has data available, the Fast Start option in the buffer descriptor can be used to transmit the data as soon as it is available. The CELL option in the buffer descriptor can be used to read entire ATM cells (except for the HEC field) from the segmentation buffer.

2.5.16 CRC-10 Generator

The segmentation coprocessor contains a CRC-10 generator that calculates a 10-bit CRC over the first 374 bits of the generated SAR-PDU. The output of the CRC-10 generator will overwrite the final 10 bits of the generated SAR-PDU if the CRC_10 option is set in the current buffer descriptor. This option would normally be set for AAL3/4 VCCs and for OAM cells.

2.5.17 Transmit Cell FIFO

The segmentation coprocessor incorporates a 128-word transmit FIFO into which 52-byte ATM cells are written before transmission to the external framer device. The transmit cell FIFO helps hide the bursty nature of the cell generation and DMA process from the ATM physical interface, and also decouples the ATM physical interface clock from segmentation coprocessor. The HEC byte is required to be calculated in the physical device.

2.6 Reassembly Coprocessor

The reassembly coprocessor is responsible for processing 52-octet cells from the receive ATM physical interface. It controls the writing of the CPCS payload to host memory and performs all necessary SAR and CPCS checks. The reassembly coprocessor maintains a hash table, hash bucket chains, and VCC tables in local memory to operate on the various CPCS connections. The hash function quickly locates the appropriate reassembly VCC table based on the received VPI/VCI value in the ATM cell header. The hashing process also allows complete freedom in assignment of VCI/VPI values.

The reassembly coprocessor uses an intelligent scatter method to write the payload portion of the ATM cell to host memory. It maintains a free-buffer queue and status queue in local memory to control the scatter operation. The free-buffer queue is updated by the host processor to point to available cell buffers in host memory. The status queue is updated by the reassembly coprocessor with information about how the cell buffers are used. A duplicate structure may be maintained to write cell payload data to local memory. The purpose of this is mainly to divert OAM, ILMI, and signaling messages to the local memory. The local processor can then process these messages without affecting the host system or being affected by host system faults. The local processor updates the local free buffer queue to point to local cell buffers, and reads the local status queue to access the local cell buffers.

The reassembly coprocessor can perform reassembly checks on AAL5, AAL3/4, AAL0, and OAM cells. For each type of cell, specific SAR and CPCS checks are performed and the results are signaled to the appropriate processor. For an AAL5 connection, only CPCS checks, including the CRC32 check, are performed. For an OAM cell, only the CRC10 check is performed. AAL3/4 traffic is checked for various SAR and CPCS protocols, including CRC10. AAL0 traffic has no checks.

The reassembly coprocessor consists of the following subunits:

- VPI/VCI Hash Function—Quickly locates connection information stored in local memory. This method uses less memory than a straight index approach while still supporting the complete VPI/VCI address space.
- AAL Reassembly Processor—Performs required SAR-PDU and CPCS-PDU checks. SAR-PDU checks are performed on AAL3/4 and OAM cells and CPCS-PDU checks are performed for AAL3/4 and AAL5 connections.
- Scatter cell payload data to host memory cell buffers through the DMA block or to local memory cell buffers. Separate free buffer queues are maintained to allocate cell buffers in host and local memory.
- Status Processing Unit—Assembles the appropriate status entry and writes it to local memory. Separate status queues are maintained to correspond with cell buffers in host and local memory.
- 64-word Receive Cell FIFO—Accepts and buffers incoming cells from the ATM physical interface prior to processing by the reassembly coprocessor.

A flow chart of the internal hardware process is shown in Figure 2-23. A flowchart of the external software process is shown in Figure 2-24

Figure 2-23. Bt8230 Receive Hardware Flow Chart

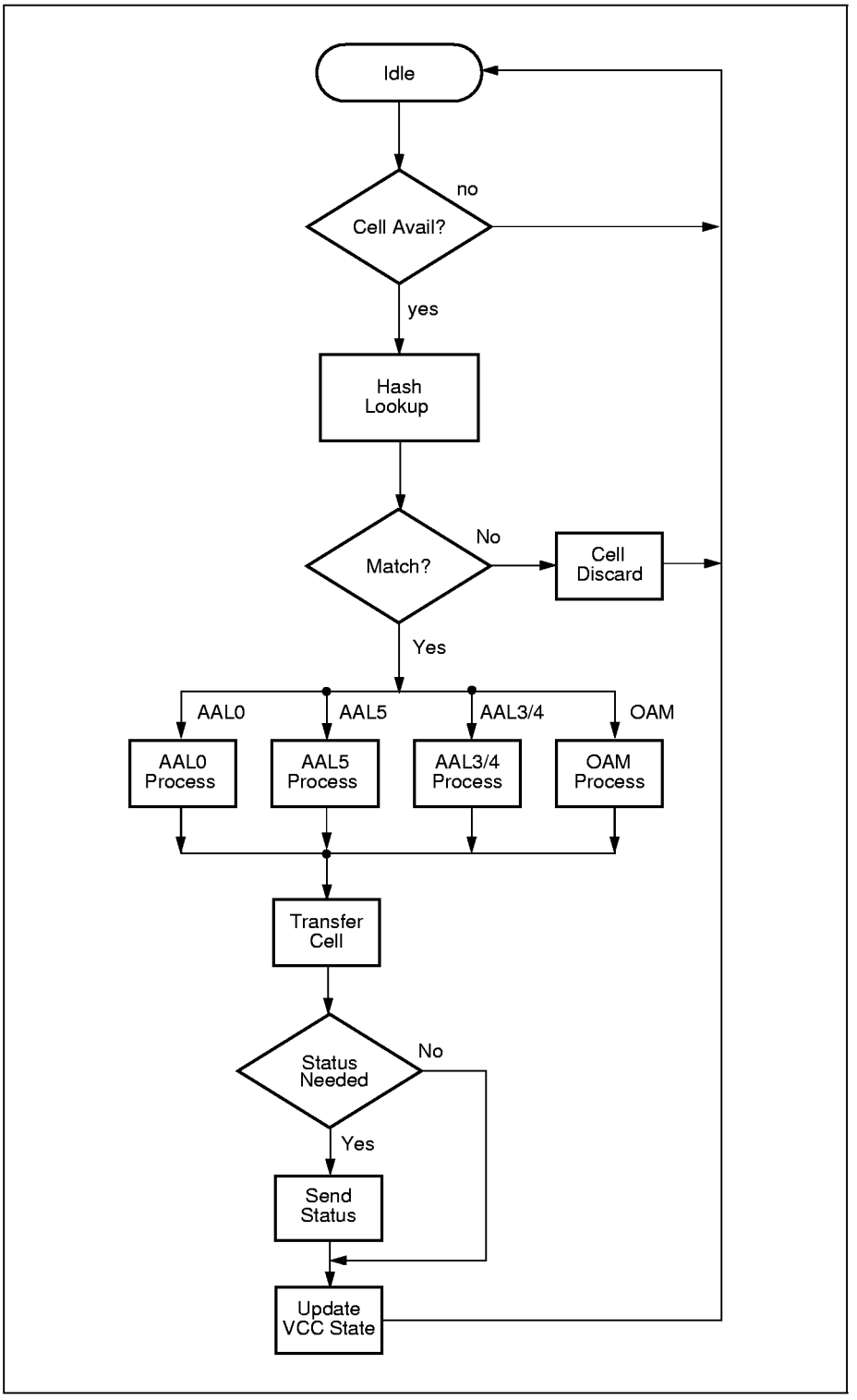
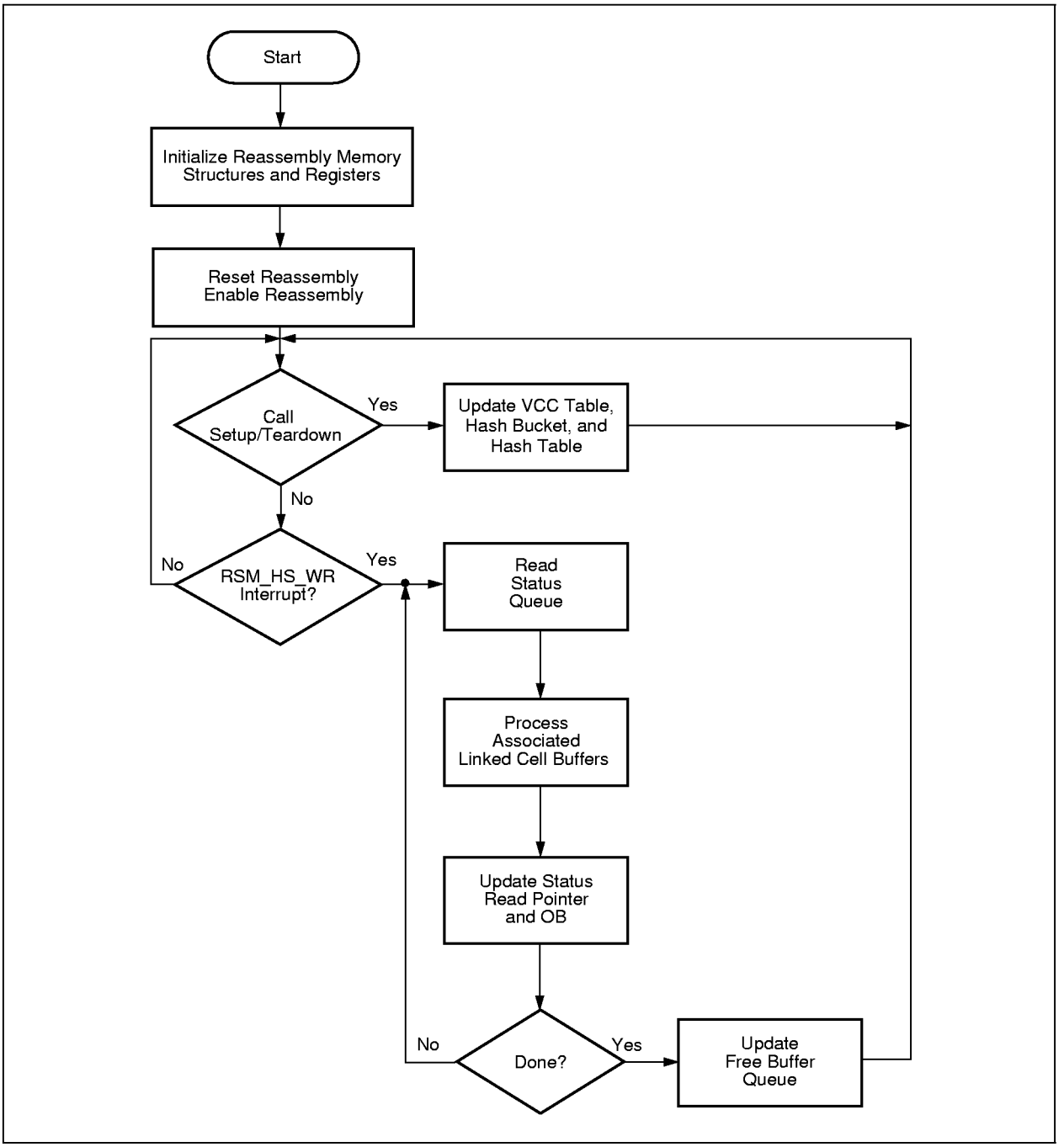


Figure 2-24. Bt8230 Receive Software Flow Chart



Various internal control registers and structures in local memory are maintained by the reassembly coprocessor to perform the functions shown in Figure 2-24. Figure 2-25 gives an overview of the structures maintained to lookup VCC connection information and illustrates how they are linked.

Figure 2-25. Bt8230 Reassembly Buffer Structure

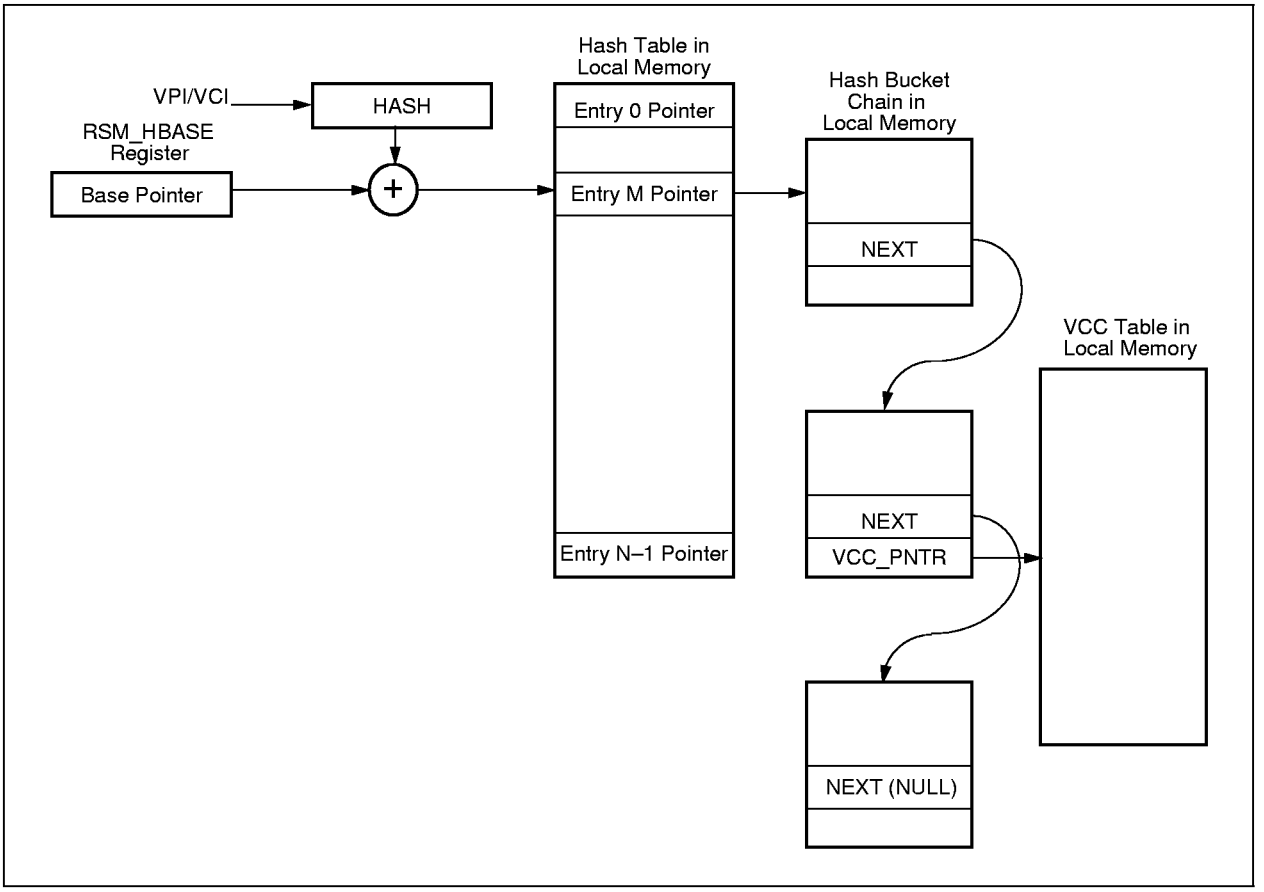
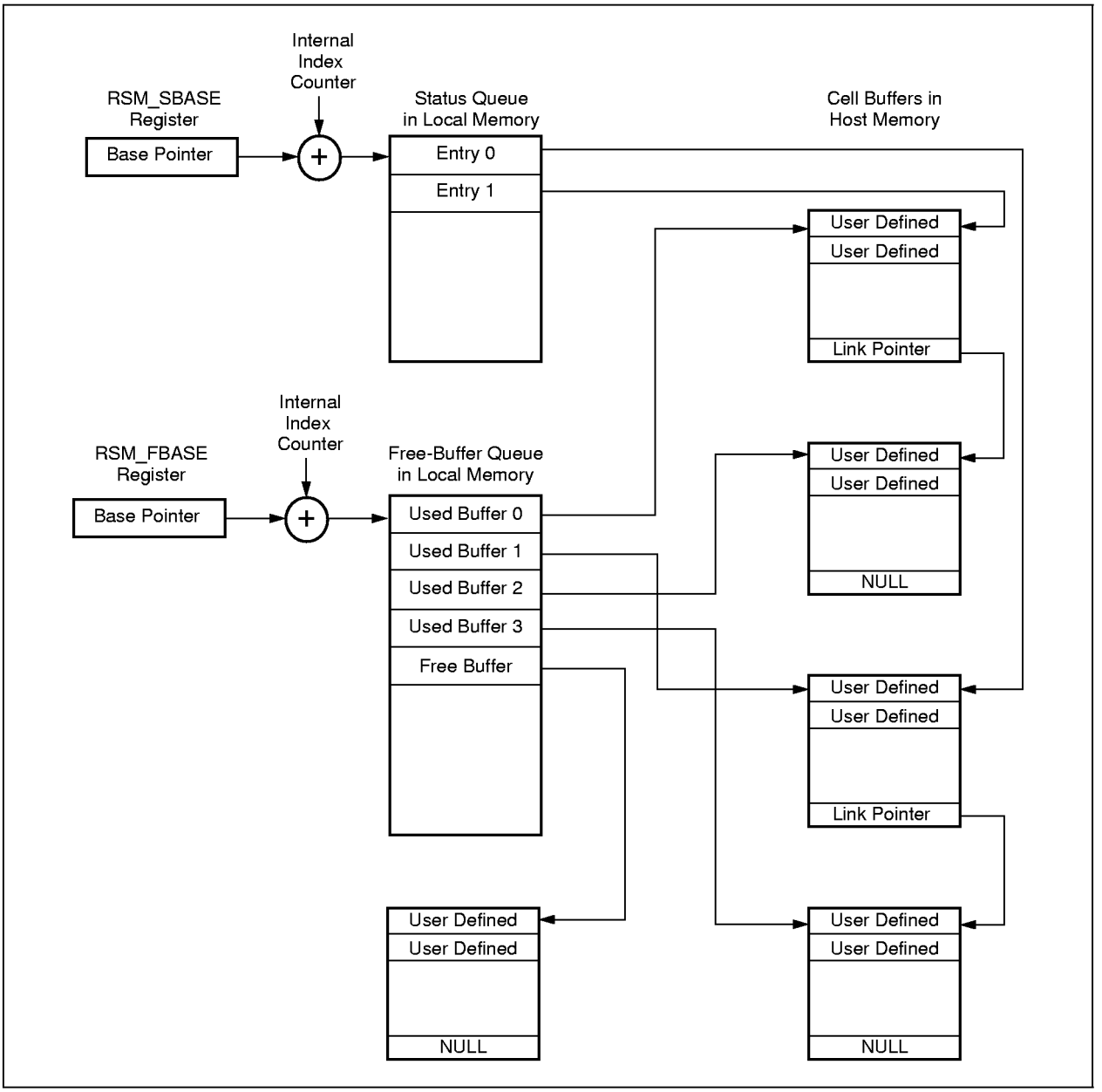


Figure 2-26 illustrates the registers and memory structures maintained to perform the intelligent scatter algorithm to host memory. The structure for local memory scatter is identical.

Figure 2-26. Bt8230 Register and Memory Structure to Perform Intelligent Scatter Algorithm



2.6.1 Local Memory Set Up

The reassembly coprocessor uses six areas of contiguous local memory (host status queue, local status queue, host free buffer queue, local free buffer queue, rsm/seg queue, and hash table) and two dereferenced areas (VCC table and hash-buckets) to store reassembly state information. Each area can be located anywhere in the memory space. The base address for the queues and hash table is set by the user in a register. Each VCC table and hash bucket is located individually within the memory space. The size of each memory area is determined by the maximum limits set in the Reassembly Control Register [RSM_CTRL;0x70]. The base pointer, size, initialization required, and purpose of each memory area are shown in Table 2-15.

Table 2-15. Bt8230 Reassembly Local Memory Configuration *(1 of 2)*

Base Pointer	Description
Host Status Queue	
HST_BASE[15:0] field in RSM_SBASE Register [0x74]	Size: HS_SIZE * 16 bytes Initialization: Set to all zero. Purpose: Indicates to the host processor that a CPCS has been received or an error condition has occurred.
Local Status Queue	
LST_BASE[15:0] field in RSM_SBASE Register [0x74]	Size: LS_SIZE * 16 bytes Initialization: Set to all-zero. Purpose: Indicates to the local processor that a CPCS has been received or an error condition has occurred.
Host Free Buffer Queue	
HFR_BASE[15:0] field in RSM_FBASE Register [0x78]	Size: HF_SIZE * 8 bytes Initialization: Set per subsection 2.6.5. Purpose: Point to available reassembly cell buffers in host memory.
Local Free Buffer Queue	
LFR_BASE[15:0] field in RSM_FBASE Register [0x78]	Size: LF_SIZE * 8 bytes Initialization: Set per subsection 2.6.5. Purpose: Point to available reassembly cell buffers in local memory.
Hash Table	
RSM_HBASE[15:0] field in RSM_HBASE Register [0x7C]	Size: (TAB_SIZE * 4bytes) + 4 words + <Number of PM-OAM channels> * 4 bytes Initialization: Set per subsection 2.6.5 Purpose: Stores pointers to hash buckets.
Hash Bucket	
Pointer in Hash Table	Size: <Number of VCCs> * 20 bytes. Add 4 bytes for each AAL3/4 nonzero MID when MID multiplexing is enabled. Initialization: Set per subsection 2.6.5 Purpose: Stores Hash match information.

Table 2-15. Bt8230 Reassembly Local Memory Configuration (2 of 2)

Base Pointer	Description
VCC Table	
VCC_PNTR in Hash Bucket	Size: <Number of VCCs> * 32 bytes Size can change as VCCs are added and removed. Initialization: Each VCC as described in subsection 2.6.5. Purpose: Stores state information for each VCC.
RSM/SEG Queue	
RS_QBASE[15:0] field in RS_QBASE Register [0x88]	Size: 2048 bytes Initialization: None Purpose: Reserved queue for reassembly to segmentation communication.

2.6.2 Hash Table/Bucket Format and Processing

The hash table data structure consists of a table of pointers, each pointing to a linked list of hash buckets. The Reassembly Hash Table Base Address Register [RSM_HBASE; 0x7C] is expected to point to the starting address of this table of pointers. Global reassembly error counters are located at the top of the table. When these counters reach the maximum value, they wrap around to zero. A table of OAM performance monitoring words is also located at the top of the hash table. The format of the hash table is given in Table 2-16

Table 2-16. Bt8230 Hash Table Data Structure

Address	Contents
RSM_HBASE + 0	Pointer to Hash Bucket Chain #0 (23 bits)
RSM_HBASE + 1	Pointer to Hash Bucket Chain #1 (23 bits)
RSM_HBASE + 2	Pointer to Hash Bucket Chain #2 (23 bits)
RSM_HBASE + (N–1)	Pointer to Hash Bucket Chain #(N–1) (23 bits)
RSM_HBASE + (N)	VPI_VCI_ERR (16 bits) Reserved(16 bits)
RSM_HBASE + (N+1)	CRC10_ERR (16 bits) MID_ERR(16 bits)
RSM_HBASE + (N+2)	LI_ERR (16 bits) SN_ERR (16 bits)
RSM_HBASE + (N+3)	BOM_EOM_ERR (16 bits) SSM_COM_ERR (16 bits)
RSM_HBASE + (N+4)	OAM channel 1 BIP16 (16 bits) BCNT (16 bits)
RSM_HBASE + (N+127)	OAM channel 124 BIP16 (16 bits) BCNT (16 bits)

The number of entries in the table (corresponding to the number of bins in the hashing function, and denoted by N above) is given by the TAB_SIZE[3:0] field [bits 20–17] in the Reassembly Control Register [RSM_CTRL;0x70], and is always a power of 2. This does not include the four global counter words and 124 OAM words at the top of the table. Each pointer in the table, as mentioned, points to a linked list of hash buckets. Unused table locations should be set to NULL.

The number of hash buckets in each linked list is essentially unlimited; the end of the linked list is denoted by a null NEXT, i.e., one with a value of zero. The format of a hash bucket is given in Table 2-17. Field descriptions are given in Table 2-18.

Table 2-17. Hash Bucket Format

MASK (32)			
HEADER (32)			
Rsvd(3)	AAL_TYPE (2)	MID_BITS (4)	NEXT (23)
MSN(8)	Rsvd(10)	AAL_EN (7)	PMINDEX (7)
Rsvd(2)	CHECK_EN (7)	VCC_PNTR #1 (23)	
Rsvd(2)	CHECK_EN (7)	VCC_PNTR #2 (23)	
:			
:			
Rsvd(2)	CHECK_EN (7)	VCC_PNTR #M (23)	

Table 2-18. Hash Bucket Format Field Descriptions *(1 of 2)*

Field Name	Description/Function
MASK	Header comparison mask
HEADER	Header value for comparison
AAL_TYPE	Indicates type of AAL to be processed 00 - AAL5 01 - AAL0 10 - AAL3/4
MID_BITS	Number of significant bits for MID field. If the MID_BITS subfield is nonzero and the AAL_TYPE is AAL3/4, CPCS-MID field processing is enabled by the reassembly coprocessor. The interpretation of the MID_BITS subfield is as follows:

MID Bits	Active Bits of MID Field	Range of MID Field Values	MID Bits	Active Bits of MID Field	Range of MID Field Values
0000	0	MID Process-ing Disabled	1000	8	0–255
0001	1	0–1	1001	9	0–511
0010	2	0–3	1010	10	0–1023
0011	3	0–7	1011	10	1–1023
0100	4	0–15	1100		Reserved
0101	5	0–31	1101		Reserved
0110	6	0–63	1110		Reserved
0111	7	0–127	1111		Reserved

Table 2-18. Hash Bucket Format Field Descriptions (2 of 2)

Field Name	Description/Function
NEXT	Pointer to next bucket, or null if end of list
MSN	Performance monitoring cell sequence number. Does not need to be initialized.
AAL_EN	Enable various cell processes. The AAL_EN field contains the following control bits: PM_EN | CRC10_EN | 52_EN | Rsvd | STM_MODE | LMEM_EN | FW_EN PM_EN — If this bit is set high, performance monitoring is enabled on this channel. CRC10_EN — If this bit is set high, enable AAL3/4 CRC10 field checking and error counting. 52_EN — If this bit is set high, the ATM and AAL overhead octets are written to the cell buffer. Rsvd — Reserved—Set to zero. STM_MODE — If this bit is set high, enable streaming mode for this channel. LMEM_EN — If this bit is set high, write CPCS-PDU to local memory. FW_EN — If this bit is set high, firewall is enabled on a per-VCC basis. Used in conjunction with FWALL_EN [bit 13] of the RSM_CTRL register. PM_EN — If this bit is set high, performance monitoring is enabled on this channel.
PMINDEX	Pointer to a PM OAM word. Index with reference to top of hash table. Minimum value is 4; maximum value is 127.
CHECK_EN	Enable various AAL3/4 cell checks. The CHECK_EN field contains the following control bits: LI_EN | ST_EN | SN_EN | CPI_EN | BAT_EN | BAH_EN | LEN_EN LI_EN — If this bit is set high, enable AAL3/4 LI field checking and error counting. ST_EN — If this bit is set high, enable AAL3/4 ST field checking and error counting. SN_EN — If this bit is set high, enable AAL3/4 SN field checking and error counting. CPI_EN — If this bit is set high, enable AAL3/4 CPI field checking. BAT_EN — If this bit is set high, enable AAL3/4 BASIZE = LENGTH checking. If low, LENGTH > BASIZE check is performed. BAH_EN — If this bit is set high, enable AAL3/4 BASIZE < 37 checking. LEN_EN — If this bit is set high, MAX_LENGTH field in the VCC table is set to BASIZE + 7 during an AAL3/4 BOM cell.
VCC_PNTR	Address pointer to beginning of VCC table. Multiple pointers are needed in AAL3/4 when MID multiplexing is enabled.

An internal hash cache is maintained to reduce the VCC table lookup time. When a cell is available to be processed, the internal hash cache is first checked for a match to the current cell header. If a match occurs, a new hash calculation and hash table lookup is not needed. The cache can be enabled be setting the CACHE_ENABLE bit in the RSM_CTRL Register to a logic high. The information in the hash cache is used instead. When the hash cache does not match the current cell header, the hardware hashing function computes the hash table index as follows:

$$(\{VPI[7:0],VCI[15:12]\} \wedge VCI[11:0]) \;\&\; (2^{TAB\_SIZE}-1)$$

Bits 0 through 7 of the VPI are concatenated with bits 12 through 15 of the VCI to form a 12-bit field that is exclusive-ORed with bits 0 through 11 of the VCI. This result is masked with a bitmask generated from the TAB_SIZE field of the RSM_CTRL Register. The table offset adder then adds the 12-bit offset to the contents of the Reassembly Hash Table Base Register [RSM_HBASE;0x7C] to produce the actual address of a pointer to a hash bucket chain. When TAB_SIZE is equal to 14, a special 14-bit hash index function is enabled. The hash table index is calculated as follows:

$$(\{VPI[7:0],VCI[15:14],'0000'\} \wedge VCI[13:0])$$

The hash bucket chain is processed sequentially for a hash bucket match. The hash bucket index comparator computes the logical-AND of the 32-bit MASK field of a hash bucket read from memory with the 32-bit ATM header and compares the result with the 32-bit HEADER field in the same hash bucket to generate a match signal. The match signal is used by the reassembly coprocessor to determine whether the required hash bucket has been found. If a match is obtained, the current hash bucket is used to process the cell; otherwise, the reassembly coprocessor reads a pointer to the next sequential bucket in the list from the NEXT field and repeats the matching process. If NEXT contains a null (zero) pointer, the end of the list is assumed to have been reached; the matching is then declared unsuccessful, the cell is discarded, and the VPI_VCI_ERR counter at the top of the hash table is incremented.

NOTE: A bit bucket can be set up for cells with unknown VPI/VCIs. Instead of a NULL NEXT pointer in the last hash bucket, a common connection can be set up to handle these cells. By setting the AAL_TYPE to AAL0, a status entry will be written for every cell with unknown VPI/VCI and the cell will be written to a cell buffer.

Once a hash bucket match occurs, the VCC_PNTR is used to access the RSM_VCC_TABLE. For AAL3/4, the pointer to the VCC table (VCC_PNTR) is found from the result of the following calculation:

$$hash\_bucket\_base\_address + 4 + (MID \;\&\; (2^{MID\_BITS} - 1))$$

where MID is the AAL3/4 MID overhead field in the receive cell and MID_BITs is the control field in the hash bucket.

The VCC_PNTR fields are used to point to the VCC table of each connection. In AAL5, only one VCC_PNTR entry is used. AAL3/4 uses a VCC_PNTR entry for every CPCS-MID connection. If consecutive cells access the same VCC table, external access to the table is not needed due to an internal VCC cache. This cache can be enabled by setting the CACHE_ENABLE bit in the RSM_CTRL Register to a logic high.

NOTE: When making changes to a hash bucket or VCC table, disable the cache by setting the CACHE_ENABLE bit to a logic low.

2.6.3 Reassembly VCC Table

A VCC table is maintained for every active VPI/VCI and MID (see Table 2-19). The table maintains state information for an entire CPCS-PDU. Nine words are currently defined. The CRC_REM (AAL5) and BTAG (AAL3/4) words share the same physical location depending on the mode. The last word is only accessed by the local or host processor and is, therefore, a user-defined word. The processor may add on to the table since the reassembly coprocessor accesses the table by a pointer in the hash bucket. Field descriptions are given in Table 2-20.

Table 2-19. Reassembly VCC Table Format

VCC_INDEX (16)	Rsvd (6)	MAX_LEN_H (2)	Rsvd (5)	PDU_FLAG[2:0] (3)
BUFF_PNTR (32)				
MAX_LEN_L (15)			TOT_PDU_LEN (17)	
FW (1)	CBUFF_LEN (15)		CBUFF_CNT (16)	
CBUFF_ADDR (32)				
Rsvd (2)	BASIZE (16)		NEXT_ST (2)	NEXT_SN (4)	BTAG (8)
CRC_REM (32)				
CELL_CNT (16)			PDU_CNT (16)	
PDU_TS (16)			LAST_TS (16)	
LAST_COUNT (16)			CREDIT_TS (16)	

Note: The dashed line is a word selector. In AAL3/4, the word above the dashed line is used. In AAL5, the word below the dashed line is used. Both words are never used at the same time.

Table 2-20. Reassembly VCC Table Format Field Descriptions *(1 of 2)*

Field Name	Description
VCC_INDEX	VCC Index—Used to reference a VPI/VCI so that the host need not perform the hash function.
MAX_LEN_H	Reference MAX_LEN_L field description below.
PDU_FLAG[2:0]	PDU_FLAG[2] Loss Priority(LP)—Loss priority parameter ORED over PDU PDU_FLAG[1] Reserved PDU_FLAG[0] Beginning of Message(BOM)—Written by the reassembly coprocessor in streaming mode to tag the first buffer of a CPCS-PDU as containing a BOM.
BUFF_PNTR	Buffer Pointer—Points to the first linked cell buffer. In streaming mode, it points to the beginning of each cell buffer.
MAX_LEN_L	Maximum allowable length of a CPCS-PDU in words—Consists of two fields: low MAX_LEN_L (15 bits) and highMAX_LEN_H (2 bits).
TOT_PDU_LEN	Total PDU length in bytes—In AAL3/4 mode, it is the summation of the SAR-LI fields across the CPCS-PDU. In AAL5 mode, it is the length of the total CPCS-PDU.
FW	Firewall Condition—Indicates that a firewall condition has occurred.

Table 2-20. Reassembly VCC Table Format Field Descriptions (2 of 2)

Field Name	Description
CBUFF_LEN	Current Buffer Length—Unused space of the current buffer in words.
CBUFF_CNT	Current Buffer Count—Buffer firewall count.
CBUFF_ADDR	Current Buffer Address—Pointer to the current unused position of the cell buffer where cell payload data may be written.
BASIZE	BaSize—Used in AAL3/4 to record the BASIZE field in order to check against the LENGTH field.
NEXT_ST	Next Segment Type—Next AAL3/4 segment type expected.
NEXT_SN	Next Sequence Number—Next AAL3/4 sequence number expected.
BTAG	BTAG—Used in AAL3/4 to record the CPCS-BTAG field in order to check against the CPCS-ETAG field.
CRC_REM	Cycle Redundancy Check Remainder—CRC32 remainder used in AAL5 only.
CELL_CNT	Cell Count—Total number of cells received.
PDU_CNT	Protocol Data Unit Count—Number of unerrored CPCS-PDUs received.
PDU_TS	Protocol Data Unit Timestamp—Records the timestamp of when the current PDU started.
LAST_TS	Last Timestamp—Records the timestamp of the last cell received. Used for message time-out function.
LAST_COUNT	Last Count—Used by the local or host processor to record the value of the CELL_CNT field when the VCC table was last read. Not required for Bt8230 operation.
CREDIT_TS	Credit Timestamp—Used by the local or host processor to record the last time that credit was checked. Not required for Bt8230 operation.

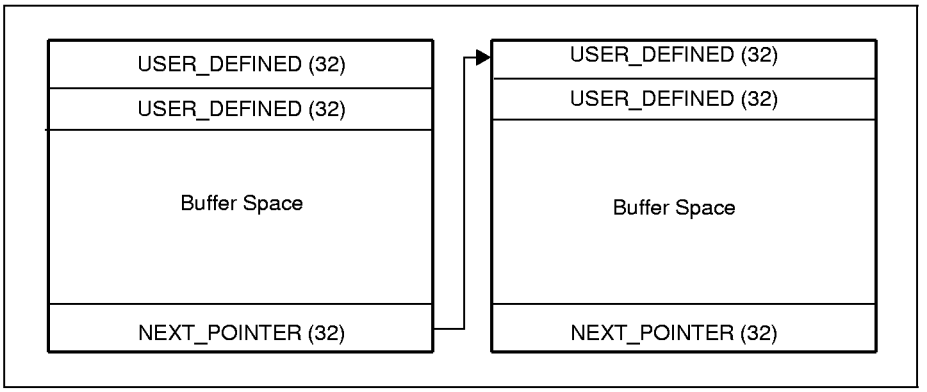
2.6.4 Scatter Method to Host Memory

The Bt8230 uses an intelligent scatter method to write cell payload data to host memory. The reassembly coprocessor maintains the scatter operation and controls the incoming DMA block during scatter DMA to host memory. Both a message and streaming mode are supported. Message mode writes a status entry at the completion of a CPCS-PDU. The host can then traverse the linked cell buffers to collect the complete CPCS-PDU. In streaming mode, a status entry is written after each cell buffer is filled or the last cell of a CPCS-PDU has been received. Three data structures are maintained, one in the host memory and two in local memory. The linked cell buffers (HCELL_BUFF) reside in host memory and the free buffer queue (HFR_BUFF_QU) and status queue (HSTAT_QU) reside in local memory. The linked cell buffers contain the payload portion of the cell. The free buffer queue is a circular buffer that contains pointers to linked cell buffers and their length. The status queue contains information that allows the processor to operate on the linked cell buffers.

2.6.4.1 Linked Cell Buffers

The variable length linked cell buffers (HCELL_BUFF), pointed to by the free buffer queue entries, reside in host memory. The reassembly coprocessor writes the payload part of the cell to this buffer space, see Figure 2-27. The first two words of the buffer are user defined and may be used to store buffer length and offset. The last word of the buffer, NEXT_POINTER, is a pointer to the next linked buffer space. This entry is written by the reassembly coprocessor after it has obtained a new buffer space for the corresponding connection.

Figure 2-27. Linked Cell Buffers



2.6.4.2 Host Free Buffer Queue, HFR_BUFF_QUE

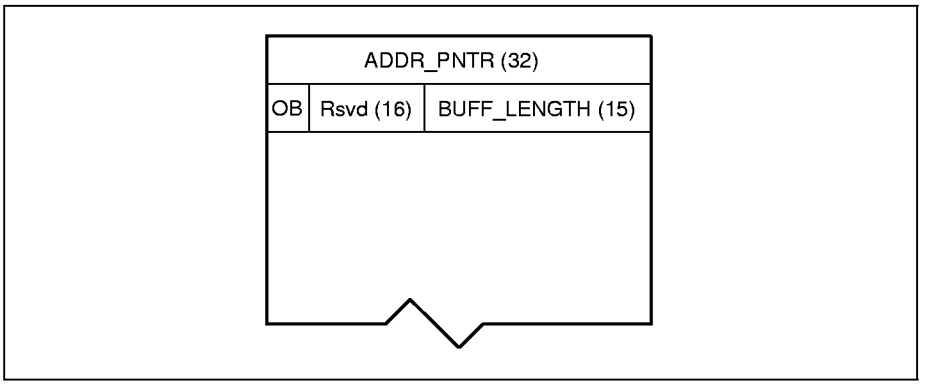
The Host Free Buffer Queue is used to process reassembled cell streams. The host initializes and maintains this queue. A free buffer queue entry consists of two words that contain the address pointer and buffer length of an available buffer space in host memory along with an owner bit (OB) (see Figure 2-28). The buffer length is in words (4 bytes). The length does not include the first two words and the last word of the buffer. This structure resides in local memory to allow low latency in setup of new buffer spaces. The owner bit is written by the reassembly coprocessor to a logical one when it has read an entry. The host processor needs to reset the owner bit to a logical zero when it writes a new entry in the queue. The host will need to keep a free list of cell buffers to supply this queue as buffers are used. Also note that only one cell buffer exists per Free Buffer Queue entry.

The reassembly coprocessor generates a free buffer queue read pointer from the HFR_BASE[15:0] field in the Reassembly Free Buffer Queue Base Register [RSM_FBASE;0x78] and the HF_SIZE field in the RSM_CTRL Register. The pointer takes on the value

```
HFR_POINTER = HFR_BASE * 128 + HFR_COUNTER * 8
```

The HFR_COUNTER is an internal counter. The size of the counter is determined by the HF_SIZE field in the RSM_CTRL Register. The counter can be 8, 10, 12, or 14 bits in length, and is reset to zero by the RSM_RESET bit in the RSM_CTRL Register.

Figure 2-28. Free Buffer Queue



2.6.4.3 Status Queue

The status queue is a circular buffer that resides in local memory. A 4-word entry is written to the queue before the reassembly coprocessor interrupts the host by setting the RSM_HS_WRITE bit in the HOST_ISTAT0 and LP_ISTAT0 registers. A status queue entry is written after a complete CPCS-PDU has been received in message mode, a cell buffer has been used in streaming mode, a PDU maximum length error condition has been detected, a message time-out detected, or an OAM cell has been received. The format of a status queue entry is shown in Figure 2-29. Field descriptions are given in Table 2-21.

The reassembly coprocessor generates a status queue write pointer from the HST_BASE field in the RSM_SBASE Register and the HS_SIZE field in the RSM_CTRL Register. The pointer takes on the value

```
HST_POINTER = HST_BASE * 128 + HST_COUNTER * 8
```

The HST_COUNTER is an internal counter. The size of the counter is determined by the HS_SIZE field in the RSM_CTRL Register. The counter can be 8, 10, 12, or 14 bits in length. The counter is reset to zero by RSM_RESET [bit 30] in the RSM_CTRL Register.

Figure 2-29. Status Queue Entry

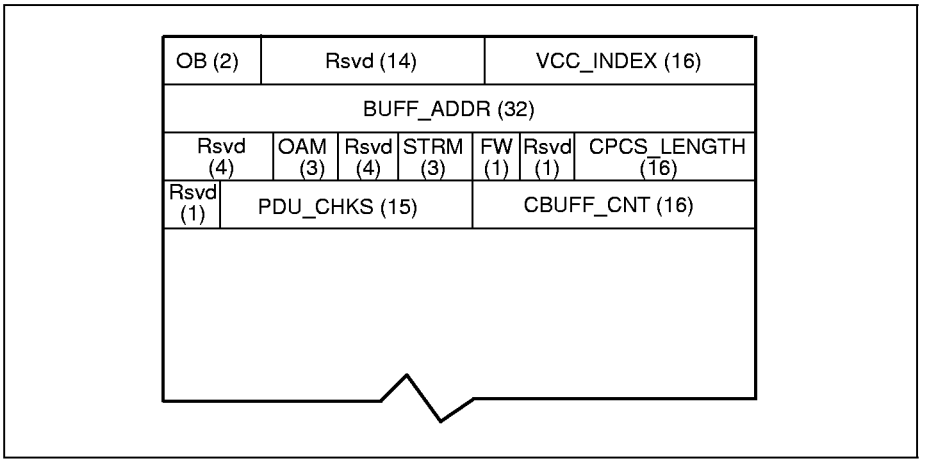


Table 2-21. Status Queue Entry Field Descriptions

Field Name	Description
OB	Owner Bits—Semaphore bits to prevent status queue overwrite and to mark used buffer locations. Written by the reassembly coprocessor to logical high. Reset by the host processor to a logical zero.
VCC_INDEX	VCC Index—Used to reference a VPI/VCI so that the host need not perform the hash function.
BUFF_ADDR	Buffer Address—In message mode, this field points to the first buffer of the linked cell buffers. In streaming mode, it points to each cell buffer used.
OAM	Operating and Maintenance—If this field is nonzero, it indicates that an OAM or management cell had been received as follows: 001 - F4 OAM 010 - F4 OAM End to End 100 - F5 OAM 101 - F5 OAM End to End 110 - PTI = 6 111 - PTI = 7
STRM	Streaming mode status field. Contains three bits as follows: Bit 2 - EOM—In streaming mode, this bit identifies that the buffer contains a EOM Bit 1 - BOM—In streaming mode, this bit identifies that the buffer contains a BOM. Bit 0 - STRM_EN—Streaming mode is enabled on this channel
FW	Firewall—Firewall error occurred.
CPCS_LENGTH	Value of CPCS-length field.
PDU_CHKS	PDU_CHKS[14] MOD_ERROR—AAL3/4 CPCS-PDU is not 32 bit aligned. PDU_CHKS[13] BA_ERROR—AAL3/4 BASIZE ≠ LENGTH. In AAL5, indicates that total pdu length exceeds MAX_LENGTH field when AUU = 1. PDU_CHKS[12] AL_ERROR—AAL3/4 AL field not equal to zero. PDU_CHKS[11] LEN_ERROR—Total PDU length exceeds maximum length and not at end of message. PDU_CHKS[10] PAD_ERROR—Pad field length is not correct length. PDU_CHKS[9] CPI_ERROR—CPI field is not all zero. PDU_CHKS[8] TAG_ERROR—BTAG does not match ETAG in AAL3/4. PDU_CHKS[7] CRC_ERROR—AAL5 CPCS or OAM CRC error. PDU_CHKS[6] ST_ERROR—AAL3/4 segment type error. PDU_CHKS[5] SN_ERROR—AAL3/4 sequence number error. PDU_CHKS[4] LI_ERROR—AAL3/4 SAR-PDU length error PDU_CHKS[3] LP—Loss priority parameter PDU_CHKS[2] CI—Congestion indication PDU_CHKS[1] ABORT—Abort function detected PDU_CHKS[0] TO—Message time out
CBUFF_CNT	Value of CBUFF_CNT field in the VCC table.

2.6.5 Structure Initialization

The structures described herein must be initialized before operation of the reassembly coprocessor is enabled. The suggested initialization is as follows:

1 Linked Cell Buffers
- First two words of each buffer may be written to user defined values
- The last word of each buffer must be written to NULL.

2 Free Buffer Queue
- Write BUFF_ADDR to point to a cell buffer configured in host memory.
- Write BUFF_LENGTH to reflect the length of the cell buffer in words not including the first two words or the last word.
- Write OB to logical zero.
- Write base address of the free buffer queue in HFR_BASE[15:0] subfield of the Reassembly Free Buffer Queue Base Register [RSM_FBASE;0x78]. Write the size of the free buffer queue in HF_SIZE subfield of the RSM_CTRL Register.

3 Status Queue
- Write OB field in each status entry to 00.
- Write base address of the status queue in HST_BASE[15:0] subfield of the Reassembly Status Queue Base Register [RSM_SBASE;0x74]. Write the size of the status queue in HS_SIZE subfield of RSM_CTRL Register.

4 Hash Table
- Write the base address of the hash table in the Reassembly Hash Table Base Register [RSM_HBASE;0x76].
- Entries not written with a pointer to a Hash Bucket chain should be written to NULL.

5 Hash Bucket
- Initialize all words in each hash bucket to the appropriate values.
- Write the global counters and PM words at the top of the hash table to NULL.

6 VCC table
- Write the following fields in each VCC table.
  VCC_INDEX = User defined number
  LP = 0
  BOM = 1
  In AAL5, MAX_LENGTH = (MAX_SDU_LENGTH + pad field length + trailer length)
  In AAL3/4, MAX_LENGTH = User defined number
  TOT_PDU_LEN = 0
  FW = 0
  CBUFF_CNT = Number of cell buffers allowed for this connection if FWALL_EN is set in the RSM_CTRL Register.
  In AAL5, CRC_REM = FFFFFFFFh.
  In AAL3/4, NEXT_ST = 10
  CELL_CNT = 0
  PDU_CNT = 0

2.6.6 Cell Buffer and Status Processing

Upon receiving an interrupt, the host will read the HOST_ISTAT0 Register to determine what event caused the interrupt. If the RSM_HS_WRITE bit is set, the status queue in local memory must be processed. The status queue supports both streaming and message modes. In message mode, a status entry is written upon the completion of the CPCS checks performed on a complete CPCS. In streaming mode, a status entry is written upon filling a cell buffer in order to allow the host to start processing a CPCS-PDU before the complete PDU is received. In this case, the STRM field in the status queue is written to indicate that the connection is in streaming mode and whether or not the buffer contains a BOM or EOM cell. The CPCS_LENGTH and PDU_CHECKS fields (not including the TO bit) are valid only for a cell buffer that contains the last cell of a CPCS-PDU. The BUFF_ADDR field points to the beginning of each buffer. The host processes a status entry by maintaining a read pointer to the circular status queue. The host reads the status entry pointed to by the read pointer and then writes the Owner Bits (OB) of the status entry to a logical zero and increments its read pointer. The host performs the necessary functions to process the cell buffer(s) referenced by the status entry. If cell buffer firewall is enabled (FWALL_EN bit in RSM_CTRL) the host must check the CBUFF_CNT field in the status entry. If the value is nearing zero, the host needs to write more credit for the channel in the Reassembly Firewall Register [RSM_FW;0x84]. It then reads the owner bits of the next status entry to determine if more status processing is needed. Both owner bits must be high for the host to process the status entry. The host continues this process until it has read an OB at a logical zero. The host must also maintain a read pointer to the free buffer queue in local memory. After processing the Status Queue, the host reads the OB of the free buffer queue entry pointed to by its read pointer. If the OB is a logical high, the host writes a new address pointer and length entry and writes the OB to a logical zero. The host continues this process until it has read an OB at a logical zero.

The host processes the linked cell buffers by reading the contents of each buffer until a NULL next pointer is encountered. This signals that the last cell buffer of the linked buffers has been reached.

2.6.7 Status Queue Full Condition

The Owner Bits (OB) of the current status entry are read before the status is written. If either bit is a logical high, the status entry is not written and an error is reported on the RSM_HS_FULL bit in the HOST_ISTAT0 and LP_ISTAT0 registers. The reassembly coprocessor also halts operations. The host may be optionally interrupted by setting the EN_RSM_HS_FULL bit in the Host Interrupt Mask Register [HOST_IMASK0;0xCC]. The host then needs to update the free buffer queue and set the RSM_EN bit in order to restart the reassembly coprocessor. The reassembly coprocessor will then reread the OB bits of the current status entry and continue processing. The full buffer protection can be disabled by setting the RSM_HS_DIS bit in the RSM_CTRL Register.

2.6.8 Cell Buffer Queue Empty Condition

The cell buffer queue empty condition occurs when there are no available cell buffers in the free buffer queue. The owner bits of the current free buffer queue are read before a cell buffer is obtained. If the bit is at a logical high, the reassembly coprocessor terminates processing and sets the RSM_HQ_EMPT bit in the HOST_ISTAT0 and LP_ISTAT0 registers. The host may be optionally interrupted by setting the EN_RSM_HF_EMPT bit in the HOST_IMASK0 Register. The host then needs to update the free buffer queue and set RSM_EN [bit 31] to start the reassembly coprocessor. The reassembly coprocessor will then reread the OB bit of the current free buffer queue and continue processing. The empty queue protection can be disabled by setting the RSM_HF_DIS bit in RSM_CTRL.

2.6.9 Firewall Condition

This option allows the host to limit the maximum number of cells it will accept on a connection. This may be used to police the channel with respect to the total number of buffers available. The free buffer queue firewall function is globally enabled by setting the FWALL_EN bit in the RSM_CTRL Register and on a per-VCC basis by setting the FW_EN bit in the hash bucket. Each time the reassembly coprocessor takes a cell buffer off the free buffer queue, it decrements the CBUFF_CNT value in the VCC table. If the CBUFF_CNT is equal to zero when the free buffer queue is accessed, a firewall condition occurs and a status entry is written with the FW bit set to a logical high. Cells on firewalled channel are then discarded until a cell buffer becomes available and a BOM or SSM cell is received. If the BUFF_ADDR field in the status entry is non-zero, the field points to a linked list of cell buffers that should be discarded.

2.6.10 Scatter Method to Local Memory

In order for the local processor to perform ILMI, signaling and flow control, the reassembly coprocessor can write cells to local memory where the local processor can reassemble the CPCS-PDU. This is controlled by the LMEM_EN bit in the hash bucket. When this bit is set, the reassembly coprocessor will write the cell to local memory rather than host memory. The local processor can then gather the cells and reassemble the message. The structure to perform local scatter is the same as host scatter. All structures reside in local memory. Table 2-22 lists the resources used in place of host memory resources.

In effect, the local processor will be interrupted when a status entry is written to the Local Status Queue. The local processor needs to perform the same functions as the host as described in the table.

Table 2-22. Local Resources Used in Place of Host Memory Resources

Local	Host
LST_BASE in RSM_SBASE[0x74]	HST_BASE in RSM_SBASE[0x74]
LFR_BASE in RSM_FBASE[0x78]	HFR_BASE in RSM_FBASE[0x78]
RSM_LF_EMP in RSM_CTRL[0x70]	RSM_HF_EMP in RSM_CTRL[0x70]
RSM_LS_FULL in RSM_CTRL[0x70]	RSM_HS_FULL in RSM_CTRL[0x70]
LF_SIZE in RSM_CTRL[0x70]	HF_SIZE in RSM_CTRL[0x70]
LS_SIZE in RSM_CTRL[0x70]	HS_SIZE in RSM_CTRL[0x70]
LSTAT_QU	HSTAT_QU
LFR_BUFF_QUE	HFR_BUFF_QUE
LCELL_BUFF	HCELL_BUFF
LFR_POINTER	HFR_POINTER
LST_POINTER	HST_POINTER
RSM_LF_DIS in RSM_CTRL[0x70]	RSM_HF_DIS in RSM_CTRL[0x70]
RSM_LS_DIS in RSM_CTRL[0x70]	RSM_HS_DIS in RSM_CTRL[0x70]
RSM_LS_WRITE in LP_ISTAT0[0xE0]	RSM_HS_WRITE in HOST_ISTAT0[0xC0]

2.6.11 AAL5 CPCS Processing

AAL5 processing is used when the AAL_TYPE field in the hash bucket indicates AAL5. Both CPCS payload and trailer are written to memory.
The following processes occur every cell:

- The LP bit in the VCC table holds the value of the LP parameter. Upon reception of a cell, the reassembly coprocessor ORs the LP bit in the VCC table with the LP bit of the cell and writes the result back to the VCC table.
- The timestamp fields in the VCC table are updated each cell. The PDU_TS is written on the first cell of a PDU. The first cell is detected when the TOT_PDU_LEN field in the VCC table is zero. The LAST_TS field is written upon reception of a cell. Also, the CELL_CNT field in the VCC table is incremented upon reception of a cell.
- The VCC table contains a TOT_PDU_LEN field. This field is constantly updated as cells of a CPCS-PDU are received. The field contains the total length of the PDU in octets. If this value ever exceeds MAX_LENGTH, the PDU is terminated and a status entry is written with the LEN_ERROR bit set high. If a cell with AUU = 1 exceeds the maximum length, the BA_ERROR bit is set high instead of the LEN_ERROR bit. The MAX_LENGTH field is written by the host or local processor at setup to reflect the AAL5 MAX_SDU_DELIVER_LENGTH parameter padding and trailer lengths in octets. It should be an integer number of cells in length. The maximum value of MAX_LENGTH in this mode is 65568.

- The VCC table contains a CRC_REM field that holds the incremental 32-bit CPCS CRC calculation. Upon receiving a cell, the reassembly coprocessor reads the CRC_REM field into the reassembly 32-bit CRC generator. The cell payload is then used to calculate the next CRC remainder. After the current cell is processed, the new CRC remainder is written back to the CRC_REM field. The CRC polynomial is:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Upon termination of a CPCS-PDU, AUU bit = 1, the following processes occur:

- The 32-bit CRC remainder is checked for a value of 0xC704DD7B. If in error, the CRC_ERROR bit in the status entry is set high.
- The SAR-CI parameter is copied into the CI field of the status entry.
- If the CPCS-LENGTH field is all zeros, the ABORT bit in the status entry is set high. The CPCS_LENGTH field in the status entry is written to the value of the CPCS-LENGTH field.
- The reassembly coprocessor performs a pad length check if LENGTH $\neq 0$ as follows: TOT_PDU_LEN –CPCS_LENGTH –8 = [0,47] octets.
- If in error, the PAD_ERROR bit is set high in the status entry. The CPI_ERROR bit in the status entry is set high if the CPCS-CPI field is not all zeros.
- The PDU_CNT field in the VCC table is also incremented if no errors have occurred.

After the status entry is written, the VCC table is initialized to start processing another CPCS-PDU.

2.6.12 AAL3/4 CPCS Processing

AAL3/4 processing is used when the AAL_TYPE field in the hash bucket indicates AAL3/4. CPCS header, payload, and trailer are written to memory. The following checks occur before the cell is accepted for further processing:

- If the CRC10_EN bit is set high in the hash bucket, the CRC10 field is checked. If in error, the cell is discarded and the CRC10_ERR counter at the top of the hash table is incremented. The CRC polynomial is:

$$x^{10} + x^9 + x^5 + x^4 + x^1 + 1.$$

- The MID field value is checked against the number of active MID bits specified by the MID_BITS field in the hash bucket. If invalid bits are set, the cell is discarded and the MID_ERR counter at the top the hash bucket is incremented.
- If the ST_EN bit is set high in the hash bucket, the ST field is checked. The NEXT_ST field in the VCC table is used for this check. A value of 1 in NEXT_ST field indicates expecting BOM/SSM. A 00 value indicates expecting COM/EOM. The following errors are detected:
  a) Receive BOM expecting COM/EOM—The current CPCS-PDU is terminated. A status entry is sent with the ST_ERROR bit set, CPCS_LENGTH = 0 and the BOM_EOM_ERR counter at the top of the hash table is incremented. A new CPCS-PDU is started with the current BOM.

b) Receive SSM expecting COM/EOM—The current CPCS-PDU is terminated. A status entry is sent with the ST_ERROR bit set and CPCS_LENGTH = 0. The current SSM is processed as a valid CPCS-PDU. The SSM_COM_ERR counter at the top of the hash table is incremented.

c) Receive COM expecting BOM/SSM—The cell is discarded and the SSM_COM_ERR counter at the top of the hash table is incremented.

d) Receive EOM expecting BOM/SSM—The cell is discarded and the BOM_EOM_ERR counter at the top of the hash table is incremented.

- If EOM and LI = 63, a status entry is sent with ABORT set and CPCS_LENGTH = 0.
- If the LI_EN bit in the hash bucket is set high, the LI field is checked. The following errors are detected:
  a) Receive BOM and LI ≠ 44—The cell is discarded and the LI_ERR counter at the top of the hash table is incremented.
  b) Receive COM and LI ≠ 44—The cell is discarded and the current CPCS-PDU is terminated. A status entry is sent with the LI_ERROR bit set, CPCS_LENGTH = 0 and the LI_ERR counter at the top of the hash table is incremented.
  c) Receive EOM and (LI < 4) or (LI > 44—The cell is discarded and the current CPCS-PDU is terminated. A status entry is sent with the LI_ERROR bit set, CPCS_LENGTH = 0 and the LI_ERR counter at the top of the hash table is incremented.
  d) Receive SSM and (LI < 8) or (LI > 44)—The cell is discarded and the LI_ERR counter at the top of the hash bucket is incremented.
- If the SN_EN bit in the hash bucket is set high, the SN field is checked. If COM or EOM is received and the SN field does not equal the NEXT_SN field in the VCC table, the cell is discarded, the current PDU is terminated and a status entry is written with the SN_ERROR bit set and CPCS_LENGTH = 0. The SN_ERR counter at the top of the hash table is also incremented.

After the cell checks are performed resulting in a valid cell, the following processes occur:

- Upon receiving a new CPCS-PDU,
  a) The CPI field is checked for a zero value if CPI_EN is set high in the hash bucket. If in error, the cell is discarded, the current PDU is terminated and a status entry is written with the CPI_ERROR bit set high and CPCS_LENGTH = 0.
  b) If a BOM and BAH_EN is set high, The BASIZE field is checked to be less than 37 octets. If less than 37 octets, the cell is discarded, the current PDU is terminated and a status entry is written with the BA_ERROR bit set high and CPCS_LENGTH = 0.
  c) If CPI and BASIZE fields are valid and LEN_EN is set high in the hash bucket, then the reassembly coprocessor writes a value of BASIZE field + 7 to the MAX_LENGTH field in the VCC table. If LEN_EN is set low, the MAX_LENGTH field must be initialized upon connection setup. The value of the BASIZE field in the VCC table is also updated.

- The LP field in the VCC table holds the value of the LP parameter. Upon reception of a cell, the reassembly coprocessor ORs the LP bit in VCC table with the LP bit of the cell and writes the result back to the VCC table.
- The timestamp fields in the VCC table are updated each cell. The PDU_TS is written on the first cell of a PDU. The first cell is detected when the TOT_PDU_LEN field in the VCC table is zero. The LAST_TS field is written upon reception of a cell. Also, the CELL_CNT field in the VCC table is incremented upon reception of a cell.
- The VCC table contains a TOT_PDU_LEN field. This field is constantly updated as cells of a CPCS-PDU are received. The field contains the accumulation of the SAR-LI fields of the PDU in octets. During a BOM or COM, if this value ever exceeds MAX_LENGTH, the PDU is terminated and a status entry is written with the LEN_ERROR bit set high and CPCS_LENGTH = 0.

Upon termination of a CPCS-PDU, ST = EOM or SSM, the following processes occur:

- The SAR-CI parameter is copied into the CI field of the status entry.
- The CPCS_LENGTH field in the status entry is written to the value of the CPCS_LENGTH field.
- The reassembly coprocessor performs a pad length check as follows:
  TOT_PDU_LEN –CPCS_LENGTH – 8 = [0,3] octets. If in error, the PAD_ERROR bit is set in the status entry.
- The reassembly coprocessor confirms that the total message length is on a 32-bit boundary. If in error, the MOD_ERROR bit in the status entry is set high.
- The AL_ERROR bit in the VCC table is set if the CPCS-AL field is not all zeros. The TAG_ERROR bit is set if the CPCS-BTAG field does not match the CPCS-ETAG field.
- If BAT_EN is set in the hash bucket, BASIZE is compared to the LENGTH field. If not equal, a status entry is sent with BA_ERROR bit set. If BAT_EN is a logic low and the LENGTH field exceeds the BASIZE field, the BA_ERROR bit is set.
- The PDU_CNT field in the VCC table is incremented if no errors occurred.

After the status entry is written, the VCC table is initialized to begin processing another CPCS-PDU.

2.6.13 AAL0 Processing

AAL0 processing is used when the AAL_TYPE field in the hash bucket indicates AAL0. Processing consists of writing the cell to host or local memory and writing a status entry for each cell. In the VCC table, the CELL_CNT field is incremented and the current timestamp is written to LAST_TS. The fields used on the status entry are VCC_INDEX, BUFF_ADDR, OB, and CBUFF_CNT.

2.6.14 OAM Processing

By global control (OAM_MEM bit in the RSM_CTRL Register), OAM cells can be written to either the local or host memory. If OAM_EN is set in the RSM_CTRL Register, the reassembly coprocessor will internally detect an OAM cell and route it accordingly. To save hash bucket locations, the following OAMs will be detected:

F4 OAM
F4 OAM End to End
F5 OAM
F5 OAM End to End
PTI = 6
PTI = 7

Connections with F5 or PTI 6 and 7 flows enabled must have the PTI field masked in the corresponding hash bucket for the connection. Connections with F4 flows enabled must configure a hash bucket for VCI 3 and 4. For this connection, only the first word of the VCC table need be configured.

The complete 52-octet OAM cell is written to either host or local memory using the scatter method. The status entry indicates an OAM cell by a non-zero OAM field. The fields used in the status entry are VCC_INDEX, BUFF_ADDR, OAM, CRC_ERROR, and OB. The reassembly coprocessor treats OAM cells as 1-cell PDUs. The 10-bit CRC is checked and if in error, the CRC_ERROR bit is set in the status entry.

Once an OAM cell is detected, the cell is checked to determine whether it is a PM cell. Note that PM detection is only performed on F4 and F5 OAM cells. The PM word pointed to by the PMINDEX field in the hash bucket is read and the PM reporting fields Block Error Result and Lost/Misinserted Cell Count are appended to the cell payload. The reporting fields are also written to the RSM/SEG Queue for further processing by the segmentation coprocessor.

2.6.14.1 PM Operation

Upon receiving a PM activation cell, the local or host processor must enable PM on the applicable channel by setting the PM_EN bit high and writing the PMINDEX to point to the memory location that contains the BIP16 and BCNT values. The PMINDEX value has a minimum value of 4 and a maximum value of 127. Both BIP16 and BCNT should initially be written to zero. For F4 flows, each VCI channel in the VPI group must have the PM_EN bit set high and the PMINDEX pointing to the same location. This is also true of the VCI 3/4 hash bucket set up to handle F4 flows. Once this has been performed, the processor must send an activation confirmed OAM cell to the originator. When a user data cell is received that has PM activated on the channel, the BIP16 and BCNT values are updated.

NOTE: The cell buffer size must be large enough to hold a complete cell when OAM detection is enabled.

2.6.15 Message Time Out

Message time out is supported by the LAST_TS field in the VCC table. The processor can peruse the VCC tables looking for active PDUs, TOT_PDU_LEN not NULL. The processor then compares the LAST_TS field against the CLOCK Register in the Bt8230. If the maximum reassembly time has elapsed, the processor alerts the reassembly coprocessor that a time-out has occurred through the RSM_TO Register.

The processor reads the Reassembly Time Out Register [RSM_TO;0x80] until it returns a NULL. The processor then writes the address of the first word of the VCC table of the timed-out channel in the TOPNTR field, the status destination to the LSTAT field and if the channel is an AAL3/4, set the TMO_AAL34 bit to a logic high. The Bt8230 will then write a status entry with the TO bit set to a logic high and initialize the VCC table for a new CPCS-PDU. The TO bit in the status entry will be set to a logic high and the BUFF_ADDR field will point to the beginning of the first buffer in the linked cell list. In streaming mode, the BUFF_ADDR field will point to the beginning of the last cell buffer.

2.6.16 Credit Check

The credit check is mainly a microprocessor function. The local or host microprocessor would peruse the VCC tables. For each active channel, TOT_PDU_LEN > 0, the LAST_TS and CREDIT_TS are read along with the CELL_COUNT and LAST_COUNT. The processor then calculates the number of cells per time interval as follows:

```
(CELL_COUNT - LAST_COUNT)  ÷ (LAST_TS - CREDIT_TS)
```

If the number of cells per time unit exceeds the channel specification, the host may be notified via the HOST_MBOX Register interrupt. The processor then copies the LAST_TS value to the CREDIT_TS field and the CELL_CNT field to the LAST_COUNT field.

2.6.17 52-Octet Mode

If the 52_EN bit is set in the hash bucket, overhead octets will be written to the cell buffer along with the payload data. In AAL0 and AAL5, the 4-octet ATM header will be written before the payload data is written. In AAL3/4, the ATM header followed by the combined AAL header and trailer will be written before the payload data. This also applies to OAM/MANAGEMENT cells for the connection if OAM_EN is set.

2.6.18 Receive Cell FIFO

The receive cell FIFO is intended to buffer incoming cells from the ATM physical interface until the channel coprocessor can complete the processing of previously received cells. It also allows the ATM physical interface clock to be decoupled from the Bt8230 clock without creating metastability problems. The receive cell FIFO is constructed from a 64-word by 32-bit dual-port RAM.

2.7 ATM Physical Interface

The ATM physical interface is responsible for communicating with and controlling the ATM link interface chip, which carries out all the transmission convergence and physical media dependent functions defined by the ATM protocol. The block performs the following functions:

- Receives and transmits ATM physical interface logic. The ATM physical interface accommodates the Brooktree Bt8222 framer device, a UTOPIA-compatible framer or a Brooktree-conceived slave UTOPIA interface and is responsible for converting between these devices and the internal data interfaces. The Slave UTOPIA interface connects the Bt8230 to a cell switched backplane.
- Receives cell synchronization logic, which validates cell boundaries in the incoming byte stream, strips off the HEC byte from the ATM header and formats the remaining 52 bytes into thirteen 32-bit words before passing them to the incoming cell FIFO. The receive cell synchronization logic ensures that only complete cells are passed down to the remainder of the reassembly controller.
- Transmits cell synchronization logic, which converts the 32-bit data read from the transmit cell FIFO into an octet stream, generates appropriate cell delineation pulses for use by the transmit ATM physical interface, and inserts the blank HEC byte into the ATM header of each cell prior to transferring it to the framer interface.
- Generates and checks odd parity on the octet transmit and receive data buses.

The ATM physical interface contains receive and transmit framer interface logic and receive error generation logic. The ATM physical interface block also interfaces with the segmentation and reassembly coprocessors.

2.7.1 ATM Physical Interface Logic

The Bt8230 ATM physical interface logic consists of the I/O drivers required to communicate with the external framer device, together with adaptation logic required to convert between either the UTOPIA, Bt8222, or Slave UTOPIA interface protocol and the internal byte streams. Configuration pins (FRCFG[1,0]) determine whether the UTOPIA, Bt8222 interface, or slave UTOPIA protocol will be used.

2.7.2 ATM Physical I/O Pins

The operational mode desired is indicated to the Bt8230 by appropriately driving the FRCFG[1,0] input according to the Table 2-23.

Table 2-23. ATM Physical Interface Mode Select

FRCFG[1,0]	ATM Physical Interface Mode
0 0	Bt8222
0 1	UTOPIA
1 0	Slave UTOPIA
1 1	Reserved - Do Not Use

The interpretation of the ATM physical interface pins on the Bt8230 and the actual signals generated or received by the framer in Bt8222 mode is shown in the Table 2-24. The Bt8222 framer needs to be configured to supply a Start of Cell indication. This is done by setting CELL_VAL[15] bit to a logic high in the Bt8222 device.

Table 2-24. Bt8230 Interface Signals

Bt8230 Signal	Bt8222 Signal	Active Polarity	Bt8230 Direction
TXD[7:0]	FDAT_IN[7:0]	-	Out
TXPAR	FDAT_IN[8]	-	Out
TXMARK	FCTRL_OUT[16]	High	In
TXFLAG*	FCTRL_IN[0]	Low	Out
TXEN*	FCTRL_OUT[12]	Low	In
RXD[7:0]	FDAT_OUT[7:0]	-	In
RXPAR	FDAT_OUT[8]	-	In
RXMARK	FCTRL_OUT[10]	High	In
RXFLAG*	FCTRL_IN[4]	Low	Out
RXEN*	FCTRL_OUT[0]	Low	In
FRCTRL	FCTRL_OUT[4]	High	In

The interpretation of the ATM physical interface pins on the Bt8230 and the actual signals generated or received by the framer in UTOPIA mode are shown in Table 2-25. Note that both the TxClk and RxClk signals of the UTOPIA interface may be derived from the CLKD3 output of the Bt8230 (which must also be connected to the FRCTRL input).

Table 2-25. UTOPIA Mode Signals

Bt8230 Signal	PHY Signal	Active Polarity	Bt8230 Direction
TXD[7:0]	TxData[7:0]	–	Out
TXPAR	TxPrty	–	Out
TXMARK	TxSOC	High	Out
TXEN*	TxEnb*	Low	Out
TXFLAG*	TxFull*	Low	In
RXD[7:0]	RxData[7:0]	–	In
RXPAR	RxPrty	–	In
RXMARK	RxSOC	High	In
RXEN*	RxEnb*	Low	Out
RXFLAG*	RxEmpty*	Low	In
FRCTRL	RxClk/TxClk	Rising Edge	In

The interpretation of the ATM physical interface pins on the Bt8230 and the actual signals generated or received by the framer in slave UTOPIA mode is shown in Table 2-26.

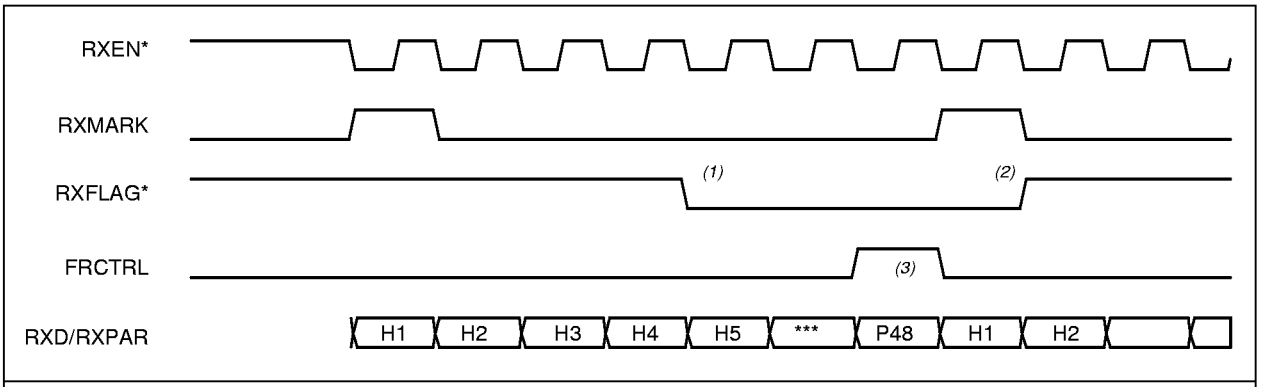
Table 2-26. Slave UTOPIA Mode Interface Signals

Bt8230 Signal	PHY Signal	Active Polarity	Bt8230 Direction
TXD[7:0]	TxData[7:0]	–	Out
TXPAR	TxPrty	–	Out
TXMARK	TxSOC	High	Out
TXEN*	TxEnb*	Low	In
TXFLAG*	TxEmp*	Low	Out
RXD[7:0]	RxData[7:0]	–	In
RXPAR	RxPrty	–	In
RXMARK	RxSOC	High	In
RXEN*	RxEnb*	Low	In
RXFLAG*	RxFull*	Low	Out
FRCTRL	RxClk/TxClk	Rising Edge	In

2.7.3 Bt8222 Mode Timing

As illustrated in Figure 2-30, received data is presented on the RXD[7:0], RXPAR, FRCTRL, and RXMARK by the framer and strobed into the Bt8230 using the RXEN* line. The adaptation logic computes the 8-bit odd parity over the RXD[7:0] lines and compares it to RXPAR. If in error, FR_PAR_ERR [bit 24] is set in the HOST_ISTAT0/LP_ISTAT0 registers. No data is discarded upon a parity error unless the RSM_PHALT bit in the RSM_CTRL Register is set to a logic high. If so, the reassembly coprocessor halts upon a parity error. The Bt8230 asserts the RXFLAG* line to indicate that its internal receive FIFO does not have room for another cell, and no more cells can be accepted; this normally results in cell loss. RXFLAG* is deasserted when there is room for a complete cell in the receive FIFO. The RXMARK input delimits the start of a cell. The FRCTRL signal is used by the adaptation logic to signal to the receive cell synchronizer that the current cell is errored or invalid and should be discarded. Even if the cell is invalid, a complete cell must be transferred or a synchronization error will occur. This signal is sampled on the last octet of each cell. The FR_RMODE bit in the CONFIG0 Register must be set to a logic high in this mode.

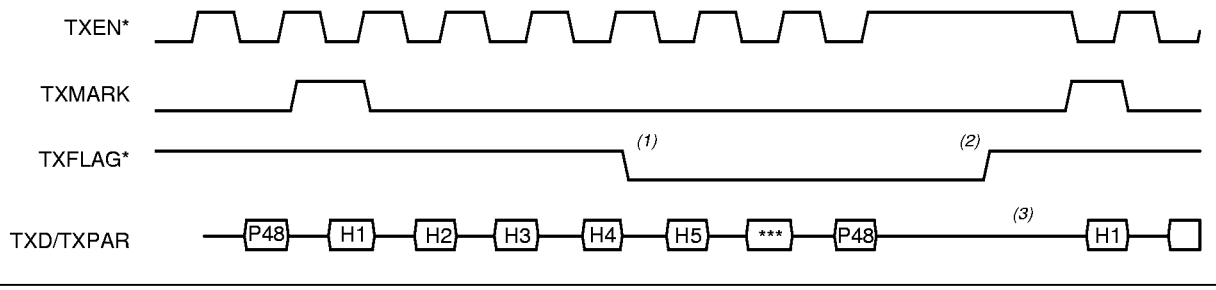
Figure 2-30. Receive Timing in Bt8222 Mode



Notes: (1). RXFLAG* goes active as soon as there is no room for another cell in the receive FIFO. However, there is still room for the current cell.
(2). RXFLAG* goes inactive when there is room for a complete cell in the receive FIFO.
(3). If FRCTRL is sampled high on the last octet of the cell, the cell is discarded.

Figure 2-31 illustrates how the TXEN* input from the framer is used as a data enable for the TXD[7:0] and TXPAR outputs and the TXMARK input. TXEN* also three-states the TXD and TXPAR outputs when a logic high so that multiple ATM SARs can be connected to the Bt8222 device. If another cell is not supplied by the transmit cell synchronization logic, the TXFLAG* output is asserted to indicate that a cell cannot be supplied in the next cell slot. The flag will be asserted after the first 4 bytes of the last cell are transmitted. As before, the TXPAR line carries the 8-bit odd parity computed over TXD[7:0]. The TXMARK line is asserted by the framer to indicate the start of each output cell.

Figure 2-31. Transmit Timing in Bt8222 Mode

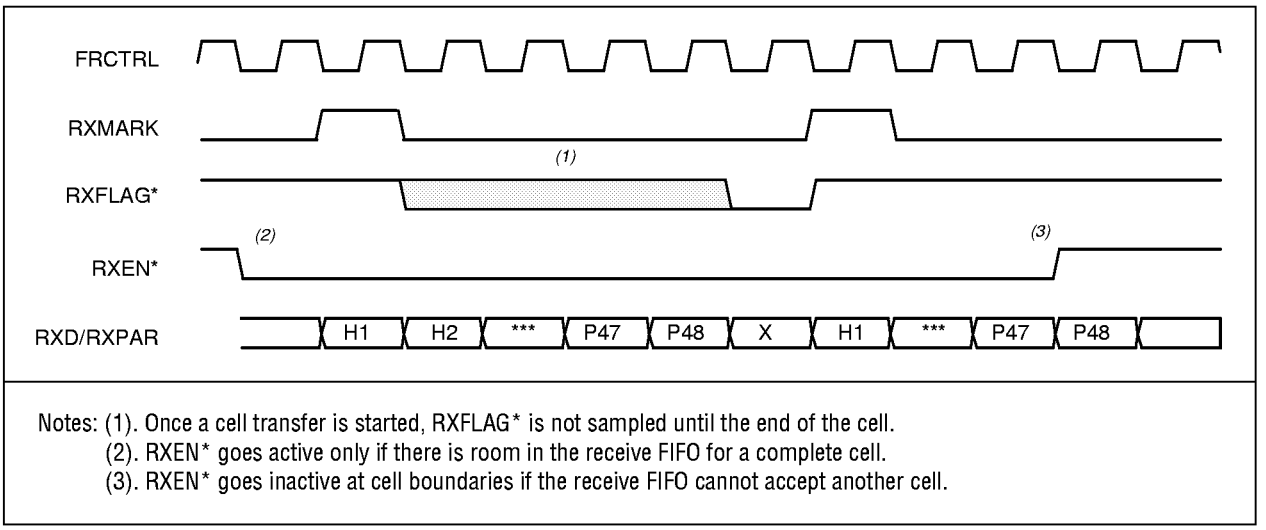


Notes: (1). TXFLAG* goes active when there is no longer a complete cell available in the transmit FIFO. This signal switches asynchronously to TXEN*.
(2). TXFLAG* goes inactive when a complete cell is in the transmit FIFO. This signal switches asynchronously to TXEN*.
(3). TXD and TXPAR are three-stated when TXEN* is a logic high.

2.7.4 UTOPIA Mode Cell Handshake Timing

If high, The UTOPIA_MODE bit in the CONFIG0 Register selects cell-level handshaking. Received data is latched from the RXD[7:0] and RXPAR lines on the rising edge of FRCTRL after RXEN* is sampled active (see Figure 2-32). The 8-bit odd parity computed over the RXD[7:0] lines is compared to the RXPAR input. If in error, the FR_PAR_ERR bit is set in the HOST_ISTAT0/LP_ISTAT0 registers. No data is discarded upon a parity error unless the RSM_PHALT bit in the RSM_CTRL Register is set to a logic high. If so, the Reassembly Coprocessor halts upon a parity error. The RXMARK signals to the Bt8230 the start of cell. The RXFLAG* input is the physical layer FIFO empty signal. When it is active, a complete cell is not present in the physical receive FIFO. The physical layer device sets RXFLAG* inactive when it has a complete cell to transfer. The Bt8230 sets RXEN* to a logic low if it can accept a complete cell. On the clock cycle after the last octet of a cell is transferred, the Bt8230 samples the RXFLAG* input. If low, the physical device does not have a cell to transfer. If RXFLAG* is high, the physical device has another cell to transfer and the Bt8230 will immediately start receiving the next cell if it can accept a complete cell. The FR_RMODE bit in the CONFIG0 Register should be set to a logic low in this mode.

Figure 2-32. Receive Timing in UTOPIA Mode with Cell Handshake

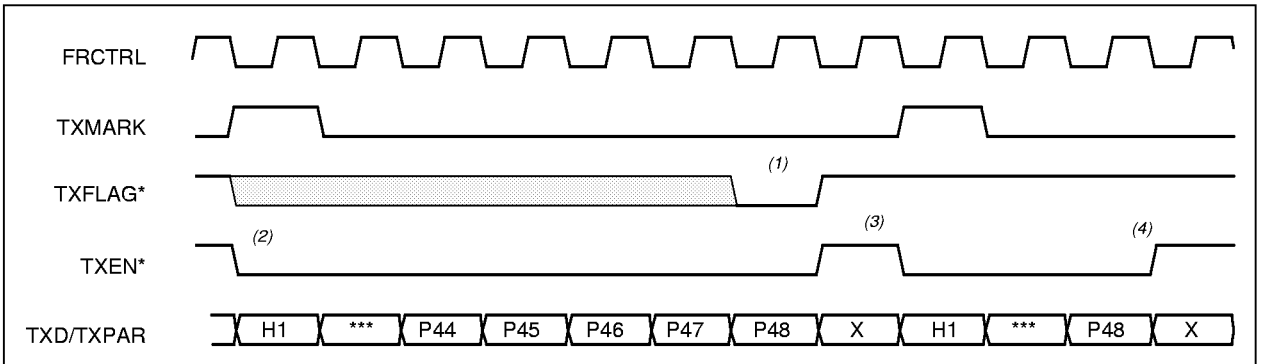


Notes: (1). Once a cell transfer is started, RXFLAG* is not sampled until the end of the cell.
(2). RXEN* goes active only if there is room in the receive FIFO for a complete cell.
(3). RXEN* goes inactive at cell boundaries if the receive FIFO cannot accept another cell.

Transmit data is driven on TXD[7:0] on the rising edge of FRCTRL when TXEN* is asserted, TXEN* is only asserted when there is data in the Bt8230 transmit FIFO. Simultaneously, the 8-bit odd parity computed over the TXD[7:0] lines is driven on to the TXPAR output. The TXMARK line is driven by the framer device to indicate start of cell. If the TXFLAG* input is asserted by the framer device, the framer device is full and another cell is not transmitted to the physical framer. (See Figure 2-33.)

In UTOPIA mode, the FRCTRL input may be connected to the Bt8230 CLKD3 output; a 50% duty cycle clock derived by dividing CLK2X by 3.

Figure 2-33. Transmit Timing in UTOPIA Mode with Cell Handshake

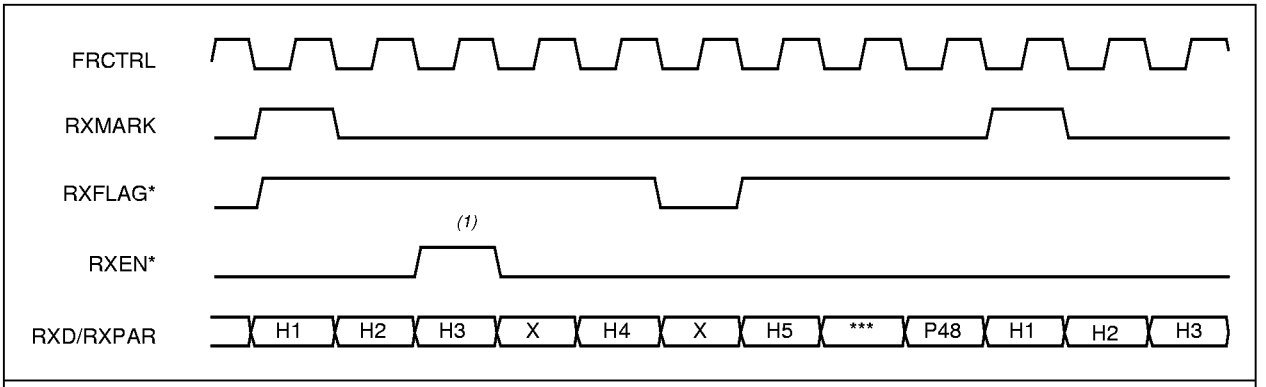


Notes: (1). Once transfer of a cell is started, TXFLAG* is sampled only on the last octet of a cell.
(2). TXEN* goes active if TXFLAG* is inactive at previous rising clock edge and a complete cell in the transmit FIFO.
(3). TXEN* goes inactive due to TXFLAG* being active on previous cycle.
(4). TXEN* goes inactive since a complete cell is not in the transmit FIFO.

2.7.5 UTOPIA Mode Octet Handshake Timing

If low, the UTOPIA_MODE bit in the CONFIG0 Register selects octet-level handshaking. Received data is latched from the RXD[7:0] and RXPAR lines on the rising edge of FRCTRL after RXEN* is sampled active (see Figure 2-34). The 8-bit odd parity computed over the RXD[7:0] lines is compared to the RXPAR input. If in error, FR_PAR_ERR in the HOST_ISTAT0/LP_ISTAT0 registers is set. No data is discarded upon a parity error unless the RSM_PHALT bit in the RSM_CTRL Register is set to a logic high. If so, the reassembly coprocessor halts upon a parity error. The RXMARK signals the start of cell to the Bt8230. The RXFLAG* input is the physical layer FIFO empty signal. When it is active, no data is present in the physical receive FIFO. The physical layer device sets RXFLAG* inactive when it has an octet to transfer. The Bt8230 sets RXEN* to a logic low if it can accept an octet in the next clock cycle. The FR_RMODE bit in the CONFIG0 Register should be set to a logic low in this mode.

Figure 2-34. Receive Timing in UTOPIA Mode with Octet Handshake

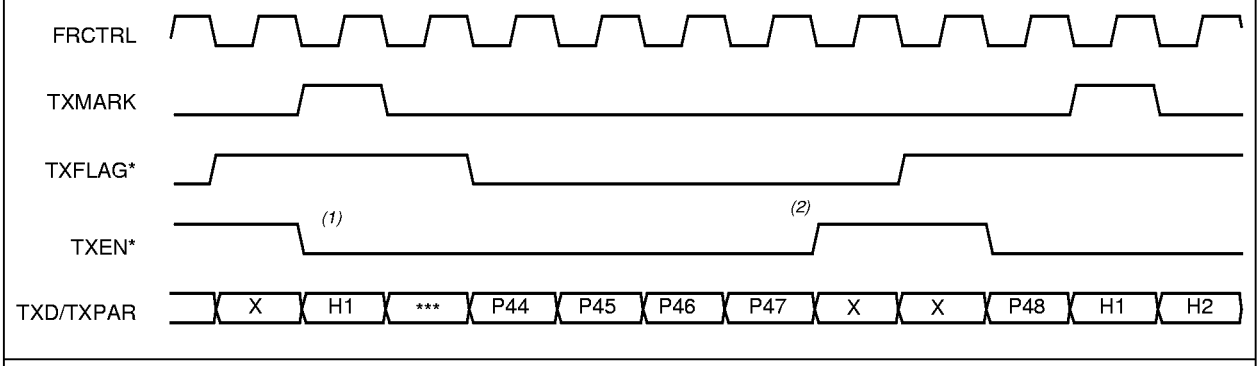


Note (1): RXEN* goes inactive only if no room for another octet in the receive FIFO.

Transmit data is driven on TXD[7:0] on the rising edge of FRCTRL when TXEN* is asserted, TXEN* is only asserted when there is data in the Bt8230 transmit FIFO. Simultaneously, the 8-bit odd parity computed over the TXD[7:0] lines is driven on to the TXPAR output. The TXMARK line is driven by the framer device to indicate start of cell. If the TXFLAG* input is asserted by the framer device, the framer device is full and can accept only one to four more octets. (See Figure 2-35).

In UTOPIA mode, the FRCTRL input may be connected to the Bt8230 CLKD3 output, which is a 50% duty cycle clock derived by dividing the CLK2X input by 3.

Figure 2-35. Transmit Timing in UTOPIA Mode with Octet Handshake



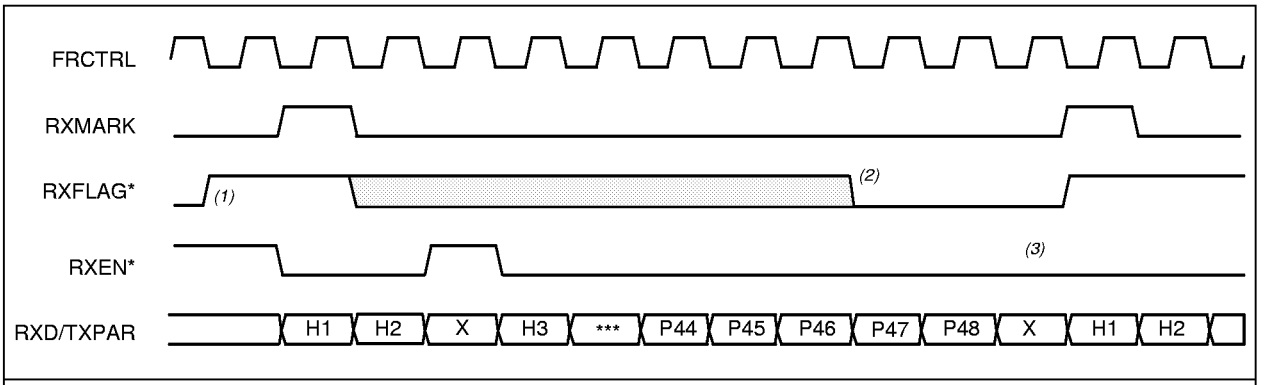
Notes: (1). TXEN* goes active if TXFLAG* is inactive at previous rising clock edge and a complete cell in the transmit FIFO.
(2). TXEN* goes inactive one to four clock cycles after TXFLAG* goes active.

2.7.6 Slave UTOPIA Mode

The slave UTOPIA mode is similar to the UTOPIA mode, except the direction of the enable signals and FIFO flags are reversed. This allows a switch fabric or backplane to directly control the physical port. The transmit and receive enable signals are generated by the physical layer instead of the Bt8230. The TxFull* signal is changed to the TxEmpty* signal and is an output of the Bt8230. The RxEmpty* signal is changed to the RxFull* signal and is also an output of the Bt8230. This mode only supports a cell-level handshake protocol.

Received data is latched from the RXD[7:0] and RXPAR lines on the rising edge of FRCTRL when RXEN* is active (see Figure 2-36). The 8-bit odd parity computed over the RXD[7:0] lines is compared to the RXPAR input. If in error, the FR_PAR_ERR bit is set in the HOST_ISTAT0/LP_ISTAT0 registers. No data is discarded upon a parity error unless the RSM_PHALT bit in the RSM_CTRL Register is set to a logic high. If so, the Reassembly Coprocessor halts upon a parity error. The RXMARK signals to the Bt8230 the start of cell. The RXFLAG* output is the receive FIFO full signal. When it is active, the Bt8230 cannot accept another cell. The Bt8230 sets RXFLAG* inactive when it has room in the receive FIFO for another cell. The physical device sets RXEN* to a logic low if it can transfer an octet. The FR_RMODE bit in the CONFIG0 Register should be set to a logic low in this mode.

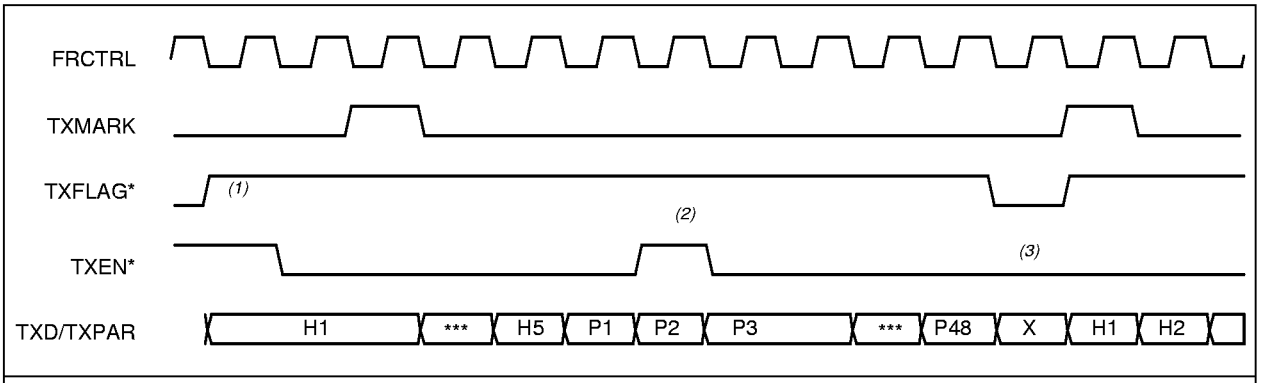
Figure 2-36. Receive Timing in Slave UTOPIA Mode



Notes: (1). RXFLAG* will go inactive when there is room in the receive FIFO for a complete cell.
(2). RXFLAG* will go active when there is no longer room in the receive FIFO for another complete cell.
(3). RXEN* need not be inactive when RXFLAG* is active.

Transmit data is driven on TXD[7:0] on the rising edge of FRCTRL after TXEN* is sampled asserted. Simultaneously, the 8-bit odd parity computed over the TXD[7:0] lines is driven on to the TXPAR output. The TXMARK line is driven by the framer device to indicate start of cell. If the TXFLAG* output is asserted by the Bt8230, the transmit FIFO does not contain a complete cell. (See Figure 2-37).

Figure 2-37. Transmit Timing in Slave UTOPIA Mode



Notes: (1). TXFLAG* goes inactive when a complete cell is in the transmit FIFO.
(2). Example of physical device deactivating TXEN*.
(3). Physical device may keep TXEN* active while TXFLAG* is active.

2.7.7 Loopback Mode

The physical interface can be internally looped by setting the FR_LOOP bit in the CONFIG0 Register to a logical high. This mode uses the internal system clock for operation; therefore, a framer clock is not needed during loopback operation.

2.7.8 Receive Cell Synchronization Logic

The receive cell synchronization logic accepts a stream of octets (together with error and cell boundary indications) from the receive ATM physical interface and performs the following functions:

- Maintains a sequence counter that marks the various components of an ATM cell: the 5-byte ATM header, the 1-byte HEC field within the header and the 48-byte payload. The sequence counter is also used by the ATM physical interface to check cell boundary synchronization.
- Extracts and discards the HEC byte from each 53-byte ATM cell, leaving 52 bytes of cell data. In Bt8222 mode, the Bt8222 will check the HEC byte.
- Formats consecutive 4-byte segments into 32-bit words; thus, the header forms the first word, the first 4 bytes of the payload form the next word, and so on. A total of 13 32-bit words are created from each 52-byte cell after the HEC byte has been removed. The bytes within each word are left-justified (big-endian format), i.e., the first byte received is the MSB of the word.
- Ensures that a complete cell (exactly 52 bytes) is always written to the FIFO. If a synchronization error occurs, the FR_SYNC_ERR bit in the HOST_ISTAT0/LP_ISTAT0 registers is set. The ATM physical interface attempts to resynchronize with the data stream.
- Sets the RSM_OVFL bit in the HOST_ISTAT0/LP_ISTAT0 registers if an octet could not be transferred due to the receive FIFO being full.

2.7.9 Transmit Cell Synchronization Logic

The transmit cell synchronization logic copies cell data from the transmit cell FIFO to the transmit ATM physical interface while performing the following functions:

- Reads 32-bit words from the transmit cell FIFO and converts them to a stream of octets, with the MSB of each 32-bit word corresponding to the first byte derived from that word (big-endian format).
- Maintains a sequence counter that delineates the various components of each ATM cell (4-byte header, 48-byte payload) in the outgoing byte stream.
- Inserts a blank (all-zero) HEC byte, used as a placeholder, into the outgoing byte stream representing each ATM cell. The HEC placeholder is inserted after the first 4 bytes (the ATM header) have been transferred. In Bt8222 mode, the Bt8222 device will calculate and insert the correct HEC value.
- Generates appropriate cell delineation pulses to the transmit ATM physical interface logic, for use in generating the TXMARK* output and also in verifying synchronization with the framer device.
- If the ATM physical transmit interface runs out of cells to transmit, the SEG_UNFL bit is set in the HOST_ISTAT0/LP_ISTAT0 registers.

The transmit cell synchronization logic supplies a continuous stream of octets to the transmit ATM physical interface unit, with cell delineation pulses at the starting byte of every cell. Only complete 53-byte cells are supplied to the ATM physical interface. If the transmit cell FIFO is empty, the transmit cell synchronization logic indicates that no more data can be transferred to the framer.

2.8 PCI Bus Interface

The PCI bus interface is compliant with PCI Local Bus Specification, Revision 2.0. This interface is completely synchronous to the PCI bus clock (HCLK). All inputs are sampled at the rising edge of HCLK, and all outputs are driven by the Bt8230 to be valid before the next rising edge of HCLK.

The maximum PCI bus clock rate guaranteed to be supported by the Bt8230 is 33 MHz. The PCI bus interface logic is clocked directly from the PCI bus clock, but the remainder of the Bt8230 logic runs off of a separate system clock. Synchronizing registers and FIFOs are implemented in the PCI bus interface to transfer data between the PCI bus clock and the system clock.

The PCI bus drivers are shared between the master and slave bus interface functional blocks. The PCI bus master logic (within the device) arbitrates with the PCI bus arbiter (external to the device) for access to the PCI bus; access to the PCI bus automatically implies access to the bus drivers, as no other master can be communicating with the slave logic. The bus master logic contends for the bus on a transaction by transaction basis.

The PCI bus interface responds to read and write requests by the host CPU, allowing access to chip resources by host software. The Bt8230 is also capable of acting as a DMA bus master on the PCI bus; as a result, the PCI bus interface implements the full set of address, data, and control signals required to drive the bus as a master, and contains the logic required to support arbitration for the PCI bus. Note that the DMA coprocessor and the PCI bus interface are closely linked and, hence, are shown as one unit.

The PCI bus interface functional blocks are:

- I/O drivers and receivers that drive the pins connected to the PCI bus signals.
- PCI bus master logic that allows the bus interface to acquire mastership of the PCI bus and act as a transaction initiator. The bus master logic also contains a command decoder that interprets access commands generated by the DMA coprocessor, and a burst controller for controlling the duration of each read or write burst. In addition, the bus master logic contains address counters that allow it to restart and retry burst transfers if required by the transaction target.
- Burst FIFO buffers that store and transfer bursts of data words between the DMA coprocessor and the PCI bus master logic.
- PCI bus slave logic that responds to transactions initiated by other masters on the PCI bus with the Bt8230 as a target. The bus slave logic also synchronizes data passed back and forth across the clock boundary between the PCI bus interface and the internal chip logic.
- Configuration registers holding initialization parameters and PCI bus status information.
- Logic that allows the host CPU to read/write the internal Bt8230 registers via the PCI slave port.
- Logic to enable read/write access to the local memory space from the host CPU, again via the PCI slave port.

2.8.1 Unsupported PCI Bus Interface Functions

The PCI bus interface on the Bt8230 does not support all the transaction types defined by the PCI bus specification; only those sections of the protocol that are necessary for slave and DMA memory accesses are implemented. In particular, the following transaction types are not implemented:

- 64-bit transfers, as well as the Dual Address Cycle command.
- Snooping and cache support. The Memory Read Multiple, Memory Read Line and Memory Write and Invalidate commands are internally aliased to the Memory Read and Memory Write commands as per the PCI specification.
- Locked and exclusive accesses: the PCI LOCK* line is not driven by the Bt8230, and the PCI slave interface does not handle locked accesses by other bus masters in any special manner.
- I/O accesses (the I/O Read and I/O Write commands).
- Interrupt acknowledge cycles, including the Interrupt Acknowledge command.
- The Special Cycle command and special cycle transactions.
- Burst transfers that do not have simple, sequentially incrementing addresses for consecutive data phases. The PCI master logic always performs sequentially incrementing burst transfers. The two LSBs of the PCI address lines (AD[1,0]) must be zero during the address phase of any transfer made to the PCI slave logic (indicating sequentially incrementing burst addresses). If AD[1,0] is not equal to zero, the slave logic will signal a type A or B target disconnect after the first data phase, forcing the external master to perform a single word transfer as per the PCI specification.

2.8.2 PCI Bus Master Logic

The PCI bus master logic block is responsible for accepting read and write commands from the DMA coprocessor (passed via the burst FIFO buffers) and in turn acquiring mastership of the PCI bus and generating transactions to perform the actual data transfers. The bus master logic contains a command decoder, which interprets commands issued from the DMA coprocessor; a burst controller that counts off read and write cycles in each burst on the PCI bus (and also latches and drives the address and command during the address phase of each transfer); arbitration logic that acquires control of the PCI bus; and a bus state machine that sequences and controls transfers.

The bus master implements a software-configurable maximum burst length counter. This counter is started at the beginning of each read or write burst transaction, and counts the actual number of words transferred during the burst. When it reaches the value set in the MAX_BRST_LEN field of the PCI configuration space, the burst is terminated and a new address phase is begun.

It is possible for the addressed slave to request a disconnect or a retry during a read or a write transfer, using the defined PCI protocol sequence. In this case, the bus master logic will terminate the current burst, maintain its bus request, and restart the transfer at the point of termination. Disconnects and retries are not regarded as errors.

Five possible sources of error are present during any PCI bus master transaction. If the any of the following five errors occur, the bus master logic will permanently terminate the transaction, flag an error, and cease to process any more commands until either the corresponding status flag in the PCI configuration space has been cleared or the PCI master has been reset.

1. Target Abort—The PCI transaction will terminate if the addressed target signals a target abort. In this case, the RTA and MERROR bits in the PCI Configuration Register space will be set and the PCI_BUS_STATUS[4] bit in the SYS_STAT Register will be set.
2. Master Abort—If the addressed target does not respond with an HDEVSEL* assertion, then a master abort is flagged. In this case, the RMA and MERROR bits in the PCI Configuration Register space will be set and the PCI_BUS_STATUS[3] bit in the SYS_STAT Register will be set.
3. Parity Error—If the data parity checked during read transfers is inconsistent with the state of the HPAR signal, then a parity error is signaled. In this case, the DPR and MERROR bits in the PCI Configuration Register space will be set and the PCI_BUS_STATUS[2] bit in the SYS_STAT Register will be set.
4. Interface Disabled—If the driver or application software on the PCI host CPU has disabled, the Bt8230 PCI bus master logic (using the M_EN bit in the Command field of the PCI bus configuration registers), then any attempt to perform a DMA transaction to the PCI bus will result in an error. In this case, the MERROR and INTF_DIS bits in the PCI configuration space will be set and the PCI_BUS_STATUS[1] bit in the SYS_STAT Register will be set.
5. Internal Failure—Upon a synchronization error between the DMA coprocessor and the PCI master logic, an internal failure will be flagged. In this case, the MERROR and INT_FAIL bits in the PCI configuration space will be set and the PCI_BUS_STATUS[0] bit in the SYS_STAT Register will be set.

As mentioned above, bus protocol errors can be cleared either by a software reset of the associated status flag or flags, i.e., RTA, RMA or DPR, or with a reset of the PCI bus master logic using the HRST* input pin. For example, a master abort error can be cleared by writing a logic one to the RMA status bit in the PCI Configuration Register space, causing the status bit to be cleared and the master to resume normal operation. The MERROR bit in the PCI Configuration Register drives the PCI_BUS_ERROR interrupt. To clear this interrupt, a logic high must be written to the MERROR bit location. The MERROR bit can also be cleared by a logic low on the HRST* input pin.

Several fields are provided in the PCI configuration space to aid in recovering from a PCI master error. The PCI host software can determine that an error occurred by checking the MERROR bit. It also can determine if the transaction was a read or write by inspecting the MRD bit, and the read or write address at which an error occurred by reading the CUR_MSTR_RD_ADDR or CUR_MSTR_WR_ADDR fields.

2.8.3 Burst FIFO Buffers

Two small FIFO buffers are implemented to support PCI burst-mode operation, to allow synchronization between the Bt8230 internal logic and the PCI bus interface, and to carry commands from the DMA coprocessor to the PCI bus logic. The incoming FIFO is 128 x 36 bits, the outgoing FIFO is 16 x 36 bits.

2.8.4 PCI Bus Slave Logic

The PCI slave logic permits the host CPU on the PCI bus to access and modify the Bt8230 resources (the external local memory and internal registers). As the control processor also has access to these resources, the PCI slave logic must arbitrate for access prior to performing any read or write transaction. The slave logic also contains the PCI configuration registers; these registers control the PCI slave and master interfaces and may be read or written at any time by the PCI host. The slave logic implements the synchronizers required for rate-matching between the PCI bus clock and the internal Bt8230 system clock. Also, small FIFOs are used to speed up burst reads and writes performed by the host processor to local resources by buffering prefetched read data and absorbing latency during consecutive writes.

In general, the PCI slave interface functions as a normal memory-mapped PCI target, responding to Memory Read, Memory Write, Configuration Read and Configuration Write commands from any initiator on the PCI bus. Note that the slave interface will only respond to Memory Read and Memory Write commands if the MS_EN bit of the Command field in the PCI Configuration Register has been set.

The PCI slave logic does not implement special cycle commands, or respond to special cycles on the PCI bus. If a master performs a special cycle on the PCI bus:

- The slave logic never asserts HDEVSEL*.
- Parity errors during the address phase of the special cycle command will be reported be asserting HSERR* in the normal fashion, if SE_EN and PE_EN in the command register are both set.
- Parity errors during the data phase are ignored.

2.8.5 PCI Host Address Map

The address map of the Bt8230 resources seen by the PCI bus are the same as seen by the local processor. Refer to Section 2.4 for further detail. The base address of the Bt8230 resource mapping is defined in the ExtMemBase field located in the PCI configuration space.

2.8.6 PCI Configuration Space

For details of the PCI configuration space, refer to Section 3.8.

3.0 Registers

3.1 Bt8230 Register Overview

Control and status registers of the Bt8230 are listed in Table 3-1. Detailed register descriptions are provided in this section. Note that byte enables are ignored by the Bt8230 when writing to control and status registers.

Table 3-1. Bt8230 Control and Status Registers *(1 of 2)*

Address	Name	Type	Description
0x00	CLOCK	R/W B, L	Real Time Clock Register
0x04	ALARM1	R/W L, NL	Alarm Register 1
0x08	Reserved	–	Not Implemented
0x0C	SYS_STAT	R/O	System Status Register
0x10	Reserved	–	Not Implemented
0x14	CONFIG0	R/W B, L	Basic Configuration And Control Register
0x18	Reserved	–	Not Implemented
0x1C	Reserved	–	Not Implemented
0x20	SEG_CTRL	R/W B, L	Segmentation Control Register
0x24	SEG_SBASE	R/W B, L	Segmentation Status Queue Base Address Register
0x28	SEG_VBASE	R/W B, L	Segmentation Virtual Connection Area Base Address Register
0x2C	SEG_ST	R/O	Segmentation Status Position Register
0x30	SEG_ABR	R/W B, L	Segmentation Available Bit Rate Parameter Register
0x34	SEG_FR_BD	R/W B, L	Seg Free Buffer Descriptor Register
0x38	SEG_HRBASE	R/W B, L	Segmentation Host Transmit Ring Base Address Register
0x3C	SEG_LRBASE	R/W B, L	Segmentation Local Transmit Ring Base Address Register
0x40	SEG_HXMIT	R/O-W/O B, L	Segmentation Host Transmit Ring Register
0x44	SEG_LXMIT	R/O-W/O B, L	Segmentation Local Transmit Ring Register
0x48–0x6C	Reserved	–	Not Implemented
0x70	RSM_CTRL	R/W B, L	Reassembly Control Register

Table 3-1. Bt8230 Control and Status Registers (2 of 2)

Address	Name	Type	Description
0x74	RSM_SBASE	R/W B, L	Reassembly Status Queue Base Address Register
0x78	RSM_FBASE	R/W B, L	Reassembly Free Buffer Queue Base Address Register
0x7C	RSM_HBASE	R/W B, L	Reassembly Hash Table Base Address Register
0x80	RSM_TO	R/W R, L	Reassembly Time Out Register
0x84	RSM_FW	R/W R, L	Reassembly Firewall Register
0x88	RS_QBASE	R/W B, L	Reassembly/Segmentation Queue Base Address Register
0x8C–0xBC	Reserved	–	Not Implemented
0xC0	HOST_ISTAT0	R/O	Host Interrupt Status Register
0xC4	Reserved	–	Not Implemented
0xC8	Reserved	–	Not Implemented
0xCC	HOST_IMASK0	R/W H, NL	Host Interrupt Mask Register
0xD0	Reserved	–	Not Implemented
0xD4	Reserved	–	Not Implemented
0xD8	HOST_MBOX	R/W L, NL	Host Mailbox Register
0xDC	Reserved	–	Not Implemented
0xE0	LP_ISTAT0	R/O	Local Processor Interrupt Status Register
0xE4	Reserved	–	Not Implemented
0xE8	Reserved	–	Not Implemented
0xEC	LP_IMASK0	R/W L, NL	Local Processor Interrupt Mask Register
0xF0	Reserved	–	Not Implemented
0xF4	Reserved	–	Not Implemented
oxF8	LP_MBOX	R/W H, NL	Local Processor Mailbox Register
0xFC	Reserved	–	Not Implemented

3.2 Register Terminology

The access type terminology given in Table 3-2 applies to all registers in this section. Each register description is prefaced with the appropriate abbreviated access types.

Table 3-2. Bt8230 Register Terminology

Abbreviation	Description
R/W B	Read and write access for both host and local processors.
R/W H	Read and write access for host, read only access for local processor.
R/W L	Read and write access for local, read only access for host processor
R/W R	Read and write access for both processors with restrictions.
R/O-W/O B	Part of this register is read only, part write only by both processors.
R/O	Read only access for both processors
L	Lockable access by LP_LOCK and HOST_LOCK control bits.
NL	Non-lockable access by any processor with write capability.

3.3 General Purpose Registers

3.3.1 0x00—Real Time Clock Register (CLOCK)

Access types: R/W B, L
The Real Time Clock Register is located at address 0x00. This register contains the 32-bit real time clock. It is incremented by the system clock (SYSCLK) which has been divided by the DIVIDER[6:0] field in the CONFIG0 register. It may be written to any value by each processor, and may generate an interrupt on the RTC_OVFL status bit in the LP_ISTAT0 register when it overflows from 0xFFFFFFFF to 0x0.

Bit	Field Size	Name	Description
31–0	32	CLOCK[31:0]	Real-time clock value.

3.3.2 0x04—Alarm Register 1 (ALARM1)

Access types: R/W L, NL
Alarm Register 1 is located at address 0x04. This register contains a 32-bit value which is compared against the CLOCK register. When the two registers are equal, the ALARM1 status bit in the LP_ISTAT0 register is latched and can be enabled to cause an interrupt to the local processor. To implement a periodic interrupt, add a constant value to this register after each interrupt.

Bit	Field Size	Name	Description
31–0	32	ALARM1[31:0]	ALARM1 comparison value.

3.3.3 0x0C—System Status Register (SYS_STAT)

Access types: R/O
The System Status Register is located at address 0x0C and provides read-only system status. This register reflects the device ID and version information for the part, as well as pin programmable options that otherwise may not be visible to the processors. It also contains expanded information for the status located in the HOST_ISTAT0 and LP_ISTAT0 registers.

Bit	Field Size	Name	Description
31–17	15	Reserved	Not implemented at this time
16–12	5	PCI_BUS_STATUS [4:0]	The status bits are as follows: 4: Target Abort 3: Master Abort 2: Parity Error 1: Interface Disabled 0: Internal Failure See Section 2.8 for more detail. Error status is also condensed into the BUS_ERROR status bit in the HOST_ISTAT0 and LP_STAT0 registers.
11	1	RAMMODE	Reflects the state of the RAMMODE input pin.
10	1	PROCMODE	Reflects the state of the PROCMODE input pin
9, 8	2	FRCFG[1,0]	Reflects the state of the FRCFG[1,0] input pins.
7–4	4	VERSION [3:0]	Version number for the Bt8230 will be 0001 for version 1.
3–0	4	DEVICE[3:0]	Device ID for the Bt8230, set to 0000.

3.3.4 0x14—Configuration Register 0 (CONFIG0)

Access types: R/W B, L

Configuration Register 0 is located at address 0x14. This register provides all of the control and configuration bits that are not associated with the reassembly and segmentation coprocessors. The majority of these bits are configuration (which occurs at initialization time) and are not changed dynamically. The assertion of the HRST* system reset pin will clear all of the bits in the CONFIG0 register except for MEMCTRL, which will be set high.

Bit	Field Size	Name	Description
31	1	LP_ENABLE	When set, this bit causes the PRST* output pin to be high. This may be used to reset the local processor.
30	1	GLOBAL_RESET	When set, this bit causes reset of the segmentation and reassembly coprocessors as well as all latched status.
29	1	PCI_MSTR_RESET	When set, this bit resets the PCI master logic. Once active, this bit must stay active for 16 cycles of the HCLK input signal.
28–26	3	Reserved	Always set to zero.
25	1	LP_LOCK	When set, this bit disables write access by the host to Bt8230 registers and local memory, with the exception of the LP_MBOX and HOST_IMASK0 registers. Read accesses are not affected. This bit cannot be set by the host processor, nor can it be set if the HOST_LOCK bit is active.
24	1	HOST_LOCK	When set, this bit disables write access by the local processor to SRC registers and local memory, with the exception of the HOST_MBOX and LP_IMASK0 registers. Read accesses are not affected. This bit cannot be set by the local processor. This bit cannot be set if the LP_LOCK bit is active.
23–21	3	Reserved	Always set to zero.
20	1	PCI_ARB	Selects PCI master arbitration scheme. When a logic low, enables round-robin between read and write requests. When a logic high, reads have priority over writes.
19–16	4	STATMODE[3:0]	Selects which internal status to output on the STAT[1,0] output pins.
15	1	FR_RMODE	Controls reassembly start of cell processing. When set low, processing starts after the first two words of a cell are received. When set high, a complete cell must be in reassembly FIFO before cell is processed. Note: In 8222 mode, this bit must set high.
14	1	FR_LOOP	When set, this bit enables loopback of cells at the ATM physical interface.
13	1	UTOPIA_MODE	Selects byte or cell UTOPIA handshake mode. 0 = Octet handshake 1 = Cell handshake
12	1	ENDIAN	Selects between Little and Big Endian host data structures. 0 = Little Endian 1 = Big Endian
11	1	LP_BWAIT	Selects zero or one wait states between consecutive data cycles during local processor burst accesses.
10	1	MEMCTRL	Selects zero or one wait states local memory (1 or 2 cycle).

Bit	Field Size	Name	Description
9–7	3	BANKSIZE[2:0]	Selects size of memory banks for contiguous memory support. See Section 2.3 for further explanation.
6–0	7	DIVIDER[6:0]	Prescaler for SYSCLK which advances the counter in the CLOCK Register. SYSCLK is divided by the divider value, if zero, divided by 128.

3.4 Segmentation Registers

3.4.1 0x20—Segmentation Control Register (SEG_CTRL)

access types: R/W B, L

The Segmentation Control Register is located at address 0x20 and contains the general control bits for the segmentation coprocessor. The assertion of the HRST* system reset pin or GLOBAL_RESET bit in the CONFIG0 register will clear the SEG_ENABLE control bit.

Bit	Field Size	Name	Description
31	1	SEG_ENABLE	Enables the Segmentation Coprocessor. If disabled, the segmentation coprocessor will halt on a cell boundary. This bit will also be set low internally when the local or host status queue overflow. In this case, the error condition should be corrected and the SEG_ENABLE bit set high to resume operation.
30	1	SEG_RESET	Resets the segmentation coprocessor and pointers.
29–17	13	Reserved	Program and read as zero.
16	1	SEG_432	When a logic zero, idle cells contain all zeros. When a logic high, idle cells have an ATM header of 0x0000_0001 and payload octets of 0x6A in compliance with the I.432 standard.
15	1	BOFFSET	When a logic zero, segmentation buffers will include 8 bytes of user-defined data at the beginning of each buffer. When a logic high, no user-defined space is available in the buffers.
14	1	SEG_XMT_ALT	When a logic zero, the host xmit ring will be emptied before the local xmit ring is processed. When a logic high, processing of the rings will alternate when both rings have entries.
13	1	SEG_LS_DIS	Disables the segmentation coprocessor halt on local status queue full condition.
12	1	SEG_HS_DIS	Disables the segmentation coprocessor halt on host status queue full condition
11, 10	2	MAXHR[1,0]	Sets the host transmit ring number of entries. See Table 3-3 for size encoding.
9, 8	2	MAXLR[1,0]	Sets the local transmit ring number of entries. See Table 3-3 for size encoding.

Bit	Field Size	Name	Description
7, 6	2	MAXI[1,0]	Maximum rate control / field (intercell interval) in VCC table for rate controlled connections. The slowest possible cell rate is 1/MAXI. See Table 3-3 for size encoding.
5, 4	2	MAXPND[1,0]	Maximum number VCCs waiting to send first cell after data (re)start. Also maximum number of ABR VCC. See Table 3-3 for size encoding.
3, 2	2	MAXHS[1,0]	Maximum number of status messages waiting to be read by the host. The size of the host status queue is MAXHS * 4 bytes. See Table 3-3 for size encoding.
1, 0	2	MAXLS[1,0]	Maximum number of status messages waiting to be read by the local processor. The size of the local status queue is MAXLS * 4 bytes. See Table 3-3 for size encoding.

Table 3-3. Segmentation MAXx Size Encoding

MAXx	Size
00	256
01	1024
10	4096
11	16384

3.4.2 0x24—Segmentation Status Queue Base Register (SEG_SBASE)

Access types: R/W B, L

The Segmentation Status Queue Base Register is located at address 0x24. This register sets the base address in the Bt8230 local memory for the local and host segmentation status queues. Both base addresses are 128-byte aligned and only the 16 most significant bits of the address are specified.

Bit	Field Size	Name	Description
31–16	16	SEG_LSBASE[15:0]	Base address for the segmentation local status queue. The size of the local status queue is MAXLS * 4 bytes.
15–0	16	SEG_HSBASE[15:0]	Base address for the segmentation host status queue. The size of the host status queue is MAXHS * 4 bytes.

3.4.3 0x28—Segmentation Virtual Connection Base Register (SEG_VBASE)

Access types: R/W B, L
The Segmentation Virtual Connection Base Register is located at address 0x28. This register sets the base address in the Bt8230 local memory for the segmentation VCC table and the segmentation reserved area. Both base addresses are 128-byte aligned and only the 16 most significant bits of the address are specified.

Bit	Field Size	Name	Description
31–16	16	SEG_RSVDB[15:0]	Base address segmentation reserved area. The size of the reserved area is (MAXI + MAXPND) * 4 bytes.
15–0	16	SEG_VCCB[15:0]	Base address for the segmentation VCC table.

3.4.4 0x2C—Segmentation Status Position Register (SEG_ST)

Access types: R/O
This Segmentation Status Position Register is located at address 0x2C. This register shows the current position of the segmentation coprocessor in the local and host status queues. The values in the register are pointer offsets from the base address of the appropriate status queue. The current host status queue entry is at address SEG_HSB * 128 + (SEG_HPTR – 1) * 4. The current local status queue entry is at address SEG_LSB * 128 + (SEG_LPTR – 1) * 4.

Bit	Field Size	Name	Description
31, 30	2	Reserved	Always read as zero.
29–16	14	SEG_LPTR[13:0]	Current segmentation coprocessor position in local status queue.
15, 14	2	Reserved	Always read as zero.
13–0	14	SEG_HPTR [13:0]	Current segmentation coprocessor position in host status queue.

3.4.5 0x30—Segmentation Available Bit Rate Register (SEG_ABR)

Access types: R/W B, L
The Segmentation Available Bit Rate Register is located at address 0x30. It sets the maximum rate parameters for the overall segmentation ABR cell stream. The I and L parameters must be limited to I + L **less than or equal** to 0x3FFFF.

Bit	Field Size	Name	Description
31	1	ABREN	Enables available bit rate control. If this bit is not set, ABR cells will be sent during all non-scheduled cell slots.
30–13	18	ABRL[17:0]	Rate Control *L* Field. This field is the corresponding parameter in the ATM UNI 3.0 GCRA specification (Section 3.6.2.4.1). This parameter is a fixed point number with 8 fractional bits and is in units of cell slots.
12–0	13	ABRI[12:0]	Rate Control/Field. This field is the corresponding parameter in the ATM UNI 3.0 GCRA specification (Section 3.6.2.4.1). This parameter is a fixed point number with 8 fractional bits and is in units of cell slots.

3.4.6 0x34—Segmentation Free Buffer Descriptor Register (SEG_FR_BD)

Access types: R/W R, L
The Segmentation Free Buffer Descriptor Register is located at address 0x34. This register can only be written if the SEG_RUN status bit in the HOST_ISTAT0 and LP_ISTAT0 registers is a logic low indicating that the segmentation coprocessor is halted.

Bit	Field Size	Name	Description
31–23	9	Reserved	Always program and read as zero.
22–2	21	NXT_FREE [20:0]	The word-aligned address of the next free buffer descriptor.
1, 0	2	Reserved	Always program and read as zero.

3.4.7 0x38—Segmentation Host Transmit Base Register (SEG_HRBASE)

Access types: R/W B, L
The Segmentation Host Transmit Base Register is located at address 0x38. This register sets the base address of the host transmit ring.

Bit	Field Size	Name	Description
31–0	32	SEG_HRBASE[31:0]	Base address of the host transmit ring. Must be word (32-bit) aligned.

3.4.8 0x3C—Segmentation Local Transmit Base Register (SEG_LRBASE)

Access types: R/W B, L
The Segmentation Local Transmit Base Register is located address 0x3C. This register sets the base address of the local transmit ring.

Bit	Field Size	Name	Description
31–16	16	Reserved	Always program and read as zero.
15–0	16	SEG_LRBAS [15:0]	Base address of the local transmit ring.

3.4.9 0x40—Segmentation Host Transmit Register (SEG_HXMIT)

Access types: R/O-W/O B, L
The Segmentation Host Transmit Register is located at address 0x40. This register contains both the current number of entries on the host transmit ring status, and the number of new entries to add to the transmit ring control. The SEG_HPND field in this register is limited to the size set in the MAXHR field in the SEG_CTRL register, it will not roll over. Exceeding this value will cause the HRING_OVFL status in the HOST_ISTAT0 and LP_ISTAT0 registers to be set.

Bit	Field Size	Name	Description
31–30	2	Reserved	Always program and read as zero.
29–16	14	SEG_HPND[13:0]	SEG_HPND is the number of entries on the host transmit ring that have not been processed. This field is read only.
15–8	8	Reserved	Always program and read as zero.
7–0	8	SEG_HNEW[7:0]	SEG_HNEW is the number of entries added by a processor to the host transmit ring. These bits are automatically added to SEG_HPND and cleared. They are read as zeros.

3.4.10 0x44—Segmentation Local Transmit Register (SEG_LXMIT)

Access types: R/O-W/O B, L
The Segmentation Local Transmit Register is located at address 0x44. This register contains both the current number of entries on the local transmit ring status, and the number of new entries to add to the transmit ring control. The SEG_LPND field in this register is limited to the size set in the MAXLR field in the SEG_CTRL register, it will not roll over. Exceeding this value will cause the LRING_OVFL status in the HOST_ISTAT0 and LP_ISTAT0 registers to be set.

Bit	Field Size	Name	Description
31–30	2	Reserved	Always program and read as zero.
29–16	14	SEG_LPND[13:0]	SEG_LPND is the number of entries on the local transmit ring that have not been processed. This field is read only.
15–8	8	Reserved	Always program and read as zero.
7–0	8	SEG_LNEW[7:0]	SEG_LNEW is the number of entries added by a processor to the local transmit ring. These bits are automatically added to SEG_LPND and cleared, they will read as zeros.

3.5 Reassembly Registers

3.5.1 0x70—Reassembly Control Register (RSM_CTRL)

Access types: R/W B, L
The Reassembly Control Register is located at address 0x70 and contains the general control bits for the reassembly coprocessor. The assertion of the HRST* system reset pin or the GLOBAL_RESET bit in the CONFIG0 register clear the RSM_ENABLE control bit.

Bit	Field Size	Name	Description
31	1	RSM_ENABLE	Reassembly enable. Initiates an incoming transfer if set, and halts it if reset. If this bit is reset while the reassembly coprocessor is running, it temporarily suspends the activities of the reassembly coprocessor logic. Suspension takes place on a cell boundary, i.e., between the completion of all processing and transfers required for the current cell and the start of processing for the next cell. The hold can be removed, and the transfer resumed by setting the RSM_ENABLE bit. This bit will also be set low internally on certain reassembly error conditions. These include free buffer queue empty, status queue full or parity error with PHALT_EN. In this case, the error condition should be corrected and the RSM_ENABLE bit set high to resume operation.
30	1	RSM_RESET	Reassembly reset. Forces a hardware reset of the reassembly coprocessor when asserted. It must be deasserted before the reassembly coprocessor will resume normal operation.

Bit	Field Size	Name	Description
29–21	9	Reserved	Not implemented at this time
20–17	4	TAB_SIZE[3:0]	VPI/VCI hash table size. The TAB_SIZE field indicates the size, in powers of 2, of the hash table used by the reassembly coprocessor to manage incoming cell streams and perform the per-connection linked-list generation task. The values in the field are interpreted according to Table 3-4.
16	1	RSM_PHALT	Reassembly coprocessor halt on parity error detect. The reassembly coprocessor will halt the incoming channel logic if a parity error is detected and the RSM_PHALT bit is set.
15	1	OAM_EN	Operating and maintenance enable. Enables detection and processing of OAM cells.
14	1	OAM_MEM	Operating and maintenance memory. OAM processing sent to local memory instead of host memory when set.
13	1	FWALL_EN	Firewall enable. Enables free buffer queue firewalling of user cells. If set, this bit enables the per connection free buffer queue firewall. Each connection that firewall is active in must have the FW_EN bit set to a logic high in the hash bucket.
12	1	CACHE_ENABLE	Cache enable. When set, this bit enables the hash table and VCC table internal caches.
11	1	RSM_LF_DIS	Local Free Buffer Queue Overwrite Disable. Disables the local Free Buffer Queue read checks if set.
10	1	RSM_HF_DIS	Host Free Buffer Queue Overwrite Disable. Disables the host free buffer queue read checks if set.
9	1	RSM_LS_DIS	Local Status Queue Overwrite Disable. Disables the local status queue write checks if set.
8	1	RSM_HS_DIS	Host Status Queue Overwrite Disable. Disables the host status queue write checks if set.
7, 6	2	HF_SIZE	Set Size Of Host Free Buffer Queue. Sets the number of entries of the host free buffer queue as shown in Table 3-5.
5, 4	2	HS_SIZE	Set Size of Host Status Queue. Sets the number of entries of the host status queue as shown in Table 3-5.
3, 2	2	LF_SIZE	Set Size of Local Free Buffer Queue. Sets the number of entries of the local free buffer queue as shown in Table 3-5.
1, 0	2	LS_SIZE	Set Size of Local Status Queue. Sets the number of entries of the local status queue as shown in Table 3-5.

Table 3-4. TAB_SIZE Encoding

TAB_SIZE	Hash Table Size, Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	Invalid
14	16384
15	Invalid

Table 3-5. Reassembly Buffer Size Coding

XX_SIZE	Buffer Size
00	256
01	1024
10	4096
11	16384

3.5.2 0x74—Reassembly Status Queue Base Register (RSM_SBASE)

Access types: R/W B, L
The Reassembly Status Queue Base Register is located at address 0x74. This register determines the base address of both the host and local status queue structures. The base address is a 16-bit number. Since both local and host status queues reside in local memory (23 bits of byte addressing) the structures can start on 128-byte boundaries.

Bit	Field Size	Name	Description
31–16	16	LST_BASE[15:0]	Local status queue base address.
15–0	16	HST_BASE[15:0]	Host status queue base address.

3.5.3 0x78—Reassembly Free Buffer Queue Base Register (RSM_FBASE)

Access types: R/W B, L
The Reassembly Free Buffer Queue Base Register is located at address 0x78. This register determines the base address of both the host and local free buffer queue structures. The base address is a 16-bit number. Since both local and host free buffer queues reside in local memory (23 bits of byte addressing) the structures can start on 128-byte boundaries.

Bit	Field Size	Name	Description
31–16	16	LFR_BASE[15:0]	Local free buffer queue base address.
15–0	16	HFR_BASE[15:0]	Host free buffer queue base address.

3.5.4 0x7C—Reassembly Hash Table Base Register (RSM_HBASE)

Access types: R/W B, L
The Reassembly Hash Table Base Register is located at address 0x7C. This register consists of a 16-bit address pointer that points to the beginning of the hash table. Since the hash table resides in local memory, the table starts on 128-byte boundaries. It contains the base address of the hash table used by the reassembly coprocessor to process incoming cell streams associated with different virtual connections. The size of this hash table is given by the TAB_SIZE field in the RSM_CTRL register, and is always a power of 2.

Bit	Field Size	Name	Description
31–16	16	Reserved	Program and read as zero.
15–0	16	RSM_HBASE[15:0]	Hash table base address.

3.5.5 0x80—Reassembly Time-out Register (RSM_TO)

Access types: R/W R, L
The Reassembly Time-out Register is located at address 0x80. This register is written by the local or host processor to point to the beginning of the VCC table of a connection that has timed out. The LSTAT bit is set high if a status entry is to be sent to the local status queue after the reassembly VCC table is reset. Otherwise, a status entry is written to the host status queue. The reassembly coprocessor will NULL this register after it has processed the time-out. The processor can only write this register when the contents are NULL. Writing when the contents are non-zero will have no effect.

Bit	Field Size	Name	Description
31–25	7	Reserved	Program and read as zero.
24	1	TMO_AAL34	If this bit is set high, the VCC table is initialized as an AAL3/4 connection; otherwise, it is initialized as an AAL5 connection.
23	1	LSTAT	If this bit is set high, status entry is sent to local status queue; otherwise, status entry is sent to host status queue.
22–2	21	TOPNTR[20:0]	Address pointer to timed out connection.
1, 0	2	Reserved	Program and read as zero.

3.5.6 0x84—Reassembly Firewall Register (RSM_FW)

Access types: R/W R, L
The Reassembly Firewall Register is located at address 0x84. It consists of a 21-bit address pointer, FWPNTR, and a credit update field, FWCREDIT. FWPNTR points to the base address of the VCC table that is to be incremented. The CBUFF_CNT field is incremented by the value in the FWCREDIT field. The reassembly coprocessor will NULL this register after it has processed the time-out. The processor should only write this register when the contents are NULL.

Bit	Field Size	Name	Description
31–23	9	FWCREDIT[8:0]	Credit update field.
22–2	21	FWPNTR[20:0]	Address pointer.
1, 0	2	Reserved	Program and read as zero.

3.5.7 0x88—Reassembly/Segmentation Queue Register (RS_QBASE)

Access types: R/W B, L
The Reassembly/Segmentation Queue Register is located at address 0x88. This register contains the 128-byte aligned base address of the reassembly/segmentation queue structure.

Bit	Field Size	Name	Description
31–16	16	Reserved	Program and read as zero.
15–0	32	RS_QBASE[15:0]	Base address of the reassembly/segmentation queue.

3.6 Host Interface Registers

3.6.8 0xC0—Host Interrupt Status Register (HOST_ISTAT0)

Access types: R/O
The Host Interrupt Status Register is located at address 0xC0. This register contains all interruptible status for the host processor. The corresponding interrupt enables are located in the HOST_IMASK0 register. Status types are defined as:

L — Level-sensitive status—A logic one on the status bit will cause an interrupt when enabled by the corresponding IMASK bit. Reading the status does not clear the status or interrupt. The source of the condition causing the status must be cleared before the status or interrupt is cleared.

E — Event driven status—A 0→1 transition on the status bit causes an interrupt when enabled. Reading the status register clears the status bit and the interrupt.

DE — Dual Event status—A 0→1 and 1→0 transition on the status bit can be enabled to cause an interrupt. Reading the status register clears the status bit and the interrupt.

NOTE: Only host reads will reset the status bits in the HOST_ISTAT0 register.

Bit	Field Size	Type	Name	Description
31	1	L	PFAIL	Reflects inverted state of processor PFAIL* input.
30	1	L	PHY_INTR	In stand-alone operation, this bit reflects the inverted state of the PDAEN* input. PHY_INTR may be connected to a PHY interrupt source.
29	1	DE	LP_LOCK_EVENT	This bit is set on a 0→1 or 1→0 transition of the LP_LOCK control bit. Read the CONFIG0 register for the current state of LP_LOCK.
28	1	E	HOST_MBOX_ WRITTEN	This bit is set upon a write to the HOST_MBOX register by the local processor, and cleared by a read of the HOST_MBOX register.
27	1	E	LP_MBOX_READ	This bit is set upon the read of the LP_MBOX register by the local processor.
26	1	L	PCI_BUS_EROR	This bit is set upon the occurrence of a PCI master bus error. Read the SYS_STAT register for more information of the type of error. Refer to Section 2.8 to reset this interrupt.
25	1	E	DMA_FULL	Set when the incoming DMA burst FIFO becomes full.
24	1	E	FR_PAR_ERR	Set on the occurrence of a parity error on the reassembly ATM physical interface.
23	1	E	FR_SYNC_ERR	Set on the occurrence of a synchronization error on the reassembly ATM physical interface.
22, 21	2	–	Reserved	Read as zero.
20	1	E	RS_QUEUE_FULL	Reassembly/segmentation queue full condition
19	1	L	RSM_RUN	Set when the reassembly machine is running. Will be high when RSM_ENABLE bit in RSM_CTRL is high or when processing the last cell after RSM_ENABLE is set low.
18	1	E	RSM_OVFL	Reassembly overflow. Indicates that a cell was lost due to a FIFO full condition.
17	1	L	RSM_HS_FULL	Set on the occurrence of a host status queue full condition.
16	1	L	RSM_LS_FULL	Set on the occurrence of a local status queue full condition.
15	1	L	RSM_HF_EMPT	Set on the occurrence of a host free buffer queue empty condition.
14	1	L	RSM_LF_EMPT	Set on the occurrence of a local free buffer queue empty condition.
13	1	E	RSM_HS_WRITE	Indicates reassembly host status has been written by the Bt8230.
12	1	E	RSM_LS_WRITE	Indicates reassembly local status has been written by the Bt8230.
11–9	3	-	Reserved	Read as zero.
8	1	L	SEG_RUN	Set when the segmentation machine is running. Will be high when SEG_ENABLE bit in SEG_CTRL is high or when processing the last cell after SEG_ENABLE is low.
7	1	E	SEG_UNFL	Segmentation underflow indicates that an idle cell was sent instead of a scheduled cell due to lack of PCI bandwidth.
6	1	E	HRING_OVFL	Host transmit ring overflow.

Bit	Field Size	Type	Name	Description
5	1	E	LRING_OVFL	Local transmit ring overflow.
4	1	L	BD_FULL	Buffer descriptor space full.
3	1	L	SEG_HS_FULL	Indicates that the segmentation host status queue is full.
2	1	L	SEG_LS_FULL	Indicates that the segmentation local status queue is full.
1	1	E	SEG_HS_WRITE	Indicates that the segmentation host status queue has been written by the Bt8230.
0	1	E	SEG_LS_WRITE	Indicates that the segmentation local status queue has been written by the Bt8230.

3.6.9 0xCC—Host Interrupt Mask Register (HOST_IMASK0)

Access types: R/W H, NL

The Host Interrupt Mask Register is located at address 0xCC. This register contains the interrupt enables that correspond to the status in the HOST_ISTAT0 register. The assertion of the HRST* system reset pin will clear all of the HOST_IMASK0 interrupt enables.

Bit	Field Size	Name	Description
31	1	EN_PFAIL	Enables interrupt when PFAIL status is a logic one.
30	1	EN_PHY_INTR	Enables interrupt when PHY_INTR status is a logic one.
29	1	EN_LP_LOCK_EVENT	Enables interrupt when LP_LOCK_EVENT status is a logic one.
28	1	EN_HOST_MBOX _WRITTEN	Enables interrupt when HOST_MBOX WRITTEN status is a logic one.
27	1	EN_LP_MBOX_READ	Enables interrupt when LP_MBOX_READ status is a logic one.
26	1	EN_PCI_BUS_ERROR	Enables interrupt when PCI_BUS_ERROR status is a logic one.
25	1	EN_DMA_FULL	Enabled interrupt when DMA_FULL status is a logic one.
24	1	EN_FR_PAR_ERR	Enables interrupt when FR_PAR_ERR status is a logic one.
23	1	EN_FR_SYNC_ERR	Enables interrupt when FR_SYNC_ERR status is a logic one.
22, 21	2	Reserved	Reserved, set to zero.
20	1	EN_RSQUEUE_FULL	Enables interrupt when RSQUEUE_FULL status is a logic one.
19	1	EN_RSM_RUN	Enables interrupt when RSM_RUN status is a logic one.
18	1	EN_RSM_OVFL	Enables interrupt when RSM_OVFL status is a logic one.
17	1	EN_RSM_HS_FULL	Enables interrupt when RSM_HS_FULL status is a
16	1	EN_RSM_LS_FULL	Enables interrupt when RSM_LS_FULL status is a logic high.
15	1	EN_RSM_HF_EMPT	Enables interrupt when RSM_HF_EMPT status is a logic high.
14	1	EN_RSM_LF_EMPT	Enables interrupt when RSM_LF_EMPT status is a logic high.

Bit	Field Size	Name	Description
13	1	EN_RSM_HS_WRITE	Enables interrupt when RSM_HS_WRITE status is a logic high.
12	1	EN_RSM_LS_WRITE	Enables interrupt when RSM_LS_WRITE status is a logic high.
11–9	3	Reserved	Set to zero.
8	1	EN_SEG_RUN	Enables interrupt when SEG_RUN status is a logic high.
7	1	EN_SEG_UNFL	Enables interrupt when SEG_UNFL status is a logic high.
6	1	EN_HRING_OVFL	Enables interrupt when HRING_OVFL status is a logic high.
5	1	EN_LRING_OVFL	Enables interrupt when LRING_OVFL status is a logic high.
4	1	EN_BD_FULL	Enables interrupt when the BD_FULL status is a logic high.
3	1	EN_SEG_HS_FULL	Enables interrupt when SEG_HS_FULL status is a logic high.
2	1	EN_SEG_LS_FULL	Enables interrupt when SEG_LS_FULL status is a logic high.
1	1	EN_SEG_HS_WRITE	Enables interrupt when SEG_HS_WRITE status is a logic high.
0	1	EN_SEG_LS_WRITE	Enables interrupt when SEG_LS_WRITE status is a logic high.

3.6.10 0xD8—Host Mailbox Register (HOST_MBOX)

Access types: R/W L, NL

The Host Mailbox Register is located at address 0xD8. This register implements a mailbox for communication between the host and local processors. The register is written by the local processor and read by the host to pass messages in that direction. Writes to this register may interrupt the host while reads may interrupt the local processor.

Bit	Field Size	Name	Description
31–0	32	HOST_MBOX[31:0]	Messages flow from local processor to host.

3.7 Local Processor Interface Registers

3.7.11 0xE0—Local Processor Interrupt Status Register (LP_ISTAT0)

Access types: R/O
The Local Processor Interrupt Status Register is located at address 0xE0 and contains all the interruptible status for the local processor. The corresponding interrupt enables are located in the LP_IMASK0 register. Status types are defined as:

L Level sensitive status—A logic one on the status bit will cause an interrupt when enabled by the corresponding IMASK bit. Reading the status does not clear the status or interrupt. The source of the condition causing the status must be cleared before the status or interrupt is cleared.

E Event driven status—A 0→1 transition on the status bit causes an interrupt when enabled. Reading the status register clears the status bit and the interrupt.

DE Dual event status—A 0→1 and 1→0 transition on the status bit can be enabled to cause an interrupt. Reading the status register clears the status bit and the interrupt.

NOTE: Only local processor reads will reset the status bits in the LP_ISTAT0 register.

Bit	Field Size	Type	Name	Description
31	1	E	RTC_OVFL	Clock register overflow.
30	1	E	ALARM1	Set when ALARM1 register matches CLOCK register.
29	1	L	HOST_LOCK_STAT	Set on a 0→1 or 1→0 transition of the HOST_LOCK control bit. Read the CONFIG0 register for the current state of HOST_LOCK.
28	1	E	LP_MBOX_WRITTEN	Set upon a write to the LP_MBOX register by the host processor.
27	1	E	HOST_MBOX_READ	Set upon the read of the HOST_MBOX register by the host processor.
26	1	L	PCI_BUS_ERROR	Set upon the occurrence of a PCI master bus error. Read the SYS_STAT register for more information of the type of error. Refer to Section 2.8 to reset this interrupt.
25	1	E	DMA_FULL	Set when the incoming DMA burst FIFO becomes full.
24	1	E	FR_PAR_ERR	Set on the occurrence of a parity error on the reassembly ATM physical interface.
23	1	E	FR_SYNC_ERR	Set on the occurrence of a synchronization error on the reassembly ATM physical interface.
22, 21	2	-	Reserved	Reserved, read as zero.

Bit	Field Size	Type	Name	Description
20	1	E	RS_QUEUE_FULL	Reassembly/segmentation queue full condition
19	1	L	RSM_RUN	Set when the reassembly machine is running. Will be high when RSM_ENABLE bit in RSM_CTRL is high or when processing the last cell after RSM_ENABLE is set low.
18	1	E	RSM_OVFL	Reassembly overflow. A cell was lost due to a FIFO full condition.
17	1	L	RSM_HS_FULL	Set on the occurrence of a host status queue full condition.
16	1	L	RSM_LS_FULL	Set on the occurrence of a local status queue full condition.
15	1	L	RSM_HF_EMPT	Set on the occurrence of a host free buffer queue empty condition.
14	1	L	RSM_LF_EMPT	Set on the occurrence of a local free buffer queue empty condition.
13	1	E	RSM_HS_WRITE	Indicates reassembly host status has been written by the Bt8230.
12	1	E	RSM_LS_WRITE	Indicates reassembly local status has been written by the Bt8230.
11–9	3	–	Reserved	Read as zero.
8	1	L	SEG_RUN	Set when the segmentation machine is running. Will be high when SEG_ENABLE bit in SEG_CTRL is high or when processing the last cell after SEG_ENABLE is set low.
7	1	E	SEG_UNFL	Segmentation underflow. Indicates that an idle cell was sent instead of a scheduled cell due to lack of PCI bandwidth.
6	1	E	HRING_OVFL	Host transmit ring overflow.
5	1	E	LRING_OVFL	Local transmit ring overflow.
4	1	L	BD_FULL	Buffer descriptor space full.
3	1	L	SEG_HS_FULL	Indicates that the segmentation host status queue is full.
2	1	L	SEG_LS_FULL	Indicates that the segmentation local status queue is full.
1	1	E	SEG_HS_WRITE	Indicates that the segmentation host status queue has been written by the Bt8230.
0	1	E	SEG_LS_WRITE	Indicates that the segmentation local status queue has been written by the Bt8230.

3.7.12 0xEC—Local Processor Interrupt Mask Register (LP_IMASK0)

Access type: R/W L, NL

The Local Processor Interrupt Mask Register is located at address 0xEC. This register contains the interrupt enables that correspond to the status in the LP_ISTAT0 register. The assertion of the HRST* system reset pin clears all of the LP_IMASK0 interrupt enables.

Bit	Field Size	Name	Description
31	1	EN_RTC_OVFL	Enables an interrupt when RTC_OVFL status is a logic high.
30	1	EN_ALARM1	Enables an interrupt when ALARM1 status is a logic high.
29	1	EN_HOST_LOCK_STAT	Enables an interrupt when HOST_LOCK_STAT status is a logic high.
28	1	EN_LP_MBOX_ WRITTEN	Enables an interrupt when LP_MBOX_WRITTEN status is a logic high.
27	1	EN_HOST_MBOX _READ	Enables an interrupt when HOST_MBOX_READ status is a logic high.
26	1	EN_PCI_BUS_ERROR	Enables an interrupt when PCI_BUS_ERROR status is a logic high.
25	1	EN_DMA_FULL	Enables an interrupt when DMA_FULL status is a logic high.
24	1	EN_FR_PAR_ERR	Enables an interrupt when FR_PAR_ERR status is a logic high.
23	1	EN_FR_SYNC_ERR	Enables an interrupt when FR_SYNC_ERR status is a logic high.
22, 21	2	Reserved	Set to zero.
20	1	EN_RSQUEUE_FULL	Enables an interrupt when RSQUEUE_FULL status is a logic high.
19	1	EN_RSM_RUN	Enables an interrupt when RSM_RUN status is a logic high.
18	1	EN_RSM_OVFL	Enables an interrupt when RSM_OVFL status is a logic high.
17	1	EN_RSM_HS_FULL	Enables an interrupt when RSM_HS_FULL status is a logic high.
16	1	EN_RSM_LS_FULL	Enables an interrupt when RSM_LS_FULL status is a logic high.
15	1	EN_RSM_HF_EMPT	Enables an interrupt when RSM_HF_EMPT status is a logic high.
14	1	EN_RSM_LF_EMPT	Enables an interrupt when RSM_LF_EMPT status is a logic high.
13	1	EN_RSM_HS_WRITE	Enables an interrupt when RSM_HS_WRITE status is a logic high.
12	1	EN_RSM_LS_WRITE	Enables an interrupt when RSM_LS_WRITE status is a logic high.
11–9	3	Reserved	Set to zero.
8	1	EN_SEG_RUN	Enables an interrupt when SEG_RUN status is a logic high.
7	1	EN_SEG_UNFL	Enables an interrupt when SEG_UNFL status is a logic high.
6	1	EN_HRING_OVFL	Enables an interrupt when HRING_OVFL status is a logic high.
5	1	EN_LRING_OVFL	Enables an interrupt when LRING_OVFL status is a logic high.
4	1	EN_BD_FULL	Enables an interrupt when the BD_FULL status is a logic high.
3	1	EN_SEG_HS_FULL	Enables an interrupt when SEG_HS_FULL status is a logic high.

Bit	Field Size	Name	Description
2	1	EN_SEG_LS_FULL	Enables an interrupt when SEG_LS_FULL status is a logic high.
1	1	EN_SEG_HS_WRITE	Enables an interrupt when SEG_HS_WRITE is a logic high.
0	1	EN_SEG_LS_WRITE	Enables an interrupt when SEG_LS_WRITE is a logic high.

3.7.13 0xF8—Local Processor Mailbox Register (LP_MBOX)

Access types: R/W H, NL

The Local Processor Mailbox Register is located at address 0xF8. This register implements a mailbox for communication between the host and local processors. LP_MBOX is written by the host processor and read by the local processor to pass messages in that direction. Writes to this register may interrupt the local processor while reads may interrupt the host processor.

Bit	Field Size	Name	Description
31–0	32	LP_MBOX[31:0]	Local processor mailbox register. Messages flow from host processor to local processor.

3.8 PCI Bus Interface Configuration Registers

In accordance with the PCI bus specification, the SRC PCI bus interface implements a 128-byte configuration register space. These configuration registers can be used by the host processor to initialize, control, and monitor the SRC bus interface logic. The complete definitions of these registers and the relevant fields within them is given in the PCI bus specification, and will not be repeated here. The implementation of these registers in the Bt8230 is shown in Table 3-6. Descriptions of these fields are given in Table 3-7.

Table 3-6. PCI Configuration Register Definition

Configuration Register Definition			Byte Address
DEVICE_ID (16 bits)	VENDOR_ID (16 bits)		0x00
STATUS (16 bits)	COMMAND (16 bits)		0x04
0x020300 (24 bits)		REV_ID (8 bits)	0x08
0x0000 (16 bits)	LAT_TIMER (8 bits)	0x00 (8 bits)	0x0c
EXT_MEM_BASE (8 bits)	0x000000 (24 bits)		0x10
0x00000000 (32 bits)			0x14–0x38
0x050201(24 bits)		ILINE (8 bits)	0x3C
0x00000 (20 bits)	SPECIAL_STATUS (4 bits)	MAX_BUR_LEN (8 bits)	0x40
CUR_MSTR_RD_ADDR (32 bits)			0x44
CUR_MSTR_WR_ADDR (32 bits)			0x48
0x00000000 (32 bits)			0x4C–0x7C

Table 3-7. PCI Configuration Registers Field Descriptions (1 of 3)

Field Name	Description/Function
DEVICE_ID	16-bit device identifier. Serves to uniquely identify the SRC to the host operating system. Set to 0x8230
VENDOR_ID	16-bit vendor identifier code, allocated on a global basis by the PCI SIG. Set to 0x109E.
STATUS	PCI bus interface status register. The PCI host may monitor its operation using the STATUS field. This field is further divided into subfields as shown below.

31	30	29	28	27–25	24	23–16
DPE	SSE	RMA	RTA	0x0	DPR	0x80

DPE	Detected Parity Error
SSE	Signaled System Error
RMA	Received Master Abort
RTA	Received Target Abort
DPR	Reported Data Parity Error

Table 3-7. PCI Configuration Registers Field Descriptions (2 of 3)

<table>
<tr><th>Field Name</th><th>Description/Function</th></tr>
<tr><td>PMINDEX</td><td>Pointer to an PM OAM word. Index with reference to top of hash table. Minimum value is 4; maximum value is 127.</td></tr>
<tr><td>COMMAND</td><td>PCI bus interface control/command register. The PCI host may configure the SRC bus interface logic using the COMMAND field. This field is further divided into subfields as shown below.
<table>
<tr><td>15–10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5–3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0x00</td><td>FB_EN</td><td>SE_EN</td><td>0x0</td><td>PE_EN</td><td>0x0</td><td>M_EN</td><td>MS_EN</td><td>0x0</td></tr>
</table>
FB_EN Fast back-to-back enable across target
SE_EN SERR* pin output enable
PE_EN Parity report enable
M_EN Master enable
MS_EN Memory space enable</td></tr>
<tr><td>REV_ID</td><td>Revision ID code for the Bt8230 chip.</td></tr>
<tr><td>LAT_TIMER</td><td>Five most significant bits of latency timer used for PCI master accesses by the DMA coprocessor</td></tr>
<tr><td>EXT_MEM_BASE</td><td>Base address of external memory being mapped into PCI address space.</td></tr>
<tr><td>ILINE</td><td>Interrupt line identifier.</td></tr>
<tr><td>SPECIAL_STATUS</td><td>Device status not defined by the PCI specification. The field is further subdivided into subfields as shown below. Detailed descriptions of the above subfields can be found in the PCI bus specification. Note that the configuration registers are accessed starting from byte address 0 in the configuration space allotted to an adapter card containing the SRC chip. Access to the configuration registers is available only to the PCI host CPU, and is independent of all other SRC logic.
<table>
<tr><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>INTF_DIS</td><td>INT_FAIL</td><td>MERROR</td><td>MRD</td></tr>
</table>
INTF_DIS If the M_EN bit in the COMMAND field is a logic low, any attempt by the Bt8230 to perform a DMA transaction to the PCI bus will result in an error. INTF_DIS and MERROR bits will both be set at a logic high. This bit may be reset by writing a logic high to itself.
INT_FAIL Set to a logic one when the an internal PCI/DMA synchronization error has occurred. The MERROR bit will also be set to a logic high. This bit may be reset by writing a logic high to itself.
MERROR Indicates that the PCI bus master has encountered a fatal error and therefor has halted operation. Set when either RTA, RMA, DPE, INTF_DIS, or INT_FAIL errors occur. This bit may be reset by writing a logic high to itself.
MRD If logic high, indicates that the errored transaction was a read and the address of the read is located in the CUR_MSTR_RD_ADDR field. If logic low, indicates a write with the corresponding address located in the CUR_MSTR_WR_ADDR field (read-only).</td></tr>
</table>

Table 3-7. PCI Configuration Registers Field Descriptions (3 of 3)

Field Name	Description/Function
MAX_BUR_LEN	Sets the maximum number of words that can be transferred in a single transaction by the SRC PCI bus master. A value of zero is invalid and can lead to erroneous and unpredictable results. When HRST* input pin is active, this register is loaded with 16 decimal.
CUR_MSTR_WR _ADDR	Current write target address used by PCI bus master (read only).
CUR_MSTR_RD _ADDR	Current read target address used by PCI bus master (read only).

4.0 Electrical and Mechanical Specifications

4.1 Timing

4.1.1 PCI Bus Interface Timing

All PCI bus interface signals are synchronous to the PCI bus clock, HCLK, except for HRST* and HINT*. Table 4-1 provides the PCI bus interface timing parameters. Figure 4-1 and Figure 4-2 illustrate this timing.

Table 4-1. PCI Bus Interface Timing Parameters *(1 of 2)*

Symbol	Parameter	Min	Max	Units
t_{cyc}	HCLK Cycle Time(1)	30	–	ns
t_{high}	HCLK High Time(1)	11	19	ns
t_{low}	HCLK Low Time(1)	11	19	ns
t_{su}	HAD Input Setup Time to HCLK(1)	7	–	ns
	HC/BE Input Setup Time to HCLK(1)	7	–	ns
	HPAR Input Setup Time to HCLK(1)	7	–	ns
	HFRAME* Input Setup Time to HCLK(1)	7	–	ns
	HIRDY* Input Setup Time to HCLK(1)	7	–	ns
	HTRDY* Input Setup Time to HCLK(1)	7	–	ns
	HSTOP* Input Setup Time to HCLK(1)	7	–	ns
	HDEVSEL* Input Setup Time to HCLK(1)	7	–	ns
	HGNT* Input Setup Time to HCLK(1)	10	–	ns
	HPERR* Input Setup Time to HCLK(1)	7	–	ns
t_h	Input Hold Time from HCLK–All Inputs(1)	0	–	ns

Table 4-1. PCI Bus Interface Timing Parameters (2 of 2)

Symbol	Parameter	Min	Max	Units
t_{val}	HCLK to HAD Valid Delay(2)	2	11	ns
	HCLK to HC/BE Valid Delay(2)	2	11	ns
	HCLK to HPAR Valid Delay(2)	2	11	ns
	HCLK to HFRAME* Valid Delay(2)	2	11	ns
	HCLK to HIRDY* Valid Delay(2)	2	11	ns
	HCLK to HSTOP* Valid Delay(2)	2	11	ns
	HCLK to HDEVSEL Valid Delay(2)	2	11	ns
	HCLK to HPERR* Valid Delay(2)	2	11	ns
	HCLK to HREQ Valid Delay(2)	2	12	ns
	HCLK to HSERR* Valid Delay(2)	2	11	ns
t_{on}	Float to Active Delay—All Three-state Outputs(2)	2	–	ns
t_{off}	Active to Float Delay—All Three-state Outputs(2)	–	28	ns
$t_{rst\text{-}off}$	Reset Active to Output Float Delay	–	40	ns

Notes: (1). See Figure 4-1 for waveforms and definitions.
(2). See Figure 4-2 for waveforms and definitions. The maximum output delays are measured with a 50 pF load and the minimum delays are measured with a 0 pF load.

Figure 4-1. PCI Bus Input Timing Measurement Conditions

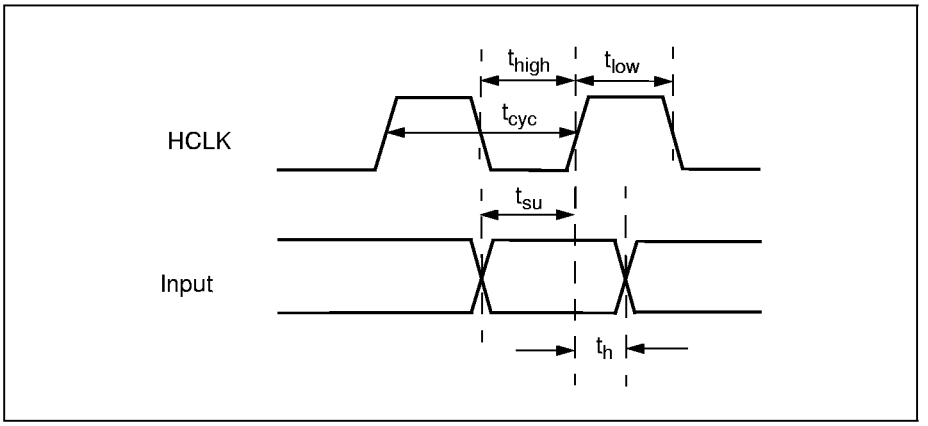
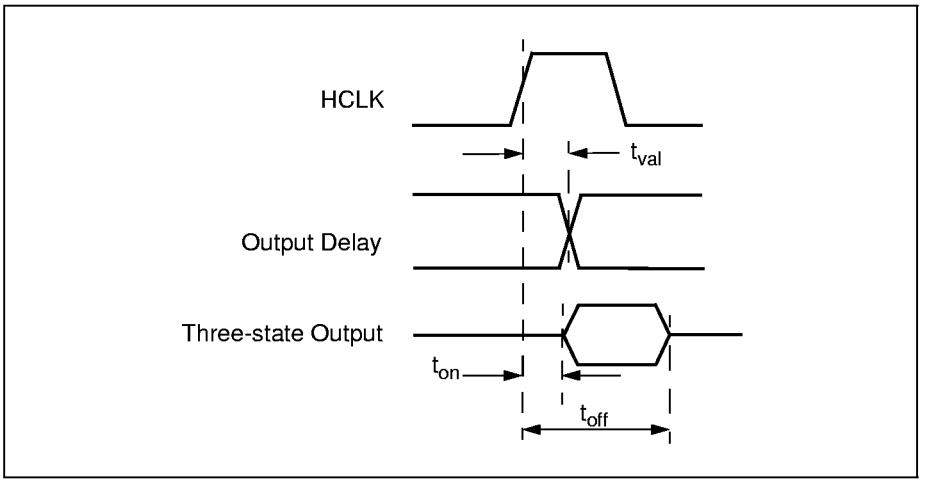


Figure 4-2. PCI Bus Output Timing Measurement Conditions



4.1.2 ATM Physical Interface Timing—UTOPIA and Slave UTOPIA

All ATM physical interface signals are synchronous to the interface clock, FRCTRL, except for TXFLAG* and RXFLAG* in the slave UTOPIA mode. Timing parameters for the UTOPIA interface are provided in Table 4-2. Table 4-3 provides the timing parameters for the slave UTOPIA interface. Timing diagrams for both interfaces are provided in Figure 4-3 and Figure 4-4.

Table 4-2. UTOPIA Interface Timing Parameters (1 of 2)

Symbol	Parameter	Min	Max	Units
t_{cyc}	FRCTRL Cycle Time(1)	40	–	ns
t_{high}	FRCTRL High Time(1)	16	24	ns
t_{low}	FRCTRL Low Time(1)	16	24	ns
t_{su}	RXD Input Setup Time to FRCTRL(1)	10	–	ns
	RXPAR Input Setup Time to FRCTRL(1)	10	–	ns
	RXMARK Input Setup Time to FRCTRL(1)	10	–	ns
	RXFLAG* Input Setup Time to FRCTRL(1)	10	–	ns
	TXFLAG* Input Setup Time to FRCTRL(1)	10	–	ns
t_h	RXD Input Hold Time from FRCTRL(1)	1	–	ns
	RXPAR Input Hold Time from FRCTRL(1)	1	–	ns
	RXMARK Input Hold Time from FRCTRL(1)	1	–	ns
	RXFLAG* Input Hold Time from FRCTRL(1)	1	–	ns
	TXFLAG* Input Setup Time from FRCTRL(1)	1	–	ns

Table 4-2. UTOPIA Interface Timing Parameters (2 of 2)

Symbol	Parameter	Min	Max	Units
t_{val}	FRCTRL to TXD Valid Delay(2)	2	18	ns
	FRCTRL to TXPAR Valid Delay(2)	2	18	ns
	FRCTRL to TXMARK Valid Delay(2)	2	18	ns
	FRCTRL to TXEN* Valid Delay(2)	2	20	ns
	FRCTRL to RXEN* Valid Delay(2, 3)	2	20	ns

Notes: (1). See Figure 4-3 for waveforms and definitions.
(2). See Figure 4-4 for waveforms and definitions. The output delays are measured with a 25 pF load.
(3). In octet handshake mode, the maximum propagations delay is either 20 nsec or setup time of RXFLAG* + 10 nsec, whichever is greatest.

Table 4-3. Slave UTOPIA Interface Timing Parameters

Symbol	Parameter	Min	Max	Units
t_{cyc}	FRCTRL Cycle Time(1)	30	–	ns
t_{high}	FRCTRL High Time(1)	12	18	ns
t_{low}	FRCTRL Low Time(1)	12	18	ns
t_{su}	RXD Input Setup Time to FRCTRL(1)	6	–	ns
	RXPAR Input Setup Time to FRCTRL(1)	3	–	ns
	RXMARK Input Setup Time to FRCTRL(1)	8	–	ns
	RXEN* Input Setup Time to FRCTRL(1)	10	–	ns
	TXEN* Input Setup Time to FRCTRL(1)	3	–	ns
t_h	RXD Input Hold Time from FRCTRL(1)	1	–	ns
	RXPAR Input Hold Time from FRCTRL(1)	1	–	ns
	RXMARK Input Hold Time from FRCTRL(1)	1	–	ns
	RXEN* Input Hold Time from FRCTRL(1)	1	–	ns
	TXEN* Input Hold Time from FRCTRL(1)	1	–	ns
t_{val}	FRCTRL to TXD Valid Delay(2)	2	16	ns
	FRCTRL to TXPAR Valid Delay(2)	2	16	ns
	FRCTRL to TXFLAG* Valid Delay(2)	2	20	ns
	FRCTRL to RXFLAG* Valid Delay(2)	2	20	ns
	FRCTRL to TXMARK Valid Delay(2)	2	17	ns

Notes: (1). See Figure 4-3 for waveforms and definitions.
(2). See Figure 4-4 for waveforms and definitions. The output delays are measured with a 25 pF load.

Figure 4-3. UTOPIA and Slave UTOPIA Input Timing Measurement Conditions

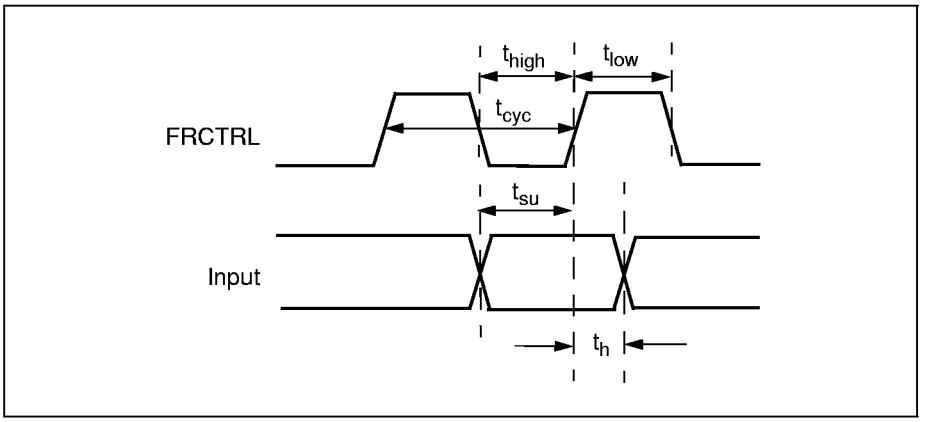
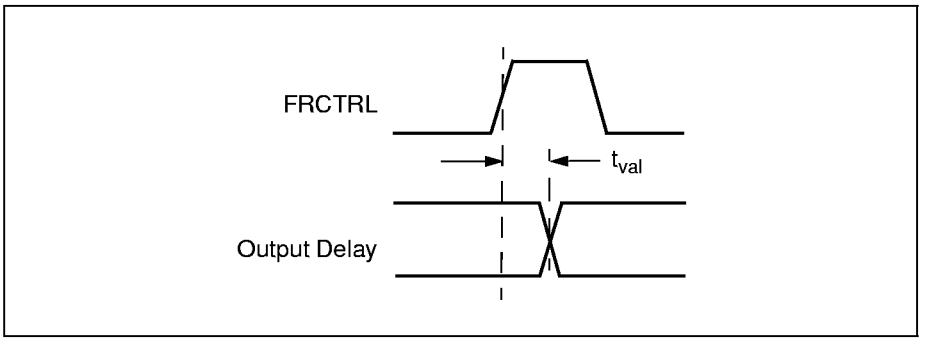


Figure 4-4. UTOPIA and Slave UTOPIA Output Timing Measurement Conditions



4.1.3 ATM Physical Interface Timing—Bt8222 Mode

All ATM physical interface signals are synchronous to the receive strobe, RXEN*, and the transmit strobe, TXEN*. Table 4-4, Figure 4-5, and Figure 4-6 provide the timing parameters for the Bt8222 interface.

Table 4-4. Bt8222 Interface Timing Parameters

Symbol	Parameter	Min	Max	Units
t_{cyc}	RXEN*/TXEN* Cycle Time(1)	50	–	ns
t_{high}	RXEN*/TXEN* High Time(1)	21	29	ns
t_{low}	RXEN*/TXEN* Low Time(1)	21	29	ns
t_{su}	RXD Input Setup Time to RXEN*(1)	13	–	ns
	RXPAR Input Setup Time to RXEN*(1)	10	–	ns
	RXMARK Input Setup Time to RXEN*(1)	8	–	ns
	FRCTRL Input Setup Time to RXEN*(1)	7	–	ns
	TXMARK Input Setup Time to TXEN*(1)	3	–	ns
t_h	RXD Input Hold Time from RXEN*(1)	1	–	ns
	RXPAR Input Hold Time from RXEN*(1)	1	–	ns
	RXMARK Input Hold Time from RXEN*(1)	1	–	ns
	FRCTRL Input Hold Time from RXEN*(1)	1	–	ns
	TXMARK Input Hold Time from TXEN*(1)	1	–	ns
t_{val}	TXEN* to TXD Valid Delay(2)	6	36	ns
	TXEN* to TXPAR Valid Delay(2)	6	39	ns
	TXEN* to TXFLAG* Valid Delay(2, 3)	4	21	ns
	RXEN* to RXFLAG* Valid Delay(2)	4	23	ns
t_{on}	Float to Active Delay—All 3s Outputs (TXD/TXPAR)(2)	4	16	ns
t_{off}	Active to Float Delay—All 3s Outputs (TXD/TXPAR)(2)	2	10	ns

Notes: (1). See Figure 4-5 for waveforms and definitions.
(2). See Figure 4-6 for waveforms and definitions. The output delays are measured with a 50 pF load.
(3). TXFLAG* going inactive is asynchronous to TXEN*.

Figure 4-5. Bt8222 Input Timing Measurement Conditions

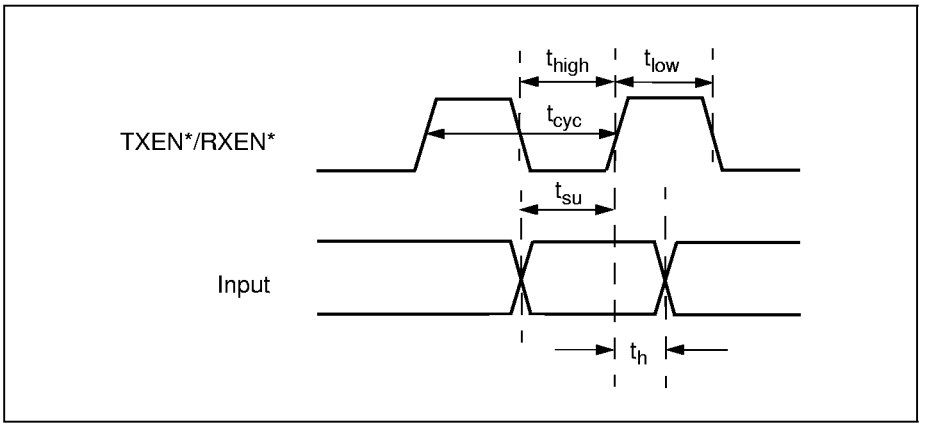
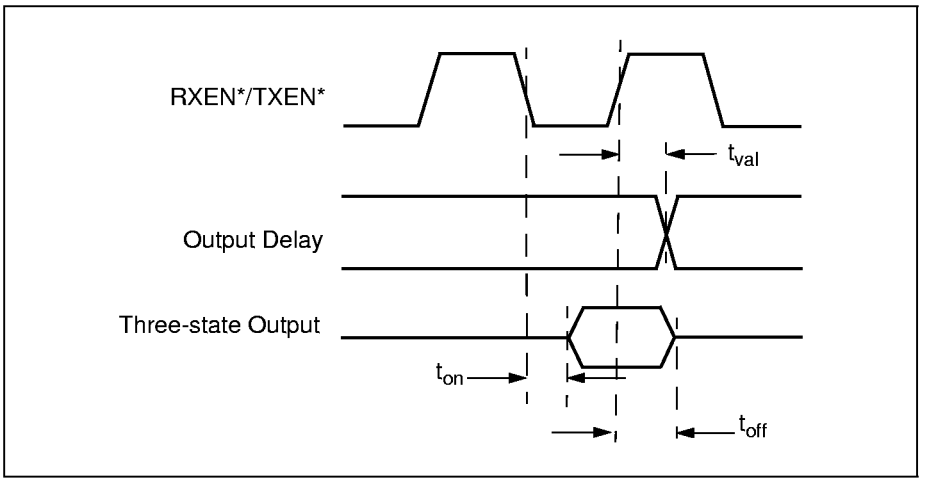


Figure 4-6. Bt8222 Output Timing Measurement Conditions



4.1.4 System Clock Timing

The system clock timing consists of the CLK2X input and the SYSCLK and CLKD3 outputs. The CLK2X input and SYSCLK outputs are used to generate the internal and external system clocks as well as local memory and processor read and write timing. The CLKD3 output may be used as the ATM Physical Interface clock. There is no defined skew relationship between the CLK2X input and SYSCLK and CLKD3 outputs. Table 4-5 specifies the timing parameters of the three clocks. Figure 4-7 and Figure 4-8 illustrates this timing.

Table 4-5. System Clock Timing

Symbol	Parameter	Min	Max	Units
	Input Clocks(1)			
t_{cf}	CLK2X Frequency	0	66	MHZ
t_c	CLK2X Period	15.15		ns
t_{cd}	CLK2X Duty Cycle	40	60	%
t_{cr}	CLK2X Rise Time	0	6	ns
t_{cf}	CLK2X Fall Time	0	6	ns
	Output Clocks(2)			
t_{sf}	SYSCLK Frequency	$t_{CF}/2$		MHz
t_s	SYSCLK Period	$2t_C$		ns
t_{sh}	SYSCLK High Time	$(t_S/2) - 2$	$(t_S/2) + 2$	ns
t_{sl}	SYSCLK Low Time	$(t_S/2) - 2$	$(t_S/2) + 2$	ns
t_{sr}	SYSCLK Rise Time	1	4	ns
t_{sf}	SYSCLK Fall Time	1	4	ns
t_{df}	CLKD3 Frequency	$t_{CF}/3$		MHz
t_d	CLKD3 Period	$3t_C$		
t_{dh}	CLKD3 High Time	$(t_D/2) - 2$	$(t_D/2) + 2$	ns
t_{dl}	CLKD3 Low Time	$(t_D/2) - 2$	$(t_D/2) + 2$	ns
t_{dr}	CLKD3 Rise Time	1	4	ns
t_{df}	CLKD3 Fall Time	1	4	ns

Notes: (1). See Figure 4-7 for waveforms and definitions.
(2). See Figure 4-8 for waveforms and definitions. The outputs are measured with a load of 35 pf.

Figure 4-7. Input System Clock Waveform

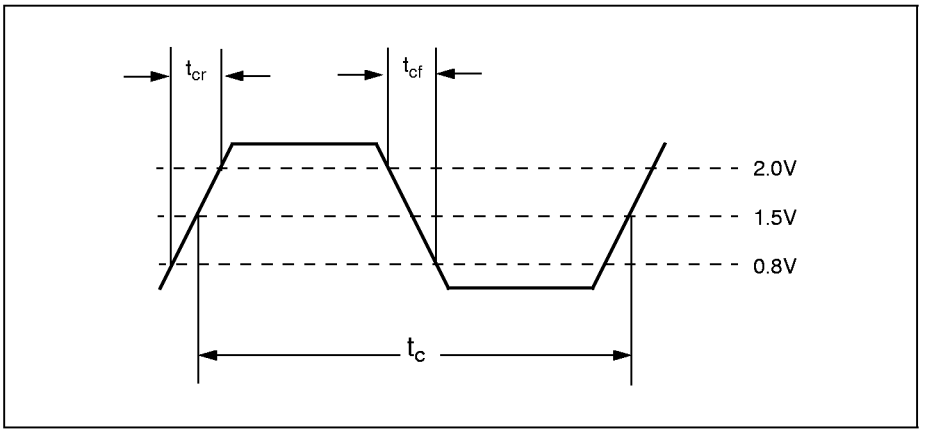
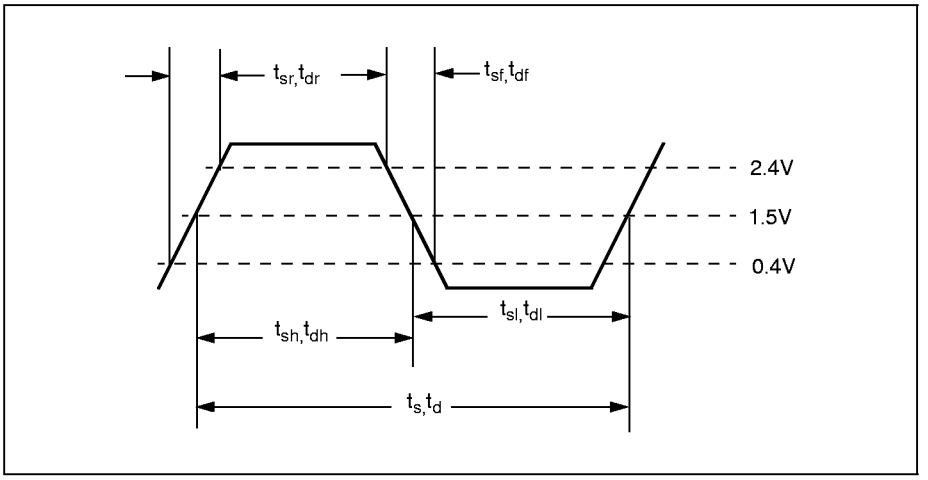


Figure 4-8. Output System Clock Waveform



4.1.5 Bt8230 Memory Interface Timing

Memory access times and other timing requirements are specified at three typical implementations of 1, 2, and 4 banks of by_8 SRAM. Table 4-6 gives the number of loads per bank for the different SRAM organizations while Table 4-7 gives the capacitive loading used in the timing specifications given for the three typical implementations. SRAM timing cross-reference is given in Table 4-8. Bt8230 Memory Interface Timing is given in Table 4-9. See Section 2.3 for details of memory bank organization.

Table 4-6. SRAM Organization Loading Dependencies

Signal	Loads/Bank(1)		
	by_16 SRAM	by_8 SRAM	by_4 SRAM
LADDR[18:0](1)	2	4	8
LDATA[31:0](1)	1	1	1
MWR*(1)	2	N/A	N/A
MOE*(1)	2	4	8
MCSx*(1, 2)	2	4	8
MWEx*(1)	1	1	2

Notes: (1). Typical input loading for SRAM is 7 pf. For exact values, consult the SRAM databook .
(2). Only connected to one bank by definition.

Table 4-7. Local Memory Output Loading Conditions

Signal	4 banks of by_8 SRAM	2 banks of by_8 SRAM	1 bank of by_8 SRAM	Units
Memory Interface Loading(1)				
LADDR[18:0](1)	150	100	50	pf
MOE*	150	100	50	pf
MWR*(2)	-	100	50	pf
LDATA[31:0]	50	35	25	pf
MWE[3:0]*	50	35	25	pf
MCS[3:0]*	50	50	50	pf

Notes: (1). In general, the LADDR loading is the most critical parameter, one bank of by_8 SRAM has the same address loading as 2 banks of by_16 and 1/2 bank of by_4 SRAM. For example, use the timing from the 2 banks of by_8 column for 4 banks of by_16 SRAM, or 1 bank of by_4 SRAM.
(2). For by_16 SRAM, the WE* input has the same loading as the address bus and OE*; therefore, 16 loads specified by the 4 banks of by_8 is not applicable, since the maximum number of by_16 SRAMs supported is 8.

Table 4-8. SRAM Timing Cross-Reference

SRC Timing Parameter	Standard SRAM Timing Parameter(1)	Description
Read Timing		
t_{rc}	t_{rc}	READ Cycle Time(3)
t_{adv}	t_{aa}	Address Access Time(2)
t_{csdv}	t_{ce}	Chip Enable Access Time(2)
–	t_{clz}	Chip Enable to Output in Low-Z(2)
–	t_{chz}	Chip Disable to Output in High-Z(4)
t_{oedv}	t_{aoe}	Output Enable Access Time(2)
–	t_{lzoe}	Output Enable to Output in Low-Z(5)
t_{doe}	t_{hzoe}	Output Disable to Output in High-Z(3)
t_{bers}	t_{abe}	Byte Enable Access Time (by_16 SRAM)(2)
Write Timing		
t_{wc}	t_{wc}	Write Cycle Time(3)
t_{w}	t_{wp}	Write Pulse Width(3)
t_{aws}	t_{aw}	Address Valid to End-of-Write(2)
–	t_{as}	Address Setup Time(5)
t_{awh}	t_{ah}	Address Hold from End-of-Write
t_{cws}	t_{cw}	Chip Enable to End-of-Write(2)
t_{dws}	t_{ds}	Data Setup Time(2)
t_{dwh}	t_{dh}	Data Hold Time
–	t_{lzwe}	Write Disable to Output in Low-Z(6)
–	t_{hzwe}	Write Enable to Output in High-Z(6)
t_{bews}	t_{bw}	Byte Enable to End-of-Write (by_16 SRAM)(2)

Notes: (1). Symbols may vary slightly from SRAM databook to databook.
(2). For proper operation, the standard SRAM timing parameter must be less than or equal to the Bt8230 timing parameter.
(3). For proper operation, the standard SRAM timing parameter must be greater than or equal to the Bt8230 timing parameter.
(4). These parameters are not critical since the Bt8230 relies on the MOE* signal to control the SRAM output enable.
(5). This parameter is not critical.
(6). The MWE*/MWR* signals are not used to control the output impedance in the Bt8230 memory.

Table 4-9. Bt8230 Memory Interface Timing

Symbol	Parameter	4 banks of by_8 SRAM		2 banks of by_8 SRAM		1 bank of by_8 SRAM		Units
		Min	Max	Min	Max	Min	Max	
Memory Read Timing								
t_{rc}	READ Cycle Time(2)	T		T		T		ns
t_{adv}	LADDR[18:0] Stable to Required Data Valid Time(1, 2, 3)	T-18		T-15		T-12		ns
t_{csdv}	MCS[3:0]* Low to Required Data Valid Time (1, 2, 3)	T-10		T-10		T-10		ns
t_{oedv}	MOE* Low to Required Data Valid Time(1, 2, 3)	T-23		T-20		T-17		ns
t_{dod}	LDATA Output Disable to MOE* Low(1)	8		8		8		ns
t_{doe}	MOE* High to LDATA Driven by SRC(1)	8		8		8		ns
t_{bers}	MWE[3:0]* Byte Enables Setup to Required Data Valid Time (RAMMODE = 1)(1)	T-10		T-8		T-6		ns
t_{berh}	MWE[3:0]* Byte Enables Hold from Required Data Valid Time (RAMMODE = 1)(1)	10		10		10		ns
Memory Write Timing								
t_{wc}	WRITE Cycle Time(2)	T		T		T		ns
t_{w}	MWR*, MWE[3:0]* Width(2, 4)	T-16	T-14	T-16	T-14	T-16	T-14	ns
t_{aws}	LADDR[18:0] Setup to MWR*, MWE[3:0]* Rising(2, 4)	T-11		T-9		T-7		ns
t_{awh}	LADDR[18:0] Hold from MWR*, MWE[3:0]* Rising(2, 4)	2		2		2		ns
t_{cws}	MCS[3:0]* Active to MWR*, MWE[3:0]* Rising(2, 4)	T-4		T-4		T-4		ns
t_{cwh}	MCS[3:0]* Hold from MWE[3:0]* Rising(4)	1		1		1		ns
t_{dws}	LDATA Setup to MWR*, MWE[3:0]* Rising Edge(1, 2, 4)	$1/2t_S$ –3		$1/2t_S$ –3		$1/2t_S$ –3		ns
t_{dwh}	LDATA Hold from MWR*, MWE[3:0]* Rising (1)	2		2		2		ns
t_{bews}	MWE[3:0]* Byte Enables Setup to MWR* Rising, (RAMMODE = 1)(1)	T-6		T-6		T-6		ns
t_{bewh}	MWE[3:0]* Byte Enables Hold from MWR* Rising, (RAMMODE = 1)(1)	4		4		4		ns

Notes: (1). See Figure 4-9 and Table 4-8 for waveforms and definitions.
(2). T = t_s for single cycle memory (no wait states), T = $2t_s$ for two cycle memory, (one wait state).
(3). Data Required Time is the time at which read data from the SRAM must be stable to meet setup requirements inside the Bt8230 to the internal version of SYSCLK. For two cycle timing, add t_s to t_{dws} time given.
(4). See Figure 4-10 and Table 4-8 for waveforms and definitions.

Figure 4-9. Bt8230 Memory Read Timing

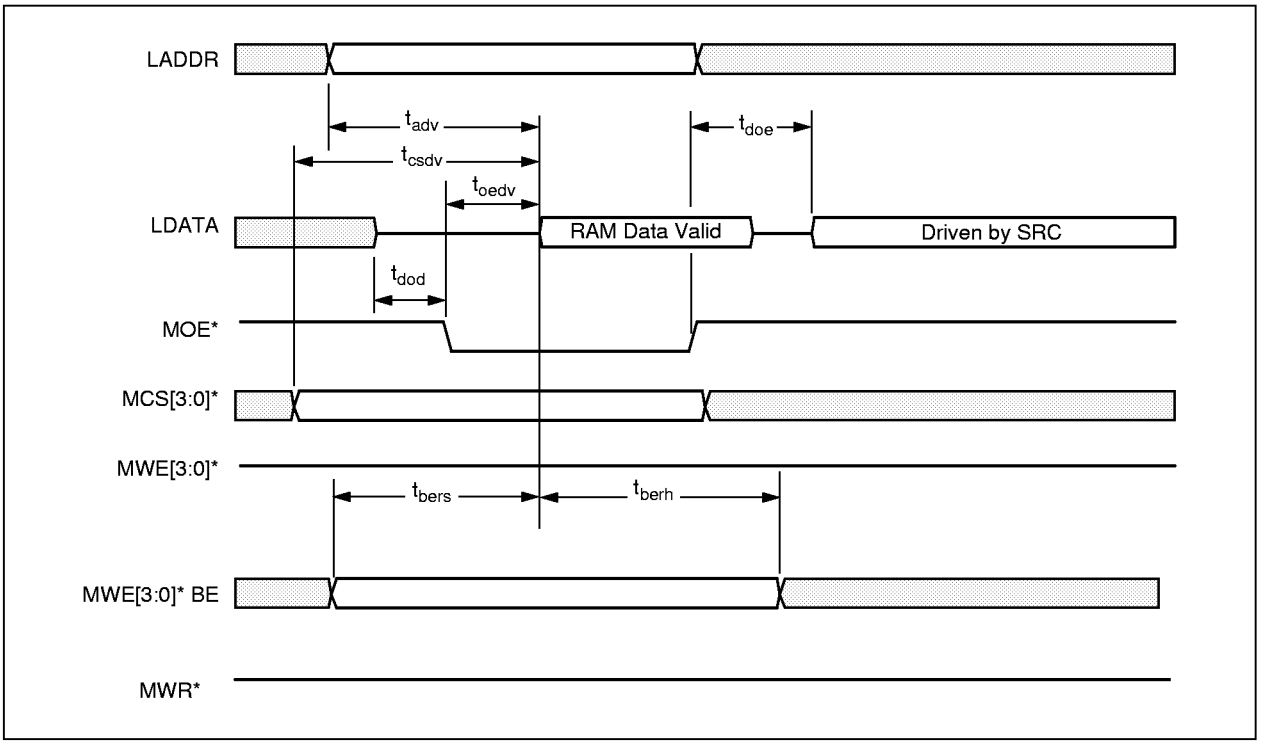
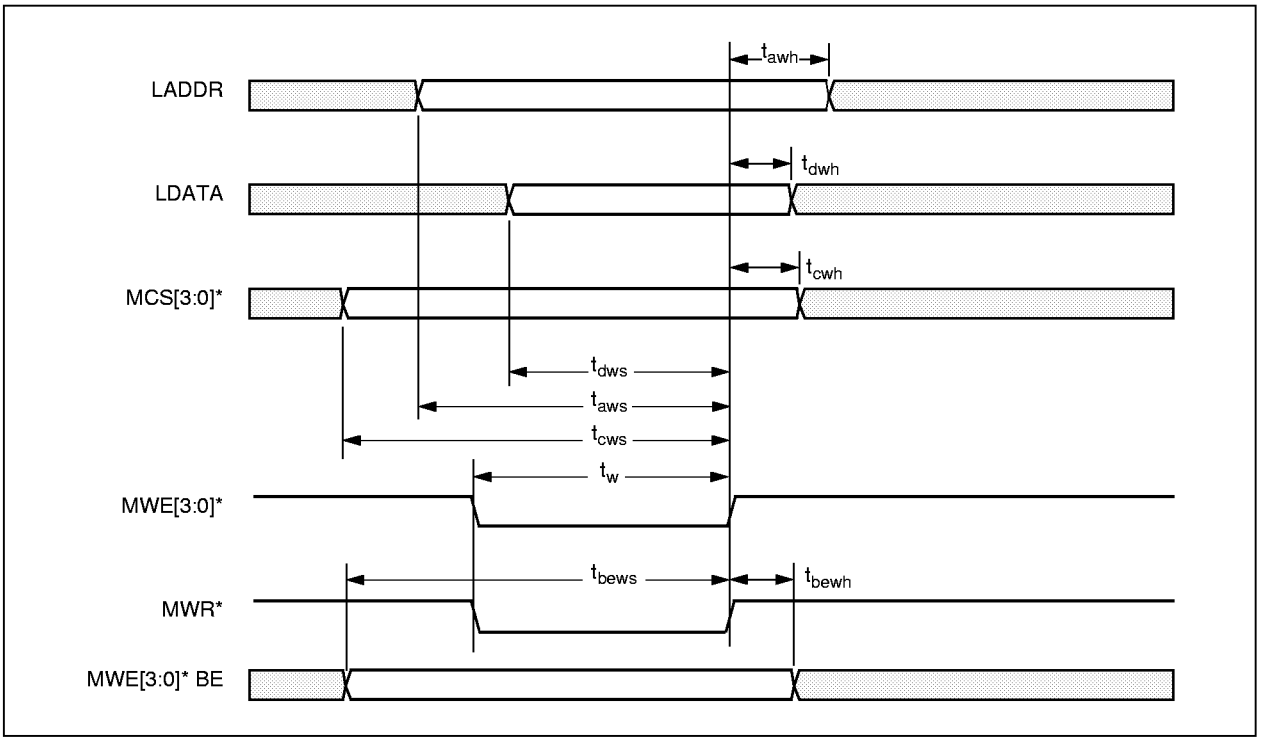


Figure 4-10. Bt8230 Memory Write Timing



4.1.6 PHY Interface Timing (Stand-Alone Mode)

The stand-alone mode of operation is entered when the PROCMODE input is at a logic high indicating that no local processor is present. In this mode, the Bt8230 changes its memory map to include the ATM physical interface device. The interface is fully synchronous to SYSCLK and is designed to interface directly to the Bt8222 ATM Receiver/Transmitter. Timing is given in Table 4-10, Figure 4-11, and Figure 4-12. See subsection 2.4.6 for details.

Table 4-10. PHY Interface Timing (PROCMODE = 1)

Symbol	Parameter	Min	Max	Units
	Synchronous Inputs			
t_{is}	PWAIT* Input Setup(1)	17		ns
t_{ih}	PWAIT* Input Hold(1)	0		ns
	Synchronous Outputs			
t_{ov}	PRDY* Output Valid Delay(2)	12		ns
	PAS* Output Valid Delay(2)	12		ns
	PCS* Output Valid Delay(2)	12		ns
	PBLAST* Output Valid Delay(2)	12		ns
	PBLAST* Output Valid Delay(2)	12		ns
t_{oh}	PRDY* Output Hold(2)		15	ns
	PAS* Output Hold(2)		15	ns
	PCS* Output Hold(2)		15	ns
	PBLAST* Output Hold(2)		15	ns
	PBLAST* Output Hold(2)		15	ns

Notes: (1). See Figure 4-11 for waveforms and definitions.
(2). See Figure 4-12 for waveforms and definitions. The outputs are measured with a load of 35 pf.

Figure 4-11. Synchronous PHY Interface Input Timing

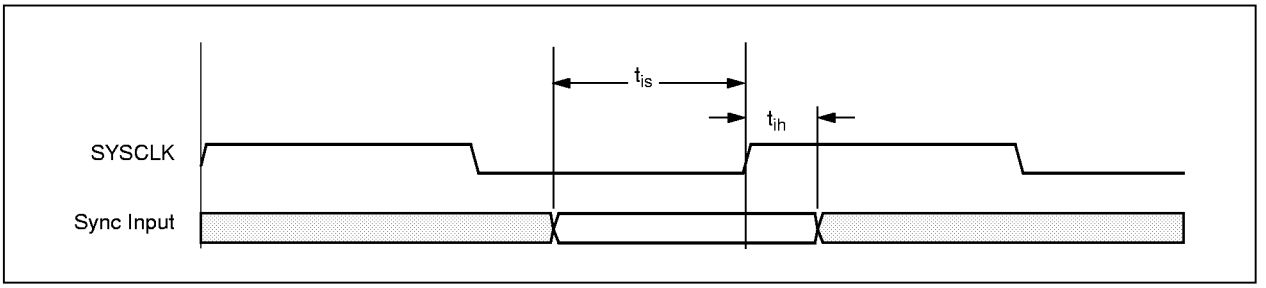
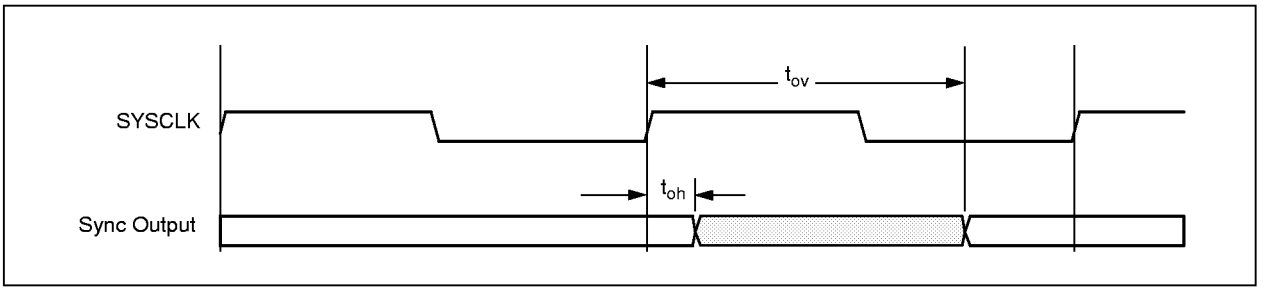


Figure 4-12. Synchronous PHY Interface Output Timing



4.1.7 Local Processor Interface Timing

Timing for the local processor interface can be broken into two sections. The first is the synchronous to SYSCLK interface of processor control signals to and from the Bt8230. The second is the interface to the local SRAM memory and the Bt8230 control and status registers, which involves both SRAM and transceiver and buffer timing parameters that are not specified and are left up to the system designer.

All of the synchronous interface signals are inputs to the Bt8230 except for the system clock (SYSCLK) and the ready output of the Bt8230 (PRDY*). The output loading of these signals is 35 pf for the timing given in Table 4-11. Memory Interface Timing is given in Table 4-12.

Table 4-11. Synchronous Processor Interface Timing (1 of 2)

Symbol	Parameter	Min	Max	Units
	Synchronous Inputs			
t_{is}	PCS* Input Setup(1)	10		ns
	PAS* Input Setup(1)	10		ns
	PBLAST* Input Setup(1)	10		ns
	PWAIT* Input Setup(1)	10		ns
	PADDR[1:0] Input Setup(1)	10		ns
	PBSEL[1:0] Input Setup(1)	8		ns
	PBE[3:0]* Input Setup(1)	9		ns
t_{ih}	PCS* Input Hold(1)	0		ns
	PAS* Input Hold(1)	0		ns
	PBLAST* Input Hold(1)	0		ns
	PWAIT* Input Hold(1)	0		ns
	PADDR[1:0] Input Hold(1)	0		ns
	PBSEL[1:0] Input Hold(1)	0		ns
	PBE[3:0]* Input Hold(1)	0		ns

Table 4-11. Synchronous Processor Interface Timing *(2 of 2)*

Symbol	Parameter	Min	Max	Units
Synchronous Outputs				
t_{ov}	PRDY* Output Valid Delay(2)	12		ns
t_{oh}	PRDY* Output Hold(2)		15	ns

Notes: (1). See Figure 4-13 for waveforms and definitions.
(2). See Figure 4-14 for waveforms and definitions. The outputs are measured with a load of 35 pf.

Figure 4-13. Synchronous Local Processor Input Timing

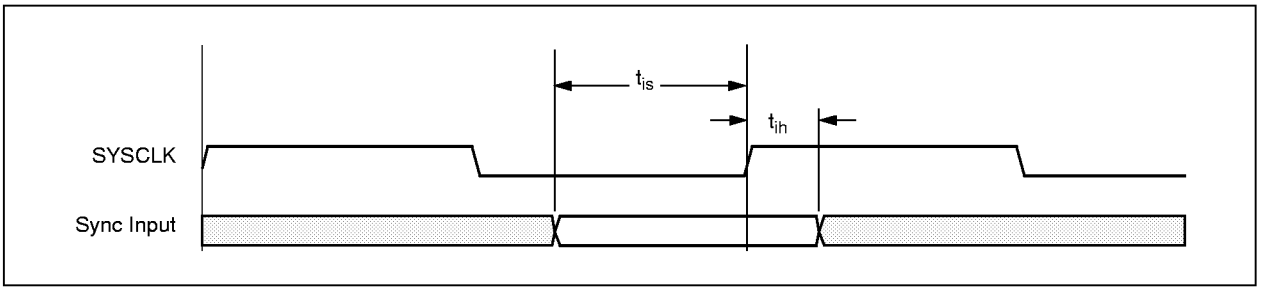


Figure 4-14. Synchronous Local Processor Output Timing

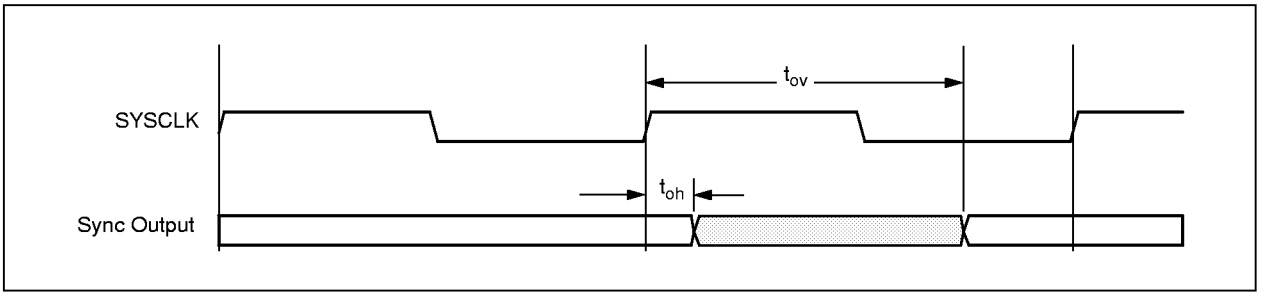


Table 4-12. Local Processor Memory Interface Timing

Symbol	Parameter	Min	Max	Units
t_{pdaenl}	SYSCLK to PDAEN* Low, Bus Recovery Cycle(1)	12	15	ns
t_{pdaenh}	SYSCLK to PDAEN* High, Bus Recovery Cycle(1)	1	4	ns
t_{lod}	LADDR[18:2], LDATA Output Disable to PDAEN* Low, Bus Recovery Cycle(1)	4		ns
t_{loe}	PDAEN* High to LADDR[18:2], LDATA Output Enable, Bus Recovery Cycle(1)	11		ns
t_{MCS}	SYSCLK to MCS*[3:0] Valid(1)	1	4	ns
t_{lav}	SYSCLK to LADDR[1,0] Valid(1, 2)	1	5	ns
t_{OEL}	MOE* Active from SYSCLK(1, 3)	8		ns
t_{OEH}	MOE* Inactive from SYSCLK(1, 3)	1		ns
t_{WL}	MWR*, MWE[3:0] Active from SYSCLK(1, 3)	12	14	ns
t_{WH}	MWR*, MWE[3:0]* Inactive to SYSCLK(1, 3)	1	3	ns
t_{BE}	MWE[3:0]* Byte Enables Valid from SYSCLK (RAMMODE = 1)(1)	1	4	ns

Notes: (1). See Figure 4-15 and Figure 4-16 for waveforms and definitions.
(2). t_{lav} is valid for second and subsequent accesses during burst transfers, see functional timing diagrams.
(3). In the case of two cycle memory, or when inserting wait states by PWAIT*, MOE*, MWE[3:0]*, and MWR* are extended across 2 or more clock cycles with the same relative timing to SYSCLK, see the functional timing diagrams in Section 2.4

Figure 4-15. Local Processor Read Timing

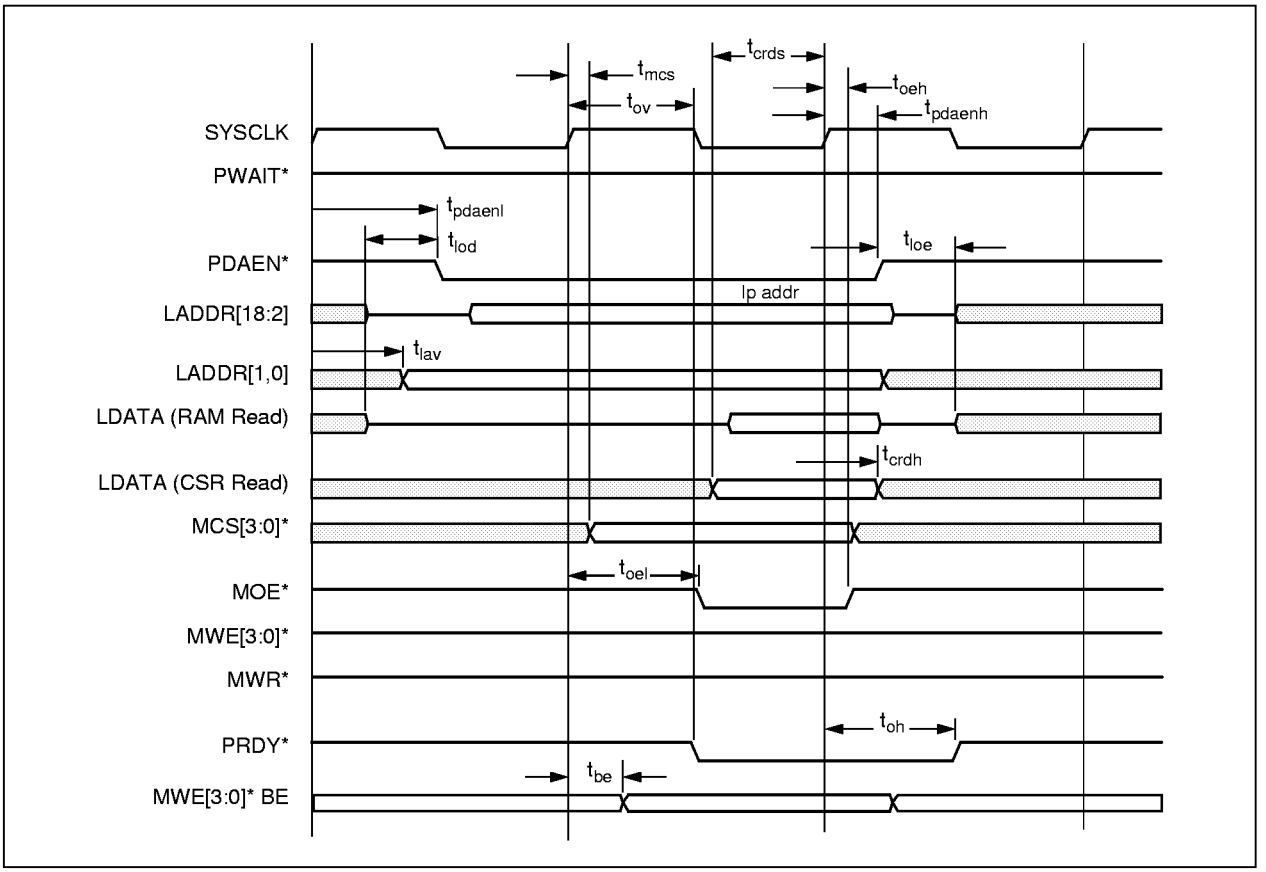
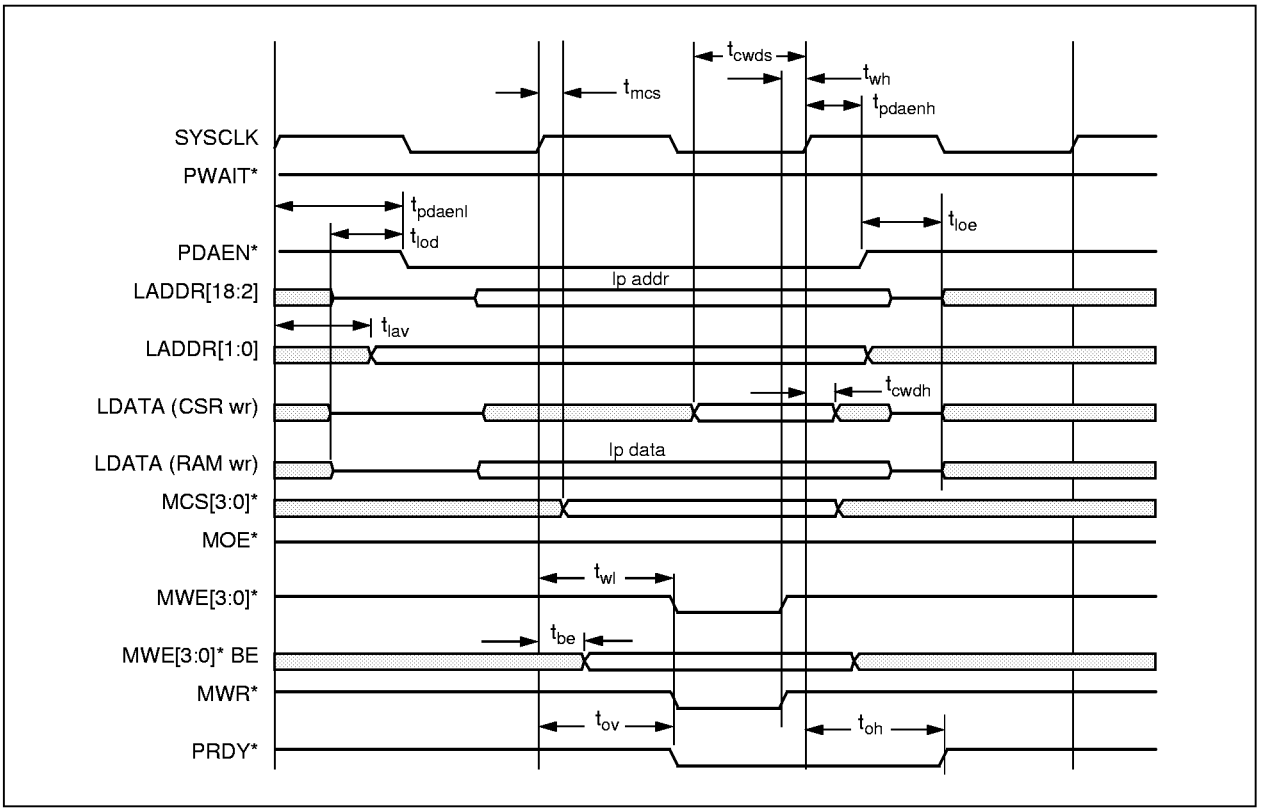


Figure 4-16. Local Processor Write Timing



4.2 Absolute Maximum Ratings

Stresses above those listed as absolute maximum ratings (Table 4-13) may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in other sections of this document is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. This device should be handled as an ESD-sensitive device. Voltage on any signal pin that exceeds the power supply voltage by more than +0.5 V can induce destructive latchup.

Table 4-13. Absolute Maximum Ratings

Parameter	Value	Unit
Supply Voltage	–0.5 to +6.0	Volts
Input Voltage	–0.5 to (Vdd + 0.5)	Volts
Output Voltage	–0.5 to (Vdd + 0.5)	Volts
Operating Temperature—No Air Flow(1)	70	C
Operating Temperature—400 Linear Feet per Minute (1)	85	C
Storage Temperature	–40 to 125	C
Operating Supply Voltage	4.75 to 5.25	Volts
Maximum Current @ Max Clock Frequencies(1)	400	mA
Note (1): This is preliminary information pending full device characterization.		

4.3 DC Characteristics

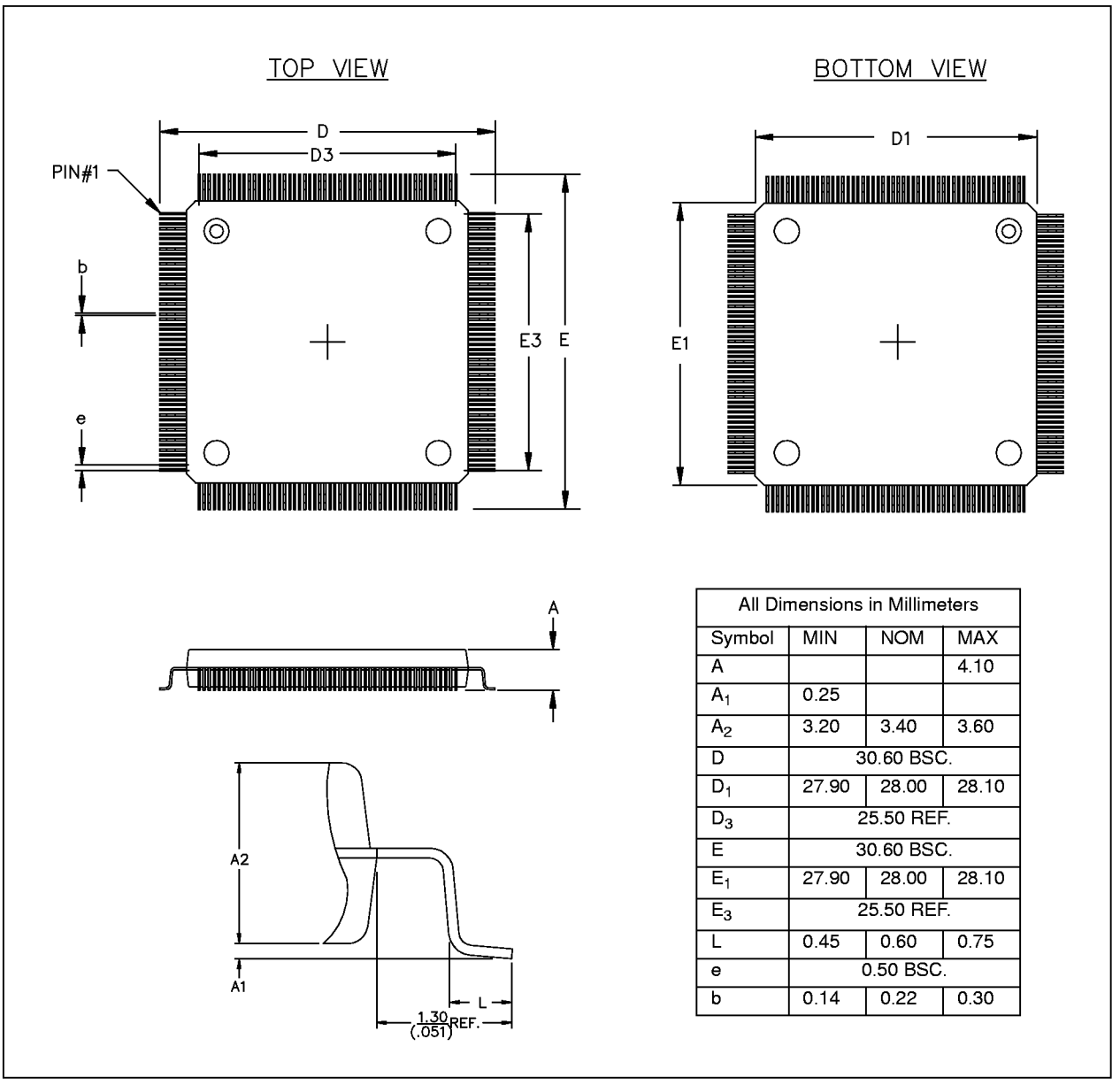
Table 4-14. DC Characteristics

Parameter	Conditions	Min	Max	Unit
Output Voltage High	I_{OH} = –0.8 mA	Vdd –0.1		Volts
Output Voltage Low	I_{OL} = 0.8 mA		Vss + 0.1	Volts
Input Voltage High		2.0	Vdd + 0.5	Volts
Input Voltage Low		–0.5	0.8	Volts
Input Leakage Current	Vin = Vdd or GND	–10	10	uA
Three-state Output Leakage Current	VOUT = Vdd or GND	–10	10	uA
Input Capacitance				pF
Output Capacitance				pF
Note: All outputs are CMOS drive levels and can be used with CMOS or TTL logic.				

4.4 Mechanical Specifications

The Bt8230 208-pin PQFP package is illustrated in Figure 4-15. Figure 4-18 is a pinout configuration of the Bt8230, and a pin listing is given in Table 4-15.

Figure 4-17. 208-Pin Plastic Quad Flat Pack (PQFP)



All Dimensions in Millimeters			
Symbol	MIN	NOM	MAX
A			4.10
A_1	0.25		
A_2	3.20	3.40	3.60
D		30.60 BSC.	
D_1	27.90	28.00	28.10
D_3		25.50 REF.	
E		30.60 BSC.	
E_1	27.90	28.00	28.10
E_3		25.50 REF.	
L	0.45	0.60	0.75
e		0.50 BSC.	
b	0.14	0.22	0.30

Figure 4-18. Bt8230 Pinout Configuration

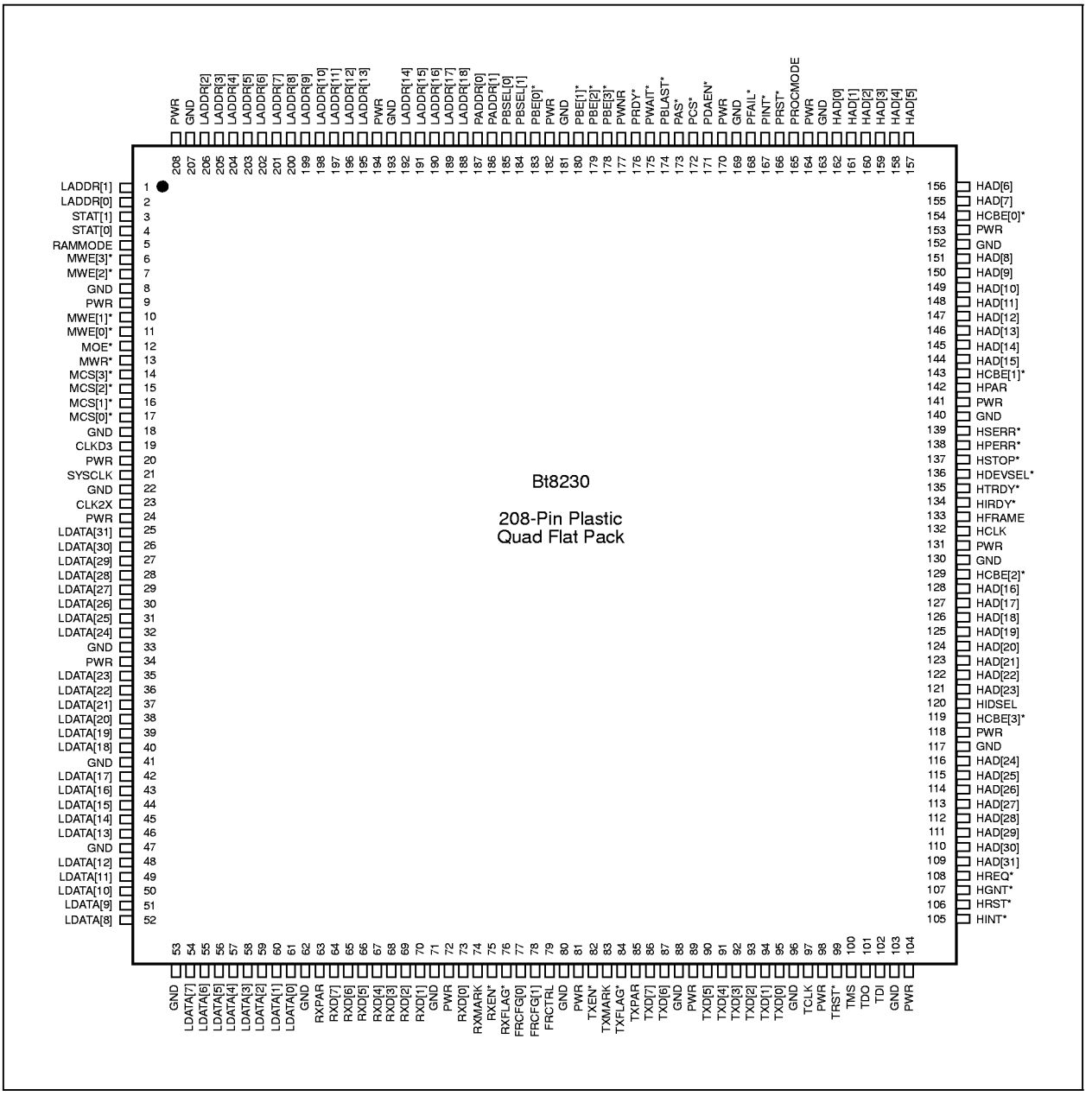


Table 4-15. Pin Descriptions *(1 of 2)*

Pin	Pin Label	I/O	Pin	Pin Label	I/O	Pin	Pin Label	I/O
1	LADDR[1]	I/O	36	LDATA[22]	I/O	71	GND	–
2	LADDR[0]	I/O	37	LDATA[21]	I/O	72	PWR	–
3	STAT[1]	O	38	LDATA[20]	I/O	73	RXD[0]	I
4	STAT[0]	O	39	LDATA[19]	I/O	74	RXMARK	I
5	RAMMODE	I	40	LDATA[18]	I/O	75	RXEN*	I/O
6	MWE[3]*	O	41	GND	–	76	RXFLAG*	I/O
7	MWE[2]*	O	42	LDATA[17]	I/O	77	FRCFG[0]	I
8	GND	–	43	LDATA[16]	I/O	78	FRCFG[1]	I
9	PWR	–	44	LDATA[15]	I/O	79	FRCTRL	I
10	MWE[1]*	O	45	LDATA[14]	I/O	80	GND	–
11	MWE[0]*	O	46	LDATA[13]	I/O	81	PWR	–
12	MOE*	O	47	GND	–	82	TXEN*	I/O
13	MWR*	O	48	LDATA[12]	I/O	83	TXMARK	I/O
14	MCS[3]*	O	49	LDATA[11]	I/O	84	TXFLAG*	I/O
15	MCS[2]*	O	50	LDATA[10]	I/O	85	TXPAR	O
16	MCS[1]*	O	51	LDATA[9]	I/O	86	TXD[7]	O
17	MCS[0]*	O	52	LDATA[8]	I/O	87	TXD[6]	O
18	GND	–	53	GND	–	88	GND	–
19	CLKD3	O	54	LDATA[7]	I/O	89	PWR	–
20	PWR	–	55	LDATA[6]	I/O	90	TXD[5]	O
21	SYSCLK	O	56	LDATA[5]	I/O	91	TXD[4]	O
22	GND	–	57	LDATA[4]	I/O	92	TXD[3]	O
23	CLK2X	I	58	LDATA[3]	I/O	93	TXD[2]	O
24	PWR	–	59	LDATA[2]	I/O	94	TXD[1]	O
25	LDATA[31]	I/O	60	LDATA[1]	I/O	95	TXD[0]	O
26	LDATA[30]	I/O	61	LDATA[0]	I/O	96	GND	–
27	LDATA[29]	I/O	62	GND	–	97	TCLK	I
28	LDATA[28]	I/O	63	RXPAR	I	98	PWR	–
29	LDATA[27]	I/O	64	RXD[7]	I	99	TRST*	I
30	LDATA[26]	I/O	65	RXD[6]	I	100	TMS	I
31	LDATA[25]	I/O	66	RXD[5]	I	101	TDO	O
32	LDATA[24]	I/O	67	RXD[4]	I	102	TDI	I
33	GND	–	68	RXD[3]	I	103	GND	–
34	PWR	–	69	RXD[2]	I	104	PWR	–
35	LDATA[23]	I/O	70	RXD[1]	I	105	HINT*	OD

Table 4-15. Pin Descriptions (2 of 2)

Pin	Pin Label	I/O	Pin	Pin Label	I/O	Pin	Pin Label	I/O
106	HRST*	I	141	PWR	–	176	PRDY*	O
107	HGNT*	I	142	HPAR	I/O	177	PWNR	I/O
108	HREQ*	O	143	HC/BE[1]*	I/O	178	PBE[3]*	I
109	HAD[31]	I/O	144	HAD[15]	I/O	179	PBE[2]*	I
110	HAD[30]	I/O	145	HAD[14]	I/O	180	PBE[1]*	I
111	HAD[29]	I/O	146	HAD[13]	I/O	181	GND	–
112	HAD[28]	I/O	147	HAD[12]	I/O	182	PWR	–
113	HAD[27]	I/O	148	HAD[11]	I/O	183	PBE[0]*	I
114	HAD[26]	I/O	149	HAD[10]	I/O	184	PBSEL[1]	I
115	HAD[25]	I/O	150	HAD[9]	I/O	185	PBSEL[0]	I
116	HAD[24]	I/O	151	HAD[8]	I/O	186	PADDR[1]	I
117	GND	–	152	GND	–	187	PADDR[0]	I
118	PWR	–	153	PWR	–	188	LADDR[18]	I/O
119	HC/BE[3]*	I/O	154	HC/BE[0]*	I/O	189	LADDR[17]	I/O
120	HIDSEL	I	155	HAD[7]	I/O	190	LADDR[16]	I/O
121	HAD[23]	I/O	156	HAD[6]	I/O	191	LADDR[15]	I/O
122	HAD[22]	I/O	157	HAD[5]	I/O	192	LADDR[14]	I/O
123	HAD[21]	I/O	158	HAD[4]	I/O	193	GND	–
124	HAD[20]	I/O	159	HAD[3]	I/O	194	PWR	–
125	HAD[19]	I/O	160	HAD[2]	I/O	195	LADDR[13]	I/O
126	HAD[18]	I/O	161	HAD[1]	I/O	196	LADDR[12]	I/O
127	HAD[17]	I/O	162	HAD[0]	I/O	197	LADDR[11]	I/O
128	HAD[16]	I/O	163	GND	–	198	LADDR[10]	I/O
129	HC/BE[2]*	I/O	164	PWR	–	199	LADDR[9]	I/O
130	GND	–	165	PROCMODE	I	200	LADDR[8]	I/O
131	PWR	–	166	PRST*	O	201	LADDR[7]	I/O
132	HCLK	I	167	PINT*	OD	202	LADDR[6]	I/O
133	HFRAME*	I/O	168	PFAIL*	I	203	LADDR[5]	I/O
134	HIRDY*	I/O	169	GND	–	204	LADDR[4]	I/O
135	HTRDY*	I/O	170	PWR	–	205	LADDR[3]	I/O
136	HDEVSEL*	I/O	171	PDAEN*	I/O	206	LADDR[2]	I/O
137	HSTOP*	I/O	172	PCS*	I/O	207	GND	–
138	HPERR*	I/O	173	PAS*	I/O	208	PWR	
139	HSERR*	OD	174	PBLAST*	I/O			
140	GND	–	175	PWAIT*	I			

Appendix A

Relevant ITU and ANSI Standards

The following is a list of the ITU and ANSI standards relevant to the Bt8230.

- ATM Forum UNI Rev 3.1: User-Network Interface Specification
- ATM Forum Utopia Level 2 Specification
- PCI Rev 2.0: PCI Local Bus Specification
- ANSI T1.627-1993:Broadband ISDN—ATM Layer Functionality and Specification
- ANSI T1.629-1993: Broadband ISDN—ATM Adaptation Layer 3/4 Common Part Functions and Specification
- ANSI T1.635-1994: Broadband ISDN—ATM Adaptation Layer Type 5 Common Part Functions and Specification
- I.363: B-ISDN ATM Adaptation Layer Specification
- I.610: B-ISDN Operation and maintenance Principles and Functions
- GR-1248: Generic Requirements for Operation of ATM Network Elements

All of the documents listed above can be obtained from the following companies:

Bellcore
Customer Service
8 Corporate Place - Room 3C-183
Piscataway, NJ 08854-4156
1-800-521-CORE

PCI Special Interest Group
P.O. Box 14070
Portland, OR 97214
1-800-433-5177
1-503-797-4207

ATM FORUM
The ATM Forum
303 Vintage Park Drive
Foster City, CA 94404-1138

ANSI
11 West 42nd Street
New York, NY 10036
1-212-642-4900

For ITU documents:

Omnicom
Phillips Business Information
1201 Seven Locks Road, Suite 300
Potomac, MD 20854
1-800 OMNICOM (666-4266)