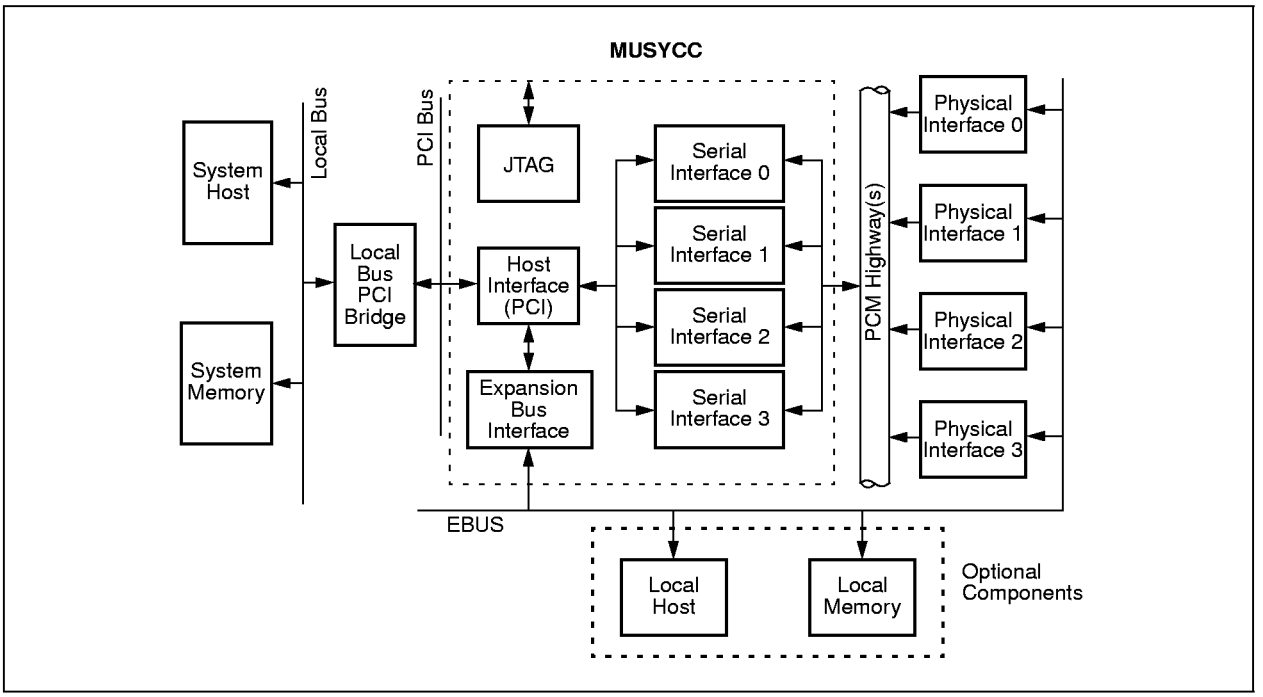
System Overview

The Brooktree Multichannel Synchronous Communications Controller (MUSYCC) multiplexes and demultiplexes up to 128 data channels. Each channel can be configured to support HDLC, Transparent, or SS7 applications. MUSYCC operates at Layer 2 (the data link protocol level) of the reference model recommended by the International Organization for Standardization (ISO) Open Systems Interconnection (OSI) and resides between multiple serial interface devices and buffer memory shared with one or more host processors.

MUSYCC is a high-throughput communications controller for synchronous link layer applications, and is available in two versions: the 64-channel Bt8472 and the 128-channel Bt8474. Each version provides the complete set of features designed into this communications controller in a common form factor which makes the Bt8474/Bt8472 pin-, hardware-, and software-compatible.

Figure 1. System Block Diagram



MUSYCC's serial ports interface to a standard PCM highway, which operates at T1, E1, 2xE1, or 4xE1 rates. Data may be formatted in the HDLC protocol or left unformatted. The protocol is specified on a per-channel and direction basis.

An on-device Peripheral Components Interface (PCI) controller known as the host interface is provided. Access to MUSYCC is available through PCI read, write, and configuration cycles.

MUSYCC also provides an on-device 32-bit local expansion bus (EBUS) controller which allows a host processor to access local memory and physical interface devices directly through MUSYCC over the PCI using configurable memory mapping features.

MUSYCC manages buffer memory for each of the active data channels with common list processing structures. The on-device features allow data transmission between buffer memory and the serial interfaces with minimum host processor intervention. This allows the host processor to concentrate on managing the higher layers of the protocol stack.

The figures below illustrates a detailed system block diagram and an example application.

Figure 2. Detailed System Block Diagram

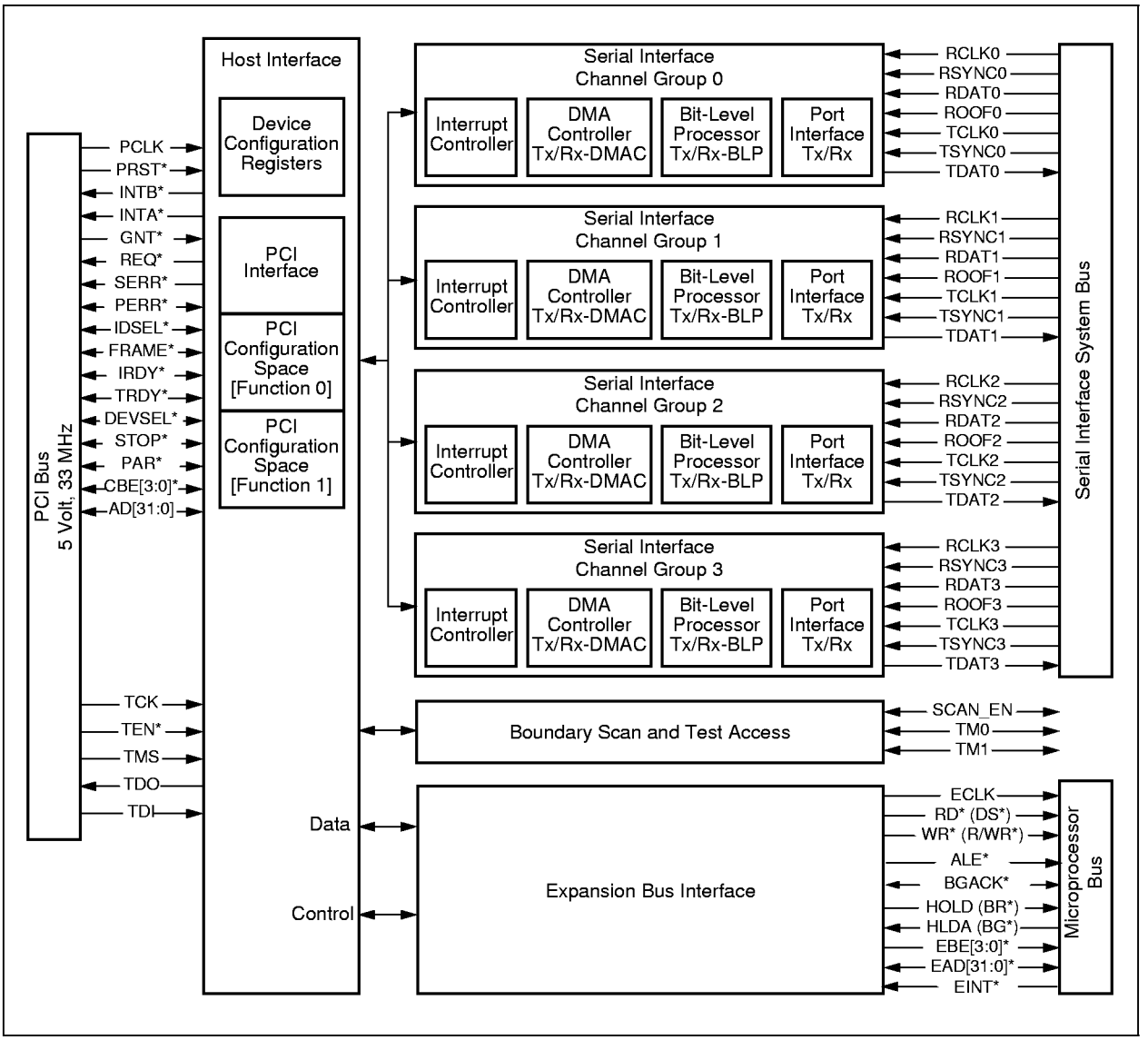
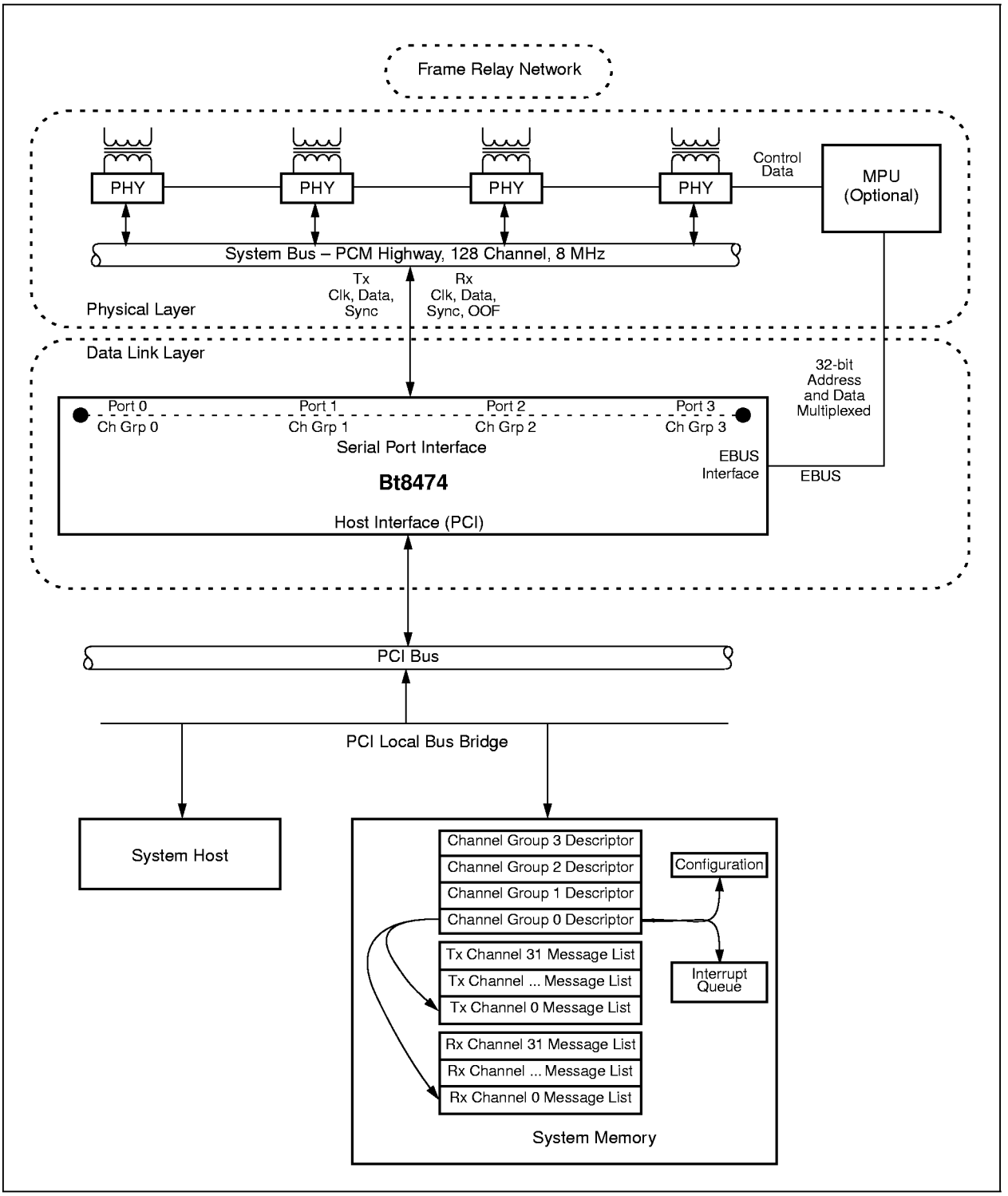


Figure 3. MUSYCC Application Example—Frame Relay Switch



Pin Descriptions

Both Bt8474 and Bt8472 are packaged in a 160-pin surface-mount Plastic Quad Flat Pack (PQFP). The figures below show the pinout diagrams and the functionally partitioned logic diagrams for both Bt8474 and Bt8472.

Bt8474 is a 128-channel MUSYCC, Bt8472 is a 64-channel MUSYCC. Both versions are pin compatible. Pin input/output functions are defined in Table 1. Pin labels, signal names, I/O functions, and signal definitions are provided in Table 2, Bt8474/8472 Hardware Signal Definitions.

An active low signal is always denoted with a trailing asterisk (*).

Figure 4. Bt8474 Pinout Configuration

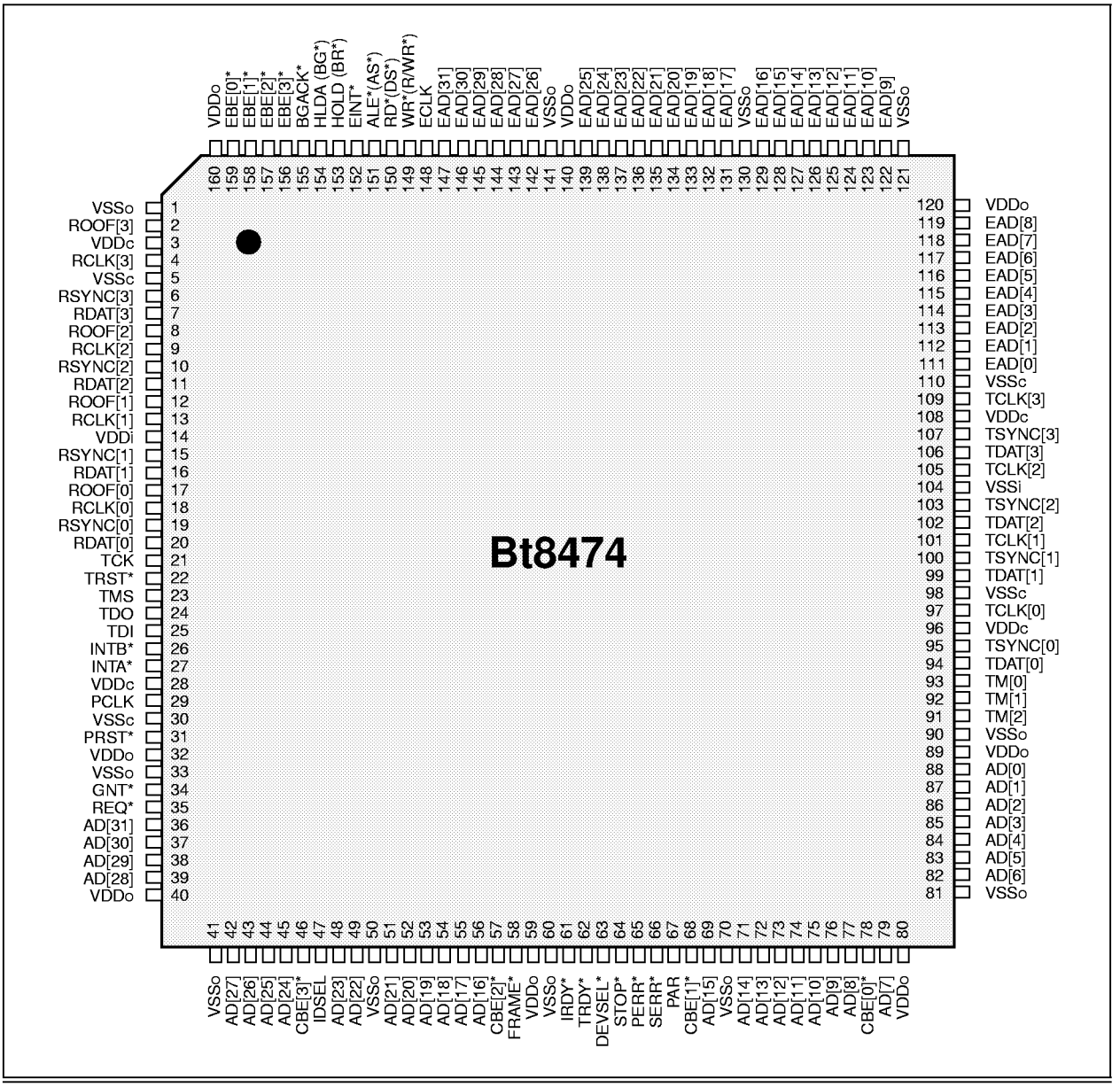
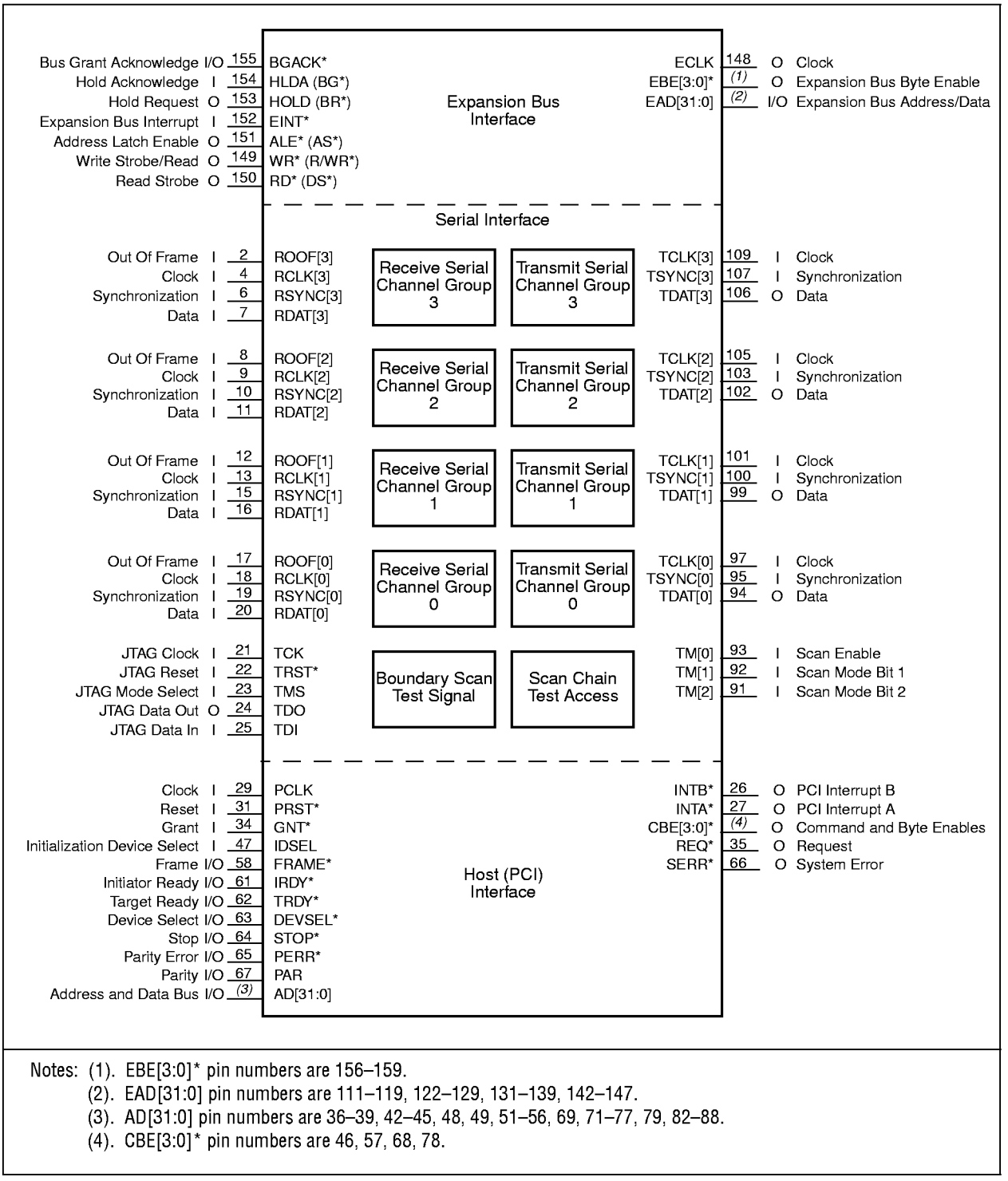


Figure 5. Bt8474 Logic Diagram



Notes: (1). EBE[3:0]* pin numbers are 156–159.
(2). EAD[31:0] pin numbers are 111–119, 122–129, 131–139, 142–147.
(3). AD[31:0] pin numbers are 36–39, 42–45, 48, 49, 51–56, 69, 71–77, 79, 82–88.
(4). CBE[3:0]* pin numbers are 46, 57, 68, 78.

Figure 6. Bt8472 Pinout Configuration

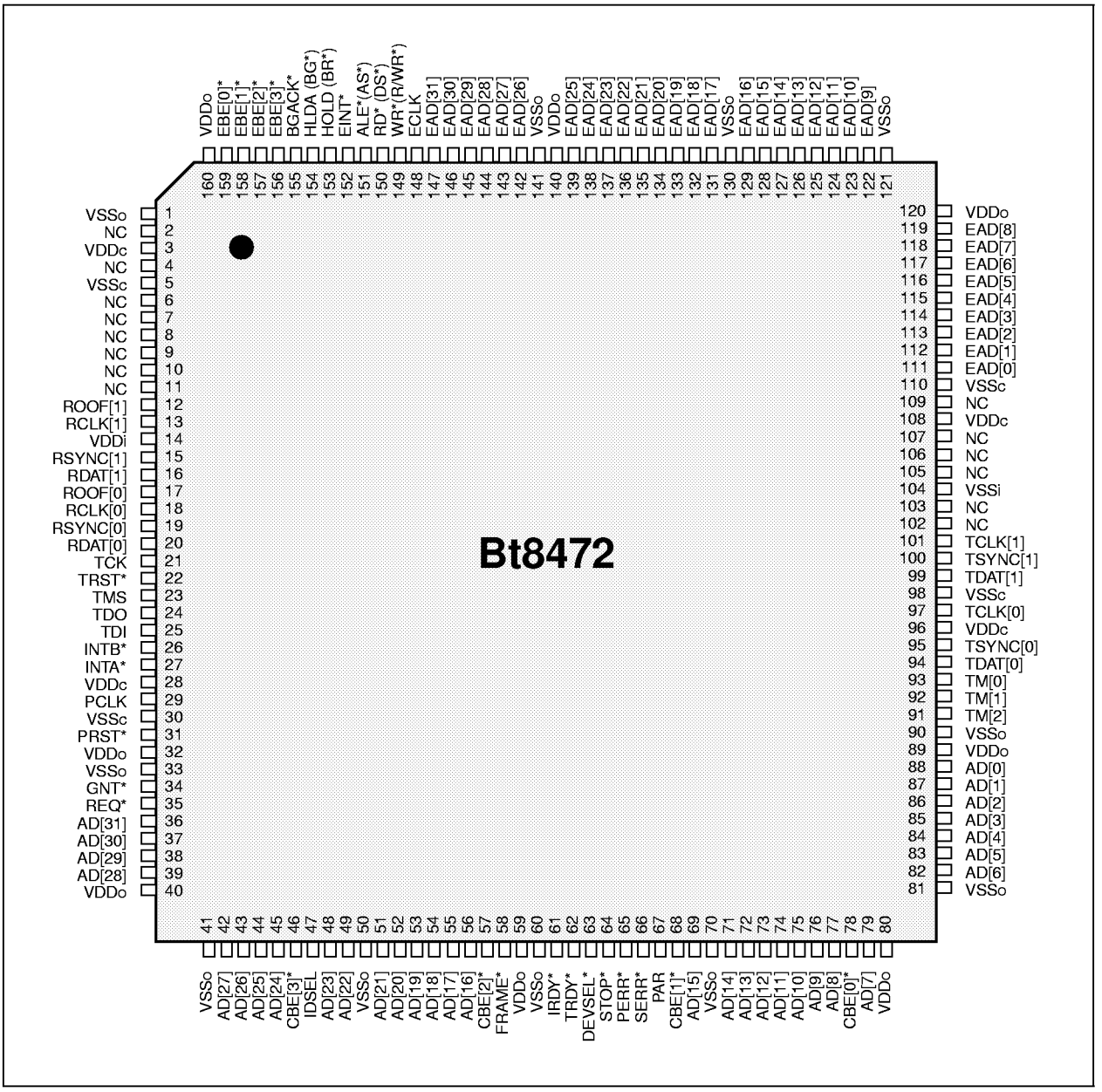
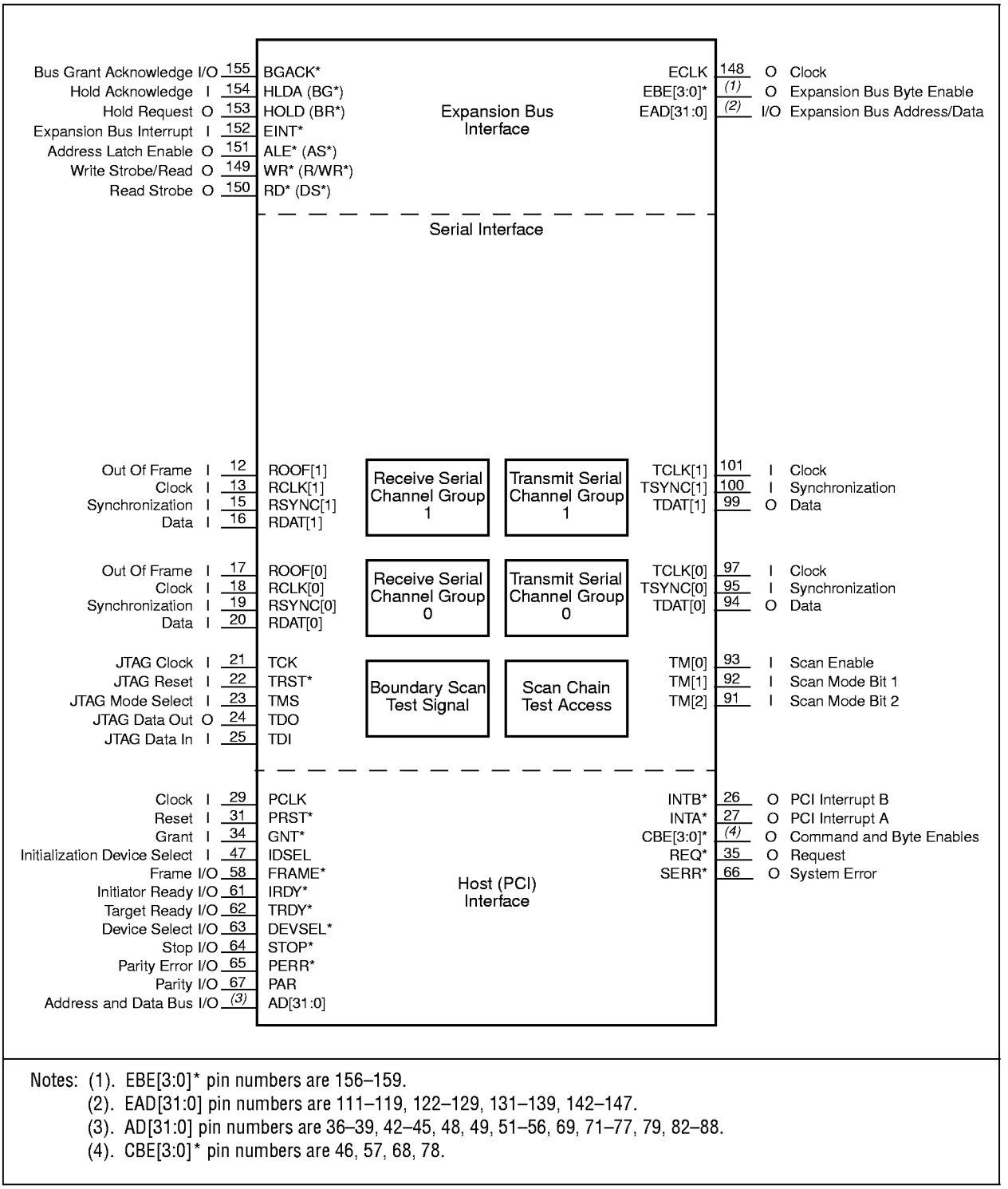


Figure 7. Bt8472 Logic Diagram



Notes: (1). EBE[3:0]* pin numbers are 156–159.
(2). EAD[31:0] pin numbers are 111–119, 122–129, 131–139, 142–147.
(3). AD[31:0] pin numbers are 36–39, 42–45, 48, 49, 51–56, 69, 71–77, 79, 82–88.
(4). CBE[3:0]* pin numbers are 46, 57, 68, 78.

Table 1. I/O Pin Types

I/O	Definition
I	Input. High impedance, TTL.
O	Output. CMOS.
I/O	Input/Output. TTL input/CMOS output.
t/s	Three-state. Bidirectional three-state input/output pin.
s/t/s	Sustained three-state. This is an active-low, three-state signal owned by only one driver at a time. The driver that drives an s/t/s signal low must drive it high for at least one clock cycle before allowing it to float. A pullup is required to sustain the deasserted value.
o/d	Open drain.
Note: All outputs are CMOS drive levels and can be used with CMOS or TTL-logic.	

Table 2. Bt8474/8472 Hardware Signal Definitions *(1 of 6)*

	Pin Label	Signal Name	I/O	Definition
Expansion Bus Interface	ECLK	Expansion Bus Clock	t/s O	ECLK is an inverted version of the PCI clock applied at the PCLK input.
	EAD[31:0]	Expansion Bus Address and Data	t/s I/O	EAD[31:0] is a multiplexed address/data bus. During the address phase, pins EAD[17:0] will contain meaningful address information. It is the same address as PCI AD[19:2] for the corresponding cycle. Pins EAD[31:18] will be driven to 0 during the address phase. This is because those upper bits are compared, during the PCI address phase, to the value in the relocatable EBUS Base Address Register to determine if the PCI cycle is in fact addressing into MUSYCC EBUS space. During data phase of an EBUS access cycle, the PCI signals AD[31:0] are transferred to the EBUS signal lines EAD[31:0] unaltered.
	EBE[3:0]*	Expansion Bus Byte Enables	t/s O	EBE* contains the same information as the PCI byte enables but driven in chip select style protocol used as active-low chip selects when MUSYCC is connected to more than one byte-wide device. All PCI accesses with byte lane 0's byte enable asserted would go to the byte-wide device connected to EAD[7:0]. Likewise, for byte lanes 1, 2, and 3 and EAD[15:8], EAD[23:16], and EAD[31:24], respectively. Only the CBE[3:0]* signals from the PCI data phase (byte-enable signals and not the command signals from the PCI address phase) are transferred to the EBE[3:0]* signal lines. EBE* is held high during all other phases of PCI access cycles.
	WR* (R/WR*)	Write Strobe	t/s O	High-to-low transition enables write data from MUSYCC into peripheral device. Rising edge defines write. (In Motorola mode, R/WR* held high throughout read operation and held low throughout write operation. Determines meaning of DS* strobe.)
	RD* (DS*)	Read Strobe	t/s O	High-to-low transition enables read data from peripheral into MUSYCC. Held high throughout write operation. (In Motorola mode, DS* transitions low for both read and write operations and held low throughout the operation.
	ALE* (AS*)	Address Latch Enable	t/s O	High-to-low transition indicates that EAD[31:0] bus contains valid address. Remains asserted low through the data phase of the EBUS access. (In Motorola mode, high-to-low transition indicates EBUS contains valid address. Remains asserted for the entire access cycle.)
	EINT*	Expansion Bus Interrupt	I	EINT* transfers interrupts from local devices to the PCI INTB* pin.
	HOLD (BR*)	Hold Request (Bus Request)	t/s O	When asserted, MUSYCC requests control of the EBUS.
	HLDA (BG*)	Hold Acknowledge (Bus Grant)	I	When asserted, MUSYCC has access to the EBUS. It is held asserted when there are no other masters connected to the bus, or asserted as a handshake mechanism to control EBUS arbitration.
	BGACK*	Bus Grant Acknowledge	t/s I/O	When asserted, MUSYCC acknowledges to the bus arbiter that the bus grant signal was detected and a bus cycle will be sustained by MUSYCC until this signal is deasserted.

Table 2. Bt8474/8472 Hardware Signal Definitions (2 of 6)

	Pin Label	Signal Name	I/O	Definition
Serial Interface	TCLK[3:0]	Transmit Clock	I	Controls the rate at which data is transmitted. Synchronizes transitions for TDATx and sampling of TSYNCx. Valid frequencies from DC to 8.192 ±10% MHz.
	TSYNC[3:0]	Transmit Synchronization	I	TSYNC is sampled on the specified active edge of the corresponding transmit clock, TCLKx. See TSYNC_EDGE bit-field in Table 36, Port Configuration Descriptor, As TSYNCx signal transitions low-to-high, start of a transmit frame is indicated. For T1 mode, the corresponding data bit latched out during the same bit time period (but not necessarily the same clock edge) is the F-bit of the T1 frame. For E1 modes, the corresponding data bit latched out during the same bit time period (but not necessarily the same clock edge) is Bit-0 of the E1 frame. For Nx64 mode, the corresponding data bit is latched out 4 bit time periods later and is Bit-0 of the Nx64 frame. TSYNCx must remain asserted high for a minimum of a setup and hold time relative to the active clock edge for this signal. If the flywheel mechanism is used, no other synchronization signal is required, as MUSYCC tracks the start of each subsequent frame. If the flywheel mechanism is not used, then a subsequent low-to-high assertion is required to indicate the start of the next frame. See SFALIGN bit-field in Table 34, Group Configuration Descriptor.
	TDAT[3:0]	Transmit Data	t/s O	Serial data latched out on active edge of transmit clock, TCLKx. If channel is unmapped to timeslot, data bit is considered invalid and MUSYCC outputs either three-state signal or logic 1 depending on value of bit-field TRITX in Table 36, Port Configuration Descriptor.
	RCLK[3:0]	Receive Clock	I	Active edge samples RDATx and RSYNCx. Valid frequencies from DC to 8.192 ± 10% MHz.
	RSYNC[3:0]	Receive Synchronization	I	RSYNCx is sampled on the specified active edge of the corresponding receive clock, RCLKx. See RSYNC_EDGE bit-field in Table 36, Port Configuration Descriptor. As RSYNCx signal transitions low-to-high, start of a receive frame is indicated. For T1 mode, the corresponding data bit sampled and stored during the same bit time period (but not necessarily the same clock edge) is the F-bit of the T1 frame. For E1 modes, the corresponding data bit sampled and stored during the same bit time period (but not necessarily the same clock edge) is Bit-0 of the E1 frame. For Nx64 mode, the corresponding data bit sampled and stored during the same bit time period (but not necessarily the same clock edge) is Bit-0 of the Nx64 frame. RSYNCx must remain asserted high for a minimum of a setup and hold time relative to the active clock edge for this signal. If the flywheel mechanism is used, no other synchronization signal is required, as MUSYCC tracks the start of each subsequent frame. If the flywheel mechanism is not used, then a subsequent low-to-high assertion is required to indicate the start of the next frame. See SFALIGN bit-field in Table 34, Group Configuration Descriptor.

Table 2. Bt8474/8472 Hardware Signal Definitions *(3 of 6)*

	Pin Label	Signal Name	I/O	Definition
Serial Interface (Continued)	RDAT[3:0]	Receive Data	I	Serial data sampled on active edge of receive clock, RCLKx. If channel is mapped to a timeslot, input bit is sampled and transferred to memory. If channel is unmapped to timeslot, data bit is considered invalid and MUSYCC ignores received sample.
	ROOF[3:0]	Receiver Out-Of-Frame	I	ROOFx is sampled on the specified active edge of the corresponding receive clock, RCLKx. See ROOF_EDGE bit-field in Table 36, Port Configuration Descriptor, As ROOFx signal transitions from low to high, an Out-of-Frame (OOF) condition is indicated. As long as ROOFx is asserted, the received serial data is considered OOF. Depending on the state of OOFABT bit-field in Table 34, Group Configuration Descriptor, receive bit processing may be disabled for the entire channel group (all channels disabled) while ROOFx remains asserted. Upon deassertion of ROOFx, bit-level processing resumes for all affected channels. If the flywheel mechanism is used, no other synchronization signal is required, as MUSYCC tracks the start of each subsequent frame during the OOF period. If the flywheel mechanism is not used, then a subsequent RSYNCx assertion is required to indicate the start of the next frame. See SFALIGN bit-field in Table 34, Group Configuration Descriptor.

Table 2. Bt8474/8472 Hardware Signal Definitions (4 of 6)

	Pin Label	Signal Name	I/O	Definition
PCI Interface	AD[31:0]	PCI Address and Data	t/s I/O	AD[31:0] is a multiplexed address/data bus. A PCI transaction consists of an address phase during the first clock period followed by one or more data phases. AD[7:0] is the LSB.
	PCLK	PCI Clock	I	PCLK provides timing for all PCI transitions. All PCI signals except PRST*, INTA*, and INTB* are synchronous to PCLK and are sampled on the rising edge of PCLK. MUSYCC supports a PCI clock up to 33 MHz.
	PRST*	PCI Reset	I	This input resets all functions on MUSYCC.
	CBE[3:0]*	PCI Command and Byte Enables	t/s I/O	During the address phase CBE[3:0]* contain command information; during the data phases these pins contain information denoting which byte lanes are valid. PCI commands are defined as follows: CBE[3:0] — Command Type Oh 0000b Interrupt Acknowledge 1h 0001b Special Cycle 6h 0110b Memory Read 7h 0111b Memory Write Ah 1010b Configuration Read Bh 1011b Configuration Write Ch 1100b Memory Read Multiple Dh 1101b Dual Address Cycle Eh 1110b Memory Read Line Fh 1111b Memory Write and Invalidate
	PAR	PCI Parity	t/s I/O	The number of 1s on PAR, AD[31:0], and CBE[3:0]* is an even number. PAR always lags AD[31:0] and CBE* by one clock. During address phases, PAR is stable and valid one clock after the address; during the data phases it is stable and valid one clock after TRDY* on reads and one clock after IRDY* on writes. It remains valid until one clock after the completion of the data phase.
	FRAME*	PCI Frame	s/t/s I/O	FRAME* is driven by the current master to indicate the beginning and duration of a bus cycle. Data cycles continue as FRAME* stays asserted. The final data cycle is indicated by the deassertion of FRAME*. For a non-burst, one-data-cycle bus cycle, this pin is only asserted for the address phase.
	IRDY*	PCI Initiator Ready	s/t/s I/O	IRDY* asserted indicates the current master's readiness to complete the current data phase.
	TRDY*	PCI Target Ready	s/t/s I/O	TRDY* asserted indicates the target's readiness to complete the current data phase.
	STOP*	PCI Stop	s/t/s I/O	STOP* asserted indicates the selected target is requesting the master to stop the current transaction.

Table 2. Bt8474/8472 Hardware Signal Definitions *(5 of 6)*

	Pin Label	Signal Name	I/O	Definition
PCI Interface	DEVSEL*	PCI Device Select	s/t/s I/O	When asserted, DEVSEL* indicates that the driving device has decoded its address as the target of the current cycle.
	IDSEL	PCI Initialization Device Select	I	This input is used to select MUSYCC as the target for configuration read or write cycles.
	SERR*	System Error	o/d O	Any PCI device can assert SERR* to indicate a parity error on the address cycle or parity error on the data cycle of a special cycle command or any other system error where the result will be catastrophic. MUSYCC will only assert SERR* if it detects a parity error on the address cycle. Since SERR* is not an s/t/s signal, restoring it to the deasserted state is done with a weak pullup (same value as used for s/t/s). Note that MUSYCC does not input SERR*. It is assumed that the host will reset MUSYCC in the case of a catastrophic system error.
	PERR*	Parity Error	s/t/s I/O	PERR* is asserted by the agent receiving data when it detects a parity error on a data phase. It is asserted one clock after PAR is driven, which is two clocks after the AD and CBE* parity was checked. If MUSYCC masters a PCI write cycle and after supplying the data during the data phase of the cycle detects this signal being asserted by the agent receiving the data, then MUSYCC generates the PERR Interrupt Descriptor towards the host. If MUSYCC masters a PCI read cycle and after receiving the data during the data phase of the cycle calculates that a parity error has occurred, then MUSYCC asserts this signal and also generates the PERR Interrupt Descriptor towards the host.
	INTA*	PCI MUSYCC Interrupt	o/d O	INTA* is driven by MUSYCC to indicate a MUSYCC Layer 2 interrupt condition to the host processor.
	INTB*	PCI Expansion Bus Interrupt	o/d O	INTB* is driven by MUSYCC to notify the host processor of an interrupt pending from the EBUS.
	REQ*	PCI Bus Request	t/s O	MUSYCC drives REQ* to notify the PCI arbiter that it desires to master the bus. Every master in the system has its own REQ*.
	GNT*	PCI Bus Grant	I	The PCI bus arbiter asserts GNT* when MUSYCC is free to take control of the bus, assert FRAME*, and execute a bus cycle. Every master in the system has its own GNT*.
Boundary Scan and Test Access	TCK	JTAG Clock	I	Used to clock in the TDI and TMS signals as well as clock out TDO signal.
	TRST*	JTAG Enable	I	An active-low input used to put the chip into a special test mode. This pin should be pulled-up in normal operation.
	TMS	JTAG Mode Select	I	The test signal input decoded by the TAP controller to control test operations.
	TDO	JTAG Data Output	t/s O	The test signal used to transmit serial test instructions and test data.
	TDI	JTAG Data Input	I	The test signal used to receive serial test instructions and test data.
	TM[0] TM[1] TM[2]	Test Mode	I	Encodes test modes. TM[0] TM[1] TM[2] 0 0 0 Normal Operation. Tie to ground. 1 1 1 All outputs three-stated.

Table 2. Bt8474/8472 Hardware Signal Definitions (6 of 6)

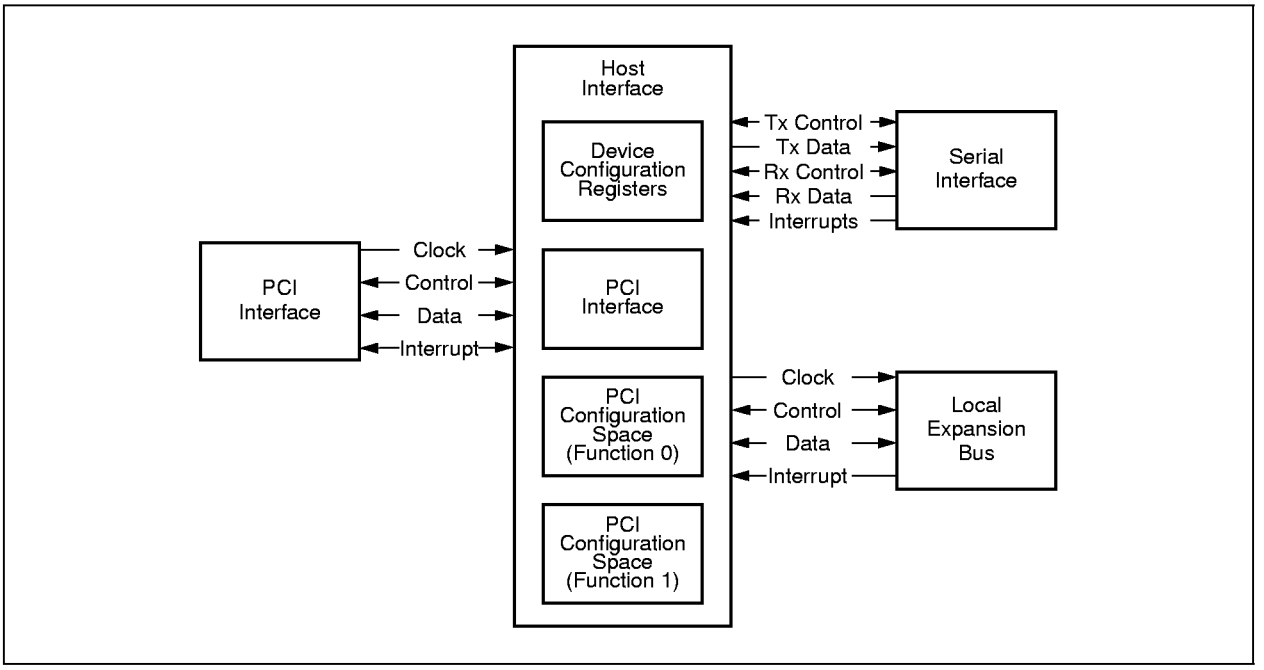
	Pin Label	Signal Name	I/O	Definition
Power and Ground	VDDc VDDi VDDo	Power	–	13 pins are provided for power. +5 V DC ± 5%. Four VDDc (core), one Vddi (input), and eight VDDo (output).
	VSSc VSSi VSSo	Ground	–	16 pins are provided for ground. 0 V DC. Four VSSc (core), one VSSi (input), and 11 VSSo (output).

Host Interface

MUSYCC's host interface performs the following major functions:

- Transfers data between the serial interface and shared memory over the PCI bus.
- Bridges system host processors to devices connected to EBUS.
- Stores configuration state information.

Figure 8. Host Interface Functional Block Diagram



PCI Interface

The host interface in MUSYCC is compliant with the PCI Local Bus Specification. MUSYCC provides a PCI interface that is specific to 5 V and 33 MHz operation.

NOTE: The PCI Local Bus Specification (Revision 2.1, June 1, 1995) is an architectural, timing, electrical, and physical interface standard that provides a mechanism for a device to interconnect with processor and memory systems over a standard bus.

The host interface can act as a PCI master and a PCI slave, and contains MUSYCC's PCI configuration space and internal registers. When MUSYCC needs to access shared memory, it masters the PCI bus and completes the memory cycles without external intervention.

MUSYCC provides the host with a PCI bridge to an on-device EBUS, and behaves as a PCI-slave when providing this access.

MUSYCC is a multifunction PCI agent. One function is mapped to the Layer 2 HDLC control logic. A second function is mapped to the Layer 1 physical interface for the expansion bus pins.

PCI Initialization

Generally, when a system initializes a module containing a PCI device, the configuration manager reads the configuration space of each PCI device on a PCI bus. Hardware signals select a specific PCI device based on a bus number, a slot number, and a function number. If a device that is addressed (via signal lines) responds to the configuration cycle by claiming the bus, then that function's configuration space is read out from the device during the cycle. Since any PCI device can be a multifunction device, every supported function's configuration space needs to be read from the device. Based on the information read, the configuration manager will assign system resources to each supported function within the device. Sometimes new information needs to be written into the function's configuration space. This is accomplished with a configuration write cycle.

MUSYCC is a multifunction device that has device-resident memory to store the required configuration information. MUSYCC supports Function 0 and Function 1 and, as such, only responds to Function 0 and Function 1 configuration cycles.

PCI Bus Operations

MUSYCC behaves either as a PCI master or a PCI slave device at any time and switches between these modes as required during device operation.

As a PCI slave, MUSYCC responds to the following PCI bus operations:

- Memory Read
- Memory Write
- Configuration Read
- Configuration Write
- Memory Read Multiple (treated like Memory Read in slave mode)
- Memory Read Line (treated like Memory Read in slave mode)
- Memory Write and Invalidate (treated like Memory Write)

All other PCI cycles are ignored by MUSYCC. The only cycles that are mapped to operations on the EBUS are the memory cycles.

As a PCI-master, MUSYCC generates the following PCI bus operations:

- Memory Read Multiple (generated only in master mode)
- Memory Write
- Dual Address Cycle

PCI Configuration Space

This section describes how MUSYCC implements the required PCI configuration register space. The intent of PCI configuration space definition is to provide an appropriate set of configuration registers which satisfy the needs of current and anticipated system configuration mechanisms, without specifying those mechanisms or otherwise placing constraints on their use. These registers allow for:

- Full device relocation, including interrupt binding
- Installation, configurations, and booting without user intervention
- System address map construction by device-independent software

MUSYCC only responds to Type 0 configuration cycles. Type 1 cycles, which pass a configuration request on to another PCI bus are ignored.

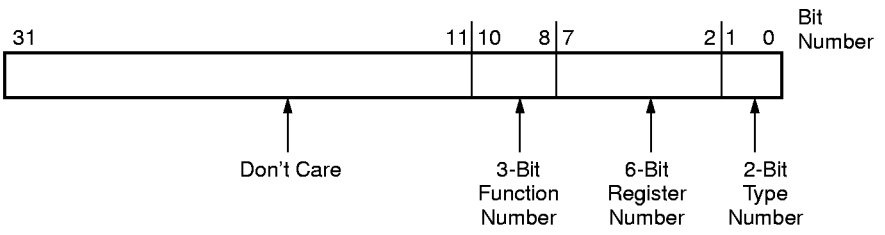
MUSYCC is a two-function PCI agent, therefore, it must implement configuration space for both functions:

- Function 0: All HDLC processing as an HDLC network controller. Can master the PCI bus or respond to slave accesses from another bus master.
- Function 1: EBUS bridge to local devices. Respond only when another bus master performs a memory access into the Function 1 address range.

The PCI controller in MUSYCC responds to configuration and memory cycles, but only memory cycles cause bus activity on the EBUS.

The address phase during a MUSYCC configuration cycle indicates the function number and register number being addressed which can be decoded by observing the status of the address lines AD[31:0].

Figure 9. Address Lines During Configuration Cycle



Notes: 1. MUSYCC supports Functions 0 and 1.
2. MUSYCC supports Registers 0 through 15, inclusively.
3. MUSYCC supports Type 0 configuration cycles.

The value of the signal lines AD[10:8] selects the function being addressed. MUSYCC supports Functions 0 and 1 and will not respond if another function is selected.

The value of the signal lines AD[7:2] during the address phase of configuration cycles selects the register of the configuration space to access. Valid values are between and including 0 and 15. Accessing registers outside this range results in an all 0s value being returned on reads, and no action being taken on writes.

The value of the signal lines AD[1:0] must be 00b for MUSYCC to respond. If these bits are zero and the IDSEL* signal line is asserted, then MUSYCC will respond to the configuration cycle.

Although there are two separate configuration spaces, one for Function 0 and one for Function 1, some internal registers are shared between the two spaces.

Registers shared between Function 0 and 1 are marked with an asterisk (*). The Base Code register contains the Class Code, Sub Class Code and Register Level Programming Interface registers.

Table 3. Function 0 Configuration Space

	Register Number	Byte Offset (hex)	31 24	16	8	0
Function Number 0	0	00h	Device Id*		Vendor Id*	
	1	04h	Status		Command	
	2	08h	Base Code			Revision Id*
	3	0Ch	Reserved	Header Type	LatencyTimer	Reserved
	4	10h	MUSYCC Base Address Register (BAR)			
	5	14h	...			
	...	...	Reserved			
	14	38h	...			
	15	3Ch	Max Latency	Min Grant	Interrupt Pin	InterruptLine

Registers shared between Function 0 and 1 are marked with an asterisk (*). The Base Code register contains the Class Code, Sub Class Code and Register Level Programming Interface registers.

Table 4. Function 1 Configuration Space

	Register Number	Byte Offset (hex)	31 24	16	8	0
Function Number 1	0	00h	Device Id*		Vendor Id*	
	1	04h	Status		Command	
	2	08h	Base Code			Revision Id*
	3	0Ch	RSVD	Header Type	Reserved	Reserved
	4	10h	EBUS Base Address Register(BAR)			
	5	14h	...			
	...	...	Reserved			
	14	38h	...			
	15	3Ch	Reserved		Interrupt Pin	InterruptLine

In summary, both configuration spaces have unique registers except for the Device Id, Vendor Id, and Revision Id which are shared between the configuration spaces for Functions 0 and 1.

MUSYCC is a multifunction device with two sources of interrupts: the serial data link interrupts and the expansion bus physical layer interrupts. Since it is not convenient to share interrupt pins for these two sources, MUSYCC supports two separate PCI interrupt pins: INTA* and INTB*.

All writable bits in the configuration space are reset to 0 by the hardware reset, PRST* asserted. After reset, MUSYCC is disabled and will only respond to PCI configuration write and PCI configuration read cycles. Write cycles to reserved bits and registers will have no effect. Read cycles to reserved bits will always result in 0 being read.

PCI Configuration Registers

Function 0 Network Controller—PCI Master and Slave

MUSYCC is a multifunction PCI device that provides the necessary configuration space for a PCI bus controller to query and configure MUSYCC's PCI interface. PCI configuration space consists of a device-independent header region (64 bytes) and a device-dependent header region (192 bytes). MUSYCC provides the device-independent header section only. Access to the device-dependent header region results in 0s being read, and no effect on writes.

Three types of registers are available in MUSYCC:

1. Read-Only (RO)—Return a fixed bit pattern if the register is used or a 0 if the register is unused or reserved
2. Read-Resetable (RR)—Can be reset to 0 by writing a 1 to the register
3. Read/Write (RW)—Retain the value last written to it.

Sixteen dword registers make up MUSYCC's Function 0 PCI Configuration Space. The following tables specify these registers.

Register 0, Address 00h

Table 5. Register 0, Address 00h

Bit Field	Name	Reset Value	Type	Description
31:16	Device Id *	847xh	RO	This unique device ident. is assigned by Brooktree. This field always returns the value 847xh where x can be 2 or 4 depending on the 64- or 128-channel version of the device, respectively.
15:0	Vendor Id *	109Eh	RO	The unique vendor ident. assigned to Brooktree. This field always returns the value 109Eh.

Register 1, Address 04h

The Status register records status information for PCI bus related events. The Command register provides coarse control to generate and respond to PCI commands.

At reset, MUSYCC sets the bits in this register to 0, meaning MUSYCC is logically disconnected from the PCI bus for all cycle types except configuration read and configuration write cycles.

Table 6. Register 1, Address 04h *(1 of 2)*

Bit Field	Name	Reset Value	Type	Description
31	Status	0	RR	Detected Parity Error. This bit is set by MUSYCC whenever it detects a parity error on a data phase when MUSYCC is a target, even if parity error response is disabled.
30		0	RR	Detected System Error. This bit is set by MUSYCC whenever it asserts SERR*.
29		0	RR	Received Master Abort. This bit is set by MUSYCC whenever a MUSYCC-initiated cycle is terminated with master-abort.
28		0	RR	Received Target Abort. MUSYCC sets this bit when a MUSYCC-initiated cycle is terminated by a target-abort.
27		0	RO	Unused.
26:25		01b	RO	DEVSEL Timing. Indicates MUSYCC is a medium-speed PCI device. This means the longest time it will take MUSYCC to return DEVSEL* when it is a target is 3 clocks.
24		0	RR	Data Parity Detected. MUSYCC sets this bit when three conditions are met: 1. MUSYCC asserted PERR* or observed PERR* 2. MUSYCC was the master for that transaction 3. Parity Error Response bit is set (see Command bit-field in this descriptor)
23		1b	RO	Fast Back-to-Back Capable. Read Only. Indicates that when MUSYCC is a target, it is capable of accepting fast back-to-back transactions when the transactions are not to the same agent.
22:16		0	RO	Unused.

Table 6. Register 1, Address 04h (2 of 2)

Bit Field	Name	Reset Value	Type	Description
15:10	Command	0	RO	Unused.
9		0	RW	Fast back-to-back enable. This bit controls whether or not MUSYCC when acting as master, can perform fast back-to-back transactions to different devices. Initialization software will set this bit if all targets in the system are fast back-to-back capable. If 1, MUSYCC can generate fast back-to-back transactions to different agents. If 0, MUSYCC can only generate fast back-to-back cycles to the same agent.
8		0	RW	SERR enable. If 1, disables MUSYCC's SERR* driver. If 0, enables MUSYCC's SERR* driver and allows reporting of address parity errors.
7		0	RO	Wait cycle control. MUSYCC does not support address stepping.
6		0	RW	Parity error response. This bit controls MUSYCC's Function 0 response to parity errors. If 1, MUSYCC takes normal action when a parity error is detected on a cycle with Function 0 as the target. If 0, MUSYCC ignores parity errors.
5		0	RO	VGA palette snoop. Unused.
4		0	RO	Memory write and invalidate. The only write cycle type MUSYCC will generate is memory write.
3		0	RO	Special cycles. Unused. MUSYCC ignores all special cycles.
2		0	RW	Bus master. If 1, MUSYCC is permitted to act as bus master. If 0, MUSYCC is disabled from generating PCI accesses.
1		0	RW	Memory space. Access control. If 1, enables MUSYCC to respond to Function 0 memory space access cycles. If 0, disables MUSYCC's response.
0		0	RO	I/O space accesses. MUSYCC does not contain any I/O space registers.

Register 2, Address 08h

This location contains the Class Code and Revision ID registers. The Class Code register contains the Base Code, Sub Class, and Register Level Programming Interface fields. These are used to specify the generic function of MUSYCC. The Revision ID register denotes the version of the device.

Table 7. Register 2, Address 08h

Bit Field	Name	Reset Value	Type	Description
31:24	Class Code	02h	RO	Function: Network Controller.
23:16	Sub Class Code	80h	RO	Type: Other.
15:8	Register Level Programming Interface	0	RO	Indicates there is nothing special about programming MUSYCC.
7:0	Revision Id *	01h	RO	Denotes the revision number of MUSYCC. Rev A = 01, Rev B = 02.

Register 3, Address 0Ch

Table 8. Register 3, Address 0Ch

Bit Field	Name	Reset Value	Type	Description
31:24	Reserved	0	RO	Unused.
23:16	Header Type	80h	RO	MUSYCC is a multifunction device with the standard layout of configuration register space.
15:11	Latency Timer	0	RW	The latency timer is an 8-bit value that specifies the maximum number of PCI clocks that MUSYCC can keep the bus after starting the access cycle by asserting its FRAME*. The latency timer insures that MUSYCC has a minimum time slot for it to own the bus, but places an upper limit on how long it will own the bus.
10:8		0	RO	
7:0	Reserved	0	RO	Unused.

Register 4, Address 10h

Table 9. Register 4, Address 10h

Bit Field	Name	Reset Value	Type	Description
31:20	MUSYCC - Function 0 Base Address Register	0	RW	Allows for 1 MByte of memory address space to be blocked off as MUSYCC space. MUSYCC will respond as a PCI slave with DEVSEL* to all memory cycles whose address bits 31:20 match the value of bits 31:20 of this register, and those upper address bits are non-zero, and memory space is enabled in the Function 0 Register 1, COMMAND bit field. Reads to addresses within this space that are not implemented will read back 0; writes have no effect.
19:4		0	RO	When appended to bits 31:20, these bits specify a 1 MByte bound memory range. 1 MByte is the only amount of address space that a MUSYCC function can be assigned.
3		0	RO	MUSYCC memory space is not prefetchable.
2:1		0	RO	MUSYCC can be located anywhere in 32-bit address space.
0		0	RO	This base register is a memory space base register, as opposed to I/O mapped.

Register 5–14, Address 14h–38h

Table 10. Register 5-14, Address 14h-38h

Bit Field	Name	Reset Value	Type	Description
31:0	Reserved	0	RO	Unused.

Register 15, Address 3Ch

Table 11. Register 15, Address 3Ch

Bit Field	Name	Reset Value	Type	Description
31:24	Maximum Latency	0Fh	RO	Specifies how quickly MUSYCC needs to gain access to the PCI bus. The value is specified in 0.25 µs increments and assumes a 33 MHz clock. 0Fh means MUSYCC needs to gain access to the PCI bus every 130 PCI clocks, expressed as 3.75 µs in this register.
23:16	Minimum Grant	0	RO	This value specifies, in 0.25 µs increments, the minimum burst period MUSYCC needs. MUSYCC does not have any special MIN_GNT requirements. In general, the more channels MUSYCC has active, the worse the bus latency and the shorter the burst cycle.
15:8	Interrupt Pin	01b	RO	Defines which PCI interrupt pin Function 0 uses. 01h means MUSYCC uses pin INTA* for HDLC controller interrupts.
7:0	Interrupt Line	0	RW	Communicates interrupt line routing. System initialization software will write a value to this register indicating which host interrupt controller input is connected to MUSYCC's INTA* pin.

Function 1 Expansion Bus Bridge, PCI Slave

MUSYCC, a multifunction PCI device, provides the necessary configuration space which allows a PCI bus or system controller to query and configure the host interface of MUSYCC as a PCI device. PCI configuration space consists of a device-independent header region (64 bytes) and a device-dependent header region (192 bytes). MUSYCC provides the 64-byte device-independent header section only. Access to the device-dependent header region will result in 0s being read, and no effect on writes.

Three types of registers are available in MUSYCC:

1. Read-Only (RO)—Return a fixed bit pattern if the register is used or a 0 if the register is unused or reserved
2. Read-Resetable (RR)—Can be reset to 0 by writing a 1 to the register
3. Read/Write (RW)—Retain the value last written to it.

Sixteen dword registers make up MUSYCC's Function 1 Configuration Space. The following tables specify these registers.

Register 0, Address 00h

Table 12. Register 0, Address 00h

Bit Field	Name	Reset Value	Type	Description
31:16	Device Id *	847xh	RO	This unique device ident. is assigned by Brooktree. This field always returns the value 847xh where x can be 2 or 4 depending on the 64- or 128-channel version of the device, respectively.
15:0	Vendor Id *	109Eh	RO	The unique vendor ident. assigned to Brooktree. This field always returns the value 109Eh.

Register 1, Address 04h

The Status register records status information for PCI bus related events. The Command register provides coarse control to generate and respond to PCI commands.

At reset, MUSYCC sets the bits in this register to 0 meaning MUSYCC is logically disconnected from the PCI bus for all cycle types except configuration read and configuration write cycles.

Table 13. Register 1, Address 04h

Bit Field	Name	Reset Value	Type	Description
31	Status	0	RR	Detected parity error. This bit is set by MUSYCC whenever it detects a parity error on a data phase.
30		0	RO	Unused.
29		0	RO	Unused.
28		0	RO	Unused.
27		0	RO	Unused.
26:25		01b	RO	DEVSEL timing. Indicates MUSYCC is a medium-speed device. This means the longest time it will take MUSYCC to return DEVSEL* when the EBUS is the target is 3 clocks.
24		0	RO	Unused.
23		1b	RO	Fast back-to-back capable. Indicates that when the EBUS is a target, it is capable of accepting fast back-to-back transactions when the transactions are not to the same agent.
22:16		0	RO	Unused.
15:7	Command	0	RO	Unused.
6		0	RW	Parity error response. This bit controls MUSYCC's Function 1 response to parity errors. If 1, MUSYCC will take normal action when a parity error is detected on a cycle with Function 1 as the target. If 0, MUSYCC will ignore parity errors.
5:2		0	RO	Unused.
1		0	RW	Memory Space access control. If 1, enables MUSYCC to respond to Function 1 memory space access cycles. If 0, disables MUSYCC's response.
0		0	RO	IO space accesses. MUSYCC does not contain any IO space registers.

Register 2, Address 08h

This location contains the Class Code and Revision ID registers. The Class Code register contains the Base Code, Sub Class, and Register Level Programming Interface fields. These are used to specify the generic function of MUSYCC. The Revision ID register denotes the version of silicon.

Table 14. Register 2, Address 08h

Bit Field	Name	Reset Value	Type	Description
31:24	Class Code	06h	RO	Function: Bridge Device.
23:16	Sub Class Code	80h	RO	Type: Other.
15:8	Register Level Programming Interface	0	RO	Indicates there is nothing special about programming MUSYCC.
7:0	Revision ID*	01h	RO	Denotes the revision number of MUSYCC. Rev A = 01, Rev B = 02.

Register 3, Address 0Ch

Table 15. Register 3, Address 0Ch

Bit Field	Name	Reset Value	Type	Description
31:24	Reserved	0	RO	Unused.
23:16	Header Type	80h	RO	MUSYCC is a multifunction device with the standard layout of configuration register space.
15:0	Reserved	0	RO	Unused.

Register 4, Address 10h

Table 16. Register 4, Address 10h

Bit Field	Name	Reset Value	Type	Description
31:20	EBUS - Function 1 Base Address Register	0	RW	Allows for 1 MByte of memory address space to be blocked off as MUSYCC expansion bus space. MUSYCC will respond as a PCI slave with DEVSEL* to all memory cycles whose address bits 31:20 match the value of bits 31:20 of this register, and bits are non-zero, and memory space is enabled in Function 1 Register 1, bit-field COMMAND. Reads to addresses within this space that are not implemented will read back 0; writes will have no effect. PCI cycles to this space will be mapped to read or write cycles on the expansion bus.
19:4		0	RO	When appended to bits 31:20, specifies a 1 MByte bound memory space. 1 MByte is the only size of address space that a MUSYCC function can be assigned.
3		0	RO	Expansion bus memory space is not prefetchable.
2:1		0	RO	Means MUSYCC expansion bus space can be located anywhere in 32-bit address space.
0		0	RO	Means this base register is a memory space base register, as opposed to I/O mapped.

Register 5 –14, Addresses 14h–38h

Table 17. Register 5 through Register 14–Addresses 14h through 38h

Bit Field	Name	Reset Value	Type	Description
31:0	Reserved	0	RO	Unused.

Register 15, Address 3Ch

Table 18. Register 15, Address 3Ch

Bit Field	Name	Reset Value	Type	Description
31:16	Reserved	0	RO	Unused.
15:8	Interrupt Pin	02h	RO	Defines which PCI interrupt pin Function 1 uses. 02h means MUSYCC uses pin INTB* for interrupts sourced by devices connected to EBUS.
7:0	Interrupt Line	0	RW	Communicates interrupt line routing. System initialization software will write a value to this register indicating which host interrupt controller input is connected to MUSYCC's INTB* pin.

PCI Reset

MUSYCC resets all internal functions when it detects the assertion of the PRST* signal line. Upon reset, the following occurs:

- All PCI output signals are three-stated immediately and asynchronously with respect to the PCI clock input, PCLK.
- All EBUS output signals are three-stated immediately and asynchronously with respect to the EBUS clock output, ECLK.
- All writable/resetable internal register bits are set to 0.
- All PCI data transfers are terminated immediately.
- All serial data transfers are terminated immediately.
- MUSYCC is disabled and responds only to PCI configuration cycles.

Host Interface

After a hardware reset, the PCI configuration space within MUSYCC needs to be configured by the host as follows:

For Function 0:

- Base address register
- Fast back-to-back enable/disable
- SERR* signal driver enable/disable
- Parity error response enable/disable
- Bus mastering enable/disable
- Memory space access enable/disable

For Function 1:

- Base address register
- Parity error response enable/disable
- Memory space access enable/disable

Function 0 provides services to the serial interfaces in MUSYCC. Function 1 provides services to the EBUS interface in MUSYCC.

After the configuration spaces are configured, MUSYCC is able to master the PCI bus when it needs to or provide slave-mode access to the host.

PCI Bus Parity

It is the responsibility of the agent driving the AD[31:0] signals during any given bus phase to also drive the even parity signal (PAR). PAR is driven one clock after AD[31:0] has been driven as follows:

- Address phase: master always drives PAR one clock after address phase.
- Read data phase: target always drives PAR one clock after read data phase.
- Write data phase: master always drives PAR one clock after write data phase.

PAR provides even parity across the AD[31:0] and CBE[3:0]* signal lines. The agent receiving the data is responsible for asserting PERR* if it detects a parity error, provided its Parity Error Response enable bit is set.

If a parity error occurs, it is the responsibility of the master that generated the cycle (whether it asserted PERR* or detected it) to report parity errors to the host. MUSYCC does this by generating an Interrupt Descriptor. It will also set the Data Parity Detected bit (for masters only) in the Status register in the appropriate function's PCI configuration space. It will also set the Detected Parity Error (for masters or targets) in the same register if MUSYCC is the agent that detected the error.

PERR* reports errors on the data phases. MUSYCC not only asserts PERR* when appropriate, but monitors PERR* for its own memory transactions and notifies the host of the parity error.

SERR* reports parity errors on the address phases. It is assumed that this open drain PCI signal is tied directly to the host's system error pin. MUSYCC will not generate an Interrupt Descriptor if it detects a parity error on an address phase, nor does MUSYCC respond to SERR* assertion in any way.

PCI Throughput and Latency Considerations

In PCI systems, achieving high bus throughput works against achieving low bus latency. As devices burst more data they keep the bus longer. Other devices waiting for the bus experience a longer acquisition latency as a result.

A PCI bus master introduces latency each time it uses the PCI bus to perform a transaction. The bus master latency is a function of the following:

- Behavior of the master
  - state of the GNT* signal
  - bus command used (read, write,...)
  - burst length
  - master data latency for each data phase
  - value of Latency Timer
- Behavior of the target
  - bus command used (read, write,...)
  - target latency

When MUSYCC requests the PCI bus, it needs the bus to transfer data between an internal FIFO and shared memory across the PCI bus with either a read or a write access. While MUSYCC waits for the bus to be granted and then additionally while MUSYCC transfers the data, another equally sized internal FIFO is simultaneously being filled or emptied at the serial interface. When MUSYCC requests the bus, it has data to transfer and also it has a finite amount of time [which is directly related to the speed of serial line clock] before a separate FIFO at the serial interface overflows or underflows.

For an application with many logical channels, MUSYCC will require a new access cycle on the PCI bus more often than an application with less logical channels. If FIFO space is evenly distributed across all channels, then more channels will result in less FIFO space per channel - and FIFO space will need to be cleared more frequently.

Conversely, an application with high data rate serial interfaces will require a new access cycle on the PCI bus more often than an application with a low data rate serial interface - FIFO will fill faster in the former.

Acquiring the PCI bus requires having to deal with arbitration latency which is defined as the number of PCI clocks a master must wait after asserting its REQ* and before having the GNT* signal asserted. This number is a function of the arbitration algorithm of the system and takes into account the sequence in which masters are given access to the bus and the latency timer of each master. Arbitration latency is also affected by the loading of the system and how efficiently the bus is being utilized.

The latency timer of a master specifies the maximum number of PCI clocks that the master can [and in the case of MUSYCC, will] keep the bus after starting the access cycle by asserting its FRAME*. The latency timer also insures that the master has a minimum time slot for it to own the bus, but places an upper limit on how long it will own the bus. In MUSYCC, the Latency Timer is reset to 0 on PRST* (PCI reset).

Once the bus is acquired and bursting begins, PCI throughput becomes the point of focus. MUSYCC is capable of multi-dword bursts (read or write). In MUSYCC, as each FIFO for a logical channel and direction is serviced on the PCI, MUSYCC relinquishes and then reacquires the bus to service the FIFO of next logical channel. If more logical channels are serviced, then bus turnover is increased and that effectively decreases throughput (but not necessarily in a service affecting way). If less logical channels are serviced, then bus turnover is decreased and that effectively increases throughput (but not necessarily to the benefit of channel processing).

Reference PCI Local Bus Specification Revision 2.1, June 1, 1995 - Chapter 3 - Bus Operation for an excellent description of bandwidth and latency considerations.

PCI Bus Latency

The latency that a PCI master encounters as it tries to gain access to the PCI bus has three components.

1. Arbitration latency: usually 2 clocks for a high priority device, but is only added into the total latency time if the bus is idle when a device requests it. Otherwise it overlaps with the bus acquisition latency.
2. Bus acquisition latency: length of time a device has to wait for the bus to become free.
3. Target latency: length of time the selected target takes to assert TRDY* for the first data transfer.

It can be shown that the longest latency MUSYCC will experience in gaining access to the PCI bus is:

$$\sum_{i=0}^{k-1} (T_i + 8)$$

or *[k * (T + 8)]* when all T_i's are equal, where *k* is the number of PCI masters in the system and *T* is the value of the latency timers in those masters. The number *8* comes from the restriction that the longest target latency allowed is 8 clocks (with the exception that the first data phase is allowed 16 clocks.)

Once a master gets the bus, it starts counting down a timer loaded with the value T, from the latency timer register. When the count reaches 0, the master must relinquish the bus as soon as its GNT* is removed and it sees TRDY* on the final data phase. As long as its GNT* is still asserted, the master is free to burst indefinitely.

The following illustrates these points:

Assume there are several MUSYCCs and one host in a system with a round robin arbitration scheme and all agents request the bus at the same time.

Table 19. PCI Latency Example

PCI Clock Increment	Bus Activity	
0	Bus is idle. Host asserts REQ*. MUSYCC asserts REQ*.	
+1	Host gets GNT*.	These 2 clocks are the arbitration latency that becomes 0 if the bus was not idle.
+1	Host asserts FRAME* to start access cycle.	
+(T+8) or [16 + (n–1)*8] whichever is smaller - host has bus-	This is the bus acquisition latency time - the amount of time the next requestor must wait for the bus because of current master, the host. During this time, assume the host loses its GNT* just +1 clock into its acquisition and MUSYCC0 receives the GNT* +1 into this time. The host's first data phase must finish within 16 PCI clocks and subsequent data phases must finish within 8 cycles each - therefore, 16+(n–1)*8 clocks is how long the host will need the bus to execute n data phases (n dword burst) assuming the host's access finishes before it's latency timer expires. As the cycle finishes, the host relinquishes the bus and one clock later, MUSYCC0 gets the GNT* and subsequently asserts it's FRAME* to start the access cycle.	
+(T+8) or [16 + (n–1)*8] whichever is smaller - MUSYCC0 has bus -	MUSYCC0 finishes with the bus and MUSYCC1 has it on the next clock. During this time, MUSYCC0 lost its GNT* and MUSYCC1 received its GNT*. MUSYCC0 behaves similarly to the host above.	
+(T+8) or [16 + (n–1)*8] whichever is smaller - MUSYCC 1 has bus -	MUSYCC1 finishes with the bus and MUSYCC2 has it on the next clock. During this time, MUSYCC1 lost its GNT* and MUSYCC2 received its GNT*. MUSYCC1 behaves similarly to the host above.	

The predictable worst case time a MUSYCC will have to wait for the bus in a system with *k* masters with equal latency timers is *[k * (T + 8)]*.

If one MUSYCC is configured with all 128 channels active, receive and transmit, running at 64 Kbps, that single device must maintain a data rate of 16 Mbps across the PCI bus.

- [128 * (64 Kbps Rx + 64 Kbps Tx)] = 16,384 Kbps
- [16,384 Kbps / (32 bits/dword)] = 512 Kdwords/sec (Kdwps)

The "16-clock" rule (PCI Specification, Revision 2.1) requires that a single access device must complete the access cycle within 16 clock periods of the FRAME* signal being asserted. For devices that are capable of burst-mode, the "16-clock" rule applies to the completion of the first data cycle.

Assuming the worst case scenario where the system only allows single dword access, then even a burst-mode device such as MUSYCC must relinquish the PCI bus within 16 clocks from receiving the bus. Using this scenario, the calculations continue:

- [1 / 512 Kdwps] = 1.95 µs per dword
- [1 / 33 MHz] = 30.303 ns
- 1.95 µs per dword /0.0303 µs per clock = 65 PCI clocks between dword transfers.
- With one MUSYCC and one host, the host can use [65 – 16] = 49 clocks.
  MUSYCC's T must be programmed to 49 – 8 target latency = 41.
  As T has a granularity of 8 units, T must be programmed to 40.
- With two MUSYCC and one host, the host can use [65 – (2 * 16)] = 33 clocks.
  The host's T must be programmed to 33 – 8 target latency = 25.
  As T has a granularity of 8 units, T must be programmed to 24.

Assuming...

- MUSYCC has enough internal buffering to buffer up 2 dwords worth of information per channel before performing a 2 dword burst cycle for every access.
- MUSYCC has a granularity of 8 for its latency timer. That is, MUSYCC is always configured to give up the bus in equal to or less than the desired timeout.
- The system will support MUSYCC burst writes and reads.
- MUSYCC, with all 128 receive and transmit channels active needs to move 512 Kdwords/sec, or one dword every 1.953 µs, or 2 dword bursts every 3.906 µs, which is 130 clocks between bursts.

The following is seen...

- The worst case time it would take each burst cycle to finish is 16 + 8 = 24 clocks.
- With one MUSYCC and one host, the host has 130 – 24 = 106 clocks.
  The host's T must be programmed to 106 – 8 = 98 (actually 96).
- For n MUSYCC and one host, the host has [130 – (n * 24) – 8] clocks.
  Hence, for four MUSYCCs and one host, a host's T of 24 would be sufficient, that is, [130 – (4 * 24) – 8] = 26 clocks, actually 24.

On reset, the value of the latency timers are reset to zero.

Expansion Bus (EBUS)

MUSYCC provides a PCI bridge to a local bus interface on MUSYCC called the Expansion Bus (EBUS) which provides a host processor across the PCI bus to access up to 1 Mbyte of peripheral memory space on the EBUS.

Although EBUS utilization is optional, the most notable application for the EBUS is the connection to peripheral devices (e.g. Bt8370 T1/E1 framers) that are local to MUSYCC's serial port. The figures below show block diagrams of the EBUS interface with and without local MPU.

Figure 10. EBUS Functional Block Diagram with Local MPU

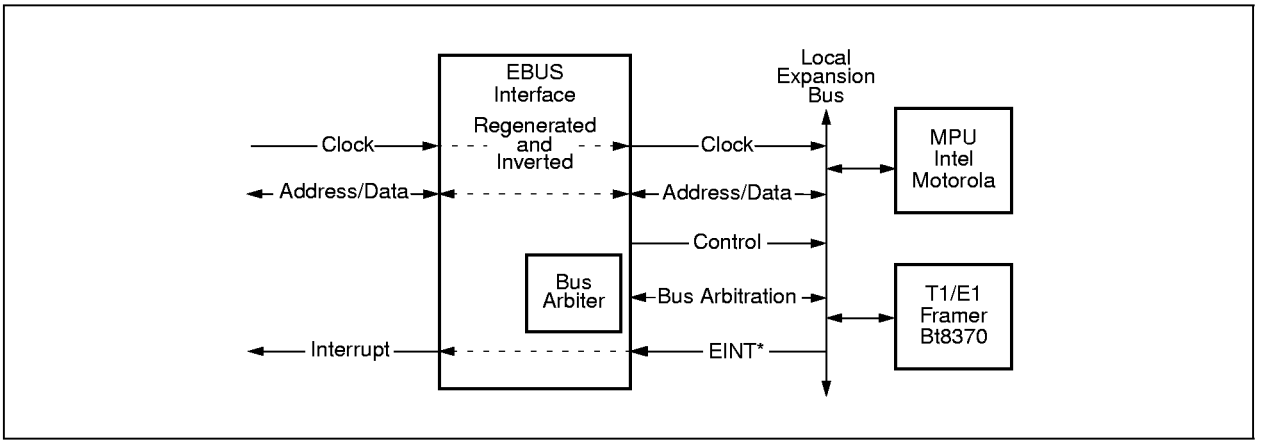
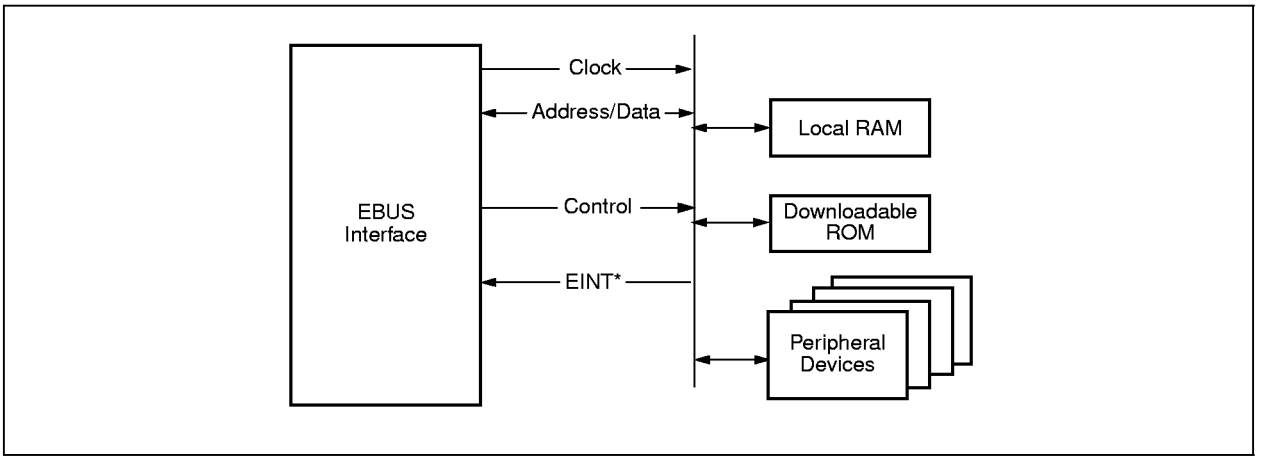


Figure 11. EBUS Functional Block Diagram without Local MPU



Operation

Initialization

At initialization, MUSYCC's PCI Function 1 Configuration Space, Table 16, Register 4, Address 10h, Expansion Bus Base Address Register is initialized with a value representing a 1 Mbyte memory range assigned to MUSYCC's EBUS. An unmapped 1 Mbyte system memory range must be specified by assigning the upper 12 bits of the memory range to the upper 12 bits of this register.

Table 13, Register 1, Address 04h Command bit-field must be properly configured for MUSYCC to respond to EBUS memory space accesses.

On reset, MUSYCC disables EBUS memory space access which results in a PCI master-abort termination if a PCI access of EBUS memory space is attempted.

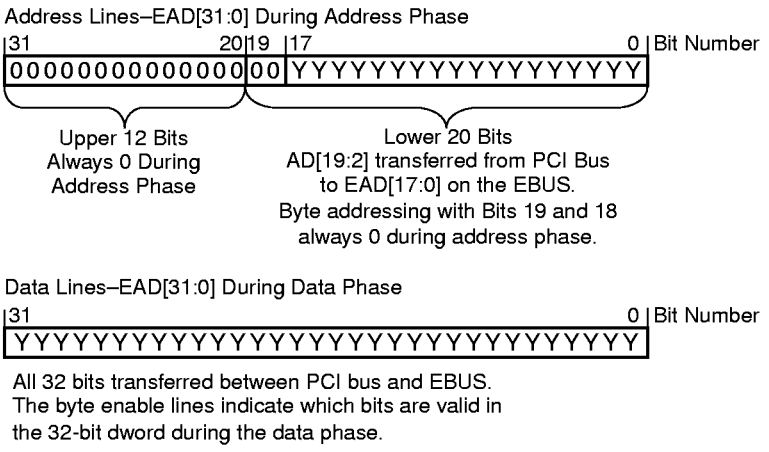
Address and Data

During a PCI access cycle when MUSYCC's host interface claims the cycle, the host interface always compares the upper 12 bits of the PCI address lines to each of its function's base address registers. If signal lines AD[31:20] are identical to the upper 12 bits of the Expansion Bus Base Address Register, MUSYCC forwards the access cycle to the EBUS interface within MUSYCC. It is important to note that only single dword PCI operations can be performed when accessing the EBUS. During the first data phase of any EBUS-addressed PCI operation, MUSYCC asserts both the PCI signal lines TRDY* and STOP* to break up an attempted burst.

Assuming the EBUS is connected to byte-wide peripheral devices, the EBUS interface uses the lower 20 bits from PCI address lines AD[19:0] to construct a byte address for the EBUS. Specifically, PCI address lines AD[19:2] are converted to EBUS address lines EAD[17:0] by shifting out the two least significant bits, AD[1:0]. This allows for byte-level addressing for up to four byte-wide devices on the EBUS. Given the above, the EBUS provides an 18-bit addressing structure which allows byte addressing of up to four banks of 256 Kbytes address space each.

The EBUS interface always transfers 32 bits of the data lines between the EBUS and the PCI bus. The byte-enable signal lines EBE[3:0]* are transferred from the PCI byte-enable signal lines CBE[3:0]* respectively to the EBUS, and indicate which byte(s) in the data dword are valid.

Figure 12. EBUS Address/Data Line Structure



Notes:
1. Byte Enable 0–EBE[0]* signals if EAD[7:0] are valid data bits during data phase.
2. Byte Enable 1–EBE[1]* signals if EAD[15:8] are valid data bits during data phase.
3. Byte Enable 2–EBE[2]* signals if EAD[23:16] are valid data bits during data phase.
4. Byte Enable 3–EBE[3]* signals if EAD[31:24] are valid data bits during data phase.

Clock

The EBUS clock is the logically inverted PCI clock. The signal is output on the ECLK signal line. It should be noted that devices on the EBUS may or may not require a synchronous interface. The ECLK signal is available at all times the PCI clock, PCLK, is available.

The EBUS clock output can be disabled by appropriately setting the ECKEN bit-field in Table 30, Global Configuration Descriptor. In the disabled state, the ECLK output is three-stated.

Interrupt

As a device connected to the EBUS drives the EINT* signal, MUSYCC carries this signal through to the PCI interrupt line, INTB*. Thus, peripheral devices can interrupt the host processor.

In MUSYCC's Function 1 PCI Configuration Space [the EBUS function] the Interrupt Pin bit-field in Table 11, Register 15, Address 3Ch indicates that the INTB* PCI interrupt be asserted for interrupts sourced by devices connected to the EBUS. Also, the Interrupt Line bit-field in the same register is set up by the system initialization software to indicate which host interrupt controller input pin is to be connected to MUSYCC's INTB* pin.

Address Duration

MUSYCC is able to extend the duration that the address bits are valid for any given EBUS address phase. This is accomplished by specifying a value between 0 and 3 in ALAPSE bit-field in Table 30, Global Configuration Descriptor. The value specifies the additional ECLK periods the address bits remain asserted. That is, a value of 0 specifies the address remains asserted for one ECLK period and a value of 3 specifies the address remains asserted for four ECLK periods. Disabling the ECLK signal output does not affect the delay mechanism.

A pre-address and a post-address cycle is always present during the address phase of an EBUS cycle. The pre-address cycle is one ECLK period lone and provides MUSYCC sufficient time to transition between the address phase and the following data phase. The pre- and post- cycles are not included in the Address Duration.

Data Duration

MUSYCC is able to extend the duration that the data bits are valid for any given EBUS data phase. This is accomplished by specifying a value between 0 and 7 in ELAPSE bit-field in Table 30, Global Configuration Descriptor. The value specifies the additional ECLK periods the data bits remain asserted. That is, a value of 0 specifies the data remains asserted for one ECLK period and a value of 7 specifies the data remains asserted for eight ECLK periods. Disabling the ECLK signal output does not affect the delay mechanism.

A pre-data and post-data cycle is always present during the data phase of an EBUS cycle. The pre-data cycle is one ECLK period long and provides MUSYCC sufficient set-up and hold time for the data signals. The post-data cycle is one ECLK period lone and provides MUSYCC sufficient time to transition between the data phase and the following bus cycle termination. The pre- and post- cycles are not included in the Data Duration.

Bus Access Interval

MUSYCC may be configured to wait a specified amount of time after it releases the EBUS and before it requests the EBUS a subsequent time. This is accomplished by specifying a value between 0 and 7 in BLAPSE bit-field in Table 30, Global Configuration Descriptor. The value specifies the additional ECLK periods MUSYCC waits immediately after releasing the bus. That is, a value of 0 specifies MUSYCC will wait for one ECLK period and a value of 5 specifies six ECLK periods. Disabling the ECLK signal output does not affect this wait mechanism.

The bus grant signal (HLDA/BG*) is deasserted by the bus arbiter only after the bus request signal (HOLD/BR*) is deasserted by MUSYCC. As the amount of time between bus request deassertion and bus grant deassertion can vary from system to system, it is possible for a misinterpretation of the "old" bus grant signal as an approval to access the EBUS. MUSYCC provides the flexibility through the bus access interval feature to wait a specific number of ECLK periods between subsequent bus requests.

Reference EBUS timing diagrams - Figure 34, EBUS Write/Read Cycle, Intel-Style and Figure 35, EBUS Write/Read Cycle, Motorola-Style.

PCI to EBUS Interaction

Using the EBUS to perform extensive polling of peripheral devices will substantially increase PCI bus utilization. The EBUS interface within MUSYCC performs single dword access without burst cycles. Also, the access time for data on the EBUS is dependent on how fast the peripherals can respond to an EBUS read or write cycle.

PCI write access cycles targeted at the EBUS are not at issue as they complete immediately. MUSYCC's host interface autonomously manages the completion of writing data to the EBUS after successfully terminating the host's PCI write access cycle.

PCI read access cycles targeted at the EBUS are at issue as they cause MUSYCC's host interface to first claim the access cycle then immediately initiate a PCI Target Retry sequence which causes the PCI bridge device to retry the same EBUS access at a later time. Concurrently, the EBUS interface is activated to access the requested data from the EBUS. As this process may take many EBUS clock cycles to complete, the host interface is capable of holding off each retry request by initiating a subsequent Target Retry sequence until the EBUS interface delivers the required data to the host interface. Target Retry sequences may occur multiple times.

As EBUS data is eventually made available to the host interface- and on the next retry from the bridge chip, the host interface checks if the retry cycle address matches the address latched in during the initial EBUS access cycle and if so forwards the EBUS data to the requester. If the addresses do not match, then MUSYCC starts a new EBUS access cycle.

The amount of time to complete a single EBUS cycle accessing a single dword at a time and the number of bus turnovers between successive retries all add up to affect PCI bus utilization. To avoid affecting the PCI bus adversely, systems need to be designed to throttle EBUS access and/or use a local microprocessor on the EBUS to filter the information from peripheral devices.

Microprocessor Interface

The MPUSEL bit-field in Table 30, Global Configuration Descriptor specifies the type of microprocessor interface to use for the EBUS.

If Intel-style protocol is selected, the following signals are effective:

- ALE*, Address Latch Enable, asserted low by MUSYCC to indicate that the address lines contain a valid address. This signal remains asserted for the duration of the access cycle.
- RD*, Read, strobed low by MUSYCC to enable data reads out of the device and is held high during writes.
- WR*, Write, strobed low by MUSYCC to enable data writes into the device and is held high during reads.
- HOLD, Hold Request, asserted high by MUSYCC when it requests the EBUS from a bus arbiter.
- HLDA, Hold Acknowledge, asserted high by bus arbiter in response to HOLD signal assertion. Remains asserted until after the HOLD signal is deasserted. If the EBUS is connected and there are no bus arbiters on the EBUS, then this signal must be asserted high at all times.

If Motorola-style protocol, the following signals are effective:

- AS*, Address Strobe, driven low by MUSYCC to indicate that the address lines contain a valid address. This signal remains asserted for the duration of the access cycle.
- DS*, Data Strobe, strobed low by MUSYCC to enable data reads or data writes for the addressed device.
- R/WR*, Read/Write, held high throughout read operation and held low throughout write operation by MUSYCC. This signal determines the meaning (read or write) of DS*.
- BR*, Bus Request, asserted low by MUSYCC when it requests the EBUS from a bus arbiter.
- BG*, Hold Acknowledge, asserted low by bus arbiter in response to BR* signal assertion. Remains asserted until after the BR* signal is deasserted. If the EBUS is connected and there are no bus arbiters on the EBUS, then this signal must be asserted low at all times.
- BGACK*, Bus Grant Acknowledge, asserted low by MUSYCC when it detects BGACK* currently deasserted. As this signal is asserted, MUSYCC begins the EBUS access cycle. After the cycle is finished, this signal is deasserted indicating to the bus arbiter that MUSYCC has released the EBUS.

Arbitration

The HOLD and HLDA (Intel) or BR* and BG* (Motorola) signal lines are used by MUSYCC to arbitrate for the EBUS.

For Intel-style interfaces, the arbitration protocol is as follows:
Reference Figure 34, EBUS Write/Read Cycle, Intel-Style.

- MUSYCC three-states EAD[31:0], EBE*[3:0]. WR*, RD*, and ALE*.
- MUSYCC requires EBUS access and asserts HOLD.
- MUSYCC checks for HLDA assertion by bus arbiter.
- If HLDA is found to be deasserted, MUSYCC waits for the HLDA signal to become asserted before continuing the EBUS operation.
- If HLDA is found to be asserted, MUSYCC continues with the EBUS access as it has control of the EBUS.
- MUSYCC drives EAD[31:0], EBE*[3:0], WR*, RD*, and ALE*.
- MUSYCC completes EBUS access and deasserts HOLD.
- Bus arbiter deasserts HLDA shortly thereafter.
- MUSYCC three-states EAD[31:0], EBE*[3:0]. WR*, RD*, and ALE*.

For Motorola-style interfaces, the arbitration protocol is as follows: Reference Figure 35, EBUS Write/Read Cycle, Motorola-Style.

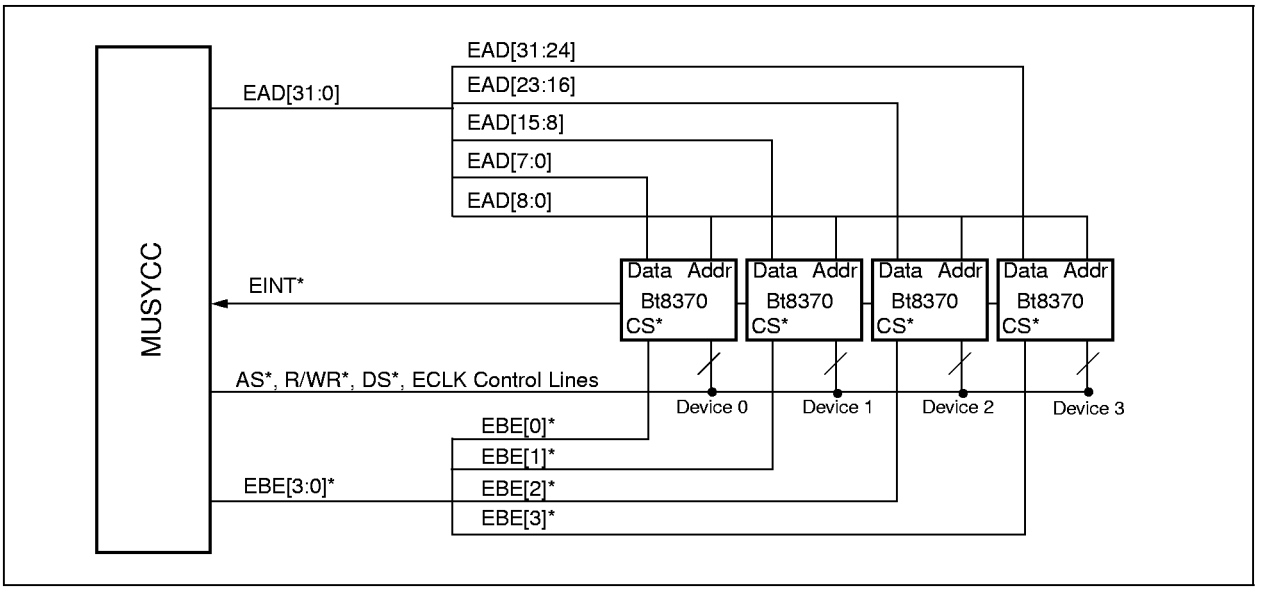
- MUSYCC three-states EAD[31:0], EBE*[3:0]. R/WR*, DS* and AS*.
- MUSYCC requires EBUS access and asserts BR*.
- MUSYCC checks for BG* assertion by bus arbiter.
- If BG* is found to be deasserted, MUSYCC waits for the BG* signal to become asserted before continuing the EBUS operation.
- If BG* is found to be asserted, MUSYCC continues with the EBUS access as it has control of the EBUS.
- If BGACK* is not asserted MUSYCC will assume control of the EBUS by asserting BGACK*.
- MUSYCC drives EAD[31:0], EBE*[3:0], R/WR*, DS*, AS*.
- Shortly after the EBUS cycle is started, MUSYCC deasserts BR*.
- Bus arbiter deasserts BG* shortly thereafter.
- MUSYCC completes EBUS cycle.
- MUSYCC deasserts BGACK*.
- MUSYCC three-states EAD[31:0], EBE*[3:0]. R/WR*, DS* and AS*.

Connection

Utilizing the EBUS address lines, EAD[17:0], and the byte enable lines, EBE[3:0]*, the EBUS can be connected in either a multiplexed or non-multiplexed address and data mode.

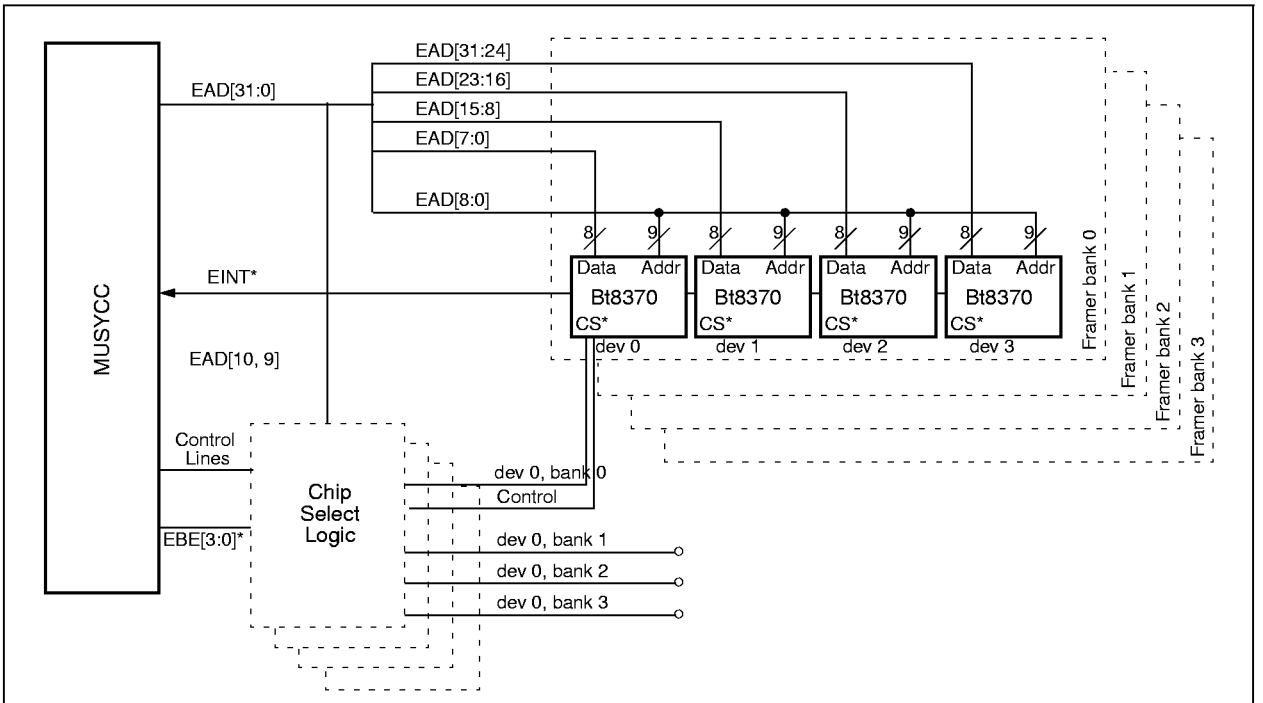
The figures below show two examples of non-multiplexed address and data modes - one shows four separate byte-wide framer devices connected to the EBUS with each byte enable line used as the chip select for separate devices. which allows a full dword data transfer over the EBUS.

Figure 13. EBUS Connection, Non-multiplexed Address/Data, 4 Framers, No Local MPU



The next figure illustrates how additional address lines can be combined with each byte enable line during the address phase to support multiple framer banks with each bank containing four byte-wide framer devices.

Figure 14. EBUS Connection, Non-multiplexed Address/Data, 16 Framers, No Local MPU

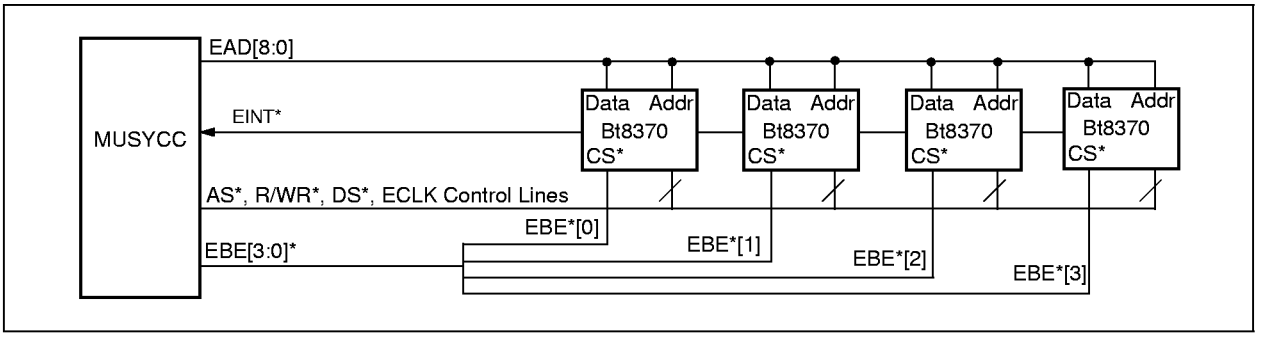


Notes: 1. EBEx[3:0]* selects device x in each framer block.
2. EAD[31:0], AS* are supplied to each framer block.
3. EBEx*, AS* are supplied to each chip select block.

In the multiplexed address and data mode, four byte-wide peripheral devices are connected to the EBUS. In this mode, 8 bits of the 32-bit EBUS transfer data to and from each device individually.

NOTE: The multiplexed address and data mode example does not allow for 4-byte data transfers.

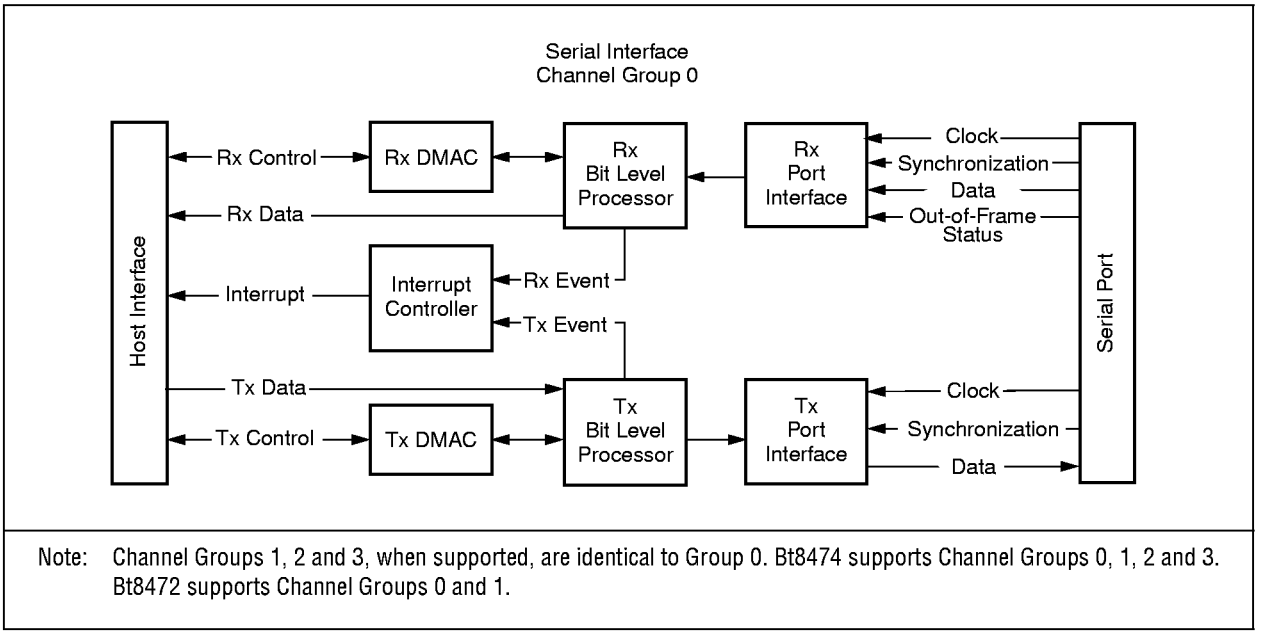
Figure 15. EBUS Connection, Multiplexed Address/Data, 4 Framers, No Local MPU



Serial Interface

This chapter describes the features of the serial port interface. Each serial interface consists of Serial Port Interfaces (SERI), Bit Level Processors (BLP), Direct Memory Access Controllers (DMAC) and an Interrupt Controller (INTC). A separate set of SERI, BLP, and DMAC services receive channels and transmit channels independently. A single INTC is shared by the receive and transmit BLP.

Figure 16. Serial Interface Functional Block Diagram, Channel Group 0



Note: Channel Groups 1, 2 and 3, when supported, are identical to Group 0. Bt8474 supports Channel Groups 0, 1, 2 and 3. Bt8472 supports Channel Groups 0 and 1.

Serial Port Interface

A receive serial port interface (Rx-SERI) connects to four input signals: RCLK, RDAT, RSYNC, and ROOF. A transmit serial port interface (Tx-SERI) connects to two input signals and one output signal: TCLK, TSYNC, and TDAT, respectively. Reference Table 2, Bt8474/8472 Hardware Signal Definitions. The SERI is responsible for receiving and transmitting data bits to FIFOs in the BLP.

The receive and transmit data and synchronization signals are synchronous to the receive and transmit line clocks, respectively. MUSYCC can be configured to sample in and latch out data signals and sample in status and synchronization signals on either the rising or falling edges of the respective line clock, namely RCLK and TCLK. This configuration is accomplished by setting the ROOF_EDGE, RSYNC_EDGE, RDAT_EDGE, TSYNC_EDGE, and TDAT_EDGE bit-fields in Table 36, Port Configuration Descriptor.

The default, after device reset, is to sample in and latch out data, synchronization and status on the falling edges of the respective line clock.

Bit Level Processor

The bit level processors (Rx-BLP and Tx-BLP) service the bits in the receive and transmit transmission path. As internal FIFOs are filled and flushed, the BLP requests memory transfers from the DMAC. The BLP coordinates all bit-level transactions between SERI and DMAC. The BLP also interacts with the INTC to notify the host of events and errors during bit-level processing.

Direct Memory Access Controller

The direct memory access controllers (Rx-DMAC and Tx-DMAC) manage all of the memory operations between a corresponding BLP and the host interface. DMAC takes requests from BLP to either fill or flush internal FIFOs, sets up an access to data buffers in shared memory and then requests access to the PCI bus through the host interface.

Interrupt Controller

The interrupt controller takes receive and transmit events from Rx-BLP and Tx-BLP, respectively. The INTC coordinates the transfer of internally queued descriptors to an interrupt queue in shared memory and also coordinates the notification to the host of pending interrupts.

Channelized Port Mode

Each SERI can be configured independently using the PORTMD bit-field in Table 36, Port Configuration Descriptor.

Channelized mode refers to a data bit stream that is segmented into frames. Each frame consists of a series of 8-bit timeslots. Typically, each timeslot recurs every 125 µs at an 8 KHz rate. MUSYCC maintains frame synchronization in both the transmit and receive directions by using the TSYNC and RSYNC input signals. In addition, the ROOF input signal may be used to notify MUSYCC of the loss of frame synchronization.

The table below describes the contents of a typical 8 KHz frame in each of the possible channelized port modes.

Table 20. Channelized Serial Port Modes

Mode	Clock Frequency	Bits per Frame	Description
T1	1.544 MHz	193	Single frame bit, followed by 24 timeslots, numbered TS0–TS23.
E1	2.048 MHz	256	32 timeslots, numbered TS0–TS31.
2 E1	4.096 MHz	512	64 timeslots, numbered TS0–TS63.
4 E1	8.192 MHz	1024	128 timeslots, numbered TS0–TS127.
Nx64	Nx64 KHz ($1 \le N \le 128$)	Nx8 ($1 \le N \le 128$)	N timeslots, numbered TS0–TSN-1.

Hyperchannels (Nx64)

A hyperchannel results from concatenating bits from one or more 8-bit timeslots within a frame. A hyperchannel can be comprised of between and including 1 and 128 timeslots. This results in one logical channel supporting an Nx64 Kbps bit rate where the actual data rate can range between 64 Kbps and 8.192 Mbps. The concatenated timeslots need not be contiguous. The following configuration is required to support hyperchannels:

- Hyperchanneled timeslots assigned to the same logical channel number within a channel group (0–31).

Table 39, Timeslot Descriptor enables and assigns a timeslot to a logical channel. The configurations for receive and transmit hyperchannels are independent.

Subchannels (Nx8)

A subchannel results from treating each bit in an 8-bit timeslot independently and assigning a logical channel number to each active bit. Not all 8 bits need to be active and any combination of bits within the 8 in a timeslot can be assigned to the same logical channel number. Similarly, multiple timeslots can supply one or more bits to comprise one subchannel. This results in one logical channel supporting an Nx8 bit rate between 8 Kbps to 64 Kbps in multiples of 8 Kbps. The following configurations are required to support subchannels:

- Each active bit is assigned a logical channel number within a channel group (0–31).
- Each timeslot with active bits must be enabled in the Timeslot Map.
- Each active bit (after the first bit, Bit-0) must be enabled in the Subchannel Map.

Table 39, Timeslot Descriptor and Table 41, Subchannel Descriptor enables and assigns a timeslot and each individual bit within the timeslot to a logical channel. The configurations for receive and transmit subchannels are independent.

The timeslot descriptor assigns bit 0 of a timeslot to a logical channel. The subchannel descriptor assigns bits 1 through 7 of a timeslot to a logical channel.

Frame Synchronization Flywheel

MUSYCC utilizes the TSYNC and RSYNC signals to maintain a timebase which keeps track of the active bit in the current timeslot. The mechanism is referred to as the frame synchronization flywheel. The flywheel counts the number of bits per frame and automatically rolls over the bit count according to the programmed mode. The TSYNC or RSYNC input marks the first bit in the frame. The mode specified in the PORTMD bit-field in Table 36, Port Configuration Descriptor determines the number of bits in the frame. A flywheel exists for both the transmit and the receive functions for every port.

The flywheel is synchronized when MUSYCC detects TSYNC = 1 or RSYNC = 1, for transmit or receive functions, respectively. Once synchronized, the flywheel maintains synchronization without further assertion of the synchronization signal.

A timeslot counter within each port is reset at the beginning of each frame and tracks the current timeslot being serviced.

For the Nx64 mode, the value of N cannot be specified and, hence, the data requires a synchronization pulse every frame period to reset the flywheel. Also, in Nx64 mode the TSYNC must precede the output of bit 0 of the frame by four line clock periods.

Figure 39, Transmit and Receive T1 Mode and Figure 40, Transmit and Receive E1 (also 2xE1, 4xE1) Mode and Figure 41, Transmit and Receive Nx64 Mode show the timing relationships between the data and the synchronization signal for various modes of operation.

Change Of Frame Alignment (COFA)

A COFA condition is defined as a frame synchronization event detected when it was not expected, and also includes the detection of the first occurrence of frame synchronization when none was present.

When a COFA condition is detected by the serial interface, an internal COFA signal is asserted for one frame period. During the frame period that the internal COFA is asserted, MUSYCC's channel group processor terminates all messages which are found to be active during the channel map processing. For each receiver channel found to be active and processing a message during the channel map processing, the corresponding message descriptor's owner bit is returned to the host and a Buffer Status Descriptor is written with the COFA error encoding. The Buffer Status Descriptor is written if configured to do so in the Group Configuration Descriptor. MUSYCC then proceeds to the next message descriptor in the list of messages.

When the internal COFA is deasserted, MUSYCC generates an Interrupt Descriptor with the COFA error encoding if the interrupt is not masked in the Group Configuration Descriptor. If a synchronization signal is received (low-to-high transition on TSYNC or RSYNC) while the internal COFA is asserted, an Interrupt Descriptor with the COFA interrupt encoding is generated immediately if this interrupt is not masked. COFA detection is not applicable to the Nx64 channelized mode.

Out Of Frame (OOF)

The Receiver Out-Of-Frame (ROOF) signal is asserted by the physical T1 or E1 interface sourcing the channelized data to MUSYCC. This signal indicates that the interface device has lost frame synchronization.

In the case of multiplexed E1 lines (2xE1 or 4xE1), the ROOF input signal on a given port may be asserted and deasserted as timeslots are received from an out-of-frame E1 followed by an in-frame E1.

The state of ROOF is evaluated on a bit-by-bit basis when processing data from a timeslot. When ROOF assertion is detected by the receiver serial interface, MUSYCC checks the OOFABT bit in the Group Configuration Descriptor. If the OOFABT bit is set (1), MUSYCC terminates any active messages for all mapped and active channels in the channel group. If the OOFABT bit is not set (0), MUSYCC continues to process the received data but still asserts the OOF Interrupt Descriptor unless it is masked.

For each receive message that was terminated during the OOF condition, the corresponding Message Descriptor's owner bit is returned to the host and a Buffer Status Descriptor is written with the OOF error encoding. The Buffer Status Descriptor is written to host memory only if configured to do so on a per group basis in the Group Configuration Descriptor.

MUSYCC then proceeds to the next Message Descriptor in the list of messages. Two frame synchronization events (via external sync or flywheel sync) after ROOF is asserted, MUSYCC generates an interrupt descriptor with the Out Of Frame (OOF) error encoding if the interrupt is not masked in the Group Configuration Descriptor.

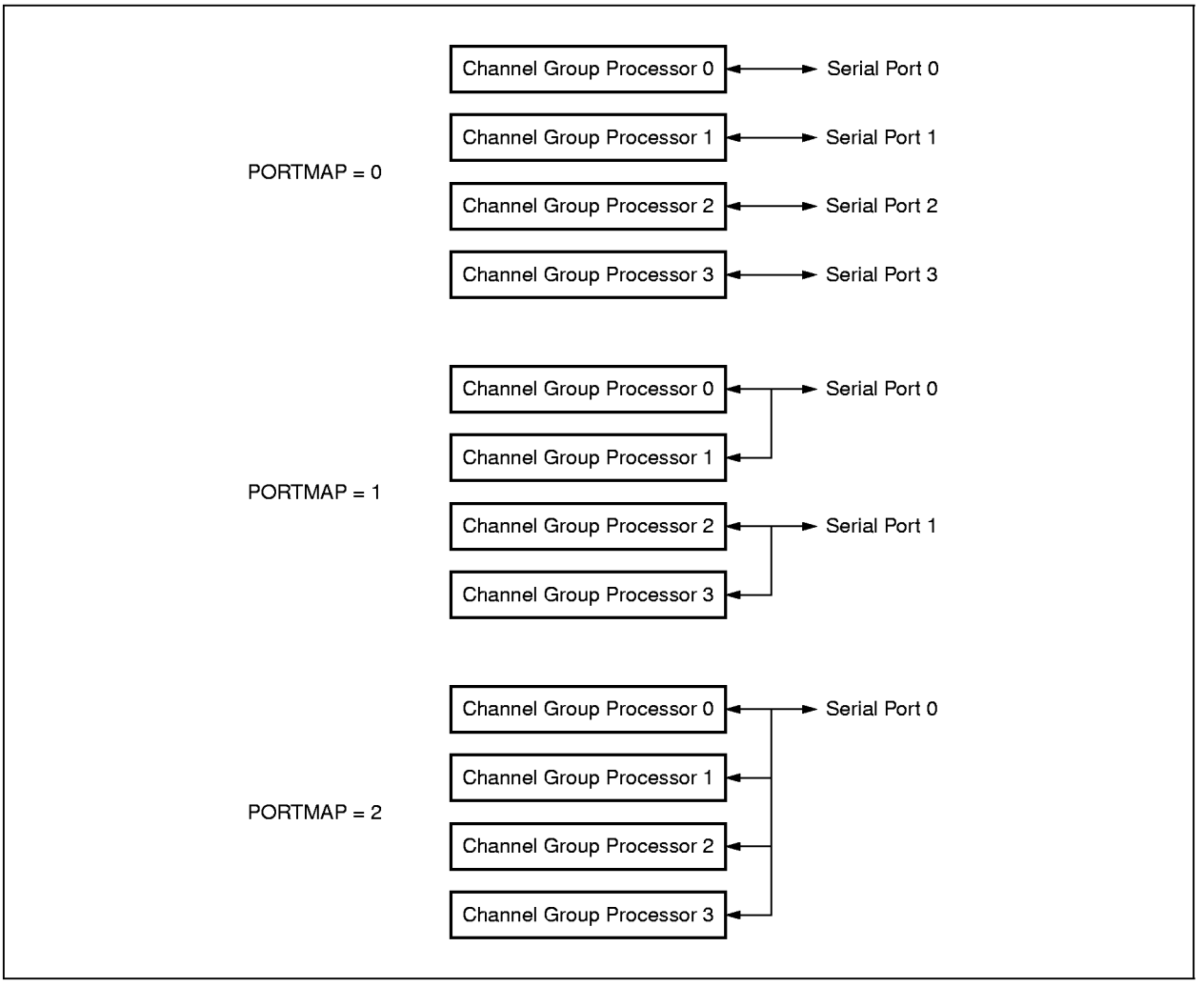
As ROOF is deasserted, MUSYCC immediately restarts normal bit level processing on all mapped and active channels. Two frame synchronization events after deassertion of ROOF is detected, MUSYCC generates an interrupt descriptor with the FREC (Frame Recovery) interrupt encoding if the interrupt is not masked in Table 34, Group Configuration Descriptor.

Serial Port Mapping

MUSYCC contains up to four serial ports with each port associated to a channel group processor that supports up to 32 logical bidirectional channels.

To manage more than 32 logical channels on a single port, the port configuration options provided in the PORTMAP bit-field in Table 30, Global Configuration Descriptor can be programmed to route the signal on one port to multiple channel groups.

Figure 17. Serial Port Mapping Options



The following port mappings are available:

- PORTMAP = 0, 1x port mode.
  Default mode after device reset. Each serial port is logically connected to one channel group and each port terminates up to 32 bidirectional channels.
- PORTMAP = 1, 2x port mode.
  Each of two serial ports is logically connected to two channel groups. In this mode, serial port 0 is connected to channel groups 0 and 1; and, serial port 1 is connected to channel groups 2 and 3. Each serial port terminates up to 64 bidirectional channels.
- PORTMAP = 2, 4x port mode.
  Serial port 0 is logically connected to channel groups 0, 1, 2, and 3. Serial ports 1, 2, and 3 are disabled.

Mapping a serial port to logical channel(s) in a channel group is one part of serial port configuration. Another important part of the configuration is indicating the channelized data rate of the serial interface. This is accomplished by configuring the PORTMD bit-field in Table 36, Port Configuration Descriptor. Additionally, the connection between each of the serial ports and the physical layer interface indicates the relationship to physical layer timeslots.

Each serial port may be configured to support 24, 32, 64, 128 or a variable number of 8-bit timeslots up to 128. Each serial port may be configured to support data rates up to and including 8.192 Mbps. Also, each serial port may be independently wired to a separate source of serial data or all four serial ports may be wired to a single source of serial data.

Note that MUSYCC always allows PORTMAP to be set to 0, 1 or 2, but only the connections to Channel Groups 0 and 1 are implemented in Bt8472. Whereas, Channel Groups 0, 1, 2 and 3 are implemented in Bt8474. Hence, PORTMAP=1 and PORTMAP=2 are identical for the 64-channel Bt8472 device.

Transmit and Receive FIFO Allocation and Management

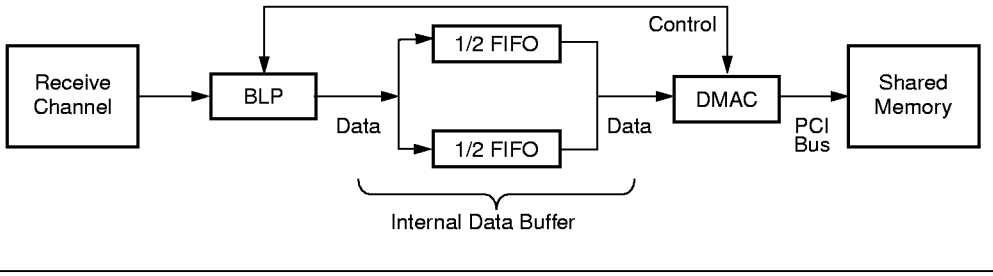
Each channel group contains a separate internal buffer memory space for transmit and receive operations. Within each of these space, separate areas are set aside for specific functions.

Table 21. Internal Buffer Memory Layout

Memory Area	Transmit	Receive
Fixed Data Buffer	64	64
Subchannel Map (or Additional Data Buffer if no Subchanneling)	64	64
Timeslot Map	32	32
Total	160 dwords 640 bytes	160 dwords 640 bytes

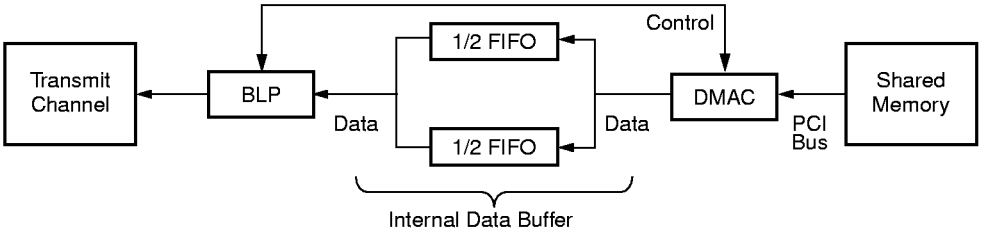
Each channel within the group must be allocated buffer space before the channel can be activated. This space acts as a holding buffer for incoming (Rx) and outgoing (Tx) data. Data buffers for each channel are allocated using the BUFFLOC and BUFFLEN bit-fields in Table 42, Channel Configuration Descriptor. Both receiver and transmitter of a channel use a data buffer scheme where half of the available FIFO services the serial interface, and the other half services data in shared memory. BUFFLEN+1 specifies half the size of the buffer space allocated to a direction of the channel.

Figure 18. Receive Data Flow



Note: 1/2 FIFO = BUFFLEN+1

Figure 19. Transmit Data Flow



Note: 1/2 FIFO = BUFFLEN+1

The allocation of internal buffers requires an understanding of the available FIFO space and configuration of channels and subchannels within a channel group.

Other important considerations to keep in mind for using the data buffer scheme include PCI bus latency, data rate of the logical channels, and the number of supported channels per channel group.

Example Channel BUFFLOC and BUFFLEN Specification

With some subchanneling, only the Fixed Data Buffer (total of 64 dwords) is the area available for Internal Data Buffer usage. If the buffer space is evenly divided across 32 channels, the BUFFLOC and BUFFLEN specifications would be as described in Table 42, Channel Configuration Descriptor

Table 22. Example 32-Channel w/Subchanneling Buffer Allocation (Receive or Transmit)

Channel Number	Within Channel Descriptor	
	BUFFLOC (dword Offset From Start Of Fixed Data Buffer)	BUFFLEN(1)
0	0	0
1	1	0
2	1	0
...	...	...
31	31	0
Notes: (1). BUFFLEN = [(Total dwords ÷ Number Of Channels) ÷ 2] - 1		

With no subchanneling, the Fixed Data Buffer area plus the Subchannel Map area are available for Internal Data Buffer usage (total of 128 dwords). If the buffer space is evenly divided across 32 channels, the BUFFLOC and BUFFLEN specification would be as described in the table below.

Table 23. Example 32-Channel w/o Subchanneling Buffer Allocation (Receive or Transmit)

Channel Number	Within Channel Descriptor	
	BUFFLOC (dword Offset From Start Of Fixed Data Buffer)	BUFFLEN(1)
0	0	1
1	2	1
2	4	1
...	...	...
31	62	1
Notes: (1). BUFFLEN = [(Total dwords ÷ Number Of Channels) ÷ 2] - 1		

If the buffer space is evenly divided across 16 channels, the BUFFLOC and BUFFLEN specification would be as described in the table below.

Table 24. Example 16-Channel w/o Subchanneling Buffer Allocation (Receive or Transmit)

Channel Number	Within Channel Descriptor	
	BUFFLOC (dword Offset From Start Of Fixed Data Buffer)	BUFFLEN(1)
0	0	3
1	4	3
2	8	3
...	...	...
15	60	3
Notes: (1). BUFFLEN = [(Total dwords ÷ Number Of Channels) ÷ 2] - 1		

Receiving Bit Stream

As a receive channel is activated, MUSYCC reads in descriptors from shared memory and prepares Rx-BLP and Rx-DMAC to service incoming serial data accordingly. This discussion assumes all configuration is proper and incoming data can be written to shared memory.

Upon channel activation, the receiver starts storing received data into a BUFFLEN+1 size of FIFO starting at BUFFLOC offset in the FIFO area. As this buffer is filled, the BLP instructs the DMAC to start a PCI data transfer cycle to shared memory of the FIFO contents and simultaneously starts filling another BUFFLEN+1 size of FIFO from the serial port. Generally, half the FIFO space for a channel is used for serial port data reception while the other half is used for shared memory data transfers.

The DMAC–initiated PCI transfer cycle requires MUSYCC to arbitrate for the PCI bus, initiate a master-write to shared memory over the PCI bus, and conclude the transfer by releasing the PCI bus. MUSYCC transfers data autonomously and always attempts to burst data to the PCI.

Transmitting Bit Stream

As a transmit channel is activated, MUSYCC reads in descriptors from shared memory and prepares Tx-BLP and Tx-DMAC to service outgoing serial data accordingly. This discussion assumes all configuration is proper and outgoing data can be read from shared memory.

Upon channel activation, the transmitter initiates a PCI data transfer cycle from shared memory of data to be output to the serial port. As the DMAC receives data over the PCI, it forwards it to the BLP which fills a BUFFLEN+1 size of FIFO starting at BUFFLOC offset in the FIFO area. Generally, half the FIFO space for a channel is used for serial port data transmission while the other half is used for shared memory data transfers.

The DMAC-initiated PCI transfer cycle requires MUSYCC to arbitrate for the PCI bus, initiate a master-read from shared memory over the PCI bus, and conclude the transfer by releasing the PCI bus. MUSYCC transfers data autonomously and always attempts to burst data from the PCI.

Transmit Data Bit Output Value Determination

The TDAT signal from MUSYCC is the only output signal in the Serial Interface. For every bit-time specified by the TCLK input signal to MUSYCC and on every active edge for a data bit specified by the TDAT_EDGE bit-field in Table 36, Port Configuration Descriptor, a value for the TDAT bit must be determined and output. The following figure describes the logic used to determine the output value.

Figure 20. Transmit Data Bit Output Value Determination

```
IF ( TRANSMITTER_NOT_ENABLED )(1)
    TDAT <= THREE-STATE
ELSE
    IF ( CHANNEL_IS_MAPPED )(2)
        IF ( CHANNEL_IS_ACTIVATED )(3)
            TDAT = BLP_OUTPUT(4)
        ELSE
            TDAT = 'LOGIC 1'
    ELSE
        IF ( THREE_STATE_OUTPUT )(5)
            TDAT = THREE-STATE
        ELSE
            TDAT = 'LOGIC 1'
```

Notes:
(1). TRANSMITTER_NOT_ENABLED. Check TXENBL bit-field in Table 34, Group Configuration Descriptor).
(2). CHANNEL_IS_MAPPED. Verify channel to timeslot mapping enabled in Figure 38, Transmit or Receive Timeslot Map.
(3). CHANNEL_IS_ACTIVATED. Verify Channel Activate Service Request issued.
(4). BLP_OUTUT. Data taken from shared memory, through the internal FIFO and ready for transmission.
(5). THREE_STATE_OUTPUT. Check TRITX bit-field in Port Configuration Descriptor.

Memory Organization

MUSYCC interfaces with a system host using a set of data structures located in a shared memory region. MUSYCC also contains a set of internal registers which the host can configure and control the MUSYCC. This section describes the various shared memory data structures and the layout of individual registers which are required for the operation of MUSYCC.

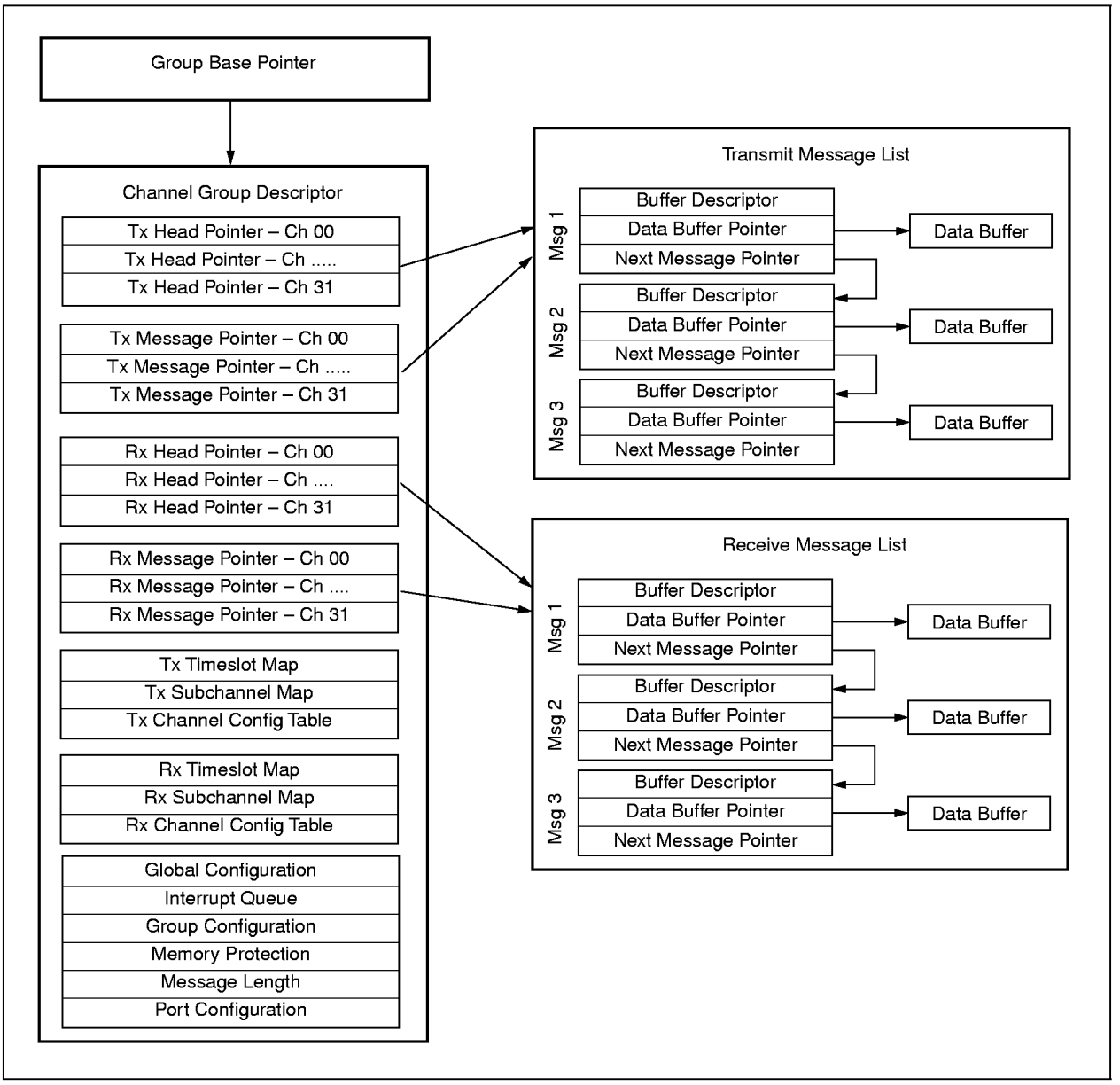
Memory Architecture

MUSYCC supports a descriptor-based memory architecture wherein data is continually moved into and out of a linked list of data buffers in shared memory for each active channel. This assumes a system topology in which a host and MUSYCC both have access to shared memory for data control and data flow. The data structures are defined in a way that the control structures and the data structures may or may not reside in the same physical memory and may or may not be contiguous. The host allocates and deallocates the required memory space as well as the size and number of data buffers within that space.

Different versions of MUSYCC support different numbers of channel groups. The host allocates shared memory regions to configure and control each group.

The figure below illustrates the memory model used by MUSYCC for control and data structures required for each supported channel group.

Figure 21. Shared Memory Model Per Channel Group



Register Map Access and Shared Memory Access

During MUSYCC's PCI initialization, the system controller allocates a dedicated 1 Mbyte memory range to each of MUSYCC's PCI functions. The memory range allocated to MUSYCC must not map to any other physical or shared memory. Instead, the system configuration manager allocates a logical memory address range and notifies the system or bus controllers that any access to these ranges must result in a PCI access cycle. MUSYCC is assigned these address ranges for each function through the PCI configuration cycle. Once configured, MUSYCC becomes a functional PCI device on the bus.

As the host accesses MUSYCC's allocated address ranges, the host initiates the access cycles on the PCI bus. It is up to individual MUSYCC devices on the bus to claim the access cycle. As MUSYCC's address ranges are accessed, it behaves as a PCI slave device while data is being read or written by the host. MUSYCC responds to all access cycles where the upper 12 bits of a PCI address match the upper 12 bits of either the EBUS Base Address Register (Function 1) or the MUSYCC Base Address Register (Function 0).

For MUSYCC's Function 1, a 1 Mbyte memory space is assigned to the EBUS Base Address Register which is written into Function 1 PCI Configuration Space in Table 16, Register 4, Address 10h. Devices connected to the EBUS can then be allocated memory addresses within this 1 Mbyte memory range. If MUSYCC claims a PCI access cycle for Function 1, MUSYCC initiates EBUS arbitration and ultimately accesses data from a device connected to the EBUS.

For MUSYCC's Function 0, a 1 Mbyte memory space is assigned to the MUSYCC Base Address Register which is written into Function 0 PCI Configuration Space in Table 9, Register 4, Address 10h. Once a base address is assigned to Function 0, a register map is used to access individual device resident registers. The register map provides the byte offset from the base address register where registers reside. The register map layout is given in Table 25, MUSYCC Register Map.

Note that the 1 Mbyte memory ranges assigned to MUSYCC functions will not restrict MUSYCC's PCI interface from attempting to access these same ranges. The host must be cognizant that MUSYCC cannot respond to an access cycle which MUSYCC itself initiates as the bus master.

Table 25. MUSYCC Register Map

Register Map	Group (Byte Offset from Base Address Register)			
	0	1	2	3
Group Base Pointer	00000h	00800h	01000h	01800h
Dual Address Cycle Base Pointer(1)	00004h			
Service Request Descriptor	00008h	00808h	01008h	01808h
Interrupt Status Descriptor(1)	0000Ch			
Transmit Timeslot Map	00200h	00A00h	01200h	01A00h
Transmit Subchannel Map	00280h	00A80h	01280h	01A80h
Transmit Channel Configuration Table	00380h	00B80h	01380h	01B80h
Receive Timeslot Map	00400h	00C00h	01400h	01C00h
Receive Subchannel Map	00480h	00C80h	01480h	01C80h
Receive Channel Configuration Table	00580h	00D80h	01580h	01D80h
Global Configuration Descriptor(1)	00600h			
Interrupt Queue Descriptor(1)	00604h			
Group Configuration Descriptor	0060Ch	00E0Ch	0160Ch	01E0Ch
Memory Protection Descriptor	00610h	00E10h	01610h	01E10h
Message Length Descriptor	00614h	00E14h	01614h	01E14h
Port Configuration Descriptor	00618h	00E18h	01618h	01E18h

Notes: (1). MUSYCC automatically maps Group 1, 2 and 3 addresses for these registers to the Group 0 address (shown). For example, accessing address 00E00h in MUSYCC (address for Group 1 Global Configuration register) automatically maps to address 00600h and the contents of 00600h is read or written.

The first set of four registers in each Channel Group (shown in bold-type in Table 25, MUSYCC Register Map) are located exclusively within MUSYCC. These registers are accessed by the host using direct reads and writes to the corresponding register map address. The remaining registers have corresponding locations within shared memory; and, the host typically accesses the shared memory image rather than the internal registers. Regardless, the values within MUSYCC are always the values used during device operation. After configuring the shared memory image of these registers, the host issues a Service Request by writing directly into the Service Request Descriptor which causes MUSYCC to copy the image from shared memory.

Each supported channel group requires its own table of registers to operate. The Dual Address Cycle Base Pointer, Interrupt Status Descriptor, Global Configuration Descriptor, and the Interrupt Queue Descriptor are shared among all supported groups and only Channel Group 0 registers are physically accessible within MUSYCC.

The Transmit Timeslot Map and the Transmit Subchannel Map are write-only areas within MUSYCC. Reading from these areas results in all one's being returned.

The Service Request Descriptors are locations within MUSYCC where commands can be directed to individual channel groups. The host writes a service request, a command, directly into the corresponding group's register. MUSYCC behaves as a PCI slave as this write is performed. The action resulting from the command may cause MUSYCC to read or write locations from shared memory. While MUSYCC accesses shared memory, it behaves as a PCI master and arbitrates for control of the bus autonomously.

MUSYCC's registers may be initialized before or after shared memory resident descriptors are initialized. The recommended sequence is to configure shared memory descriptors first then copy the relevant information to MUSYCC's registers, via the Service Request mechanism.

It is critically important to note that upon channel activation, shared memory and internal registers must be initialized, valid, and available to MUSYCC. MUSYCC will use the information within the shared memory descriptors to transfer data between the serial interface and shared memory. MUSYCC assumes the information is valid once a channel is activated.

The first four sets of pointers for each channel group (shown in bold-type in Table 26, Channel Group Memory Map) are pointer locations exclusive to shared memory. MUSYCC does not keep these values internally although they are accessed regularly during channel processing. The remaining locations have a corresponding register within MUSYCC.

Table 26. Channel Group Memory Map

Channel Group Memory Map	Byte Offset from Respective Group Base Pointer	Length (bytes)
Transmit Head Pointers	00000h	128
Transmit Message Pointers	00080h	128
Receive Head Pointers	00100h	128
Receive Message Pointers	00180h	128
Transmit Timeslot Map	00200h	128
Transmit Sub Channel Map	00280h	256
Transmit Channel Configuration Table	00380h	128
Receive Timeslot Map	00400h	128
Receive Sub Channel Map	00480h	256
Receive Channel Configuration Table	00580h	128
Global Configuration Descriptor	00600h	4
Interrupt Queue Descriptor	00604h	8
Group Configuration Descriptor	0060Ch	4
Memory Protection Descriptor	00610h	4
Message Length Descriptor	00614h	4
Port Configuration Descriptor	00618h	4
	Total Space Required	1564

Memory Access Illustration

Assume the system memory controller (or the host) allocates addresses for MUSYCC's PCI functions as shown below.

Table 27. MUSYCC PCI Function Memory Allocation

System Allocated MUSYCC Memory Ranges	Start Address	End Address	Length
MUSYCC - Function 0- Base Address Register	0240 0000h	024F FFFFh	1 Mbyte
EBUS - Function 1- Base Address Register	0340 0000h	034F FFFFh	1 Mbyte

The Base Address is written into MUSYCC by the host initiated PCI configuration access write cycles. After MUSYCC functions are memory mapped to PCI space, the host allocates shared memory space for each of the supported channel group descriptors. It is required that each Group Base Pointer start on a 2 Kbyte boundary.

Table 28. Shared Memory Allocation - Group Descriptors

Channel Group Descriptors	Start Address	End Address	Length
Group 0 Base Pointer	0090 0000h	0090 061Bh	1564 bytes
Group 1 Base Pointer	0090 0800h	0090 161Bh	1564 bytes
Group 2 Base Pointer	0090 1000h	0090 261bh	1564 bytes
Group 3 Base Pointer	0090 1800h	0090 361B	1564 bytes

The Group Base Pointer value is written into MUSYCC by the host via PCI write access cycles. The location of the Group Base Pointer register for each group within MUSYCC is shown in Table 25, MUSYCC Register Map.

For this illustration, the host would have to perform the following write operations:

Table 29. Host Assigns Group Base Pointers

Host Writes Pointer Value	To PCI Address
0090 0000h	0240 0000h
0090 0800h	0240 0800h
0090 1000h	024F 1000
0090 1800h	034F 1800

The next step is for the host to allocate the required shared memory for transmit and receive messages. Assume for this illustration that the host needs 8 message descriptors for each channel and direction and each corresponding data buffer per message is 100h (256) bytes in length.

Memory for Message Descriptors =
32 channels/group *
2 directions/channel *
12 bytes/message descriptor *
8 buffers/channel *
= 1800h bytes/group
= 6144 bytes/group

Memory for Data Buffers =
32 channels/group *
2 directions/channel *
8 buffers/channel *
256 bytes/buffer =
= 131, 072 bytes/group
= 20,000h bytes/group

Further, the host may choose to allocate all the memory contiguously or it may allocated the memory for message descriptors separately from data buffers - in which case, message descriptors for four Channel Groups may be merged into a contiguous block of memory [(1800h * 4 = 6000h) bytes in length].

Descriptors

This section further details the descriptors specified in the MUSYCC memory model. The following descriptors are expanded upon:

- Host Interface Level Descriptors
  - Global Configuration
  - Dual Address Cycle Base Pointer
- Channel Group Level Descriptors
  - Group Base Pointer
  - Service Request
  - Group Configuration
  - Memory Protection
  - Port Configuration
  - Message Length
  - Timeslot Map
  - Subchannel Map
- Channel Level Descriptors
  - Channel Configuration
- Message Level Descriptor
  - Head Pointer
  - Message Pointer
  - Message Descriptor
  - Buffer Descriptor
  - Buffer Status Descriptor
  - Next Message Pointer
  - Data Buffer Pointer
  - Message Descriptor Handling
- Interrupt Level Descriptors
  - Interrupt Queue Descriptor
  - Interrupt Descriptor
  - Interrupt Status Descriptor

Each of the above entities are allocated, deallocates, read from and written to by the host. MUSYCC can read all of these entities as well, but can only write to the:

- Message Pointers
- Buffer Status Descriptors
- Receive Data Buffers

Host Interface Level Descriptors

Global Configuration Descriptor

This descriptor specifies configuration information which applies to the entire device including all channel groups, all serial ports, and all channels.

Memory space is reserved for the Global Configuration Descriptor within each Channel Group Descriptor. By convention, the values corresponding to Channel Group 0 (a group present in all versions of MUSYCC) provides the correct data. The host coordinates how this data is transferred into MUSYCC; either by instructing MUSYCC to read the Channel Group 0 Global Configuration Descriptor when setting the global data or by copying the Channel Group 0 Global Configuration Descriptor to all other supported Channel Group Descriptors and then requesting a global initialization service request operation for any of the supported channel groups.

Table 30. Global Configuration Descriptor *(1 of 2)*

Bit Field	Name	Value	Description
31:24	RSVD	0	Reserved.
23 22 21 20 19 18 17 16	TCLKACT0 TCLKACT1 TCLKACT2 TCLKACT3 RCLKACT0 RCLKACT1 RCLKACT2 RCLKACT3		Transmit and Receive Line Clock Activity Indicator. 0,1,2, or 3 corresponds to a channel group number. Read Only. Reset to 0 after each read. Each indicator bit is cleared when the respective channel group is reset via PCI Reset, Soft Group Reset, or Soft Chip Reset. The TCLKACTx corresponds to TCLKx line clock. The RCLKACTx corresponds to RCLKx line clock. The indicator is set to 1 on the second rising edge of the corresponding serial interface line clock and the previous value for the indicator bit was 0. If multiple channel groups are mapped to a single serial port, then one clock is driving each channel group. The indicator bits reflect the activity of the clock driving the channel group. If MUSYCC does not detect a line clock, then the value of the indicator bit(s) remain at the reset value 0. Reading from channel group RAM during the absence of a line clock will result in the dword DEADACCEh (stands for “dead access”) being returned. Writing to channel group RAM during the absence of a line clock will result in the write being ignored.
15	RSVD	0	Reserved.
14:12	BLAPSE[2:0]	0–7	Expansion Bus Access Interval. MUSYCC waits BLAPSE+1 number of ECLK periods immediately after relinquishing the bus. This wait ensures that all the bus grant signals driven by the bus arbiter have sufficient time to be deasserted as a result of bus request signals being deasserted by MUSYCC.
11	ECKEN	0	Expansion Bus Clock Disabled. ECLK output is three-stated.
		1	Expansion Bus Clock Enabled. MUSYCC redrives and inverts PCLK input onto ECLK output pin.
10	MPUSEL	0	Expansion Bus Microprocessor Selection - Motorola-style. Expansion bus supports the Motorola-style microprocessor interface and uses Motorola signals: Bus Request (BR*), Bus Grant (BG*), Address Strobe (AS*), Read/Write (R/WR*), and Read Strobe (RD*).
		1	Expansion Bus Microprocessor Selection - Intel-style. Expansion bus supports the Intel-style microprocessor interface and uses Intel signals: Hold Request (HOLD), Hold Acknowledge (HLDA), Address Latch Enable (ALE*), Write Strobe (WR*), and Data Strobe (DS*).

Table 30. Global Configuration Descriptor (2 of 2)

Bit Field	Name	Value	Description
9:8	ALAPSE[1:0]	0–3	Expansion Bus Address Duration. MUSYCC extends the duration of valid address bits during an EBUS address phase to ALAPSE+1 number of ECLK periods. The control lines ALE* (Intel) or AS* (Motorola) indicate that the address bits have had the desired setup time.
7		0	Reserved.
6:4	ELAPSE[2:0]	0–7	Expansion Bus Data Duration. MUSYCC extends the duration of valid data bits during an EBUS data phase to ELAPSE+1 number of ECLK periods. The control lines RD* and WR* (Intel) or DS* and R/WR* (Motorola) indicate that the data bits have had the desired setup time.
3:2		0	Reserved.
1:0	PORTMAP[1:0]	0	Default. Port 0 mapped to Channel Group 0. Port 1 mapped to Channel Group 1. Port 2 mapped to Channel Group 2. Port 3 mapped to Channel Group 3.
		1	Port 0 mapped to Channel Groups 0 and 1. Port 1 mapped to Channel Groups 2 and 3.
		2	Port 0 mapped to Channel Groups 0, 1, 2, and 3.
		3	Reserved.

Dual Address Cycle Base Pointer

MUSYCC supports 32-bit and 64-bit memory addressing. The Dual Address Cycle Base Pointer is used to supports 64-bit memory addressing.

Table 31. Dual Address Cycle Base Pointer

Bit Field	Name	Value	Description
31:0	DACBASE[31:0]		Dual Address Cycle Base Pointer. A 32-bit base register when non-zero causes all MUSYCC master operations (read/write) to use PCI Dual Address Cycle. The value in this register would be the upper 32-bits of the 64-bit addressing.

If the value of Dual Address Cycle Base Pointer (DACBASE) is zero, then MUSYCC initiates all memory access cycles without dual-addressing. If the value of the pointer is non-zero, then MUSYCC initiates all memory access cycles with dual-addressing.

For cycles without dual-addressing, MUSYCC uses the AD[31:0] signal lines to indicate the address of the memory access. During the address phase, MUSYCC encodes the type of access cycle (e.g. read, write,...) in the Command/Byte Enable signal lines, CBE[3:0]*. The address phase lasts for one PCLK period.

For cycles with dual-addressing, MUSYCC multiplexes a 64-bit address onto the AD[31:0] signal lines and adds an additional PCLK period to the address phase. To indicate 64-bit addressing, MUSYCC encodes the dual address code onto the CBE[3:0]* signal lines during the first PCLK period of the address phase. MUSYCC encodes the access type code (e.g. read, write) onto the CBE[3:0]* signal lines during the second PCLK period of the address phase.

When MUSYCC accesses a 64-bit memory address using dual-addressing, the upper 32 bits of the address are fixed to a non-zero value from DACBASE. To change from 64-bit addressing to 32-bit addressing, the value of DACBASE must be zeroed. Though MUSYCC is capable of initiating 64-bit addressing when in master mode, it only responds to 32-bit access cycles without dual-addressing.

Channel Group Level Descriptors

Channel Group Descriptors contain all of the information necessary to configure one channel group and the associated 32 logical channels while maintaining pointers to buffer descriptors for each channel and direction. The contents of the Channel Group Descriptor are listed in Table 26, Channel Group Memory Map.

Group Base Pointer

This register (GBASE) per channel group within the host interface contains a 2 Kbyte aligned pointer to a corresponding Channel Group Descriptor in shared memory.

Table 32. Group Base Pointer

Bit Field	Name	Value	Description
31:11	GBASEx[20:0]		These 21 bits are appended with 11 zeros to form a 2 K block-aligned 32-bit address pointing to the first dword of the channel group structure for Channel Group x.
10:0	GBASEx[10:0]	0	These 11 bits appended to GBASE ensure 2 Kbyte block alignment.

Service Request

This register per channel group within the host interface contains a bit-field where instructions are written to MUSYCC by the host. The following instructions are supported:

- Perform device reset and initialization
- Perform channel group reset and initialization
- Configure a channel
- Read specific descriptors from within a Channel Group Descriptor
- Activate a channel
- Deactivate a channel
- Jump (re)activate a channel
- No-operation command

A service request is issued to a specific channel group within MUSYCC. That service request is subsequently acknowledged by the channel group that issues a service request acknowledge interrupt descriptor back to the host.

The soft-chip reset service request is the only service request that is not acknowledged by MUSYCC.

Issuing multiple service requests to the same channel group successively without first receiving acknowledgments from each request may cause the host to lose track of which service request has been acknowledged as there is no way for MUSYCC to uniquely acknowledge service requests for the same channel group. As well, issuing multiple simultaneous request to the same channel group will cause indeterminate results within the channel group.

Issuing a single service requests to each supported channel group simultaneously is supported as MUSYCC will acknowledge each one uniquely with the Group ID bit-field.

Table 33. Service Request Descriptor *(1 of 2)*

Bit Field	Name	Value	Description
31:13	RSVD	0	Reserved.
12:8	SREQ[4:0]	0	No-Op. This service request performs no action other than to facilitate a Service Request Acknowledge Interrupt (SACK). This would be used as a "UNIX Ping-like" operation to detect the presence of a channel group processor.
		1	Soft Chip Reset. This is identical to a hardware reset. Set PORTMAP = 0, disable all supported ports (both directions), and deactivate all 32 channels of all supported groups (both directions). Note that the Interrupt Status Descriptor is reset to point to the first dword in the queue and all indicator bits are reset. Note: This service request is not acknowledged by MUSYCC.
		2	Soft Group Reset. This is similar to a hardware reset for a specified group and direction. Disable all specified ports (both directions), and deactivate all 32 channels of specified group (both directions).
		3	Reserved.
		4	Global Initialization. For the entire device, read the Global Configuration Descriptor and the Interrupt Queue Descriptor from shared memory. This initialization is performed following a hardware or soft-chip reset. Note that the Interrupt Status Descriptor is reset to point to the first dword in the queue and all indicator bits are reset.
		5	Group Initialization. For this group and direction, read the following from shared memory: – Timeslot Map – Subchannel Map – Channel Configuration Descriptor Table – Group Configuration Descriptor – Memory Protection Descriptor – Message Length Descriptor – Port Configuration Descriptor This initialization is performed for each group and each direction immediately following any reset or global initialization. See Note(1).
		6–7	Reserved.
		8	Channel Activation. For a specified channel and direction initialize the channel, read the head pointer, jump to the first message descriptor, and start processing the data buffer. This request starts processing a new message list. The effect of this request is destructive to the current message being processed if the channel was already activated and processing a message. This request is useful in aborting the current message and starting a new message list.
		9	Channel Deactivation. For a specified channel and direction suspend channel activity.
		10	Jump. For the receiver, this request is the same as the Channel Activation request. For the transmitter, this request is useful for switching to a new message list after the current message is completely transferred.
		11	Channel Configure. For a specified channel and direction read the Channel Configuration Descriptor.
		12–15	Reserved.

Table 33. Service Request Descriptor (2 of 2)

Bit Field	Name	Value	Description
12:8	SREQ[4:0]	16	Read Global Configuration Descriptor.
		17	Read Interrupt Queue Descriptor. Note that the Interrupt Status Descriptor is reset to point to the first dword in the queue and all indicator bits are reset.
		18	Read Group Configuration Descriptor.
		19	Read Memory Protection Descriptor
		20	Read Message Length Descriptor
		21	Read Port Configuration Descriptor.
		22–23	Reserved.
		24	Read Timeslot Map. For this group and specified direction read the Timeslot Map(1).
		25	Read Subchannel Map. For this group and specified direction read the Subchannel Map(1).
		26	Read Channel Configuration Table. For this group and specified direction read the Channel Configuration Table(1).
		27–31	Reserved.
7:6	RSVD	0	Reserved.
5	DIR	0	Receive direction.
		1	Transmit direction.
4:0	CH[4:0]		Channel number.

Notes: (1). Timeslot Map, Subchannel Map, and Channel Configuration Description Table are read into MUSYCC only if a serial clock is provided to the Serial Interface being configured

Group Configuration Descriptor

This descriptor contains configuration bits that apply to all 32 logical channels within a given channel group.

Table 34. Group Configuration Descriptor *(1 of 3)*

Bit Field	Name	Value	Description
31:22	RSVD	0	Reserved.
21:16	SUET[5:0]		Signal Unit Error Threshold. Sets maximum value of SUERM counter. When SUERM exceeds this count a SUERR interrupt is generated.
15	SFALIGN	0	Super Frame Alignment. Flywheel Mechanism. Select roll over to 0 of timeslot counter (the flywheel mechanism in the serial interface) as a frame synchronization event. For a non-HDLC channel, wait for the flywheel to roll over to start message processing after channel activation. For descriptor polling, use the flywheel roll over as a frame synchronization event. The polling frequency is determined by using the poll-throttle field elsewhere in this descriptor.
		1	Super Frame Alignment. External Signal. Select detection of frame synchronization signal (TSYNC or RSYNC) assertion as frame synchronization event. For a non-HDLC channel wait for assertion of signal to start message processing after channel activation. For descriptor polling, use assertion of signal as frame synchronization event. Polling frequency is determined by using poll-throttle field elsewhere in this descriptor.
14:12	RSVD	0	Reserved.
11:10	POLLTH[1:0]	0	Poll Throttle. Poll at every frame synchronization event. This mechanism can be used in avoiding bus saturation when multiple logical channels are active and idle and waiting for buffers to service.
		1	Poll at every 16th frame synchronization event.
		2	Poll at every 32nd frame synchronization event.
		3	Poll at every 64th frame synchronization event.
9	INHTBSD	0	Inhibit Transmit Buffer Status Descriptor Disabled. At end of each transmitted data buffer, do not inhibit (allow) over writing of Tx Buffer Descriptor with a Tx Buffer Status Descriptor.
		1	Inhibit Transmit Buffer Status Descriptor. As the Tx Buffer Status Descriptor is being inhibited, the host must rely on an interrupt for status information regarding transmitted data message.
8	INHRBSD	0	Inhibit Receive Buffer Status Descriptor Disabled. At the end of each Receive Data Buffer, do not inhibit (allow) over writing of Rx Buffer Descriptor with a Rx Buffer Status Descriptor.
		1	Inhibit Receive Buffer Status Descriptor. As the Rx Buffer Status Descriptor is being inhibited, the host must rely on an interrupt for status information regarding received data message.

Table 34. Group Configuration Descriptor (2 of 3)

Bit Field	Name	Value	Description
7	MEMPVA	0	Memory Protection Violation Action. Reset Group. On memory protection violation error, group reset is performed. As a result, all 32 channels are deactivated in both receive and transmit directions.
		1	Memory Protection Violation Action. Deactivate Channel. On memory protection violation error, only channel being serviced during violation is deactivated in both receive and transmit directions.
6	MCENBL	0	Message Configuration Bits Copy Disable. Refer to “enable” description.
		1	Message Configuration Bits Copy Enable. A set of configuration bits are copied from the last transmit buffer descriptor of a message (with EOM = 1) into internal configuration space and subsequently acted upon. The bits are used in specialized data communications applications requiring non-default idle code transmission on a per-message basis, inter-message pad fill, and/or message retransmission. For applications involving high data rates including back-to-back message transmission, this bit is set to its disable state. Note that direct writes into MUSYCC to change any of the message configuration bits remain available. Only automatic copying from a transmit buffer descriptor (with the bit-field EOM set to 1) is disabled. For a thorough discussion on message configuration bits see the Message Configuration Bits subsection in the Basic Operation section.
5	MSKCOFA	0	Change Of Frame Alignment Interrupt Enabled. If COFA is detected, generate Interrupt Descriptor indicating COFA.
		1	Change Of Frame Alignment Interrupt Disabled. If COFA is detected, do not generate Interrupt Descriptor.
4	MSKOOF	0	Out Of Frame/Frame Recovery Interrupt Enabled—Receive Only. If OOF/FREC is detected, generate Interrupt Descriptor indicating OOF/FREC.
		1	Out Of Frame/Frame Recovery Interrupt Disabled—Receive Only.
3	OOFABT	0	OOF Message Processing Enabled—Receive Only. When OOF condition is detected, continue processing incoming data.
		1	OOF Message Processing Disabled—Receive Only. When OOF condition is detected, abort incoming message processing and suspend activity on channel.

Table 34. Group Configuration Descriptor *(3 of 3)*

Bit Field	Name	Value	Description
2	SUBDSBL	0	Subchanneling Enabled. Overrides Subchannel Enable bit for all timeslots and allows subchanneling.
		1	Subchanneling Disabled. Overrides Subchannel Enable bit for all timeslots and disallows subchanneling. Using this field to disable subchanneling frees Subchannel Descriptor Map memory space for use as an extended data buffer space.
1	TXENBL	0	Transmitter Disabled. Logically resets timeslot enable bits for all timeslots. Transmit data lines are three-stated. This logical, channel-group-wide state does not affect the bit values in any timeslot map.
		1	Transmitter Enabled. Logically allows all channels with timeslot enable bits set to start processing data. This logical, channel-group-wide state does not affect the bit values in any timeslot map.
0	RXENBL	0	Receiver Disabled. Logically resets timeslot enable bits for all timeslots. This logical, channel-group-wide state does not affect the bit values in any timeslot map.
		1	Receiver Enabled. Logically allows all channels with timeslot enable bits set to start processing data. This logical, channel-group-wide state does not affect the bit values in any timeslot map.

Memory Protection Descriptor

This descriptor per channel group describes a contiguous region in memory that is legal address space for MUSYCC's PCI bus master mode addressing.

When the memory protection feature is enabled, the 32-bit address of each master mode memory access initiated by MUSYCC is checked by the host interface to ensure that it is within the region defined by the Memory Protection Descriptor for the channel group requesting the transaction.

In the case where a memory protection violation is detected, the PCI read or write operation is inhibited and the PROT interrupt is generated toward the host. Also, depending on the value of MEMPVA bit-field in Table 34, Group Configuration Descriptor, either the channel group is reset or a single channel is deactivated in both directions.

Memory protection checks are automatically disabled during writes to the interrupt queue because memory protection violations during this time are considered catastrophic and only occur due to device failure. For that reason, memory protection checks are automatically disabled during interrupt queue memory operations, so that delivery of interrupt descriptors to the host can continue.

The protected memory regions are based on a 1 Mbyte bound memory region and are specified using low and high addresses of 1 Mbyte segments.

To protect 1 Mbyte of memory, both the low and high address' upper 12 bits must be the same.

Table 35. Memory Protection Descriptor

Bit Field	Name	Value	Description
31	PROTENBL	0	Memory Protection Disabled.
		1	Memory Protection Enabled.
30:28	RSVD	0	Reserved.
27:16	PROTHI[11:0]		Memory Protection High Address. Upper 12 bits (inclusive) of address for highest memory location under protection.
15:12	RSVD	0	Reserved.
11:0	PROTLO[11:0]		Memory Protection Low Address. Upper 12 bits (inclusive) of the address for lowest memory location under protection.

Port Configuration Descriptor

This descriptor per port defines how MUSYCC interprets and synchronizes the transmit and receive bit streams associated with that port. Note that there is one descriptor per port. Hence, the descriptor is used for both transmit and receive directions for a single port.

It is imperative to note when a port is being used in conjunction with another port as is the case when PORTMAP=1 or PORTMAP=2 in Table 30, Global Configuration Descriptor, that the unused port descriptor be mapped identically to the used port descriptor. That is, if PORTMAP=1, then the port 1 descriptor must be bit-for-bit identical with the port 0 descriptor; and, the port 3 descriptor must be bit-for-bit identical with the port 2 descriptor. In the case of PORTMAP=2, then the ports 1, 2, and 3 descriptors must be bit-for-bit identical with the port 1 descriptor.

Table 36. Port Configuration Descriptor

Bit Field	Name	Value	Description
31:10	RSVD	0	Reserved.
9	TRITX	0	Transmit Three-state Enabled. When a channel group is enabled, but a timeslot within the group is not mapped via the Timeslot Map, the transmitter three-states the output data signal.
		1	Transmit Three-State Disabled. When a channel group is enabled, but a timeslot within the group is not mapped via the Timeslot Map, the transmitter outputs a logic 1 on the output data signal.
8	ROOF_EDGE	0	Receiver Out Of Frame—Falling Edge. ROOF input sampled in on falling edge of RCLK.
		1	Receiver Out Of Frame—Rising Edge.
7	RSYNC_EDGE	0	Receiver Frame Synchronization—Falling Edge. RSYNC input sampled in on falling edge of RCLK.
		1	Receiver Frame Synchronization—Rising Edge.
6	RDAT_EDGE	0	Receiver Data—Falling Edge. RDAT input sampled in on falling edge of RCLK.
		1	Receiver Data—Rising Edge.
5	TSYNC_EDGE	0	Transmitter Frame Synchronization—Falling Edge. TSYNC input sampled in on falling edge of TCLK.
		1	Transmitter Frame Synchronization—Rising Edge.
4	TDAT_EDGE	0	Transmitter Data—Falling Edge. TDAT output latched out on falling edge of TCLK.
		1	Transmitter Data—Rising Edge.
3	RSVD	0	Reserved.
2:0	PORTMD[2:0]	0	T1 Mode. 24 timeslots and T1 signaling.
		1	E1 Mode. 32 timeslots and E1 signaling.
		2	2xE1 Mode. 64 timeslots and E1 signaling.
		3	4xE1 Mode. 128 timeslots and E1 signaling.
		4	Nx64 Mode. Frame synchronization flywheel disabled. COFA detection disabled. Every synchronization signal assertion resets timeslot counter to zero.
		5–7	Reserved.

Message Length Descriptor

Each channel group can have two separate values for maximum message length (MAXFRM1 or MAXFRM2 in Table 37, Message Length Descriptor). The maximum message length is 16,384 octets. The minimum message length is either 3 or 5 depending on 16- or 32-bit FCS support, respectively. Each receive channel either selects one of these message length values or disables message length checking altogether.

The MAXSEL bit-field in Table 42, Channel Configuration Descriptor selects which, if any, register is used for received-message length checking. If MUSYCC receives a message that exceeds the allowed maximum, the current message processing is discontinued and further transfer of data to shared memory is terminated. In addition, a Receive Buffer Status Descriptor corresponding to the partially received message indicates this error condition, too long of a message, and an Interrupt Descriptor is generated towards the host indicating the same error condition.

If MUSYCC receives a message whose bit count is less than 3 or 5 octets, depending on 16- or 32-bit FCS support, respectively, then the message is considered too short. In the case of a short message, data is not transferred into shared memory and is discarded. In addition, an Interrupt Descriptor is generated towards the host indicating the same error condition.

Table 37. Message Length Descriptor

Bit Field	Name	Value	Description
31:30	RSVD	0	Reserved.
29:16	MAXFRM2[13:0]		Defines a limit for the maximum number of octets allowed in a received HDLC message. Valid values for the register range from 2 to 16383. As this value is 0-based, the range of values specifies a minimum of 3 octets to a maximum 16384 octets.
15:14	RSVD	0	Reserved.
13:0	MAXFRM1[13:0]		Defines a limit for the maximum number of octets allowed in a received HDLC message. Valid values for the register range from 2 to 16383. As this value is 0-based, the range of values specifies a minimum of 3 octets to a maximum 16384 octets.

Timeslot Map

This map is used when the serial port associated with a channel group is configured in one of the channelized modes: T1, E1, 2xE1, 4xE1, or Nx64. MUSYCC supports mapping of up to 128 timeslots from a channelized bit stream with up to 32 logical channels in each channel group.

Numerous mappings of timeslots to channels are possible. Multiple timeslots can be mapped to a single channel; however, each timeslot can only map to one channel at a time. In cases where the number of active timeslots exceeds the number of channels in a group (greater than 32 timeslots) and each timeslot requires a separate channel, MUSYCC can be configured to internally connect 2 or 4 channel groups together to provide up to 64- or 128- channel support, respectively.

For each channel group, two timeslot maps are required, one for transmit functions and one for receive functions. The two separate maps are configured independently. Each map consists of 128 successive 8-bit fields, each corresponding to one timeslot. The bit-field includes the following information:

- Timeslot Enabled/Timeslot Mode of Operation (64 Kbps, 56 Kbps, subchannel) indicator (TSEN)
- Logical Channel Number (0–31) associated with timeslot (CH)

For disabled timeslots, modes and logical channels may be assigned but the information does not apply to operation of the channel.

For enabled timeslots, the valid modes of operation include the following:

- 64 Kbps Mode: All 8 bits of timeslot are assigned to one channel.
- 56 Kbps Mode: First 7 bits of timeslot are assigned to one channel. Last bit (msb) is unassigned and considered disabled.
- Subchannel Mode, Bit 0 Disabled: First bit (lsb) of timeslot is unassigned. Each of next 7 bits in timeslot may be individually enabled and independently mapped to any channel in a channel group.
- Subchannel Mode, Bit 0 Enabled: First bit (lsb) of timeslot is always enabled and assigned to a channel in a channel group. Each of the next 7 bits in timeslot may be individually enabled and independently mapped to any channel in a channel group.

The logical channel number represents the channel in the channel group that will handle the bit stream from the timeslot(s) assigned to it. The value of the channel number ranges from 0 to 31, inclusively.

A timeslot map consists of 32 timeslot descriptors. One descriptor maps four timeslots. The entire map contains configuration information for 128 separate timeslots. Note that the serial port does not need to support 128 timeslots all the time. Any number of timeslots from 1 to 128 is supportable.

Table 38. Transmit or Receive Timeslot Map

Byte Offset	Timeslot Descriptor Relative Location			
	MSB			LSB
00h	TS03	TS02	TS01	TS00
...	...	...	...	...
1Ch	TS31	TS30	TS29	TS28
...	...	...	...	...
3Ch	TS63	TS62	TS61	TS60
...	...	...	...	...
5Ch	TS95	TS94	TS93	TS92
...	...	...	...	...
7Ch	TS127	TS126	TS125	TS124

Accessing the Timeslot Map within MUSYCC requires that a serial line clock be present at the Serial Interface. If a clock is not present, writes are ignored and reads return all ones.

The host can read and write Receive Timeslot Map from within MUSYCC; however, the host can only write Transmit Timeslot Map into MUSYCC. The transmit maps are stored in write-only registers. Reading transmit maps results in all ones being returned.

Table 39. Timeslot Descriptor

Bit Field	Name	Value	Description
31:29	TSEN3[2:0]	0	Timeslot Disabled. Default.
		1–3	Reserved.
		4	Timeslot Enabled. 64 Kbps mode.
		5	Timeslot Enabled. 56 Kbps mode.
		6	Timeslot Enabled. Subchannel mode w/o first bit.
		7	Timeslot Enabled. Subchannel mode w/ first bit.
28:24	CH3[4:0]	0–31	Channel number assigned to this timeslot.
23:21	TSEN2[2:0]	0	Timeslot Disabled. Default.
		1–3	Reserved.
		4	Timeslot Enabled. 64 Kbps mode.
		5	Timeslot Enabled. 56 Kbps mode.
		6	Timeslot Enabled. Subchannel mode w/o first bit.
		7	Timeslot Enabled. Subchannel mode w/ first bit.
20:16	CH2[4:0]	0–31	Channel number assigned to this timeslot.
15:13	TSEN1[2:0]	0	Timeslot Disabled. Default.
		1–3	Reserved.
		4	Timeslot Enabled. 64 Kbps mode.
		5	Timeslot Enabled. 56 Kbps mode.
		6	Timeslot Enabled. Subchannel mode w/o first bit.
		7	Timeslot Enabled. Subchannel mode w/ first bit.
12:8	CH1[4:0]	0–31	Channel number assigned to this timeslot.
7:5	TSCN0[2:0]	0	Timeslot Disabled. Default.
		1–3	Reserved.
		4	Timeslot Enabled. 64 Kbps mode.
		5	Timeslot Enabled. 56 Kbps mode.
		6	Timeslot Enabled. Subchannel mode w/o first bit.
		7	Timeslot Enabled. Subchannel mode w/ first bit.
4:0	CH0[4:0]	0–31	Channel number assigned to this timeslot.

Subchannel Map

To provide the subchanneling feature, MUSYCC shares the timeslot mapping function between a Timeslot Map and a Subchannel Map. The transmit and receive function each have a separate pair of maps.

The Timeslot Map indicates if the timeslot is enabled and configured for subchanneling. The TSEN bit-field in the Timeslot Descriptor specifies the mode of operation for the timeslot. If TSEN = 6, timeslot is in subchannel mode with bit 0 disabled. If TSEN = 7, timeslot is in subchannel mode with bit 0 enabled and mapped to the channel specified in the CH bit-field in the same Timeslot Descriptor. The CHx bit-field is also used in the treatment of the remaining 7 bits in the timeslot, specifically it is used as part of an index into Subchannel Map. The Timeslot Map always manages (disables or enables/maps) bit 0 of the timeslot. The Subchannel Map configures each of the remaining 7 bits of a timeslot.

The CHx bit-field (0–31) bit-field from the Timeslot Descriptor is concatenated with the bit number (1–7) of the timeslot being processed to form an 8-bit index (0–255) into the Subchannel Map for each of the bit numbers. The Subchannel Map associates each bit of a timeslot to a channel.

For a timeslot configured for subchannel mode, MUSYCC considers each bit of that timeslot independently of any other bit in the same timeslot. Any bit may be enabled or disabled. The timeslot is considered to consist of 8 individual 8 Kbps bit streams, or subchannels, which may be left separated or concatenated together to provide one or more subchannels running at bit rates between 8 Kbps and 64 Kbps, inclusive, in multiple of 8 Kbps.

Note the following special cases for subchannel map assignments:

- Bit 0 is always disabled or enabled/mapped by the Timeslot Map. Therefore, configuring the Subchannel Map for bit 0 is not required.
- Assigning the same channel number for all bits (including bit 0) is equivalent to disabling the subchanneling feature and treating the timeslot as if in 64 Kbps mode.
- Disabling all bits in subchannel mode is equivalent to disabling the timeslot using the Timeslot Map.

The Subchannel Map can map bits 1 through 7 from each timeslot to any one of 32 channels in a channel group. The map for bit 0 is allocated, but unused (recall bit 0 is mapped by the timeslot map). The Subchannel Map consists of 256 descriptors representing 32 channels and 8 bits per channel. Two subchannel descriptors (2 dwords) are required to describe the treatment of 8 bits of one channel. The first dword describes the treatment of the lower 4 bits, and the second dword describes the treatment of the upper 4 bits.

Table 40. Transmit or Receive Subchannel Map

Byte Offset	MSB			LSB
00h	Ch0, Bit 3	Ch0, Bit 2	Ch0, Bit 1	Unused
04h	Ch0, Bit 7	Ch0, Bit 6	Ch0, Bit 5	Ch0, Bit 4
08h	Ch1, Bit 3	Ch1, Bit 2	Ch1, Bit 1	Unused
0Ch	Ch1, Bit 7	Ch1, Bit 6	Ch1, Bit 5	Ch1, Bit 4
...	....	....	....	...
...	....	....	....	...
F8h	Ch31, Bit 3	Ch31, Bit 2	Ch31, Bit 1	Unused
FCh	Ch31, Bit 7	Ch31, Bit 6	Ch31, Bit 5	Ch31, Bit 4

Table 41. Subchannel Descriptor

Bit Field	Name	Value	Description
31	BITEN3/7	0	Bit disabled.
		1	Bit enabled.
30:29	RSVD	0	Reserved.
28:24	CH3[4:0]	0–31	Channel number assigned to this bit.
23	BITEN2/6	0	Bit disabled.
		1	Bit enabled.
22:21	RSVD	0	Reserved.
20:16	CH2[4:0]	0–31	Channel number assigned to this bit.
15	BITEN1/5	0	Bit disabled.
		1	Bit enabled.
14:13	RSVD	0	Reserved.
12:8	CH1[4:0]	0–31	Channel number assigned to this bit.
7	BITEN0/3	0	Bit disabled.
		1	Bit enabled.
6:5	RSVD	0	Reserved.
4:0	CH0[4:0]	0–31	Channel number assigned to this bit.

To enable the subchanneling feature, both the Timeslot Map and the Subchannel Map must be copied into MUSYCC's internal registers as it is from here that timeslot to channel mapping and channel to subchannel mapping is decoded. The host can instruct MUSYCC to read in the maps from shared memory by issuing the appropriate service request, otherwise the host must perform multiple direct writes into MUSYCC's internal registers by appropriately addressing PCI access cycles for MUSYCC.

Accessing the Timeslot Map or Subchannel Map within MUSYCC requires that a serial line clock be present at the Serial Interface. If a clock is not present, writes are ignored and reads return all ones.

The host can read and write Receive Timeslot Map from within MUSYCC; however the host can only write Transmit Timeslot Map into MUSYCC. The transmit maps are stored in write-only registers. Reading transmit maps results in all ones being returned.

The host can read and write Receive Subchannel Map from within MUSYCC; however, the host can only write Transmit Subchannel Map into MUSYCC. The transmit maps are stored in write-only registers. Reading transmit map results in all ones being returned.

Channel Level Descriptors

Channel Configuration Descriptor

This descriptor configures aspects of the channel that are common to all messages passing through the channel. One descriptor exists for each logical channel direction.

Table 42. Channel Configuration Descriptor *(1 of 2)*

Bit Field	Name	Value	Description
31	PADJ	0	Pad Count Adjustment Disabled. No adjustment is made to number of pad fill octets if Zero Insertions is detected.
		1	Pad Count Adjustment Enabled.
30	RSVD	0	Reserved
29:24	BUFFLOC[5:0]	00h–3Fh	Channel Buffer Location Index. Starting location of internal FIFO data buffer for this channel and direction.
23	INV	0	Data Inversion Disabled. All data bits in message are not inverted between shared memory and MUSYCC.
		1	Data Inversion Enabled.
22:16	BUFFLEN[6:0]	00h–7Fh	Internal Data Buffer Length. Number of internal FIFO data buffer locations allocated to this channel and direction equals 2 * (BUFFLEN+1) dwords.
15	EOPI	0	End Of Padfill Interrupt Disabled. Transmit Only. After outputting last padfill code, do not generate interrupt indicating condition.
		1	End of Padfill Interrupt Enabled.
14:12	PROTOCOL[2:0]	0	TRANSPARENT
		1	SS7-HDLC-FCS16
		2	HDLC-FCS16
		3	HDLC-FCS32
		4–7	Reserved.
11:10	MAXSEL[1:0]	0	Message Length - Disable message length check.
		1	Message Length - Use Register 1. Use MAXFRM1 bit-field in the message Length Descriptor for maximum receive message length limit.
		2	Message Length - Use Register 2. Use MAXFRM2 bit-field in the message Length Descriptor for maximum receive message length limit.
		3	Reserved
9	FCS	0	FCS Transfer Normal. For receive, do not transfer received FCS into shared memory along with data message. For transmit, do transmit calculated FCS out serial port after last bit in last data buffer has been transmitted.
		1	FCS Transfer Debug. For receive, transfer received FCS into shared memory along with data message. For transmit, do not transmit calculated FCS out serial port.
8	MSKSUERR	0	SUERR Interrupt Enabled. Receive Only. For SS7-HDLC-FCS16 mode, Signal Unit Error Rate Monitor function generates interrupt when signal unit error count crosses signal unit error threshold.
		1	SUERR Interrupt Disabled.

Table 42. Channel Configuration Descriptor (2 of 2)

Bit Field	Name	Value	Description
7	MSKSINC	0	SINC Interrupt Enabled. Receive Only. For SS7-HDLC-FCS16 mode, SUERM function generates interrupt when signal unit error count increments.
		1	SINC Interrupt Disabled.
6	MSKSDEC	0	SDEC Interrupt Enabled. Receive Only. For SS7-HDLC-FCS16 mode, SUERM function generates interrupt when signal unit error count decrements.
		1	SDEC Interrupt Disabled.
5	MSKSFILT	0	SFILT Interrupt Enabled. Receive Only. For SS7-HDLC-FCS16 mode, interrupt generated when two consecutive received messages found to be identical. Second message discarded.
		1	SFILT Interrupt Disabled. For SS7-HDLC-FCS16 mode, interrupt is not generated when two consecutive received messages found to be identical. Second message discarded.
4	MSKIDLE	0	CHABT, CHIC, SHT Interrupt Enabled. Receive Only. When receiver detects change to abort code, change to idle code or too-short message, generate interrupt to indicate condition.
		1	CHABT, CHIC, SHT Interrupt Disabled.
3	MSKMSG	0	LNG, FCS, ALIGN, ABT Interrupt Enabled. Receive Only. When receiver detects too-long message, FCS error, message alignment error or abort condition, generate interrupt to indicate condition.
		1	LNG, FCS, ALIGN, ABT Interrupt Disabled.
2	MSKEOM	0	EOM Interrupt Enabled. Receive and Transmit. Interrupt generated when end of message detected.
		1	EOM Interrupt Disabled.
1	MSKBUFF	0	BUFF/ONR Interrupt Enabled. Receive and Transmit. Interrupt generated when transmitter underflow buffer or receiver overflows buffer internally to MUSYCC.
		1	BUFF/ONR Interrupt Disabled.
0	RSVD	0	Reserved

Message Level Descriptor

One message descriptor defines one data buffer where all or part of a message is stored in shared memory. By linking Message Descriptors, numerous data buffers are linked together to support high-speed data links or large messages spread across a number of smaller data buffers.

Depending on the transmission and reception rate of individual channels, the numbers and sizes of message buffers can vary between channels and/or applications. For high-speed lines, more and larger buffers may be employed to provide ample data storage while the host processes each message in the list of messages. For low-speed lines, less and smaller buffers may be employed as the host may be able to process each message faster and the need to store messages is lessened.

Multiple smaller data buffers can be linked together using message descriptors to store one large message. In utilizing multiple buffers, the importance of keeping the sequence of data buffers in order is obvious.

MUSYCC's operation allows for the following:

- multiple buffer lists
- multiple and variable size buffers within a buffer list
- multiple buffers storing a single message
- sequencing of individual data buffers for a multi-buffer message

Table 43. Message Descriptor

Byte Offset	Field Name	Dwords	Bytes
00h	Buffer Descriptor (Host Writes) Buffer Status Descriptor (MUSYCC Writes)	1	4
04h	Data Pointer	1	4
08h	Next Pointer	1	4
	TOTAL	3	12

A message descriptor is intentionally designed to be usable by both the transmit and receive functions in MUSYCC. In providing this symmetry, a mechanism known as self-servicing buffers is available. This mechanism allows the reuse of a single descriptor for both the transmit and receive portions of a channel, and is designed for diagnostics and loopback capabilities.

Using Message Descriptors

MUSYCC checks certain data from a message descriptor before processing the associated data buffer. When a data buffer is completely processed (either transmitted or received), MUSYCC overwrites the buffer descriptor field (the first dword in a message descriptor) with a buffer status descriptor.

The buffer status descriptor specifies the number of bytes transferred, an end of message indicator, and a buffer owner-bit indicator which assigns control of associated buffers back to the host.

The owner-bit mechanism transfers control of a data buffer between MUSYCC and the host. The message descriptor may be assigned before an associated data buffer is allocated in memory. In this case, MUSYCC is instructed to poll the contents of the buffer descriptor until the host grants ownership of a data buffer to MUSYCC. After MUSYCC processes the data buffer, it grants the ownership back to the host.

The owner-bit mechanism prevents MUSYCC from processing the same buffer twice without intervention from the host. If MUSYCC detects an opening flag of a received message but does not have ownership of the current data buffer (via the current message descriptor), then an interrupt is generated toward the host indicating that MUSYCC needed a data buffer and did not have access to one.

Additionally, the host may append additional information beyond the end of a data buffer as long as the longest message length can be fitted first into the data buffer. In the case of additional information, MUSYCC would not know about the information nor would it ever read from or write to that space.

For simplicity, the message level descriptions that follow are made in reference to one channel. Each channel is serviced independently of another channel, and separate descriptor lists are maintained for each supported channel.

Similarly, the transmit section and the receive section of a channel service the descriptor lists identically, and separate descriptor lists are maintained for each section. Also, the size of data fields in the descriptors are identical, however, the layout of fields between receive and transmit descriptors are different.

Head Pointer

This pointer points to the first message descriptor of a list of descriptors assigned to a channel's transmitter or receiver.

Using a head pointer mechanism allows the host to specify a new list of descriptors to be used for channel processing. This mechanism may be used after a reset to kick-start or reactivate channel processing which has completely processed the current list of descriptors.

A head pointer mechanism also allows the host to generate a new list of descriptors in memory before performing a list transition. That is, while MUSYCC works on processing data in one list, the host can be working on processing data in a separate list, and when appropriate, the host can switch the lists.

Table 44. Head Pointer

Bit Field	Name	Value	Description
31:2	HEADPTR[29:0]		These 30 bits are appended with 00b to form a dword aligned 32-bit address. This address points to the first Message Descriptor in a list of descriptors.
1:0	HEADPTR[1:0]	0	Ensures dword alignment.

Message Pointer

This pointer points to the current message descriptor being serviced. This pointer is maintained in a fixed memory location relative to a Group Base Pointer in shared memory.

Table 45. Message Pointer

Bit Field	Name	Value	Description
31:2	MSGPTR[29:0]		These 30 bits are appended with 00b to form a dword aligned 32-bit address. This word pointer points to the first dword of a Message Descriptor.
1:0	MSGPTR[1:0]	0	Ensures dword alignment.

Message Descriptor

This descriptor is pointed to by the Message Pointer and the Head Pointer and is maintained in a variable location in shared memory. A Message Descriptor includes the following fields:

- Buffer Descriptor (when host writes)
  Buffer Status Descriptor (when MUSYCC writes)
- Data Buffer Pointer
- Next Descriptor Pointer

Buffer Descriptor

This descriptor is the first dword of a Message Descriptor after the host has prepared the data structures in memory. All Buffer Descriptors include the following fields:

- Owner Indicator Bit (OWNER)
- No Poll/Poll Indicator (NP)
- End of Buffer Interrupt Enable (EOBI)
- Buffer Length (BLEN)

The OWNER bit is a generic term for any descriptor. In a Transmit Buffer Descriptor it is called MUSYCC. In a Receive Buffer Descriptor it is called HOST. The names are different to indicate that the active sense of the owner bit is different between transmit and receive functions.

In addition to the above list of fields, Transmit Buffer Descriptors also include the following fields:

- End of Message Indicator (EOM)
- Idle Code Selection (IC)
- Pad Enable (PADEN)
- Pad Count (PADCNT)
- Repeat Packet Enable (REPEAT)

IC, PADEN, PADCNT, and REPEAT are valid only when EOM = 1.

Table 46. Transmit Buffer Descriptor

Bit Field	Name	Value	Description
31	OWNER	0	HOST Owns Buffer. Channel is to remain in idle mode while polling this bit periodically (if NP = 0) until host relinquishes control to MUSYCC by setting OWNER = 1.
		1	MUSYCC Owns Buffer. Continue processing data buffer normally.
30	NP	0	Poll Enabled. If OWNER = 0, not MUSYCC, then poll descriptor periodically while in idle mode until OWNER = 1.
		1	Poll Disabled. If OWNER = 0, then enter idle mode and wait for a Channel Activate or Jump Service Request from host.
29	EOM	0	Data Buffer w/o End of Message.
		1	Data Buffer w/ End of Message.
28	EOBI	0	End Of Buffer Interrupt Disabled. When no more data can be taken from or put into a data buffer, an EOB interrupt is not generated.
		1	End Of Buffer Interrupt Enabled.
27		0	Reserved.
26:25	IC[1:0]	0	Idle Code Select – 7Eh
		1	Idle Code Select – FFh
		2	Idle Code Select – 00h
		3	Reserved.
24	PADEN	0	Pad Fill Disabled. One shared opening/closing flag (7Eh) is inserted before sending next message.
		1	Pad Fill Enabled. Also, see PADCNT bit-field.
23:16	PADCNT[7:0]		Pad Count. When PADEN = 1, PADCNT indicates the minimum number of idle codes to be inserted between the closing flags and the next opening flag (7Eh). If PADCNT = 2 and IC = 1, for example, MUSYCC outputs the bit pattern 7Eh..FFh..FFh..7Eh. Note that there is no indication by MUSYCC if more than PADCNT number of idle codes are inserted.
15	REPEAT	0	Repeat message transmission disabled.
		1	Repeat message transmission enabled.
14	RSVD	0	Reserved.
13:0	BLEN[13:0]		Buffer Length. The number of octets in data buffer to be transmitted. Note: In general, this would equal the allocated buffer size.

Table 47. Receive Buffer Descriptor

Bit Field	Name	Value	Description
31	OWNER	0	MUSYCC Owns Buffer. Continue processing data buffer normally.
		1	HOST Owns Buffer. Channel is to remain in idle mode while polling this bit periodically (if NP=0) until host relinquishes control to MUSYCC by setting OWNER = 0.
30	NP	0	Poll Enabled. If OWNER = 1, host owns, then poll message descriptor periodically until OWNER = 0.
		1	Poll Disabled. If OWNER = 1 then wait for a Channel Activate Service Request from host.
29	RSVD	0	Reserved.
28	EOBI	0	End Of Buffer Interrupt Disabled. When more data cannot be taken from or put into a data buffer, an EOB interrupt is not generated.
		1	End Of Buffer Interrupt enabled.
27:14	RSVD	0	Reserved.
13:0	BLEN[13:0]		Buffer Length. Actual number of received data octets might be less than this. This number indicates how many will fit into data buffer.

Buffer Status Descriptor

This descriptor contains status information regarding the servicing of a data buffer. If configured to do so, MUSYCC writes a Buffer Status Descriptor over a Buffer Descriptor. This descriptor includes the following fields:

- Owner Indicator Bit (OWNER)
- End of Message Indicator (EOM)
- Data Length (DLEN)

In addition to the above list of fields, a Receive Buffer Status Descriptor also includes the Error Status (ERROR) bit-field.

Note that Buffer Status Descriptors were designed to work with Buffer Descriptors to allow a self-servicing buffer mechanism where a transmit channel will empty a list of data buffers immediately after a receive channel fills the same buffers, all without host intervention. Note the opposite bit value for the OWNER bit-field in both transmit and receive buffer descriptors.

Table 48. Transmit Buffer Status Descriptor

Bit Field	Name	Value	Description
31	OWNER	0	Host Owns Buffer. MUSYCC relinquished control of buffer back to host. MUSYCC is done processing buffer.
		1	MUSYCC Owns Buffer. Until MUSYCC relinquishes control, the data in this descriptor is being used by MUSYCC.
30	RSVD	0	Reserved.
29	EOM	0	End Of Message Indicator. Copied from Transmit Buffer Descriptor.
		1	End Of Message Indicator. Copied from Transmit Buffer Descriptor.
28:14	RSVD	0	Reserved.
13:0	BLEN[13:0]		Buffer Length. The number of octets from data buffer transmitted. In general this would equal the allocated buffer size.

Table 49. Receive Buffer Status Descriptor (1 of 2)

Bit Field	Name	Value	Description
31	OWNER	0	MUSYCC Owns Buffer. Until MUSYCC relinquishes control, the data in this descriptor is being used by MUSYCC.
		1	Host Owns Buffer. MUSYCC relinquished control of buffer back to host. MUSYCC is done processing buffer.
30	RSVD	0	Reserved.
29	EOM	0	End Of Message Indicator. The last octet for this message is not in this buffer.
		1	End Of Message indicator. The last octet for this data message is in this buffer either because a valid closing flag (7Eh) was detected or the receiver terminated due to an error condition.
28:20	RSVD	0	Reserved.

Table 49. Receive Buffer Status Descriptor *(2 of 2)*

Bit Field	Name	Value	Description
19:16	ERROR[3:0]	0	OK: No error detected in this receive buffer.
		1	BUFF: Buffer Error. Data is lost. For receive: Internal data buffer overflow.
		2	COFA: Change Of Frame Alignment. TSYNC or RSYNC signal is misaligned with the fly-wheel in the serial interface.
		3–7	Reserved.
		8	OOF: Out of Frame. ROOF signal is asserted.
		9	FCS: Frame Check Sequence Error. Received HDLC frame is terminated with proper 7Eh flag, but computed FCS does not match received FCS.
		10	ALIGN: Octet Alignment Error. Message payload size, after zero extraction, is not multiple of 8 bits. This error takes precedence over an FCS error.
		11	ABT: Abort Flag Termination. Received message is terminated with abort sequence, seven sequential ones, instead of a closing flag (7Eh).
		12	LNG: Long Message. Message payload size greater than selected limit was received. Message processing is terminated and transfer to shared memory is discontinued. Channel resumes scanning for HDLC flags or idle codes.
		13–15	Reserved.
15:14	RSVD	0	Reserved.
13:0	DLEN[13:0]		Received Octets.

Next Message Pointer

A Next Message Pointer is a 32-bit dword aligned address that points to the first dword of a Message Descriptor which is next in a list of descriptors.

Table 50. Next Descriptor Pointer

Bit Field	Name	Value	Description
31:2	NEXTPTR[29:0]		These 30 bits are appended with 00b to form a dword aligned 32-bit address. This address points to the next message descriptor in the list.
1:0	NEXTPTR[1:0]	0	Ensures dword alignment.

Data Buffer Pointer

A Data Buffer Pointer is a 32-bit address to the first byte of a data message in shared memory. This pointer does not have to be word or dword aligned.

Table 51. Data Pointer

Bit Field	Name	Value	Description
31:0	DATAPTR[31:0]		The 32-bit address in this descriptor serves as a byte pointer to the first octet of a data buffer.

Message Descriptor Handling

The bit-fields Inhibit Buffer Status Descriptor (INHTBSD for transmitters or INHRBSD for receivers) in the Group Configuration Descriptor specify whether or not MUSYCC writes a Buffer Status Descriptor to shared memory after the end of the current message has been detected.

If INHTBSD/INHRBSD is set to 0, MUSYCC:

- Assumes the Message Pointer points to the current Message Descriptor.
- Overwrites the Buffer Descriptor field with the Buffer Status Descriptor field.
- Fetches Next Message Pointer from the descriptor.
- Reads in the next Message Descriptor
- Writes the pointer to the new descriptor into the Message Pointer in shared memory.

If INHTBSD/INHRBSD is set to 1, MUSYCC:

- Assumes the Message Pointer points to the next Message Descriptor.
- Reads in the next Message Descriptor
- Writes the Next Message Pointer from the descriptor into the Message Pointer in shared memory.

Interrupt Level Descriptors

MUSYCC generates interrupts for a variety of reasons. Interrupts are events or errors detected by MUSYCC during bit-level processing of the incoming serial data streams. Interrupts are generated by MUSYCC and forwarded to the host for servicing. Individual types of interrupts may be masked from being generated by setting the appropriate interrupt mask or interrupt disable bit-fields in various descriptors. The interrupt mechanism, each individual interrupt, and interrupt controlling mechanisms are discussed in this section.

Interrupt Queue Descriptor

MUSYCC employs a single Interrupt Queue Descriptor to communicate interrupt information to the host. This descriptor is stored within MUSYCC in an internal register. The descriptor in this register space stores the location and the size of an interrupt queue in shared memory. MUSYCC requires this information to transfer interrupt descriptors it generates to shared memory for the host to use. MUSYCC writes Interrupt Descriptors directly into the shared memory queue using PCI bus master mode; MUSYCC's PCI interface must be configured to allow bus mastering.

The Interrupt Queue Descriptor is initialized by the host issuing a service request to MUSYCC to read of a copy of the Interrupt Queue Descriptor from shared memory. Another method of initialization is for the host to directly write the information into the appropriate register space within MUSYCC.

Table 52. Interrupt Queue Descriptor

Byte Offset	Field Name	Dwords	Octets
00h	Interrupt Queue Pointer	1	4
04h	Interrupt Queue Length	1	4
TOTAL		2	8

Table 53. Interrupt Queue Pointer

Bit Field	Name	Value	Description
31:2	IQPTR[30:0]		These 30 bits are appended with 00b to form a dword aligned 32-bit address. This address points to the first word of the Interrupt Queue buffer.
1:0	IQPTR[1:0]	0	Ensures dword alignment.

Table 54. Interrupt Queue Length

Bit Field	Name	Value	Description
31:15	RSVD	0	Reserved.
14:0	IQLEN[14:0]		This 15-bit number specifies the length of the Interrupt Queue buffer in dwords. The maximum size for an interrupt queue is 32,768 dwords. This is a 0-based number. A value of 1 indicates the queue length is 2 descriptors long, the required minimum.

Interrupt Descriptor

The Interrupt Descriptor describes the format of data transferred into the queue. This 32-bit word includes fields for:

- Identifying source of interrupt from within MUSYCC. Channel group number (0–3), channel number (0–31), and direction (Rx or Tx) are provided. There are 256 possible channel sources.
- Events that assist host in synchronizing channel activities.
- Errors and unexpected conditions that may have lost data, discontinued message processing, or prevented successful completion of a service request.
- Number of bytes transferred to or from shared memory for cases where a memory buffer has been completely processed.

All interrupts are associated with a channel group, channel number, and direction of the channel with the following exceptions:

- When an OOF or COFA condition is detected on a serial port, only one interrupt is generated for the entire group until the condition is cleared and the condition reoccurs. The group is identified by the GRP field and the direction is identified by the DIR field. The CH field is the channel number currently being serviced when this condition is detected.
- The ILOST interrupt bit indicates that interrupt(s) have been lost internally due to a lack of internal queuing space. This occurs when MUSYCC generates more Interrupt Descriptors than can be stored in the Interrupt Queue in shared memory. The latency of host processing of the Interrupt Queue can also be a factor. This condition is conveyed by MUSYCC overwriting the ILOST bit-field in the last interrupt descriptor in an internal queue prior to being transferred out to shared memory. The bit-field is not specific to or associated with the Interrupt Descriptor being overwritten. Only one bit is overwritten and the integrity of the original descriptor is maintained.
- The PERR interrupt bit indicates that a parity error was detected by MUSYCC during a PCI access cycle. This condition is conveyed by MUSYCC overwriting the PERR bit-field in the last interrupt descriptor in an internal queue prior to being transferred out to shared memory. The bit-field is not specific to or associated with the Interrupt Descriptor being overwritten. Only one bit is overwritten and the integrity of the original descriptor is maintained.

It is important to note that Interrupt Descriptors can convey certain combinations of Events and Errors - but always at most one Event and at most one Error. Always take into account that multiple information can be conveyed via a single interrupt descriptor - that is, always look at the Event, Error, ILOST and PERR fields when servicing interrupt descriptors. The following are possible combinations of events and errors:

The following items are issued separately in their own interrupt descriptors:

- Events
  - SACK
  - EOP
  - CHABT
  - CHIC
  - FREC
  - SINC
  - SDEC
  - SFILT
- Errors
  - PROT
  - SUERR

A single event listed below can combine with a single error listed below within the same interrupt descriptor:

- Events:
  - EOB
  - EOM
- Errors:
  - BUFF
  - COFA
  - ONR
  - OOF
  - FCS
  - ALIGN
  - ABT
  - LNG
  - SHT

The ILOST error is always piggy-backed onto an existing interrupt descriptor which may have an event, an error or both bit-fields set.

The PERR will always be issued as a separate interrupt descriptor.

Table 55. Interrupt Descriptor *(1 of 5)*

Bit Field	Field Name	Value	Interrupt Name	Group	Channel	Direction		Maskable	Description
						Tx	Rx		
31	DIR	0							Receive
		1							Transmit
30:29	GRP[1:0]	0–3							Group Number
28:24	CH[4:0]	0–31							Channel Number

Table 55. Interrupt Descriptor (2 of 5)

Bit Field	Field Name	Value	Interrupt Name	Group	Channel	Direction		Maskable	Description
						Tx	Rx		
23:20	EVENT[3:0]	0	NONE	√	√	√	√		No event to report in this interrupt.
		1	SACK	√	√	√	√		Service Request Acknowledge. Generated at conclusion of service request which was processed successfully. In case of an error as a result of a service request being executed, other interrupts may be generated - PERR, for example.
		2	EOB	√	√	√	√	√	End-Of-Buffer. Generated when current data buffer has been completely processed, and EOBI bit-field in associated Transmit Buffer Descriptor or Receive Buffer Descriptor is set.
		3	EOM	√	√	√	√	√	End-Of-Message. Generated when data buffer which was just processed contained last octet of message. Note: this interrupt also implies an end-of-buffer as this interrupt must occur at the end of a buffer.
		4	EOP	√	√	√	√	√	End-Of-Padfill. Generated when the pad-count is enabled with non-zero value in a transmit channel and last idle code octet is sent. This interrupt is conditioned on the end-of-padfill-enabled bit being set in the Transmit Channel Configuration Descriptor.
		5	CHABT	√	√		√	√	Change To Abort Code. Generated when a received pad fill code changes from 7Eh to FFh – abort code.
		6	CHIC	√	√		√	√	Change to Idle Code. Generated when a received pad fill code changes from FFh to 7Eh – idle code.
		7	FREC	√	√		√	√	Frame Recovery. Generated when serial port transitions from Out-Of-Frame (OOF) back to in-frame.
		8	SINC	√	√		√	√	SS7 SUERM Octet Count Increment. Generated when in SS7 mode and Signal Unit Error Rate Monitor counter increments.
		9	SDEC	√	√		√	√	SS7 SUERM Octet Count Decrement. Generated when in SS7 mode and Signal Unit Error Rate Monitor counter decrements.
		10	SFILT	√	√		√	√	SS7 Filtered Message. Generated when in SS7 mode and just received message is identical to one previous message. The current message is not written to shared memory.
		11–15							Reserved

Table 55. Interrupt Descriptor *(3 of 5)*

Bit Field	Field Name	Value	Interrupt Name	Group	Channel	Direction		Maskable	Description
						Tx	Rx		
19:16	ERROR[3:0]	0	NONE	√	√	√	√		No error to report in this interrupt.
		1	BUFF*	√	√	√	√	√	Buffer Error. Data is lost. MUSYCC has no place to read or write data internally. If from transmitter, then internal buffer underflow. If from receiver, internal buffer overflow.
		2	COFA*	√		√	√	√	Change Of Frame Alignment. Generated when serial port is configured in channelized mode and transmitter or receiver synchronization signal assertion is detected during a bit-time, and the synchronization signal is misaligned with internal flywheel mechanism in serial interface. This condition affects all channels in the group. All receive channels abort their current channel activity. Note that this interrupt is sent to host after all receive channels have been aborted. This condition is also flagged in Buffer Status Descriptors.
		3	ONR	√	√	√	√	√	Owner-bit Error. Generated when next message pointer/descriptor is not available to MUSYCC when expected. This error is similar to BUFF error except that MUSYCC has no place to read or write data in shared memory. For example, when MUSYCC can not write out a Buffer Status Descriptor.
		4	PROT	√	√	√	√	√	Memory Protection Violation. Generated when memory protection is enabled and MUSYCC attempts a PCI master mode access to an address outside the memory region specified in a group's Memory Protection Descriptor. The memory access is inhibited.
		5–7							Reserved.
		8	OOF*	√			√	√	Out Of Frame. Generated when serial port is configured in channelized mode and receiver-out-of-frame (ROOF) input signal assertion is detected.
		9	FCS*	√	√		√	√	Frame Check Sequence Error. Generated when received HDLC frame is terminated with octet aligned 7Eh flag but computed FCS does not match received FCS.

Table 55. Interrupt Descriptor (4 of 5)

Bit Field	Field Name	Value	Interrupt Name	Group	Channel	Direction		Maskable	Description
						Tx	Rx		
19:16 cont'd	ERROR[3:0] cont'd	10	ALIGN*	√	√		√	√	Octet Alignment Error. Generated when message payload size, after zero extraction, is not a multiple of 8 bits. This generally occurs with an FCS error. This interrupt also implies an FCS error. The FCS interrupt will not be generated if the ALIGN interrupt is issued.
		11	ABT*	√	√		√	√	Abort Termination. Generated when received message is terminated with an abort sequence—seven sequential ones —instead of a specific closing flag – 7Eh.
		12	LNG*	√	√		√	√	Long Message. Generated when received message length (after zero extraction) is greater than selected maximum message size. Message reception is terminated and transfer to shared memory is not performed.
		13	SHT	√	√		√	√	Short Message. Generated when received message length (after zero extraction) is less than or equal to number of bits in FCS field. The message data is not transferred to shared memory.
		14	SUERR	√	√		√	√	SS7 Signal Unit Error Rate Interrupt. Generated when in SS7 mode and error monitor, SUERM, value rises past the threshold value, SUET.
15	ILOST	0	ILOST						No interrupts have been lost.
		1							Interrupt Lost. Generated when internal interrupt queue is full and more interrupt conditions are detected. As MUSYCC has no way to store the newest interrupt descriptors, it discards the new interrupts and overwrites this bit in the last interrupt in an internal queue prior to that interrupt being transferred out to shared memory. The integrity of the descriptor being overwritten is maintained completely.
14	PERR	0	PERR					√	No parity errors have been detected.
		1							PCI Bus Parity Error. Generated when MUSYCC detects a parity error on data being transferred into MUSYCC either from another PCI agent writing into MUSYCC or from MUSYCC reading data from shared memory. This error is specific to the data phase of a PCI transfer while MUSYCC is receiving data. Note: PCI system error signal, SERR*, is ignored by MUSYCC. Note: To mask the PERR interrupt, MUSYCC's PCI Configuration Space, Function 0, Register 1, Parity Error Response field must be set to 0.

Table 55. Interrupt Descriptor *(5 of 5)*

Bit Field	Field Name	Value	Interrupt Name	Group	Channel	Direction		Maskable	Description
						Tx	Rx		
13:0	BLEN[13:0]								This field is relevant when EVENT field is EOB (Rx and Tx) or EOM (Rx only). For Rx, it is equal to the number of octets received. For Tx, it is the size of the buffer length targeted for transmission and not necessarily the number of octets transmitted. This field is 0 all other times.
Note: Interrupt names marked with an asterisk (*) indicate that these conditions are also reported in an error field within a Buffer Status Descriptor which indicates the transfer status of a message currently being processed on a channel. The order of appearance in shared memory of a Buffer Status Descriptor and an Interrupt Descriptor carrying the same error condition information is not always determinable. For interrupts marked with the asterisk, the host should confirm that both an Interrupt Descriptor and a Buffer Status Descriptor reports the error condition.									

Interrupt Status Descriptor

This descriptor is located in a fixed position within MUSYCC's internal registers. MUSYCC updates this descriptor after each transfer of Interrupt Descriptors from its internal queue to the Interrupt Queue in shared memory. The host is required to read this descriptor from MUSYCC registers before it processes any interrupts. The contents of the Interrupt Status Descriptor are reset on hardware reset or soft chip reset or whenever any field in the Interrupt Queue Descriptor is modified.

Table 56. Interrupt Status Descriptor

Bit Field	Name	Value	Description
31	RSVD	0	Reserved.
30:16	NEXTINT[14:0]		Next Interrupt Index. 15-bit dword index from start of Interrupt Queue up to where the host has serviced Interrupt Descriptors. The host may read this value to get the location of the first unserviced descriptor in the queue. As the queue is circular, care must be taken to ensure roll-over cases at beginning and end of queue. Only the host updates this value. After the host services descriptors in the queue, it must write a new NEXTINT value into this descriptor to indicate a location up to where MUSYCC may write additional descriptors without overwriting unserviced descriptors still in the queue. This is a 0-based number and equals the dword offset from Interrupt Queue Pointer.
15	INTFULL	0	Interrupt Queue Not Full—shared memory.
		1	Interrupt Queue Full—shared memory.
14:0	INTCNT[14:0]		Interrupt Count. 15-bit value indicates the number of interrupts pushed into the Interrupt Queue since the last reading of the Interrupt Status Descriptor.

Interrupt Handling

Initialization

Interrupt management resources are automatically reset upon:

- Hardware reset
- Soft-chip reset service request
- Global initialization service request
- Read Interrupt Queue Descriptor service request
- Direct memory write to Interrupt Queue Pointer
- Direct memory write to Interrupt Queue Length

MUSYCC uses two interrupt queues. One is internal to MUSYCC and is controlled exclusively by the interrupt controller logic. The other is the Interrupt Queue in shared memory which is allocated and administered by the host, but written to by MUSYCC.

Upon initialization, the data in the status descriptor is reset to all zeros, indicating first location for next descriptor, queue not full and no descriptors are currently in the queue. Any existing descriptors in the internal queue are discarded.

The Interrupt Status Descriptor stores the location of the next descriptor to be read by the host, a queue full indicator and count of interrupts last written into shared memory since the last read of the Interrupt Status Descriptor.

It is the responsibility of the host to allocate sufficient shared memory space for the Interrupt Queue. Up to 32,768 dwords of queue space are accessible by MUSYCC; this sets the upper limit for the queue size. MUSYCC requires a minimum of 2 dwords of queue space; this sets the lower limit for the queue size.

The host must store the pointer to the queue and the length in dwords of the queue in MUSYCC within the Interrupt Queue Descriptor register. This can be done using direct memory writes into MUSYCC or by issuing the appropriate service request to MUSYCC. As MUSYCC takes in the new values, it automatically resets the controller logic as indicated above. Note that this mechanism can also be used to switch interrupt queues while MUSYCC is in full operation.

Interrupt Descriptor Generation

Interrupt conditions are detected in both error and non-error cases. MUSYCC makes a determination based on channel group, channel, and device configuration whether reporting of the condition is to be masked or whether an Interrupt Descriptor is to be sent to the host. If the interrupt is not masked, then MUSYCC generates a descriptor and stores it internally prior to transfer to the Interrupt Queue in shared memory.

The internal queue is capable of holding 128 descriptors while MUSYCC arbitrates to master the PCI bus and transfer the descriptors into the Interrupt Queue in shared memory.

As the PCI bus is mastered and after descriptors are transferred out to shared memory, MUSYCC updates the Interrupt Status Descriptor. In making the INTCNT field in the descriptor non-zero, MUSYCC asserts the PCI INTA* signal line.

If during the transfer of descriptors, the Interrupt Queue in shared memory becomes full, then MUSYCC stops transferring descriptors until the host indicates that more descriptors can be written out. MUSYCC indicates that it cannot transfer more descriptors into shared memory by setting the bit-field INTFULL in the Interrupt Status Descriptor. Recall, MUSYCC has enough internal space to store 128 additional descriptors.

In cases where both shared memory queue and internal queue are full and new descriptors are generated, the new descriptors are discarded. MUSYCC indicates that it has lost interrupts internally by overwriting the bit-field ILOST in the last Interrupt Descriptor in the internal queue. The ILOST indication represents one or more lost descriptor.

INTA* Signal Line

The host must monitor the INTA* signal line at all times. The assertion of this signal always signifies that the INTCNT filed in the Interrupt Status Descriptor is non-zero. Non-zero INTCNT always signifies that Interrupt Descriptors have been written to the Interrupt Queue in shared memory.

Upon detection of the INTA* assertion, the host must perform a direct read of the Interrupt Status Descriptor from within MUSYCC. This descriptor provides the offset to the location of the first unserviced descriptor in the queue, the number of unserviced descriptors, and determines if the queue is full.

The INTCNT field is reset on each read of the Interrupt Status Descriptor. As the INTCNT is reset, the INA* signal is deasserted.

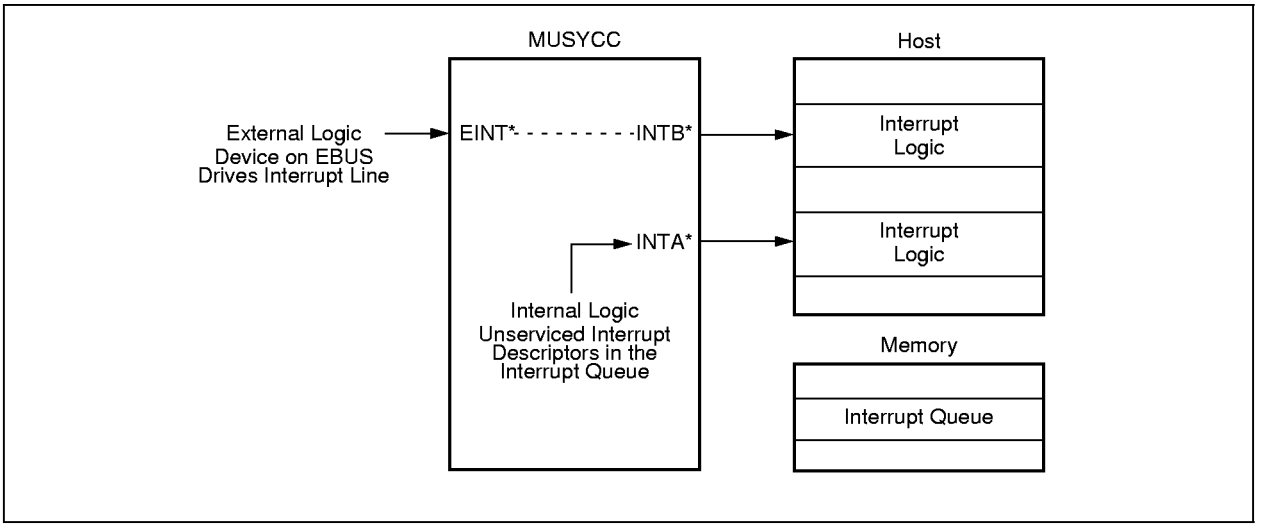
The host applies its interrupt service routines to service each of the descriptors. As the host finishes servicing a number of descriptors, it is required to write the offset to the location of the last serviced descriptor back into the Interrupt Status Descriptor: the NEXTINT. A write to this field indicates to MUSYCC that the descriptor locations previously unserviced now have been serviced and new descriptors can be written. Note that MUSYCC continues to write to available space regardless of whether the host updates the NEXTINT field.

It is critically important to note that after reading the Interrupt Status Descriptor, the host is solely responsible for servicing all unserviced descriptors (count of INTCNT starting at NEXTINT) in the queue at the time of the read. If the host is unsuccessful in servicing this set of descriptors, then the host must provide an alternate method to keep track of unserviced descriptors. Every read of the status descriptors only provides information on new descriptors placed in the queue autonomously by MUSYCC since the last time the status descriptor was read.

INTB* Signal Line

A second interrupt signal line, the PCI INTB* signal line, is asserted by MUSYCC when it detects the assertion of the EBUS EINT* signal line. MUSYCC does not generate descriptors or use the interrupt queue for this condition because it does not know the source or the reason for the interrupt; the reason is external to MUSYCC. This signal acts as an interrupt line passthrough for devices connected to the EBUS. The EINT* signal line may be tied to interrupt output pin(s) of one or more peripheral devices. As MUSYCC detects EINT* assertion, MUSYCC asserts the INTB* towards the host as long as the EINT* remains asserted. The figure below illustrates the operation of EINT*.

Figure 22. Interrupt Notification To Host



Reset

There are several levels of reset state. These are:

- Hard PCI Reset
- Soft Chip Reset
- Soft Group Reset
- Channel Activation
- Channel Deactivation

There are two ways to assert a reset:

1. Assert the PCI reset signal pin, PRST*.
2. Assert a service request through the host interface to perform the soft chip reset, soft group reset, or channel activation, or channel deactivation.

Hard PCI Reset

The PCI reset is the most thorough level of reset in MUSYCC. All subsystems enter into their initial states. PCI reset is accomplished by asserting the PCI signal, PRST*.

The PRST* signal is an asynchronous signal on the PCI bus. The reset signal can be activated in several ways. The system must always assert the reset signal on power-up. Also, a host bus to PCI bus bridging device should provide a way for software to assert the reset signal. Additionally, software-controlled circuitry can be included in the system design to specifically assert the reset signal on demand.

Asserting PRST* towards MUSYCC guarantees that data transfer operations and PCI device operations will not commence until after MUSYCC has been properly initialized for operation. Upon entering PCI reset state, MUSYCC outputs a three-stated signal on all output pins and halts activity on all subsystems including the host interface, serial interface, and expansion bus

The effects of a PCI reset signal within MUSYCC will take one PCI clock cycle to complete. After this time the host may communicate with MUSYCC using the PCI configuration cycles.

Soft Chip Reset

A soft chip reset is a device-wide reset without the host interface's PCI state being reset. Serial interface operations and EBUS operations are halted. The soft chip reset state is entered in one of two ways:

- As a result of the PCI reset
- As a result of a soft chip reset service request.

A soft chip reset causes:

- Transmit data signals, TDAT, to be three-stated.
- EBUS address-data lines to be three-stated and read enable and write enable outputs to be deasserted; halting all memory operations on EBUS.
- Every channel-direction in all supported channel groups to enter the channel deactivated state.
- DMA controllers to be reset; halting all PCI transactions.

When a soft chip reset is requested by the host, the service request mechanism is used. Normally, every service request is acknowledged by MUSYCC with a Service Request Acknowledge (SACK) Interrupt Descriptor. However, in this case, the host must follow up the reset request with another non-reset service request to effectively remove MUSYCC from the soft chip reset state.

It is recommended that the host either allow two line clock periods of the clock connected to the associated serial port to elapse for this reset to complete before issuing any other service request or periodically poll the transmit and receive line clock activity indicator bit-fields, TCLKACTx and RCLKACTx in Table 30, Global Configuration Descriptor, to see if the serial interface is active.

When MUSYCC acknowledges a follow up service request with a SACK, the device is out of the reset state.

Soft Group Reset

Every supported channel group within MUSYCC has the ability to reset (or deactivate) a specific direction for all the channels in the group using a single service request: the soft group reset service request.

When a soft group reset is requested, a direction (either transmit or receive) is specified in the request and all channels in the specified group and direction are deactivated. For the transmit direction, the output signal, TDAT, is three-stated.

It is recommended that the host allow 2 line clock periods of the clock connected to the associated serial port to elapse for this reset to complete before issuing any other service request.

When a soft group reset is requested by the host, the service request mechanism is used. Normally, every service request is acknowledged by MUSYCC with a SACK Interrupt Descriptor. In this case, the host must follow up the reset request with another non-reset service request to effectively remove the channel group from the soft group reset state. When MUSYCC acknowledges the follow up service request with a SACK, the channel group is out of the reset state.

Configuration

A sequence of hierarchical initialization must occur after resets. The levels of hierarchy are:

- PCI Configuration—only after hardware reset
- Global Configuration
- Interrupt Queue Configuration
- Channel Group(s) Configuration

PCI Configuration

After power-up or a PCI reset sequence, MUSYCC enters a holding pattern waiting for PCI configuration cycles directed specifically for MUSYCC, actually directed at the PCI bus and PCI slot that MUSYCC resides in.

PCI configuration involves PCI read and write cycles. These cycles are initiated by the host and performed by a host-bus-to-PCI-bus bridge device. The cycles are executed at the hardware signal level by the bridge device. The bridge device polls all possible slots on the bus it controls for a PCI device and then iteratively reads the configuration space for all supported functions on each device. All information from the basic configuration sequence is forwarded to the system controller or host processor controlling the bridge device.

During PCI configuration, the host may perform the following configuration for MUSYCC's Function 0, HDLC Network Controller function:

- Read PCI configuration space (Device Identification, Vendor Identification, Class Code, and Revision Identification).
- Allocate 1 Mbyte system memory range and assign the Base Address Register using this memory range.
- Allow fast back-to-back transactions.
- Enable PCI system error signal line, SERR*.
- Allow response for PCI parity error detection.
- Allow PCI bus-master mode.
- Allow PCI bus-slave mode.
- Assign latency.
- Assign interrupt line routing.

During PCI configuration, the host may perform the following configuration for MUSYCC's Function 1, PCI to EBUS bridge:

- Read PCI configuration space (Device Identification, Vendor Identification, Class Code, and Revision Identification).
- Allocate 1 Mbyte system memory range and assign the Base Address Register using this memory range.
- Allow response for PCI parity error detection.
- Assign latency.
- Assign interrupt line routing.

Global Configuration

After PCI configuration is complete, a set of hierarchical configuration sequences must be executed to begin operation at the channel level. Global configuration is initiated by the host issuing a service request or by the host performing slave writes into MUSYCC resident Global Configuration Descriptor.

Global configuration specifies information used across the entire device including all supported channel groups, all channels and the EBUS.

The relevant references are:

- Table 30, Global Configuration Descriptor

NOTE: Device identification at the PCI Level Configuration must be used to identify the number of supported channel groups and channels in MUSYCC - which in turn will affect how MUSYCC is eventually configured.

Interrupt Queue Configuration

Part of global configuration involves interrupt queue configuration.

The relevant references are:

- Table 52, Interrupt Queue Descriptor

Channel Group(s) Configuration

After global configuration, more specific group configuration must be performed for each supported channel group.

The relevant references are:

- Table 34, Group Configuration Descriptor
- Table 35, Memory Protection Descriptor
- Table 37, Message Length Descriptor
- Table 36, Port Configuration Descriptor
- Table 38, Transmit or Receive Timeslot Map
- Table 40, Transmit or Receive Subchannel Map
- Table 42, Channel Configuration Descriptor

Service Request Mechanism

The service request mechanism requires the host to perform a direct memory write operation (slave write) into the appropriate channel group's service request descriptor which is within MUSYCC's internal registers.

The relevant references are:

- Table 25, MUSYCC Register Map
- Table 33, Service Request Descriptor

MUSYCC Internal Memory

MUSYCC has two areas of host accessible internal memories. One of the area is referred to as the Internal RAM (IRAM) and is accessed through MUSYCC's Direct Memory Access Controller (DMAC). The IRAM area contains the following descriptors and maps:

- Table 38, Transmit or Receive Timeslot Map
- Table 40, Transmit or Receive Subchannel Map
- Table 42, Channel Configuration Descriptor (transmit or receive)

A second area of internal memories makes up the Host Interface Registers. This area is not accessed through MUSYCC's DMAC. The Host Interface Register area contains the following descriptors:

- Table 30, Global Configuration Descriptor
- Table 31, Dual Address Cycle Base Pointer
- Table 32, Group Base Pointer
- Table 33, Service Request Descriptor
- Table 52, Interrupt Queue Descriptor
- Table 34, Group Configuration Descriptor
- Table 35, Memory Protection Descriptor
- Table 37, Message Length Descriptor
- Table 36, Port Configuration Descriptor
- Table 38, Transmit or Receive Timeslot Map
- Table 40, Transmit or Receive Subchannel Map
- Table 42, Channel Configuration Descriptor

Read Operation—Slave Reads

Read operations to any of the IRAMs listed above may affect the operation of a currently active channels. It is recommended that reads from IRAM be used for diagnostic purposes only and/or with all channels deactivated. Also, IRAMs are readable only if the corresponding channel group's line clocks (TCLK, RCLK) are active. Reading RAMs without line clocks active results in the pattern DEAD ACCEh being returned - conveying "dead access".

Read operations to the Host Interface Registers can occur at any time and do not affect the operation of active channels.

Read operations are started by the host processor accessing the memory map of MUSYCC registers through PCI read access cycles.

Read operations to invalid (unsupported or reserved) addresses or write-only registers result in all ones being returned.

Write Operation—Slave Writes

Write operations to the IRAM are allowed and do not affect the operation of active channels. Internal RAMs are writable only if the corresponding channel group's line clock (TCLK, RCLK) are active. Writing RAMs without line clocks active results in the writes being ignored.

Write operations to the Host Interface Registers can occur at any time and do not affect the operation of active channels.

Write operations to invalid (unsupported or reserved) addresses or read-only register bits results in the write to that bit location being ignored.

Channel Operation

To start any channel processing a series of shared memory segments must be obtained by the host and initialized as specific descriptors which MUSYCC can use to control its channel processing operations.

To illustrate the required MUSYCC configuration, assume the following:

- Port 0 physically wired to an PCM carrying E1 signal (2.048 Mbps)
- EBUS not used.
- PCI config in Table 27, MUSYCC PCI Function Memory Allocation
- Memory Protection enabled for range 0x00100000 to 0x001FFFFF
- Application:

  Port 0 configured for 32 channel operation, E1 signal, 2.048 Mbps
  Transmit and Receive timeslots mapped identically.
  Timeslot 0 map to logical channel 0 (64 Kbps)
  Timeslot 1 bits 0-3 map to logical channel 1 (32 Kbps subchannel)
  Timeslot 2-3 mapped to logical channel 2 (128 Kbps hyperchannel)
  16-bit FCS HDLC
  Maximum message length 1024 octets for channel 0
  Maximum message length 512 octets for channel 1
  Maximum message length check disabled for channel 2
  No SS7 functions
  Idle Code = 7Eh
  Pad Fill Count = 0
  If feature not specified here, accept reset value (generally 0)

- C-Language support
- Each section below builds on the previous sections

Group Structure

A group structure (one per supported group) is required to be allocated in shared memory. A data structure is extremely instrumental in keeping the memory spaces for the various descriptors required for a Channel Group configuration in a sequential order and at exact offsets from the beginning of the group structure.

Once a group structure is allocated in shared memory, then by definition, all descriptor spaces are allocated within the group structure.

Service request handling within MUSYCC requires the group structure and the descriptor contents be at an exact offsets within the structure.

For this example, the following group structure declaration is used:

```
/* Reference: Chapter "Memory Organization" */

#define SIZE_OF_GROUP_STRUCTURE 1564
#define NUM_GROUPS 1
#define BOUNDARY 2048

#define MUSYCC_FUNC_0_BAR 0x00900000 /* system usually assigns this ............*/

/* declare variable ..........................................................*/

typedef struct tGROUP_STRUCTURE
{
unsigned long *pGroupBase;
unsigned long *pDualAddressCycleBase;
unsigned long ServiceRequestDescr;
unsigned long InterruptStatusDescr;
unsigned char TxTimeslotMap[128];
unsigned char TxSubchannelMap[256];
unsigned char TxChannelConfigDescr[128];
unsigned char RxTimeslotMap[128];
unsigned char RxSubchannelMap[256];
unsigned char RxChannelConfigDescr[128];
unsigned long GlobalConfigDescr;
unsigned long InterruptDescr[2];
unsigned long GroupConfigDescr;
unsigned long MemoryProtectDescr;
unsigned long MessageLengthDescr;
unsigned long PortConfigDescr;
} tGROUP_STRUCTURE;

/* IMPORTANT NOTE: Byte padding within the structure would cause descriptor ....*/
/* offsets from the beginning of the structure to move. MUSYCC requires every ..*/
/* offset to be fixed at all times. Byte padding is an automatic function of ...*/
/* many compilers. ...........................................................*/

/* allocate space ............................................................*/

tGROUP_STRUCTURE GroupStr0; /* one per supported group */
```

```
/* fixed descriptor offsets into the group structure.........................*/

#define GROUP_BASE_OFFSET..............0x00000000
#define DUAL_ADDRESS_CYCLE_BASE_OFFSET..0x00000004
#define SERVICE_REQUEST_OFFSET.........0x00000008
#define INTERRUPT_STATUS_OFFSET........0x0000000C
#define TX_TIMESLOT_MAP_OFFSET.........0x00000200
#define TX_SUBCHANNEL_MAP_OFFSET........0x00000280
#define TX_CHANNEL_CONFIG_DESCR_OFFSET..0x00000380
#define RX_TIMESLOT_MAP_OFFSET.........0x00000400
#define RX_SUBCHANNEL_MAP_OFFSET........0x00000480
#define RX_CHANNEL_CONFIG_DESCR_OFFSET..0x00000580
#define GLOBAL_CONFIG_DESCR_OFFSET......0x00000600
#define INT_QUEUE_DESCR_OFFSET.........0x00000604
#define INT_QUEUE_POINTER_OFFSET........0x00000604
#define INT_QUEUE_LENGTH_OFFSET........0x00000608
#define GROUP_CONFIG_DESCR_OFFSET.......0x0000060C
#define MEMORY_PROTECT_DESCR_OFFSET.....0x00000610
#define MESSAGE_LENGTH_DESCR_OFFSET.....0x00000614
#define PORT_CONFIG_DESCR_OFFSET........0x00000618
```

Group Base Pointer

For the Group Base Pointer the host must allocate a 2K-byte bound memory segment for Channel Group 0. The value calculated as the address for a Group Base Structure must be written into MUSYCC's Group 0 Base Pointer register using a PCI write cycle - after PCI configuration, this is a simple memory access by the host.

A service request does not exist to update this pointer in MUSYCC due to the fact that all service requests reference this pointer value to gain access to the shared memory resident group structure and the descriptors within it.

```
#define SIZE_OF_GROUP_STRUCTURE 1564
#define GROUP_STR_BOUNDARY 2048

GroupStr0.pGroupBase = malloc( SIZE_OF_GROUP_STRUCTURE + GROUP_STR_BOUNDARY );
GroupStr0.pGroupBase =
...........( GroupStr0.pGroupBase + GROUP_STR_BOUNDARY) & ~(GROUP_STR_BOUNDARY -1);

/* group base pointer pointers must be a 2K byte aligned address ..............*/
/* above, there is enough space to first move forward 2K bytes, then ...........*/
/* lop off the 2K automatically. This will bring the pointer back ..............*/
/* to the original address or give us the next 2K byte boundary address.........*/

/* IMPORTANT NOTE: be sure to save away the original pointer returned by the ...*/
/* memory access routine as that same value will be required to free the space. */

/* must write directly into MUSYCC register ....................................*/
*(MUSYCC_FUNC_0_BAR + GROUP_BASE_OFFSET) = GroupStr0.pGroupBase;
```

Descriptor	Component Of Descriptor	Value Of Components
Group Base Pointer	Pointer to a shared memory segment large enough for all configuration descriptors for Channel Group 0.	Return pointer from "malloc()" adjusted to a 2K boundary.

Global Configuration Descriptor

```
/* CLOCK activity indicators - read only, writes are ignored ..................*/
/* MPU control - assume EBUS is not used and default values are fine............*/
/* PORTMAP = 0, PORT 0 mapped to CHANNEL GROUP 0 ..............................*/
GroupStr0.GlobalConfigDescr = 0x00000000;

/* either write directly into MUSYCC register - or - use a service request .....*/
*(MUSYCC_FUNC_0_BAR + GLOBAL_CONFIG_DESCR_OFFSET) = GroupStr0.GlobalConfigDescr;
```

Descriptor	Component Of Descriptor	Value Of Components
Global Configuration Descriptor	TXCLKACT0 TXCLKACT1 TXCLKACT2 TXCLKACT3 RXCLKACT0 RXCLKACT1 RXCLKACT2 RXCLKACT3	Read only values. Writes are ignored. It would be ideal to read this descriptor first and verify that MUSYCC detects clocks at the Rx and Tx ports being used.
	BLAPSE ECKEN MPUSEL ALAPSE ELAPSE	0, don't care 0, EBUS clock output disabled. 0, Motorola-style protocol supported 0, don't care 0, don't care
	PORTMAP	use 0 where the mapping is defined as: Port 0 -> Channel Group 0 Port 1 -> Channel Group 1 Port 2 -> Channel Group 2 Port 3 -> Channel Group 3

Interrupt Queue Descriptor

```
#define BYTES_PER_INT_DESCR 4 /* recall, dword = 32-bits or 4 bytes .....*/
#define NUM_INT_DESCR_NEEDED 10 /* assumption (min = 2, max = 32768 ).....*/
#define SIZE_OF_INTERRUPT_QUEUE (BYTES_PER_INT_DESCR * NUM_INT_DESCR_NEEDED)
#define INT_QUEUE_BOUNDARY 4

/* local variables ..........................................................*/
unsigned long *pIntQueue;
unsigned long IntQueueLen;

pIntQueue = malloc( SIZE_OF_INTERRUPT_QUEUE + INT_QUEUE_BOUNDARY );
pIntQueue = (pIntQueue + INT_QUEUE_BOUNDARY ) & ~(INT_QUEUE_BOUNDARY - 1);

/* interrupt queue pointers must be a dword aligned address .................*/
/* above, there is enough space to first move forward 4 bytes, then .....*/
/* lop off the 4 automatically. This will bring the pointer back ...........*/
/* to the original address or give us the next 4 byte boundary address.....*/

/* IMPORTANT NOTE: be sure to save away the original pointer returned by the ...*/
/* memory access routine as that same value will be required to free the space. */
IntQueueLen = NUM_INT_DESCR_NEEDED - 1; /* 0-based ..........................*/

GroupStr0.IntQueueDescr[0] = pIntQueue;
GroupStr0.IntQueueDescr[1] = IntQueueLen;

/* either write directly into MUSYCC register - or - use a service request .....*/
/* for this descriptor, 2 dwords need to be written, so 2 write accesses........*/
*(MUSYCC_FUNC_0_BAR + INT_QUEUE_POINTER_OFFSET)= GroupStr0.IntQueueDescr[0];
*(MUSYCC_FUNC_0_BAR + INT_QUEUE_LENGTH_OFFSET) = GroupStr0.IntQueueDescr[1];
```

Descriptor	Component Of Descriptor	Value Of Components
Interrupt Queue Descriptor	IQPTR	pIntQueue, provided by memory allocation functions and adjusted to be dword bound
	IQLEN	IntQueueLen, specified by #define (0-based)

Group Configuration Descriptor

```
/* signal unit error threshold = 0, no SS7 support required.....*/
/* sf alignment = 0, use internal flywheel mechanism after initial frame sync...*/
/* poll throttle = 0, poll every frame.....*/
/* inhibit tx bsd = 0, do not inhibit.....*/
/* inhibit rx bsd = 0, do not inhibit.....*/
/* memory protection violation action = 0, reset group on detection
/* message config bit copy = 0, disable.....*/
/* mask cofa interrupt = 0, do not mask.....*/
/* mask oof interrupt = 0, do not mask.....*/
/* oof message processing = 0, continue processing incoming messages.....*/
/* subchanneling = 0, enabled.....*/
/* transmitter enabled = 1, enabled.....*/
/* receiver enabled = 1, enabled.....*/
GroupStr0.GroupConfigDescr = 0x00000003;

/* either write directly into MUSYCC register - or - use a service request .....*/
*(MUSYCC_FUNC_0_BAR + GROUP_CONFIG_DESCR_OFFSET) = GroupStr0.GroupConfigDescr;
```

Descriptor	Component Of Descriptor	Value Of Components
Group Configuration Descriptor	SUET	0, SS7 not being used
	SFALIGN	0, use internal flywheel mechanism
	POLLTH	0, poll buffer ownership every frame per channel
	INHTBSD	0, allow tx buffer status descriptor writes
	INHRBSD	0, allow rx buffer status descriptor writes
	MEMPVA	0, on memory protection violation, reset group.
	MCENBL	0, message configuration bits copy disabled
	MSKCOFA	0, do not mask COFA interrupt
	MSKOOF	0, do not mask OOF interrupt
	OOFABT	0, on OOF detection, continue processing channel
	SUBDSBL	0, subchanneling enabled
	TXENBL	1, transmitter enabled
	RXENBL	1, receiver enabled

Memory Protection Descriptor

```
/* memory protection enabled = 1.............................................*/
/* memory protection lo 1 Mbyte bound = 0x00100000...........................*/
/* memory protection hi 1 Mbyte bound = 0x00100000 (protects up to 0x001FFFFF...*/
GroupStr0.MemoryProtectDescr = 0x80010001;

/* either write directly into MUSYCC register - or - use a service request .....*/
*(MUSYCC_FUNC_0_BAR + GROUP_CONFIG_DESCR_OFFSET) = GroupStr0.MemoryProtectDescr;
```

Descriptor	Component Of Descriptor	Value Of Components
Memory Protection Descriptor	PROTENBL	1, memory protection enabled
	PROTHI	001h, upper 12-bits of high memory 1Mbyte
	PROTLO	001h, upper 12-bits of low memory 1Mbyte

Port Configuration Descriptor

```
/* three-state transmitter output = 0........................................*/
/* rx out of frame signal active edge = 0, falling edge.....................*/
/* rx synchronization signal active edge = 0, falling edge..................*/
/* rx data signal active edge = 0, falling edge.............................*/
/* tx synchronization signal active edge = 0, falling edge..................*/
/* tx data signal active edge = 0, falling edge.............................*/
/* port mode = 1, E1 32 timeslot............................................*/
GroupStr0.PortConfigDescr = 0x00000001;

/* either write directly into MUSYCC register - or - use a service request .....*/
*(MUSYCC_FUNC_0_BAR + PORT_CONFIG_DESCR_OFFSET) = GroupStr0.PortConfigDescr;
```

Descriptor	Component Of Descriptor	Value Of Components
Port Configuration Descriptor	TRITX	0, three-state output when transmitter enabled and timeslot is not mapped
	ROOF_EDGE RSYNC_EDGE RDAT_EDGE TSYNC_EDGE TDAT_EDGE	0, active edge of signals is falling edge
	PORTMD	1, 32 channel with E1 signalling

Message Length Descriptor

```
/* maximum frame length register 2 = 0x200.....*/
/* maximum frame length register 1 = 0x400.....*/
GroupStr0.MessageLengthDescr = 0x02000400;

/* either write directly into MUSYCC register - or - use a service request .....*/
*(MUSYCC_FUNC_0_BAR + MESSAGE_LENGTH_DESCR_OFFSET) = GroupStr0.MessageLengthDescr;
```

Descriptor	Component Of Descriptor	Value Of Components
Message Length Descriptor	MAXFRM2	0x200, register 2 to 512 octets
	MAXFRM1	0x400, register 1 to 1024 octets

Transmit Timeslot Map—Channel 0

```
/* each timeslot descriptor contains 4 timeslot assignments.....................*/
/* each byte in the dword descriptor is a timeslot assignment...................*/
/* byte 0/dword 0 is for timeslot 0.............................................*/
/* byte 1/dword 0 is for timeslot 1.............................................*/
/* byte 2/dword 0 is for timeslot 2.............................................*/
/* byte 3/dword 0 is for timeslot 3.............................................*/
/* for demonstration, assign each byte separately...............................*/
GroupStr0.TxTimeslotMap[0] = 0; /* zero it out for demo purposes................*/

/* timeslot 0, channel number assigned = 0......................................*/
/* timeslot 0, timeslot enabled code = 4, enabled and 64 Kbps...................*/
GroupStr0.TxTimeslotMap[0] |= 0x00000080;

/* timeslot 1, channel number assigned = 1......................................*/
/* timeslot 1, timeslot enabled code = 7, subchannel w/ 1st bit active..........*/
GroupStr0.TxTimeslotMap[1] |= 0x0000D100;

/* timeslot 2, channel number assigned = 2......................................*/
/* timeslot 2, timeslot enabled code = 4, enabled and 64 Kbps...................*/
GroupStr0.TxTimeslotMap[2] |= 0x00820000;

/* timeslot 3, channel number assigned = 2......................................*/
/* timeslot 3, timeslot enabled code = 4, enabled and 64 Kbps...................*/
GroupStr0.TxTimeslotMap[3] |= 0x82000000;

/* either write directly into MUSYCC register - or - use a service request .....*/
*(MUSYCC_FUNC_0_BAR + TX_TIMESLOT_MAP_OFFSET) = GroupStr0.TxTimeslotMap[0];

/* the value for the first dword becomes Tx Timeslot Map = 0x8282D180 ..........*/
```

Descriptor	Component Of Descriptor	Value Of Components
Timeslot Map	TSEN3/7	4, timeslot enabled w/ 64 Kbps
	CH3/7	2, logical channel 2
	TSEN2/6	4, timeslot enabled w/ 64 Kbps
	CH2/6	2, logical channel 2
	TSEN1/5	7, timeslot enabled w/ subchanneling and 0-bit
	CH1/5	1, logical channel 1
	TSEN0/4	4, timeslot enabled w/ 64 Kbps
	CH0/4	0, logical channel 0

Transmit Subchannel Map

```
/* each subchannel descriptor is made up of 2 dwords.........................*/
/* dword 0 defines the configuration of 1st 4 subchannel bits................*/
/* dword 1 defines the configuration of 2nd 4 subchannel bits................*/
/* for demonstration, assign each byte separately............................*/
/* for this example, only logical channel 1 is being subchanneled............*/
/* dword 0 and 1, for logical channel 0......................................*/
/* dword 2 and 3, for logical channel 1......................................*/
/* dword 4 and 5, for logical channel 2......................................*/
GroupStr0.TxSubchannelMap[4] = 0; /* zero it out for demo purposes............*/
GroupStr0.TxSubchannelMap[5] = 0; /* zero it out for demo purposes............*/

/* timeslot 1, bit-0 enabled and assigned to channel 1 in timeslot map.......*/

/* timeslot 1, bit-1 enabled and assigned to channel 1 in subchannel map.....*/
GroupStr0.TxSubchannelMap[4] |= 0x00008100;

/* timeslot 1, bit-2 enabled and assigned to channel 1 in subchannel map.....*/
GroupStr0.TxSubchannelMap[4] |= 0x00810000;

/* timeslot 1, bit-3 enabled and assigned to channel 1 in subchannel map.....*/
GroupStr0.TxSubchannelMap[4] |= 0x81000000;

/* either write directly into MUSYCC register - or - use a service request .....*/
*(MUSYCC_FUNC_0_BAR + TX_SUBCHANNEL_MAP_OFFSET + 4) = GroupStr0.TxSubchannelMap[0];
*(MUSYCC_FUNC_0_BAR + TX_SUBCHANNEL_MAP_OFFSET + 5) = GroupStr0.TxSubchannelMap[0];

/* note +4 and +5 dword offsets into the subchannel map are for channel 1 bits .*/
/* the value for the 4th dword becomes Tx Subchannel Map = 0x81818100 .......*/
/* the value for the 5th dword becomes Tx Subchannel Map = 0x00000000 .......*/
```

Descriptor	Component Of Descriptor	Value Of Components
Subchannel Map (dword 4)	BITEN3/7	1, bit 3 enabled
	CH3/7	1, logical channel 1
	BITEN2/6	1, bit 2 enabled
	CH2/6	1, logical channel 1
	BITEN1/5	1, bit 1 enabled
	CH1/5	1, logical channel 1
	BITEN0/4	0, bit 0 not assigned here, see Timeslot Map
	CH0/4	0, bit 0 not assigned here set Timeslot Map

Descriptor	Component Of Descriptor	Value Of Components
Subchannel Map (dword 5)	TSEN3/7	0, bit 7 disabled
	CH3/7	0, don't care
	TSEN2/6	0, bit 6 disabled
	CH2/6	0, don't care
	TSEN1/5	0, bit 5 disabled
	CH1/5	0, don't care
	TSEN0/4	0, bit 4 disabled
	CH0/4	0, don't care

Transmit Channel Configuration Descriptor

```
/* each transmit channel descriptor is made up of 1 dwords.....*/
/* need to define channel 0, 1, and 2 - 3 dwords total.....*/
/* dword 0 for logical channel 0.....*/
/* dword 1 for logical channel 1.....*/
/* dword 2 for logical channel 2.....*/

/* for logical channel 0.....*/
/* pad count adjustment = 0, disabled.....*/
/* buffer location index = 0.....*/
/* data inversion = 0, disabled.....*/
/* internal buffer length = 0.....*/
/* end of padfill interrupt = 0, disabled.....*/
/* protocol = 2, hdlc-16-fcs.....*/
/* message length check register = 1, use register 1.....*/
/* fcs transfer = 0, normal, do not transfer rx fcs into shared memory.....*/
/* mask suerr interrupt = 0, do not mask, enable interrupt.....*/
/* mask sinc. interrupt = 0, do not mask, enable interrupt.....*/
/* mask sdec. interrupt = 0, do not mask, enable interrupt.....*/
/* mask sfilt interrupt = 0, do not mask, enable interrupt.....*/
/* mask idle. interrupt = 0, do not mask, enable interrupt.....*/
/* mask msg.. interrupt = 0, do not mask, enable interrupt.....*/
/* mask eom.. interrupt = 0, do not mask, enable interrupt.....*/
/* mask buff. interrupt = 0, do not mask, enable interrupt.....*/
GroupStr0.TxChannelConfigDescr[0] = 0x000024000;

/* for logical channel 1.....*/
/* everything same except as logical channel 0.....*/
/* buffer location index = 1.....*/
/* internal buffer length = 0.....*/
/* message length check register = 2, use register 2.....*/
GroupStr0.TxChannelConfigDescr[1] = 0x01002800;

/* for logical channel 2.....*/
/* everything same except as logical channel 0.....*/
/* buffer location index = 2.....*/
/* internal buffer length = 0.....*/
/* message length check register = 0, do not check.....*/
GroupStr0.TxChannelConfigDescr[2] = 0x02002000;


/* either write directly into MUSYCC register - or - use a service request .....*/
*(MUSYCC_FUNC_0_BAR + TX_CHANNEL_CONFIG_DESCR_OFFSET +0) =
        GroupStr0.TxChannelConfigDescr[0];

*(MUSYCC_FUNC_0_BAR + TX_CHANNEL_CONFIG_DESCR_OFFSET +1) =
        GroupStr0.TxChannelConfigDescr[1];

*(MUSYCC_FUNC_0_BAR + TX_CHANNEL_CONFIG_DESCR_OFFSET +2) =
        GroupStr0.TxChannelConfigDescr[2];
```

Descriptor	Component Of Descriptor	Value Of Components
Transmit Channel Configuration Descriptor	PADJ	0, pad count adjustment disabled
	BUFFLOC	0, for logical channel 0 1, for logical channel 1 2, for logical channel 2
	INV	0, data inversion disabled
	BUFFLEN	0, total FIFO = (0+1)*2 = 2 dwords
	EOPI	0, end-of-padfill interrupt disabled
	PROTOCOL	2, HDLC w/ 16-bit FCS
	MAXSEL	1 for logical channel 0 application 2 for logical channel 1 application 0 for logical channel 2 application
	FCS	0, FCS transfer normal
	MSKSUERR MSKSINC MSKSDEC MSKSFILT MSKIDLE MSKMSG MSKEOM MSKBUFF	0, interrupt masking disabled therefore enabling these interrupts

Receive Timeslot Map

Same as Transmit Timeslot Map

Receive Subchannel Map

Same as Transmit Subchannel Map

Receive Channel Configuration Descriptor

Same as Transmit Channel Configuration Descriptor

Message Lists

Message lists contain data that are transmitted or received by MUSYCC. Message lists always reside in shared memory. Upon channel activation, MUSYCC traverses the list and either takes data from data buffers (tx) or puts data into data buffers (rx). Each direction of a channel must be assigned a message list before that direction of the channel is activated.

A message list is a singly-linked list of message descriptors. A message descriptor consists of a Buffer Descriptor (1 dword), a Data Pointer (1 dword), and a Next Descriptor Pointer (1 dword) - reference Table 43, Message Descriptor.

The Buffer Descriptor contains a set of bit-fields which instructs MUSYCC on how to behave after the data is put in or taken out of a particular data buffer.

The Data Pointer contains an address in shared memory where MUSYCC may take data to be transmitted or may store data just received from the corresponding channel.

The Next Descriptor Pointer contains an address of a message descriptor in shared memory where MUSYCC may access the "next" message descriptor in the linked list.

To terminate a message list, the contents of the Next Descriptor Pointer in the last descriptor in a list may either point to the address of the last descriptor itself or may point to a general purpose "terminate" message descriptor that can be used by any message list to represent the end of the list. The idea being that the OWNER bit-field in the last descriptor's Buffer Descriptor must [eventually] indicate that the host owns the buffer - this bit value is opposite for receive and transmit buffer ownership.

The reason why the OWNER bit-field mechanism works in controlling the termination of the message list is as MUSYCC reads in each message descriptor in the linked list, it first checks the OWNER bit-field to see if it, and not the host, owns the buffer. If it does own the descriptor, then after servicing the contents of this descriptor, MUSYCC reverses the OWNER bit-field to hand the descriptor back to the host. If it does not own the buffer, then, the end of the message list is automatically concluded. Note that the channel stays active and depending on other bit-field values in the Buffer Descriptor, MUSYCC may either poll this last descriptor regularly to see if the OWNER bit value has changed or it may idle the channel and await another channel activation or channel jump request. The former would be useful in continuing a message list while retaining the original list; and, the latter would be useful in starting a message list from the top element in the list.

The following describes a general sequence for setting up the **transmit** message descriptor for a single channel:

```
/* assume transmit channel is currently deactivated ..........................*/
/* assume that a 1024-byte message is separated into four 256-byte data buffers..*/
/* Because four buffers will be used, four 12-byte segments [or one 48-byte......*/
/* of shared memory is required...............................................*/

/* declare structures */

typedef tMSG_DESCR
{
unsigned long BufferDescr;
unsigned char *pDataBuffer;
unsigned long *pNextMsgDescr;
} tMSG_DESCR;

typedef tDATA_BUFFER
{
unsigned char Data[256];
} tDATA_BUFFER;

/* allocate space */
tMSG_DESCR pTxMsgDescr[4];
tDATA_BUFFER pDataBuf[4];

/* get memory for the descriptors - note: can get one 48-byte segment also .....*/
pTxMsgDescr[0] = malloc(12);
pTxMsgDescr[1] = malloc(12);
pTxMsgDescr[2] = malloc(12);
pTxMsgDescr[3] = malloc(12);

/* get memory for the data buffers - note: can get one 1024-byte segment also...*/
pDataBuf[0] = malloc( 256 );
pDataBuf[1] = malloc( 256 );
pDataBuf[2] = malloc( 256 );
pDataBuf[3] = malloc( 256 );

/* link the message descriptors together. Terminate the message list by.........*/
/* assigning the "next" pointer in the last descriptor to point to the last.....*/
/* descriptor itself..........................................................*/

pTxMsgDescr[0]->pNextMsgDescr = pTxMsgDescr[1];
pTxMsgDescr[1]->pNextMsgDescr = pTxMsgDescr[2];
pTxMsgDescr[2]->pNextMsgDescr = pTxMsgDescr[3];
pTxMsgDescr[3]->pNextMsgDescr = pTxMsgDescr[3];

/* Note: last descriptor points to itself.....................................*/


/* assign each message descriptor a data buffer */
pTxMsgDescr[0]->pDataBuf[0];
pTxMsgDescr[1]->pDataBuf[1];
pTxMsgDescr[2]->pDataBuf[2];
pTxMsgDescr[3]->pDataBuf[3];
```

```
/* set the value for each buffer descriptor in each message descriptor.........*/
/* OWNER bit to MUSYCC, for tx buffers set to 1, for rx set to 0..............*/
/* NP bit to enable polling, set to 0.........................................*/
/* EOM bit if the last data buffer is associated with this descriptor..........*/
/* EOBI bit to enable end-of-buffer interrupt, set to 1.......................*/
/* IC field to set the idle-code to 7Eh, set to 0.............................*/
/* PADEN field to disable pad fill, set to 0..................................*/
/* PADCNT field to specify 0 pad fill codes, set to 0.........................*/
/* REPEAT bit to disable message retransmission, set to 0.....................*/
/* BLEN field set to the length of the data buffer, set to 256................*/

/* msg descr 0 */
pTxMsgDescr[0]->BufferDescr = 0x90000200;

/* msg descr 1 */
pTxMsgDescr[1]->BufferDescr = 0x90000200;

/* msg descr 2 */
pTxMsgDescr[2]->BufferDescr = 0x90000200;

/* msg descr 3 */
/* only difference is EOM bit */
pTxMsgDescr[3]->BufferDescr = 0x92000200;

/* fill data buffer with outbound traffic. each buffer contains 256-bytes of data */

/* set the head pointer for channel 0, for example, to point to the top of......*/
/* the just formed message descriptor list....................................*/

GroupStr0.TxHeadPointer[0] = pTxMsgDescr[0];

/* now activate transmit channel by issuing a service request - for example.....*/

ServiceRequest( ACTIVATE_CHANNEL, Group=0, Channel=0, Direction=TX);
```

Channel Activation

After the previous levels of configuration are completed, individual channels within a channel group are ready to be activated. Service requests are used to activate channels.

Each channel within a channel group consists of a transmitter and a receiver section. Each section is independent of the other and maintains its own state machine, configuration registers, and internal resources. To activate both transmitter and receiver sections, two separate service requests are required, one directed to the transmitter and one to the receiver. MUSYCC responds to each service request with the SACK Interrupt Descriptor which notifies the host that the task was completed.

Channel Activation is an asynchronous command from the host interface to a transmit or receive section of a channel to jump to a new message. Message Descriptors in shared memory describe the attributes of the new message, what to do between messages, and identify the location of message data buffers in memory to use for transmit data or receive data.

The following describes what MUSYCC does when **transmit** channel is activated:

1 Read Tx Head Pointer for channel from shared memory and store in internal channel descriptor map.
2 Reads Message Descriptor pointed to by Tx Head Pointer and store in internal channel descriptor map.
3 Check bit-field OWNER and NP in Buffer Descriptor.
If OWNER = 1, MUSYCC is buffer owner. Load channel descriptor map with data buffer pointer and data buffer length. Go to 4.
If OWNER = 0, MUSYCC is not buffer owner. May enter polling mode until MUSYCC owns buffer or receives another service request to activate channel or jump to new message. Repeat 3.
4 Check bit-field INHTBSD in Group Configuration Descriptor.
If INHTBSD = 0 (i.e., MUSYCC is allowed to overwrite Buffer Descriptor with a Buffer Status Descriptor), then the address of the Buffer Descriptor is stored away in the Message Pointer slot in shared memory. After current message is completely processed, MUSYCC will read in Next Message Pointer and overwrite Buffer Descriptor. Go to 3.
If INHTBSD = 1 (i.e., MUSYCC is not allowed to overwrite Buffer Descriptor), then the address of Next Message Pointer is pre-fetched from the current Message Descriptor and stored in the Message Pointer slot in shared memory. After the current message is completely processed, MUSYCC jumps to this pointer and starts processing a new message.
Go to 5 when current message is completely processed, otherwise repeat 4.
Simultaneously, MUSYCC masters the PCI bus and reads data into transmit FIFO from shared memory.
Simultaneously, the serial port outputs data using the control lines TCLK, TSYNC, and TDAT as appropriate.
5 At end of message, Interrupt Descriptors and Buffer Status Descriptors may be written out to shared memory (see 3)—depending on the masking of interrupts and allowance of Buffer Descriptor overwrites.
6 Read Next Message Descriptor.
7 Go to 3.

The following describes what MUSYCC does when **receive** channel is activated:

1. Read Rx Head Pointer for specified channel from shared memory and store in internal channel descriptor map.
2. Read Message Descriptor pointed to by Head Pointer and store in channel descriptor map.
3. Check bit-field OWNER and NP in Buffer Descriptor.
   If OWNER = 1, MUSYCC is buffer owner. Load channel descriptor map with data buffer pointer and data buffer length. Go to 4.
   If OWNER = 0, MUSYCC is not buffer owner. May enter polling mode until MUSYCC owns buffer or receives another service request to activate channel. Go to 4.
4. Check bit-field INHRBSD in Group Configuration Descriptor.
   If INHRBSD = 0 (i.e., MUSYCC is allowed to overwrite Buffer Descriptor with a Buffer Status Descriptor), then the address of the Buffer Descriptor is stored away in the Message Pointer slot in shared memory. After current message is completely processed, MUSYCC will read in Next Message Pointer and overwrite Buffer Descriptor. Go to 3.
   If INHRBSD = 1 (i.e., MUSYCC is not allowed to overwrite Buffer Descriptor), then the address of Next Message Pointer is pre-fetched from the current Message Descriptor and stored in the Message Pointer slot in shared memory. After current message is completely processed, MUSYCC jumps to this pointer and starts processing new message.
   Go to 5 when current message is completely processed, otherwise repeat 4.
   Simultaneously, the receiver is configured and data is sampled in from serial port using control lines RCLK, RSYNC, RDAT, and ROOF as appropriate.
   Simultaneously, MUSYCC masters the PCI bus and transfers data from internal FIFO to shared memory.
5. At end of message, Interrupt Descriptors and Buffer Status Descriptors may be written out to shared memory (see 3)—depending on the masking of interrupts and allowance of Buffer Descriptor overwrites.
6. Read Next Message Descriptor.
7. Go to 3.

Channel Deactivation

After the channel has been activated, channel deactivation via a service request suspends activity on an individual channel-direction.

Each channel within a channel group consists of a transmitter and a receiver section. Each section is independent of the other and maintains its own state machine and configuration registers. To deactivate both transmitter and receiver sections, two separate service requests are required, one directed to the transmitter and one to the receiver. MUSYCC responds to each service request with the SACK Interrupt Descriptor which notifies the host that the task was completed.

A Channel Deactivation is an asynchronous command from the host interface to a transmit or receive section of a channel to suspend bit-level processing and halt memory transfers into shared memory.

The following describes what MUSYCC does when **transmit** channel is deactivation:

1. Current message processing is terminated destructively. That is, data can be lost and messages prematurely aborted.
2. The bit-level processor responsible for handling outbound bits to the serial port is immediately and asynchronously disabled. The data output pin, TDAT, is three-stated or held at logic '1' depending on the bit-field TRITX in the Port Configuration Descriptor. Data transfers from shared memory are halted.
3. The channel direction remains in the suspended state until the channel is activated. The current channel direction configuration is maintained.

The following describes what MUSYCC does when **receive** channel is deactivation:

1. Current message processing is terminated destructively. That is, data can be lost and messages prematurely aborted.
2. The bit-level processor responsible for handling inbound bits from the serial port is immediately and asynchronously disabled. Data transfers to shared memory are halted.
3. The channel direction remains in the suspended state until the channel is activated. The current channel direction configuration is maintained.

Channel Jump

A channel jump request is issued by the host via a service request. For a receiver, channel jumps are the same as channel activation.

For a transmitter, channel jumps are non-destructive to currently serviced messages. The channel state is not reset as in the channel activate sequence. Hence, a transmitter channel must be activated first, then subsequent jump requests can be made using the channel jump service request. For a transmitter, channel jumps provide a non-destructive way to start transmitting a new message list. MUSYCC waits until the completion of the current message before jumping to a message list pointed to by a new Head Pointer..

A jump request is issued by the host via a service request towards a channel in a channel group in MUSYCC. It is imperative to note that the service request acknowledge (via the SACK interrupt descriptor) for the jump service request - specifically for the transmit direction - will not be output towards the host until after the current message is completely transmitted. For applications with long messages to transmit, the jump service request for the needs to be used with care.

Frame Alignment

The serial data stream that MUSYCC can manage consists of either packetized data or unpacketized data. MUSYCC supports two types of data-stream modes: HDLC and transparent.

In transparent mode, bit-level processing begins after a frame synchronization event. The method of frame alignment is specified on a per channel group basis with the SFALIGN bit-field in the Group Configuration Descriptor and applies to all HDLC channels within the channel group.

In HDLC mode, bit level processing begins immediately upon channel activation. Any type of frame alignment is unnecessary and MUSYCC disables its frame synchronization mechanism.

For a channel configured for HDLC mode, the transmit and receive section of the channel can either wait for a synchronization signal from an internal frame synchronization flywheel or wait for a received synchronization signal from the serial port before beginning message processing after channel activation. By default, the sections wait for the synchronization signal from the frame synchronization flywheel.

If external alignment is preferred, then the SFALIGN bit-field must be set to expect the synchronization signal from the serial port and the external framer device must be configured to strobe the MUSYCC input signals TSYNC and RSYNC only on superframe boundaries.

By design, MUSYCC must be provided an external frame synchronization signal one. After this event, MUSYCC keeps track of subsequent frame bit location with its flywheel mechanism.

Descriptor Polling

Upon channel activation and any necessary frame alignment, MUSYCC must fetch Message Descriptors from shared memory to start the flow of message bits into and out of shared memory.

As a Buffer Descriptor is fetched, MUSYCC checks the owner-bit to verify if the buffer is serviceable by MUSYCC. If the owner-bit indicates that the host still owns the buffer, then the host has not yet prepared the data in the buffer for processing. This may or may not be an error condition. In this case, MUSYCC also must check the no-poll bit in the same descriptors to determine if polling for MUSYCC ownership is enabled.

If the host owns the buffer and polling is disabled, the channel-direction is suspended from processing messages until the host intervenes with a subsequent channel activation or channel jump request. The channel is not capable of leaving this suspended state autonomously.

If the host owns the buffer and polling is enabled, the channel-direction is suspended from processing messages and MUSYCC periodically polls the owner-bit in the Buffer Descriptor to verify if the buffer is ready for MUSYCC. The channel is capable of leaving this suspended state autonomously.

The frequency of polling is controlled independently for each channel group by the SFALIGN (superframe alignment) and POLLTH (poll throttle) bit-fields in the Group Configuration Descriptor.

The SFALIGN bit-field defines the source of the synchronization event to be used by MUSYCC. The source is either an internal flywheel method or an external signal at the serial port.

NOTE: Timeslot counter or flywheel time base method uses a 7-bit counter. As every bit is serviced, over 32 channels with 8 bits per channel in a 2.048 MHz data stream, the counter is incremented. When the counter rolls over to 0, a beginning of frame is declared. 256 bits in a 2.048 MHz represents 125 ms

The POLLTH bit-field specifies how often MUSYCC checks the owner-bit in a host-owned Buffer Descriptor. The values correspond to 1, 16, 32 or 64 frame periods, or 125 µs, 2 ms, 4 ms, and 8 ms, respectively. The POLLTH bit-field is always used in conjunction with the SFALIGN bit-field.

If the serial port is configured for Nx64 mode (a variable number of 64 Kbps channels concatenated together to form one logical channel), the Timeslot Counter is reset only when the inputs TSYNC or RSYNC are asserted. In this case, the SFALIGN bit-field is always ignored.

Table 57. Polling Frequency Using A Timeslot Counter Method

Standard Channelized Input			Poll Throttle Value (multiples of frames)			
Name	Rate (MHz)	Bits Per Frame	0 (x1)	1 (x16)	2 (x32)	3 (x64)
T1	1.536	192	125 µs	2 ms	4 ms	8 ms
E1	2.048	256	125 µs	2 ms	4 ms	8 ms
2 E1	4.096	512	125 µs	2 ms	4 ms	8 ms
4 E1	8.192	1024	125 µs	2 ms	4 ms	8 ms

Repeat Message Transmission

MUSYCC provides a mechanism to repeatedly transmit a single message. A transmitter channel enters the repeat mode if the REPEAT bit-field is 1 and the EOM bit-field is 1 in a Transmit Buffer Descriptor. It is essential that the message being repeatedly transmitted be specified completely by a single Message Descriptor and not by a linked list of descriptors.

A repeating message either fits entirely in the internal FIFO space allocated for the transmitter channel or the message is accessed in pieces over the PCI bus and then re-accessed from the beginning when the end of buffer is reached. The determination of whether the message fits entirely in the FIFO or not is automatically performed each time MUSYCC enters repeat mode. MUSYCC compares the BLEN bit-field [which specifies the number of bytes in the message] from the Transmit Buffer Descriptor to (BUFFLEN+1)*2 [which specifies the number of dwords in the FIFO] from the Channel Configuration Descriptor.

To exit repeat mode after the current message is completely transmitted and before the next repetition (gracefully or non-destructively), a channel jump service request must be issued. Prior to the jump request, the host must initialize the channel's Transmit Head Pointer with a new Message Descriptor.

To exit repeat mode regardless of the message being processed, a channel activate or a channel deactivate service request may be issued. Either is considered destructive as the current message transmission is aborted.

Protocol Support

Frame Check Sequence

MUSYCC is configurable to calculate either a 16- or 32-bit Frame Check Sequence (FCS) for HDLC packets ranging in sized from a minimum of 2 octets to a maximum of 16384 octets. The FCS always applies to the entire packet length.

For all HDLC modes which require FCS calculations, the polynomials used to calculate the FCS are according to ITU-T Q.921 and ISO 3309-1984.

- CRC-16

$$x^{16} + x^{12} + x^5 + 1$$

- CRC-32

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

Opening/Closing Flags

For HDLC modes only, MUSYCC supports the use of opening and closing message flags. The 7Eh (01111110b) flag is the opening and closing flag. An HDLC message is always bounded by this flag at the beginning and the end of the message.

MUSYCC supports receiving a shared flag where the closing flag of one message can act as the opening of the next message. MUSYCC also supports receiving a shared-zero bit between two flags—that is, the last zero bit of one flag is used as the first zero bit of the next flag.

MUSYCC can be configured to transmit a shared flag between successive messages by configuring the bit-field PADEN in each transmit buffer descriptor. MUSYCC does not transmit shared-zero bits between successive flags.

Abort Codes

Seven consecutive 1s constitute an abort flag. Receiving the abort code causes the current frame processing to be aborted and further data transfer into shared memory to terminate. After detecting the abort code, MUSYCC enters a mode searching for a new opening flag.

Notification of this detected condition is provided by the receive buffer status descriptor and/or an interrupt descriptor indicating the error condition Abort Flag Termination.

In cases where received idle codes transition to an abort code, an interrupt descriptor is generated toward the host indicating the informational event Change To Abort Code. All received abort codes are discarded.

Zero-Bit Insertion/Deletion

MUSYCC provides zero-bit insertion and deletion when it encounters five consecutive ones within a frame. In the receiver, a zero-bit is de-inserted and in the transmitter, zero-bit is inserted after five ones are seen.

Message Configuration Bits

A group of bits specified in a Transmit Buffer Descriptor specifies the data to be transmitted at the transmit channel after the end of a current message has been transmitted. The bits are collectively known as Table 59, Message Configuration Descriptor and include the specification for the following:

- Idle Code specification, IC
- Inter-message Pad Fill Enable, PADEN
- Inter-message Pad Fill Count, PADCNT
- Repeat Message Transmission, REPEAT

Idle Code

The Idle Code (IC) specification allows one of a set of idle codes to be chosen to be transmitted after the current message in case the next message is not available to be transmitted or inter-message pad fill is requested via PADEN.

Inter-message Pad Fill

The Pad Enable (PADEN) and Pad Count (PADCNT) specification allows pad fill octets (a sequence of one or more specified idle codes) to be transmitted between messages. PADEN enables or disables the pad octet transmission feature. PADCNT is the minimum number of fill octets to be transmitted between closing flag of one message and the opening flag of the next message.

Repeat Message Transmission

The Repeat Message Transmission (REPEAT) specification allows a single message to be transmitted repeatedly without additional host intervention. This feature is required to support SS7 message retransmission. In the SS7 application, a retransmitted message is usually 3, 4, or 5 octets in length. Message retransmission in MUSYCC requires that the entire message be held in a single shared memory message buffer versus being spread across multiple buffers. The message retransmission feature is not limited to SS7 applications.

Message Configuration Bits Copy Enable/Disable

The message configuration bits described above are used in special data communication applications requiring the following attributes specified on a per-message basis:

- Specific idle code is to be transmitted between messages
- Inter-message pad fill is required
- Repeat message transmission is required

If at least one transmit channel in a channel group is supporting an application which requires any one of the above attributes, then the Message Configuration Bits Copy must be enabled for the entire channel group by setting the MCENBL bit-field to 1.

Setting MCENBL to 0 prevents MUSYCC from copying the Message Configuration Bits from the Transmit Buffer Descriptor and has the following effect on transmit channel operations throughout the channel group:

- If the protocol for a channel is specified to be transparent, then the idle code is automatically defaulted to 00h (0000 0000b). Otherwise, for any of the HDLC protocols, the idle code is defaulted to 7Eh (0111 1110b).
- PADEN and PADCNT are set to 0; thereby facilitating back-to-back message transmission.
- REPEAT is set to 0; thereby disabling automatic message retransmission.

Any of the Message Configuration Bits can be changed by writing new values for these bit-fields directly into a channel group's Transmit Configuration Table. However, the exact time the new bit-field values are applied by the transmitter is unrelated to the message being serviced. If the channel is idle, then any change to the message configuration bits in the Configuration Table will apply to subsequent messages.

To write new configuration bits into the Transmit Message Configuration Table, a PCI dword operation is required. The address map and the Message Configuration Descriptor layouts follow.

Table 58. Memory Map for Message Configuration Descriptor Table

Channel Number (0–31)	Location of Message Configuration Descriptor in Specific Group (0–3) (byte offset from Base Address Register)			
	0	1	2	3
0	06180h	06980h	07180h	07980h
1	06184h	06984h	07184h	07984h
x	Address of Message Configuration Descriptor for a channel in a group 06180h + (Group_Number[3:0] * 00800h) + (Channel_Number[31:0] * 00004h)			
31	061FCh	069FCh	071FCh	079FCh

Table 59. Message Configuration Descriptor

Bit Field	Name	Value	Description
31:27	RSVD	0	Reserved.
26:25	IC	0	Idle Code Select –7Eh (0111 1110b)
		1	Idle Code Select –FFh (1111 1111b)
		2	Idle Code Select –00h (0000 0000b)
		3	Idle Code Select –Reserved.
24	PADEN	0	Pad Fill Disabled. One shared opening/closing flag (7Eh) is inserted before sending next message
		1	Pad Fill Enabled. Also, see PADCNT bit-field.
23:16	PADCNT[7:0]	0	Pad Count. When PADEN = 1, PADCNT indicates the minimum number of idle codes to be inserted between the closing flags and the next opening flag (7Eh). If PADCNT = 2 and IC = 1, for example, yields the bit pattern 7Eh..FFh..FFh..7Eh Note that there is no indication by MUSYCC if more than PADCNT number of idle codes are inserted.
15	REPEAT	0	Repeat Message Transmission Disabled.
		1	Repeat Message Transmission Enabled.
14:0	RSVD	0	Reserved.

Bit-Level Operation

Each channel group provides two separate Bit Level Processors (BLP) to service the transmit and receive directions separately. Also, each channel group provides two separate direct memory access controllers to service the transmit and receive directions separately. BLP and DMAC work in conjunction to transfer serial data between the serial interface and shared memory. BLPs perform the required bit-level processing based on the protocol mode assigned to the channel and direction.

DMACs access the data to transmit or store the received data to shared memory (via the host interface). Each DMAC seeks out the (next) message descriptor to service for each active channel and direction. This information is available from the Channel Group Descriptor in shared memory which has information locating the message list for each channel direction. A Buffer Descriptor within each Message Descriptor along with the protocol mode set for the channel-direction drives the treatment of the receive and transmit bit stream.

Bit-level operations vary between HDLC and transparent modes. The differences relate to protocol-specific support as well as to the treatment of the bit stream during abnormal conditions. Additionally, bit-level operations are independent and sometimes differ between the transmitter and the receiver.

The following apply to all event and error handling which are described in the sections to follow:

- During bit level operations, events and errors can occur which may affect the outcome message processing. Unless masked, all events and errors cause Interrupt Descriptors to be generated within MUSYCC. Interrupt Descriptors identify the error or event condition, the transmit or receive direction, and the channel and channel group number affected.
- If a channel is suspended, enters an idle mode, enters an abort mode, and/or is autonomously turned off by MUSYCC during bit-level operations, then a channel reactivation must occur by either a Channel Activation Service Request or a Channel Jump Service Request. This is referred to as "requiring reactivation."
- The bit-fields INHTBSD and INHRBSD in the Group Configuration Descriptor specify if MUSYCC can write a Buffer Status Descriptor into a Message Descriptor to indicate that MUSYCC has completed servicing the descriptor. BLP and DMAC.
- In cases where bit-level operations continue normally, the DMAC accesses the Next Message Pointer from the current Message Descriptor. This accesses the next Message Descriptor in the chain of descriptors for a particular channel and direction. Each Message Descriptor indicates whether the host or MUSYCC owns the descriptor.

Transmit

The transmitter initiates data transfer from shared memory to the serial interface only if the following conditions are true:

- TXENBL bit set to 1 in Group Configuration Descriptor.
- Transmit channel mapped to logical channel(s) in Transmit Timeslot Map.
- Transmit channel (re)activated (via service request).

If the TXENBL bit is set to 0, then the output signal is a "three-state" signal. If the channel is not mapped or the channel is not active, then the transmitter either outputs a "three-state" or an "all ones" signal depending on the state of the bit-field TRITX in the Group Configuration Descriptor.

Receive

The receiver transfers data from the serial interface to memory buffers in shared memory only if the all of the following conditions are true:

- RXENBL bit set to 1 in the Group Configuration Descriptor.
- Receive channel mapped to logical channel(s) in Receive Timeslot Map.
- Receive channel (re)activated (via service request).

If any one of the above conditions is not true, then the receiver ignores the incoming data stream.

Data transfer consists of MUSYCC first seeking out the (next) message descriptor from the Channel Group Descriptor in shared memory for each active channel. The buffer descriptor in each message descriptor plus the protocol mode set for the channel dictates the treatment of the incoming bit stream.

HDLC Mode

MUSYCC supports three HDLC modes. The modes are assigned on a per-channel and direction basis by setting the PROTOCOL bit-field within the Channel Configuration Descriptor. The HDLC modes are:

- SS7-HDLC-16CRC: specific SS7 support, HDLC support, 16-bit CRC.
- HDLC-16CRC: HDLC support, 16-bit CRC.
- HDLC-32CRC: HDLC support, 32-bit CRC.

HDLC support by the transmitter includes:

- Generate opening/closing/shared flags.
- Zero bit insertion after five consecutive 1s are transmitted.
- Generate pad fill between frames and adjust for zero insertions.
- Generate 16- or 32-bit FCS.
- Generate abort sequences upon data corruption in message.

HDLC support by the receiver includes:

- Detection and extraction of opening/closing/shared flags.
- Detection of shared-0 between successive flags.
- Zero bit extraction after five consecutive 1s are received.
- Detect changes in pad fill idle codes.
- Check and extract 16- or 32-bit FCS.
- Check frame length.
- Check for octet alignment.
- Check for abort sequence reception.

Bit-fields within the Transmit Buffer Descriptor specify inter-message bit level operations. Specifically, when the EOM bit-field is set to 1 within a Message Descriptor by the host, it signifies that the descriptor represents the last buffer for the current message being transmitted and the bit-fields IC, PADEN, PADCNT, and REPEAT take effect. These bits are collectively known as Table 59, Message Configuration Descriptor.

Additionally, the bit-field NP in both the Receive and Transmit Buffer Descriptors enables a polling scheme in case MUSYCC discovers that it does not own the (next) message descriptor.

Transmit Events

Transmit events are informational in nature and do not require channel recovery actions.

End Of Buffer [EOB]

Reasons:

- DMAC reached the end of a buffer by servicing a number of octets equal to the bit-field BLEN in the Transmit Buffer Descriptor. Note that the [last] EOB and an EOM are coincident and result in two separate events being generated.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = EOB, DIR = 1 (if EOBI = 1 in Transmit Buffer Descriptor).
- BLP and DMAC continue with normal message processing. If the DMAC does not get more data from shared memory before the BLP needs to output the next data bit, then the BLP outputs another octet of idle code.

End Of Message [EOM]

Reasons:

- BLP has transmitted the last bit of a data buffer and the Transmit Buffer Descriptor signified the end of a message by the bit-field EOM in a Transmit Buffer Descriptor. Note that the [last] EOB and an EOM are coincident and result in two separate events being generated.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = EOM, DIR = 1 (if MSKEOM = 0 in Transmit Channel Configuration Descriptor).
- BLP and DMAC continue with normal message processing. If the DMAC does not get more data from shared memory before the BLP needs to output the next data bit, then the BLP outputs another octet of idle code.

End Of Padfill [EOP]

Reasons:

- BLP has transmitted the specified number of pad fill octets.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = EOP, DIR = 1 (if EOPI = 1 in Transmit Channel Configuration Descriptor).
- BLP and DMAC continue with normal message processing. If the DMAC does not get more data from shared memory before the BLP needs to output the next data bit, then the BLP outputs another octet of idle code.

Receive Events

Receive events are informational in nature and do not require channel recovery actions.

End Of Buffer [EOB]

Reasons:

- One message is stored across multiple message buffers. MUSYCC reached the end of a buffer by servicing a number of octets equal to the bit-field BLEN in a Receive Buffer Descriptor.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = EOB, DIR = 0 (if EOBI = 1 in Receive Buffer Descriptor).
- BLP and DMAC continue with normal message processing.

End Of Message (EOM)

Reasons:

- BLP has detected the end of a message (closing flag or an error condition) in the received data stream. Error conditions include ABT, LNG, ALIGN, BUFF, ONR errors.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = EOM, DIR = 0 (if MSKEOM = 0 in Receive Channel Configuration Descriptor).
- DMAC sets bit-field EOM = 1 in Receive Buffer Status Descriptor (if INHRBSD = 0 in Receive Channel Configuration Descriptor).
- BLP and DMAC continue with normal message processing.

Change To Abort Code (CHABT)

Reasons:

- BLP detected received data changed from pad fill (7Eh) octets to abort code (zero followed by seven consecutive ones).

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = CHABT, DIR = 0 (if MSKIDLE = 0 in Receive Channel Configuration Descriptor).
- BLP and DMAC continue with normal message processing.

Change To Idle Code (CHIC)

Reasons:

- BLP detected received data changed from abort code (0 followed by seven ones) to idle code (7Eh) octets.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = CHIC, DIR = 0 (if MSKIDLE = 0 in Receive Channel Configuration Descriptor).
- BLP and DMAC continue with normal message processing.

Frame Recovery (FREC)

Reasons:

- BLP detected that the serial interface has transitioned from an out-of-frame to in-frame condition.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = FREC, DIR = 0 (if MSKOOF = 0 in Group Configuration Descriptor).
- BLP and DMAC continue with normal message processing.

SS7 SUERM Octet Count Increment (SINC)

Reasons:

- BLP incremented the SUERM counter. The channel is in SS7 mode. The reasons for SUERM counter to increment include reception of a short message, octet alignment error, FCS mismatch or an accumulation of octet count errors. Each of these conditions may also generate an interrupt.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = SINC, DIR = 0 (if MSKSINC = 0 in Receive Channel Configuration Descriptor).
- BLP and DMAC continue with normal message processing.

SS7 SUERM Octet Count Decrement (SDEC)

Reasons:

- BLP decremented the SUERM counter. The channel is in SS7 mode. The SUERM counter decrements by one when MUSYCC receives 256 consecutive unerrored messages.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = SDEC, DIR = 0 (if MSKSDEC = 0 in Receive Channel Configuration Descriptor).
- BLP and DMAC continue with normal message processing.

SS7 Filtered Message (SFILT)

Reasons:

- BLP detected an unerrored 3, 4, or 5 octet message which is identical to the previous message. The channel is in SS7 mode.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = SFILT, DIR = 0 (if MSKSFILT = 0 in Receive Channel Configuration Descriptor).
- BLP discards the received message in the FIFO.
- BLP and DMAC continue with normal message processing.

Transmit Errors

Transmit errors are service-affecting and require a corrective action by a controlling device to resume normal bit-level processing.

Underflow due to Host Ownership of Buffer (ONR)

In this case, as MUSYCC attempts to access the [next] Message Descriptor to access a message or part of a message to transmit, it finds that the ownership of the descriptor has not been granted by the host.

This error results when currently transmitting an HDLC message and no additional descriptors are available in a timely manner.

Once a descriptor is granted, however, MUSYCC assumes ownership of the message buffer and continues reading data until the end of buffer is reached. If the host reclaims the buffer without MUSYCC granting ownership back to the host, then this is considered a host error and the effects are indeterminate.

Reasons:

- Degradation of the host subsystem or application software performance.

Effects:

- Partial HDLC message transmission has occurred.
- Interrupt Descriptor in Interrupt Queue with ERROR = ONR, DIR = 1 (if MSKBUFF = 0 in Transmit Channel Configuration Descriptor).
- Transmit channel enters abort state where the BLP transmits a repetitive abort sequence of seven consecutive ones.
- Transmit Buffer Status Descriptor cannot be written.
- Message polling is automatically disabled.
- Transmit channel enters abort state.

Channel Level Recovery Actions:

- Transmit channel reactivation is required.

Underflow due to internal FIFO buffer under-run (BUFF)

In this case, the internal FIFO becomes empty when MUSYCC must transmit data bits (per serial interface clock rate) and MUSYCC has ownership of a message buffer in shared memory.

Reasons:

- Degradation of the host subsystem or application software performance.
- Congestion of the PCI bus.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = BUFF, DIR = 1 (if MSKBUFF = 0 in Transmit Channel Configuration Descriptor).
- Transmit channel enters abort state where the BLP transmits a repetitive abort sequence of 16 consecutive ones.
- Message polling is automatically disabled.
- Transmit Buffer Status Descriptor is not written.

Channel Level Recovery Actions:

- Transmit channel reactivation is required.

Change of frame alignment while transmitting HDLC message (T1/E1 modes) (COFA)

In this case, the TSYNC input signal transitioned from low to high when not expected to do so by the "frame synchronization flywheel mechanism." This error only applies to ports configured for T1, E1, 2xE1 or 4xE1 signal. Frame synchronization indicates the location of timeslot 0 in the serial data stream. Lacking frame synchronization, the transmitter cannot map and align timeslots. This error affects all active channels in the channel group.

Reasons:

- T1/E1 signal failure detected by the physical interface providing the serial data, clock, and synchronization to the serial interface on MUSYCC.

Effects:

- Causes serial interface to enter COFA mode for one T1/E1 frame period (125 µs)—not necessarily on a frame boundary.
- For every activated channel transmitting an HDLC message Transmit channel enters abort state where the BLP transmits a repetitive abort sequence of 16 consecutive ones.
- For every activated channel transmitting an HDLC message DMAC accesses Next Message Pointer from current Message Descriptor. The ownership of current Message Descriptor is granted back to the host by writing the Transmit Buffer Status Descriptor with ONR = HOST (if INHTBSD = 0 in Transmit Channel Configuration Descriptor).
- After all activated channels are serviced, Interrupt Descriptor in Interrupt Queue with ERROR = COFA, DIR = 1 (if MSKCOFA = 0 in Group Configuration Descriptor).
- After the COFA condition subsides, normal bit-level operations continues.
- The BLP transmits idle code or message data provided by the DMAC.
- Simultaneously, DMAC checks for message descriptor ownership before transferring message data from shared memory to the internal FIFO.

Channel Level Recovery Actions:

None required.

Receive Errors

Receive errors are service-affecting and require a corrective action by the host to resume normal bit-level processing.

Overflow due to host ownership of buffer while receiving HDLC message (ONR)

In this case, as MUSYCC attempts to access the [next] Message Descriptor to store a message or part of a message, it finds that the ownership of the descriptor has not been granted by the host.

This error results when currently receiving an HDLC message and no additional descriptors are available in a timely manner.

Once a descriptor is granted, however, MUSYCC assumes ownership of the message buffer and continues writing data until the end of buffer is reached. If the host reclaims the buffer without MUSYCC granting ownership back to the host, then this is considered a host error and the effects are indeterminate.

Reasons:

- Degradation of host subsystem or application software performance.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = ONR, DIR = 0 (if MSKBUFF=0 in Receive Channel Configuration Descriptor).
- The received data in the internal FIFO is discarded and lost to the host.
- The remainder of HDLC message currently being received is discarded.
- Receive Buffer Status Descriptor cannot be written.
- Channel is deactivated.

Channel Level Recovery Actions:

- Provide sufficient amount of shared memory to store received data using lists of Message Descriptors with ownership granted to MUSYCC.
- Reactivate channel.

Overflow due to internal FIFO buffer overrun [BUFF]

In this case, the internal FIFO buffer has not been completely copied to shared memory before more received data bits needed to be stored in the FIFO. MUSYCC has access to a message buffer space in shared memory in this case.

Reasons:

- Degradation of host sub system performance
- Congestion of the PCI bus

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = BUFF, DIR = 0 (if MSKBUFF = 0 in Receive Channel Configuration Descriptor).
- The received data in the internal FIFO is discarded and lost to the host.
- The remainder of HDLC message currently being received is discarded.
- Access the Next Message Pointer from the Current Message Descriptor.
- Return ownership of current Message Descriptor by writing the Receive Buffer Status Descriptor with ONR = HOST, ERROR = BUFF (if INHRBSD = 0 in Receive Channel Configuration Descriptor).
- BLP scans for the opening flag of the next HDLC message.
- Simultaneously, DMAC checks for message descriptor ownership before transferring received data to shared memory.

Channel Level Recovery Actions:

- If possible, increase internal FIFO space for this channel. For this action, the channel must be deactivated first.
- If required, alleviate congestion of the PCI bus.

Change of frame alignment while receiving HDLC message (T1/E1 modes) (COFA)

In this case, the RSYNC input signal transitioned from low to high when not expected to do so by the "frame synchronization flywheel mechanism." This error only applies to ports configured for T1, E1, 2xE1 or 4xE1 signal. Frame synchronization indicates the location of timeslot 0 in the serial data stream. Lacking frame synchronization, the received channelized data becomes unaligned and unmappable. This error affects all active channels in the channel group.

Reasons:

- T1/E1 signal failure detected by the physical interface providing the serial data, clock, and synchronization to the serial interface on MUSYCC.

Effects:

- Causes serial interface to enter COFA mode for one T1/E1 frame period (125 µs).
- For every activated channel receiving an HDLC message the remainder of the HDLC message currently being received is discarded and the receiver scans for the opening flag of the next HDLC message before attempting to fill the channel's FIFO again.
- For every activated channel receiving an HDLC message The ownership of current Message Descriptor is granted back to the host by writing the Receive Buffer Status Descriptor with ONR = HOST and ERROR = COFA (if INHRBSD = 0 in Receive Channel Configuration Descriptor).
- After all activated channels are serviced, Interrupt Descriptor in Interrupt Queue with ERROR = COFA, DIR = 0 (if MSKCOFA = 0 in Group Configuration Descriptor).
- After the COFA condition subsides normal bit-level operations continue.
- The BLP scans for the opening flag of the next HDLC message.
- Simultaneously, DMAC checks for message descriptor ownership before transferring received data to shared memory.

Channel Level Recovery Actions:

- None required.

Out Of Frame (OOF)

Out-of-frame or loss-of-frame indicates that the entire serial data stream is invalid and data cannot be recovered from such a signal. In this case, out of frame of the incoming signal occurred while in the midst of receiving an HDLC message and copying the data to shared memory.

Reasons:

- T1/E1 signal failure detected by the physical interface providing the serial data, clock, and synchronization to the serial interface on MUSYCC.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = OOF, DIR = 0 (if MSKOOF = 0 in Group Configuration Descriptor).
- If bit-field OOFABT = 0, BLP and DMAC continue as if no errors and transfer received data into shared memory buffers normally.
- If bit-field OOFABT = 1 and is currently receiving an HDLC message, then the received data in the internal FIFO is discarded and lost to the host. DMAC accesses Next Message Pointer from current Message Descriptor. MUSYCC returns ownership of current Message Descriptor by writing the Receive Buffer Status Descriptor with ONR = HOST, ERROR = OOF (if INHRBSD = 0 in Receive Channel Configuration Descriptor).
- Regardless, the BLP continues scanning for opening flag.
- Simultaneously, DMAC checks for message descriptor ownership before transferring received data to shared memory.
- Receive channel recovers automatically.

Channel Level Recovery Actions:

- None required.

Frame Check Sequence (FCS) Error

In this case, the frame check sequence (or CRC) calculated for the received HDLC message by MUSYCC did not match the FCS which was sent within the HDLC message.

Reasons:

- Bit errors during transmission.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = FCS, DIR = 0 (if MSKMSG = 0 in Channel Configuration Descriptor).
- The entire HDLC message already copied to shared memory buffers.
- DMAC accesses Next Message Pointer from current Message Descriptor.
- Return ownership of current Message Descriptor by writing the Receive Buffer Status Descriptor with ONR = HOST, ERROR = FCS (if INHRBSD = 0 in Channel Configuration Descriptor).
- The BLP scans for the opening flag of the next HDLC message.
- Simultaneously, DMAC checks for message descriptor ownership before transferring received data to shared memory.

Channel Level Recovery Actions:

- None required.

Octet Alignment Error (ALIGN)

In this case, the HDLC message size after zero-bit extraction was not a multiple of 8 bits.

Reasons:

- Bit errors during transmission.
- Incorrect transmission of HDLC messages from distant end.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = ALIGN, DIR = 0 (if MSKMSG = 0 in Receive Channel Configuration Descriptor).
- The entire HDLC message would be transferred to shared memory.
- DMAC accesses Next Message Pointer from current Message Descriptor.
- Return ownership of current Message Descriptor by writing the Receive Buffer Status Descriptor with ONR = HOST, ERROR = ALIGN (if INHRBSD = 0 in Receive Channel Configuration Descriptor).
- The BLP scans for the opening flag of the next HDLC message.
- Simultaneously, DMAC checks for message descriptor ownership before transferring received data to shared memory.

Channel Level Recovery Actions:

- None required.

Abort Termination (ABT)

In this case, the receiver detected an abort sequence from the distant end. An abort sequence is defined as 0 followed by 7 consecutive ones.

Reasons:

- Distant end could not complete transmission of HDLC message.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = ABT, DIR = 0 (if MSKMSG = 0 in Receive Channel Configuration Descriptor).
- Partial HDLC message would be transferred to shared memory.
- DMAC accesses Next Message Pointer from current Message Descriptor.
- Return ownership of current Message Descriptor by writing the Receive Buffer Status Descriptor with ONR = HOST, ERROR = ABT (if INHRBSD = 0 in Receive Channel Configuration Descriptor).
- The BLP scans for the opening flag of the next HDLC message.
- Simultaneously, DMAC checks for message descriptor ownership before transferring received data to shared memory.

Channel Level Recovery Actions:

- None required.

Long Message (LNG)

In this case, the received HDLC message size was determined to be greater than the maximum allowed message size (per MAXSEL in Channel Configuration Descriptor).

Reasons include:

- Incorrect transmission of HDLC messages from distant end.

Effects of this error include:

- Interrupt Descriptor in Interrupt Queue with ERROR = LNG, DIR = 0 (if MSKMSG = 0 in Receive Channel Configuration Descriptor).
- DMAC accesses Next Message Pointer from current Message Descriptor.
- Return ownership of current Message Descriptor by writing the Receive Buffer Status Descriptor with ONR = HOST, ERROR = LNG (if INHRBSD = 0 in Receive Channel Configuration Descriptor).
- The BLP scans for the opening flag of the next HDLC message.
- Simultaneously, DMAC checks for message descriptor ownership before transferring received data to shared memory.

Channel Level Recovery Actions:

- None required.

Short Message (SHT)

In this case, the total received HDLC message size (including FCS) was determined to be less than the number of FCS bits specified for the receive channel. That is, for a channel configured for 16-bit FCS, a minimum of an 8-bit payload must be received to avoid a short message error. For this example, three octets must be received—minimum 1 octet for payload and 2 for FCS. Receiving 2 octets would be considered a short message.

Reasons:

- Bit errors during transmission.
- Incorrect transmission of HDLC messages from distant end.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = SHT, DIR = 0 (if MSKIDLE = 0 in Receive Channel Configuration Descriptor).
- Maintain ownership of current Message Descriptor.
- The BLP resumes scanning for opening flag of the next HDLC message.
- Simultaneously, MUSYCC checks for message descriptor ownership before proceeding with bit-level operations.

Channel Level Recovery Actions:

- None required.

SS7 Signal Unit Error Rate Interrupt (SUERR)

In this case, an error was detected in SS7 mode which caused a counter for SS7 related errors to equal or exceed the permitted threshold value. The threshold is stored on a per-channel group basis in the bit-field SUET in a Group Configuration Descriptor.

Reasons:

- Short SS7 message increments counter.
- FCS error in SS7 message increments counter.
- Octet alignment error in SS7 message
- Accumulation of 16 "octet count" type errors increments counter.

NOTE: Receiving 256 unerrored SS7 messages decrements the counter.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = SUERR, DIR = 0 (if MSKSUERR = 0 in Receive Channel Configuration Descriptor).
- The BLP scans for the opening flag of the next HDLC message.
- Simultaneously, DMAC checks for message descriptor ownership before transferring received data to shared memory.

Channel Level Recovery Actions:

- None required.

Transparent Mode

MUSYCC supports a transparent mode where no distinction is made between information and non-information bits in the data bit stream. This mode is assigned on a per channel and direction basis by the bit-field PROTOCOL in the Channel Configuration Descriptor.

In transparent mode, the following characteristics apply:

- All data bits are transferred between shared memory and the serial interface without protocol support such as those listed for the HDLC mode.
- Host must maintain the necessary data transfer rates at all times by providing message descriptors and data buffers for both the transmit and receive channels.
- The host must always set the bit-field EOM to 0 in each Transmit Buffer Descriptor. Setting EOM to 1 will cause indeterminate results. Due to EOM = 0, the other Transmit Buffer Descriptor bit-fields, IC, PADEN, PADCNT, and REPEAT, are ineffective.
- Bit-field NP in either the Receive or Transmit Buffer Descriptor are effective until an error condition causes the channel to automatically suspend bit-level operations.

Transmit Events

Transmit events are informational in nature and require no recovery actions.

End Of Buffer (EOB)

Reasons include:

- DMAC reached the end of a buffer by servicing a number of octets equal to the BLEN bit-field in a Transmit Buffer Descriptor. Note that for the transparent mode, EOM is not a valid event as there is no concept of messages.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = EOB, DIR = 1 (per EOBI in Transmit Buffer Descriptor).
- MUSYCC continues with normal message processing.
- MUSYCC requires more data buffers. The transmit channel autonomously outputs idle code if waiting for new data buffers.

Receive Events

Receive events are informational in nature and require no recovery actions.

End Of Buffer (EOB)

Reasons:

- BLP reached the end of a buffer by transferring into shared memory a number of octets equal to the BLEN bit-field in a Receive Buffer Descriptor.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = EOB, DIR = 0 (per EOBI in Receive Buffer Descriptor).
- MUSYCC continues with normal message processing.

Frame Recovery (FREC)

Reasons:

- BLP detected that the serial interface has transitioned from an out-of-frame to an in-frame condition.

Effects:

- Interrupt Descriptor in Interrupt Queue with EVENT = FREC, DIR = 0 (per MSKOOF in Group Configuration Descriptor).
- MUSYCC continues with normal message processing.

Transmit Errors

Transmit Errors are service-affecting and require a corrective action by the host to resume normal bit-level processing.

Underflow due to Host ownership of buffer (ONR)

In this case, sufficient data throughput from shared memory was not maintained to support the data rate of the serial interface. That is, ownership of messages was not handed over to MUSYCC in a timely manner.

Reasons:

- Degradation of the host subsystem or application software performance.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = ONR, DIR = 1.
- Idle Code transmission.
- Data from Data Buffer not being read.
- Buffer Descriptor not overwriting Buffer Status Descriptor.
- No additional activity on the PCI bus for this channel-direction.
- Transmit channel activity is suspended.

Channel Level Recovery Actions:

- Provide sufficient amount of data for transmission using lists of Message Descriptors with ownership given to MUSYCC.
- Reactivate transmit channel.

Underflow due to internal FIFO buffer under-run [BUFF]

In this case, the internal FIFO becomes empty when MUSYCC needed to output data bits (per serial interface clock rate) and MUSYCC had ownership of a message buffer in shared memory.

Reasons:

- Degradation of the host subsystem or application software performance.
- Congestion of the PCI bus.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = BUFF, DIR = 1.
- Continuous idle code transmission.
- Data from data buffer not being read.
- Buffer Descriptor not overwriting Buffer Status Descriptor.
- No additional activity on the PCI bus for this channel direction.
- Transmit channel activity is suspended.

Channel Level Recovery Actions:

- Provide sufficient amount of data for transmission using lists of Message Descriptors with ownership given to MUSYCC.
- Reactivate transmit channel.

Receive Errors

Receive errors are service-affecting and may require a corrective action by a controlling device to resume normal bit-level processing.

Overflow due to Host ownership of the buffer (ONR)

In this case, the host has not provided sufficient data buffer space to store received data from the serial interface and the internal FIFO overflows with received data bits.

Reasons:

- Degradation of the host subsystem or application software performance.
- Congestion of the PCI bus.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = ONR, DIR = 0.
- The received data in the internal FIFO is discarded and lost to the host.
- The channel continues to wait for a message buffer in shared memory to become available—ownership granted to MUSYCC.
- No additional activity on the PCI bus for this channel direction.
- The receive channel remains activated and resumes data bit reception into the internal FIFO at the serial interface line clock rate.

Channel Level Recovery Actions:

- Provide sufficient amount of shared memory for received data to be copied into using lists of Message Descriptors with ownership granted to MUSYCC.

Overflow due to internal FIFO buffer overrun (BUFF)

In this case, the internal FIFO buffer has not been completely copied to shared memory before more received data bits need to be stored in the FIFO. MUSYCC has access to a shared memory buffer in this case.

Reasons:

- Degradation of the host subsystem or application software performance.
- Congestion of the PCI bus.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = BUFF, DIR = 0.
- The received data in the internal FIFO is discarded and lost to the host.
- No additional activity on the PCI bus for this channel direction.
- Receive channel activity is suspended.

Channel Level Recovery Actions:

- If possible, increase internal FIFO space for this channel.
- Alleviate loading of the PCI bus.
- Reactivate receive channel with a channel activate or channel jump service request or with a slave write into the Receive Channel Configuration Table.

Change Of Frame Alignment (T1/E1 modes) (COFA)

In this case, loss of frame synchronization of a T1, E1, 2xE1 or 4xE1 occurred. As frame synchronization indicates the location of Timeslot 0 in the serial data stream, lacking frame synchronization causes received channelized data to become unaligned and unmappable.

Reasons:

- Signal failure detected by the physical interface providing the serial data, clock, and synchronization to the serial interface on MUSYCC.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = COFA, DIR = 0.
- If OOFABT bit-field is set to 0 in the Group Configuration Descriptor, then continue channel activity. That is, received data bits are sampled and eventually copied into shared memory.
- If bit-field OOFABT is set to 1 in the Group Configuration Descriptor, then suspend channel activity. Received data bits are discarded and lost to the host processor.

Channel Level Recovery Actions:

- If OOFABT = 1, once the received signal has recovered, reactivate receive channel with a Channel Activate or Channel Jump Service Request or with a slave write into the Receive Channel Configuration Table.

Out Of Frame (OOF)

Out-of-frame or loss-of-frame indicates that the entire serial data stream is invalid and data cannot be recovered from such a signal.

Reasons:

- Signal failure detected by the physical interface providing the serial data, clock, and synchronization to the serial interface on MUSYCC.

Effects:

- Interrupt Descriptor in Interrupt Queue with ERROR = OOF, DIR = 0.
- The received data in the internal FIFO is discarded and lost to the host.
- If OOFABT bit-field is set to 0 in the Group Configuration Descriptor, then continue channel activity but transfer "all 1s" data into shared memory for the duration of the OOF. When the OOF condition no longer applies, normal bit-level processing resumes automatically without host intervention.
- If OOFABT bit-field is set to 1 in the Group Configuration Descriptor, then suspend channel activity without transferring any data to shared memory.

Channel Level Recovery Actions:

- If OOFABT is set to 1 and once the OOF condition no longer applies, then the host must reactivate the receive channel.

Signaling System 7

SS7 Repeat Message Transmission

Signaling System 7 (SS7) requires the ability to continuously repeat a message under certain circumstances. The Repeat Message Transmission section of this document describes the repeat feature fully and is usable for an SS7 application.

Message Filtering

Message filtering is always enabled for a receive channel in which is configured for SS7-HDLC-CRC16 mode in Table 42, Channel Configuration Descriptor.

When reception of an unerrored message which has a payload length of 3, 4, or 5 octets and the required 2-octet FCS, MUSYCC will transfer the data into the memory buffer as usual and then enter the message filtering mode. This mode does not apply to messages with payloads greater than 5 octets.

In this mode, the Next Message Descriptor is processed in the normal manner (Receive Buffer Status Descriptor is written, Next Pointer is used to read Next Message Descriptor, ONR and NP bits are checked, and so on...). In this case, the current memory buffer is not filled until MUSYCC exits the message filtering mode.

The 3-, 4-, or 5-octet payload and the 2-octet FCS is stored inside MUSYCC and is considered a golden message by being 3, 4, or 5 octets long. Subsequent received messages are compared to this message.

Each bit of the subsequent message will be bit-wise compared to the contents of the golden message. The following can occur:

- Match: Payload and FCS of golden message match the current message before a closing flag is received for the current message. The maskable interrupt MSKSFILT is generated to the host. MUSYCC remains in message filtering mode and the golden message is retained.
- Mismatch: A bit-wise mismatch is detected between a bit in the golden message and the corresponding bit in the current message before a closing flag is received for the current message. MUSYCC immediately exits the Filter Mode.

The treatment for the current message becomes a normal message treatment. If the current message is unerrored and qualifies to become a golden message, it is transferred to the current Receive Data Buffer and becomes the new golden message internally to MUSYCC.

If the current message is unerrored and greater than 6 octets total, it cannot become a golden message, MUSYCC continues normal message treatment.

Signal Unit Error Rate Monitoring (SUERM)

The Signal Unit Error Rate Monitor (SUERM) facility provides a 6-bit counter which serves as a real time figure-of-merit for the receive link integrity. It is incremented and decremented in a leaky-bucket style based upon integration of good message and bad octet periods on the receive channel.

SUERM Counter Incrementing

The SUERM counter is incremented when any signal unit error occurs. A signal unit error is defined as one of the following events:

- SHT: Short Frame error.
- ALIGN: Octet Alignment error.
- CRC: FCS Mismatch error.
- Accumulation of 16 octet count errors.

Short Frame errors, Octet Alignment errors, or CRC/FCS Mismatch errors generate a maskable interrupt, SHT, LNG, CRC respectively, toward the host and cause the SUERM counter to be incremented. Each time the SUERM counter is incremented, the maskable interrupt SINC is generated to the host to indicate this condition.

SUERM Octet Counting

Octet counting mode is entered if seven consecutive 1 are detected (abort condition), or the received message length exceeds the selected maximum received-frame length register value (long frame error)

When in Octet counting mode, a 4-bit bad octet counter is incremented for every received octet until a condition is met to exit this mode. As the counter rolls over from 15 to 0, the SUERM counter is incremented by one. This mode is exited when a correctly checked signal unit (unerrored message) is detected.

When not in octet counting mode, the value of the 4-bit bad octet counter is maintained from the last octet counting mode entered and starts to increments again from that value when the mode is entered again.

SUERM Counter Decrementing

The SUERM counter is decremented when 256 unerrored messages are received. An unerrored message indicates that a short frame or long frame error was not detected, no octet alignment error was detected, and no CRC error was detected. Each unerrored message increments an 8-bit good message counter. When this counter rolls over from 255 to 0, the SUERM counter is decremented by one.

While in octet counting mode, the value of the 8-bit good message counter is maintained from the last non-octet counting mode and starts to increment again from that value when a good message causes an exit from the octet counting mode.

When the SUERM counter is decremented, the maskable interrupt SDEC is generated to the host to indicate this condition.

Self-Servicing Buffers

The transmit and receive Buffer Descriptors and Buffer Status Descriptors are designed to facilitate a mechanism know as "self-servicing buffers." This mechanism allows the host to configure MUSYCC to fill a linked list of data buffers as it receives a complete message through a receive channel, then empty the same list of data buffers through a transmit channel without any further host intervention.

The mechanism works as follows:

1. Host initializes linked list of message descriptors in shared memory.
2. Host configures receive channel to point to first message descriptor.
3. Host configures transmit channel to point to the same message descriptor.
4. The OWNER bit-field in the buffer descriptor in the message descriptor is set to 0. For the transmitter, this means the buffer is owned by the host. For the receiver, this means the buffer is owned by MUSYCC.
5. Both receive and transmit channel are activated.
6. As the receiver detects a valid incoming message. it begins filling the first data buffer from the linked list. The transmitter remains idle, polling the OWNER bit in the Transmit Buffer Descriptor.
7. As the receiver fills the first buffer, it writes the Receive Buffer Status Descriptor (and sets OWNER to 1) and moves to the Next Message Pointer which identifies the next Receive Data Buffer on the linked list.
8. The transmit channel detects the OWNER set to 1 for the first Transmit Data Buffer and assumes ownership of the buffer and begins emptying data to the serial port.
9. Upon detecting the end of a message, the receiver writes the Receive Buffer Status Descriptor and marks this last buffer as containing the End of Message and sets the buffer length field, BLEN to indicate the amount of data received in this last buffer.
10. Upon the transmitter detecting the End of Message marking in the last buffer, the transmitter sends the final BLEN amount of data out the serial port and writes the Transmit Buffer Status Descriptor (and sets OWNER to 0) and moves into the idle state again.
11. Go to step 6 to continue processing the next message.

It is important to note that for self-servicing buffers, the host does not need to write to any descriptors for receive or transmit operations. MUSYCC writes the Receive Buffer Status Descriptor which is subsequently used as the Transmit Buffer Descriptor.

Electrical and Environmental Specifications

Absolute Maximum Ratings

Stressing the device parameters above absolute maximum ratings may cause permanent damage to the device. This is a stress rating only. Functional operation of the device at these or any other conditions beyond those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 60. Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Supply Voltage	V_{dd}	–0.3 to 7	V
DC Input Voltage	V_{in}	–0.5 to V_{dd} +0.5	V
Continuous Power Dissipation	P_d	tbd	mW
Operating Junction Temperature	T_{jc}	125	°C
Storage Temperature	T_s	–55 to +125	°C

Recommended Operating Conditions

Table 61. Recommended 5 V Operating Conditions

Parameter	Symbol	Value	Unit
Supply Voltage	V_{dd}	4.75 to 5.25	V
Ambient Operating Temperature KPF EPF	T_{ac}	 0 to +70 –40 to +85	 ˚C ˚C
High-Level Input Voltage	V_{ih}	2.0 to V_{dd} +0.3	V
Low-Level Input Voltage	V_{il}	–0.3 to 0.8	V
High-Level Output Current Source	I_{oh}	200 to 400	µA
Low Level Output Current Sink	I_{ol}	2 to 3	mA
Output Capacitive Loading	C_{ld}	60	pF

Electrical Characteristics

Table 62. Electrical Characteristics for 5 V Operation

Parameter	Symbol	Value	Units
High-Level Output Voltage	V_{oh}	2.4	V
Low-Level Output Voltage	V_{ol}	0.4	V
Input Leakage Current	I_{l}	–10 to 10	µA
Three-state Leakage Current	I_{oz}	–10 to 10	µA
Resistive Pullup Current	I_{pr}	100 to 500	µA
Supply Current	I_{dd}	tbd	mA

Timing and Switching Specifications

Overview

This section defines the timing and switching characteristics of MUSYCC. The major subsystems include the host interface, the expansion bus interface, and the serial interface. The host interface is Peripheral Component Interface (PCI) compliant. For other references to PCI, see the PCI Local Bus Specification, Revision 2.1, June 1, 1995. The expansion bus and serial bus interfaces are similar to the host interface timing characteristics; the differences and specific characteristics common to either interface are further defined.

Host Interface (PCI) Timing and Switching Characteristic

Reference the PCI Local Bus Specification, Revision 2.1, June 1, 1995 for information on:

- Indeterminate inputs and metastability.
- Power requirements, sequencing, decoupling.
- PCI DC specifications.
- PCI AC specifications.
- PCI V/I curves.
- Maximum AC ratings and device protection.

Table 63. PCI Interface DC Specifications

Symbol	Parameter	Condition	Min	Max	Units
V_{dd}	Supply Voltage		4.75	5.25	V
V_{ih}	Input High Voltage		2.0	V_{dd}+ 0.5	V
V_{il}	Input Low Voltage		–0.5	0.8	V
I_{ih}	Input High Leakage Current(1)	V_{in} = 2.7 V		70	µA
I_{il}	Input Low Leakage Current(1)	V_{in} = 0.5 V		–70	µA
V_{oh}	Output High Voltage	I_{out} = –2 mA	2.4		V
V_{ol}	Output Low Voltage(2)	I_{out} = 3 mA I_{out} = 6 mA		0.55	V
C_{in}	Input Pin Capacitance			10	pF
C_{clk}	PCLK Pin Capacitance		5	12	pF
C_{idsel}	IDSEL Pin Capacitance(3)			8	pF
L_{pin}	Pin Inductance			20	nH

Notes: (1). Input leakage currents include hi-Z output leakage for all bidirectional buffers with three-state outputs.
(2). Signals without pull-up resistors must have 3 mA low output current. Signals requiring pull-up must have 6 mA; the latter include FRAME*, TRDY*, IRDY*, DEVSEL*, STOP*, SERR*, and PERR*.
(3). Lower capacitance on this input-only pin allows for non-resistive coupling to AD[xx].

Table 64. PCI Clock (PCLK) Waveform Parameters, 5 V Clock

Symbol	Parameter	Min	Max	Units
T_{cyc}	Clock Cycle Time(1)	30	•	ns
T_{high}	Clock High Time	11		ns
T_{low}	Clock Low Time	11		ns
–	Clock Slew Rate(2)	1	4	mV/ns
V_{ptp}	Peak-to-Peak Voltage	2		V

Notes: (1). MUSYCC works with any clock frequency between DC and 33 MHz, nominally. The clock frequency may be changed at any time during operation of the system as long as clock edges remain monotonic, and minimum cycle and high and low times are not violated. The clock may only be stopped in a low state.
(2). Rise and fall times are specified in terms of the edge rate measured in V/ns. This slew rate must be met across the minimum peak-to-peak portion of the clock waveform

Figure 23. PCI Clock (PCLK) Waveform, 5 V Clock

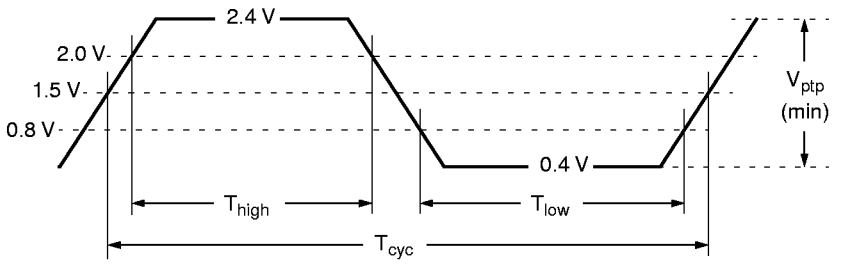


Table 65. PCI Reset Parameters

Symbol	Parameter	Min	Max	Units
T_{rst}	Reset Active Time after Power Stable	1		ms
T_{rst_clk}	Reset Active Time after Clock Stable	100		µs
V_{nom}	Nominal Voltage Level(1)			V
–	RST* Slew Rate(2)	50	–	mV/ns
T_{fail}	Power Failure Detect Time(3)			
$T_{rst-off}$	Reset Active to Float Delay			

Notes: (1). The nominal voltage level refers to a voltage test point in the power up curve where the system can declare start of a "power good" signal.
(2). The minimum RST* slew rate applies only to the rising (deassertion) edge of the reset signal, and ensures that system noise cannot render an otherwise monotonic signal to appear to bounce in the switching range.
(3). The value of T_{fail} is the minimum of:
a. 500 ns (max) from power rail going out of specification by exceeding specified tolerances by more than 500 mV
b. 100 ns (max) from 5 V rail falling below 3.3 V rail by more than 300 mV.

Figure 24. PCI Reset Timing

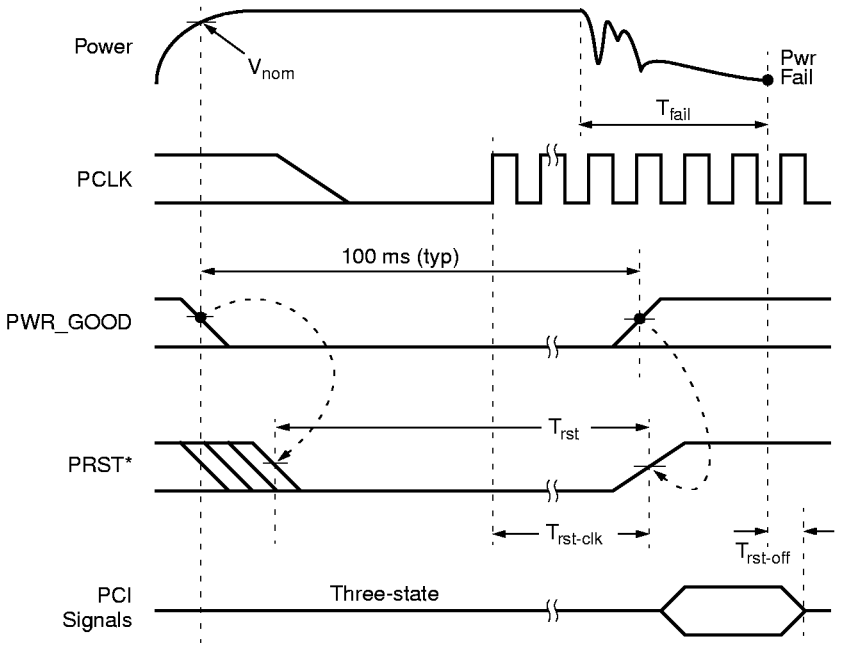


Table 66. PCI Input/Output Timing Parameters

Symbol	Parameter	Min	Max	Units
T_{val}	PCLK to Signal Valid Delay – Bused Signal(1, 2)	2	11	ns
T_{val} (ptp)	PCLK to Signal Valid Delay – Point To Point(1, 2)	2	12	ns
T_{on}	Float to Active Delay(3)	2		ns
T_{off}	Active to Float Delay(3)		28	ns
T_{ds}	Input Setup Time to Clock– Bused Signal(2)	7		ns
T_{su} (ptp)	Input Setup Time to Clock – Point To Point(2)	10, 12		ns
T_{dh}	Input Hold Time from Clock	0		ns

Notes: (1). Minimum and maximum times are evaluated at 80 pF equivalent load. Actual test capacitance may vary, and results should be correlated to these specifications.
(2). REQ* and GNT* are the only point-to-point signals, and have different output valid delay and input setup times than do bused signals. GNT* has a setup of 10; REQ* has a setup of 12.
(3). For purposes of active/float timing measurements, the hi-z or off state is defined to be when the total current delivered through the component pin is less than or equal to the leakage current specification at 80 pF equivalent load.

Table 67. PCI Input/Output Measure Conditions

Symbol	Parameter	Value	Units
V_{th}	Voltage Threshold High(1)	2.4	V
V_{tl}	Voltage Threshold Low(1)	0.4	V
V_{test}	Voltage Test Point	1.5	V
V_{max}	Maximum Peak-to-Peak(2)	2.0	V
–	Input Signal Edge Rate	1	V/ns

Notes: (1). The input test for the 5 V environment is done with 400 mV of overdrive (over V_{ih} and V_{il}). Timing parameters must be met with no more overdrive than this. Production testing may use different voltage values, but must correlate results back to these parameters.
(2). V_{max} specifies the maximum peak-to-peak voltage waveform allowed for measuring input timing. Production testing may use different voltage values, but must correlate results back to these parameters.

Figure 25. PCI Output Timing Waveform

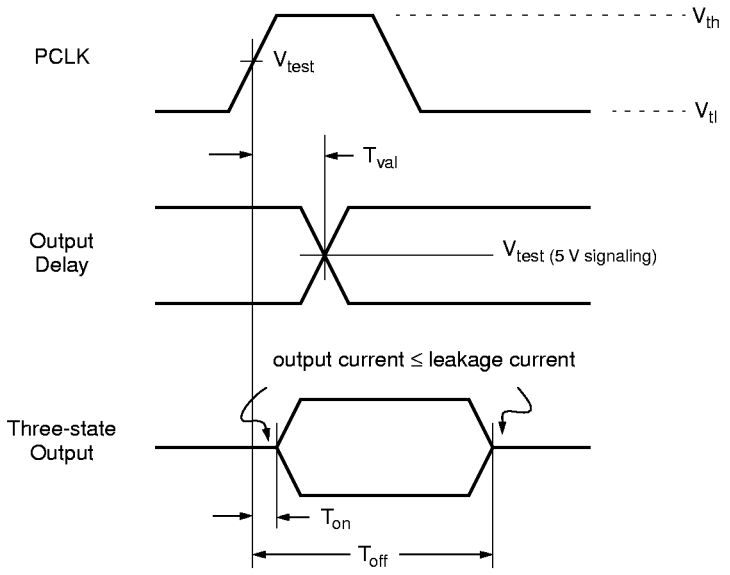


Figure 26. PCI Input Timing Waveform

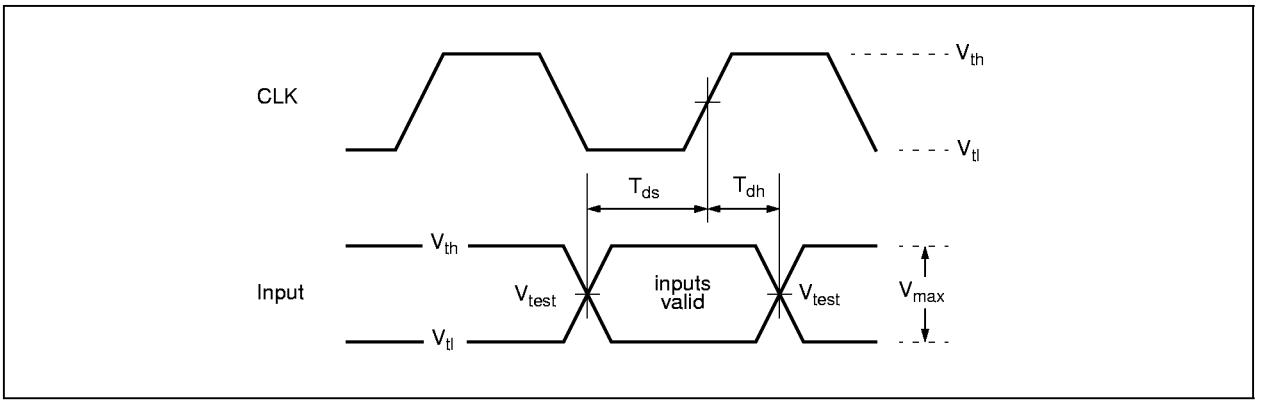


Figure 27. PCI Read Multiple Operation

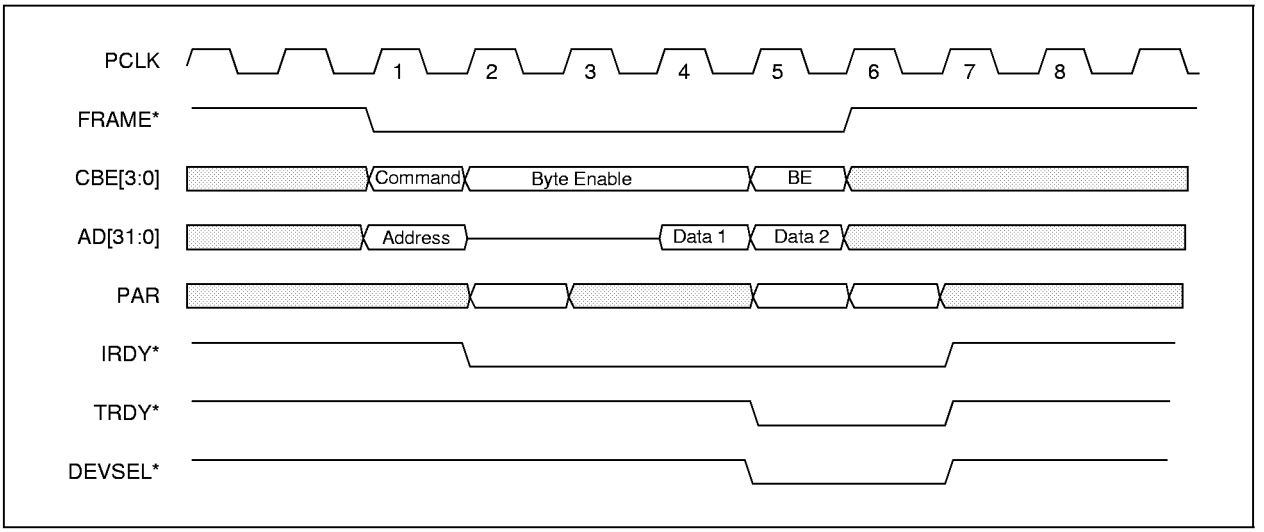


Figure 28. PCI Write Multiple Operation

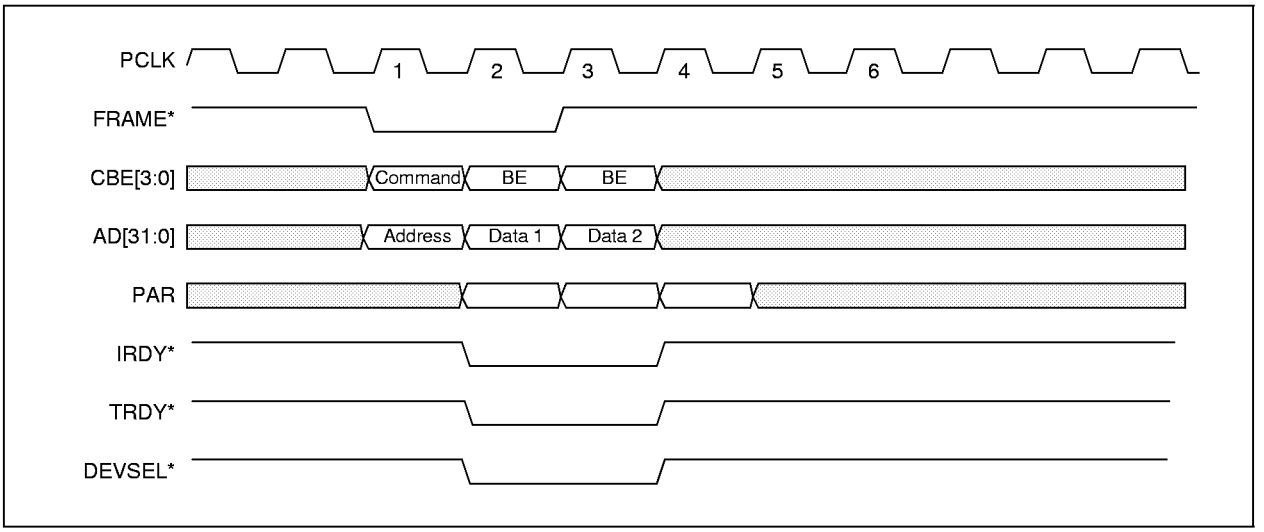
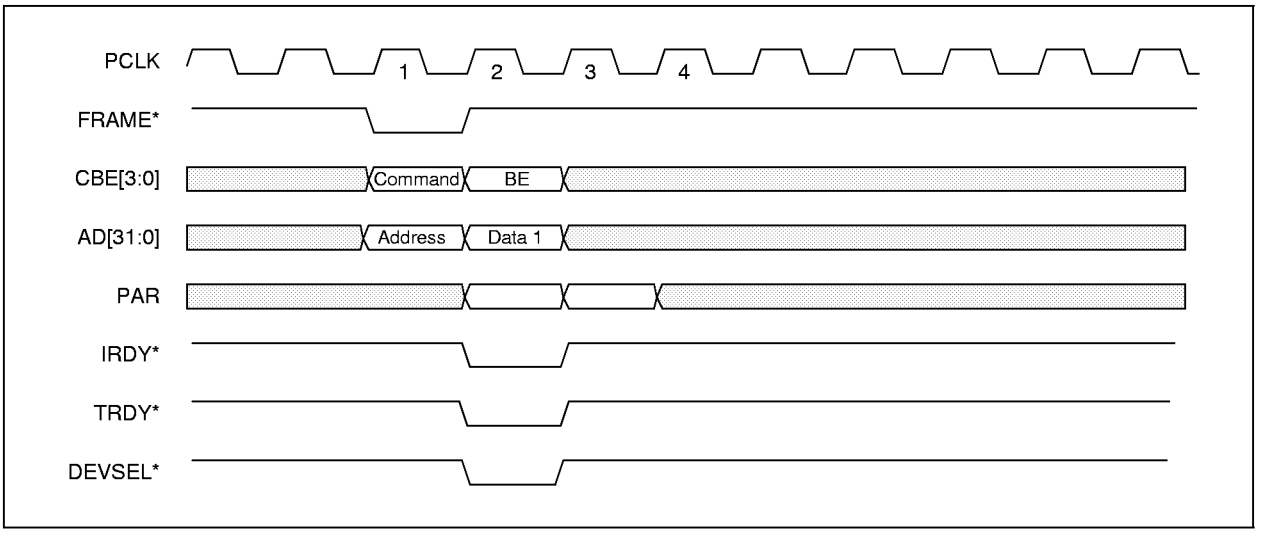


Figure 29. PCI Write Single Operation



Expansion Bus (EBUS) Timing and Switching Characteristic

The EBUS timing is derived directly from the PCI clock (PCLK) input to MUSYCC. The ECLK output is output as an inverted and a half-clock phase shifted PCLK.

The EBUS input/output timing characteristics are identical to the PCI input/output timing characteristics.

The EBUS clock waveform characteristics are identical to the PCI clock waveform characteristics.

Figure 30. ECLK to PCLK Relationship

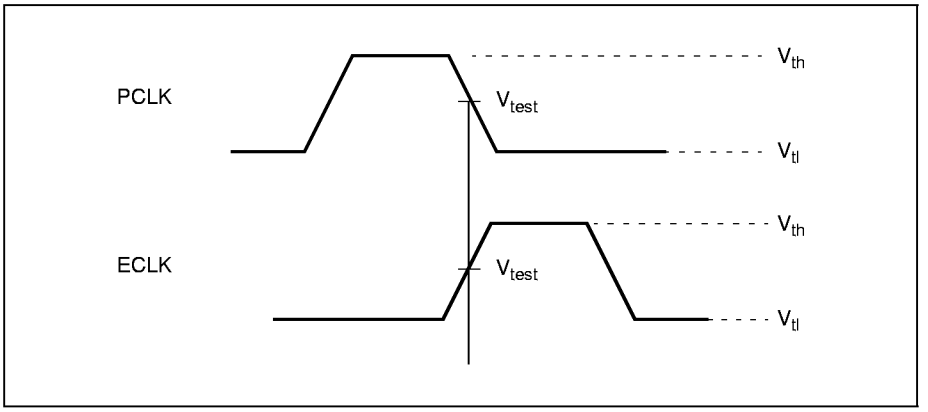
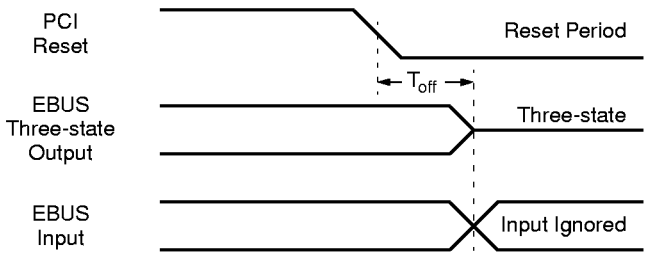


Table 68. EBUS Reset Parameters

Symbol	Parameter	Min	Max	Units
T_{off}	Active to Inactive Delay(1)		28	ns

Notes: (1). For purposes of active/float timing measurements, the hi-z or off state is defined to be when the total current delivered through the component pin is less than or equal to the leakage current specification.

Figure 31. EBUS Reset Timing



Note: The EBUS reset is dependent on the PRST* (PCI Reset) signal being asserted low.

Table 69. EBUS Input/Output Timing Parameters

Symbol	Parameter	Min	Max	Units
T_{val}	ECLK to Signal Valid Delay – Bused Signal(1)	2	11	ns
T_{val} (ptp)	ECLK to Signal Valid Delay – Point To Point(1)	2	12	ns
T_{on}	Float to Active Delay(2)	2		ns
T_{off}	Active to Float Delay(2)		28	ns
T_{ds}	Input Setup Time to Clock – Bused Signal	7		ns
T_{ds} (ptp)	Input Setup Time to Clock – Point To Point	10		ns
T_{dh}	Input Hold Time from Clock	0		ns

Notes: (1). Minimum and maximum times are evaluated at 80 pF equivalent load. Actual test capacitance may vary, and results should be correlated to these specifications.
(2). For purposes of active/float timing measurements, the hi-z or off state is defined to be when the total current delivered through the component pin is less than or equal to the leakage current specification at 80 pF equivalent load.

Table 70. EBUS Input/Output Measure Conditions

Symbol	Parameter	Value	Units
V_{th}	Voltage Threshold High(1)	2.4	V
V_{tl}	Voltage Threshold Low(1)	0.4	V
V_{test}	Voltage Test Point	1.5	V
V_{max}	Maximum Peak-to-Peak(2)	2.0	V
–	Input Signal Edge Rate	1	V/ns

Notes: (1). The input test for the 5 V environment is done with 400 mV of overdrive (over V_{ih} and V_{il}). Timing parameters must be met with no more overdrive than this. Production testing may use different voltage values, but must correlate results back to these parameters.
(2). V_{max} specifies the maximum peak-to-peak voltage waveform allowed for measuring input timing. Production testing may use different voltage values, but must correlate results back to these parameters.

Figure 32. EBUS Output Timing Waveform

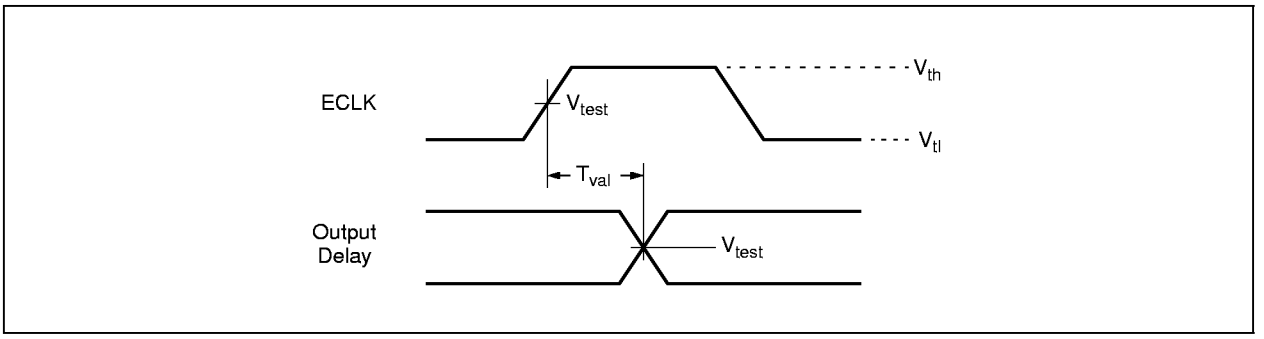
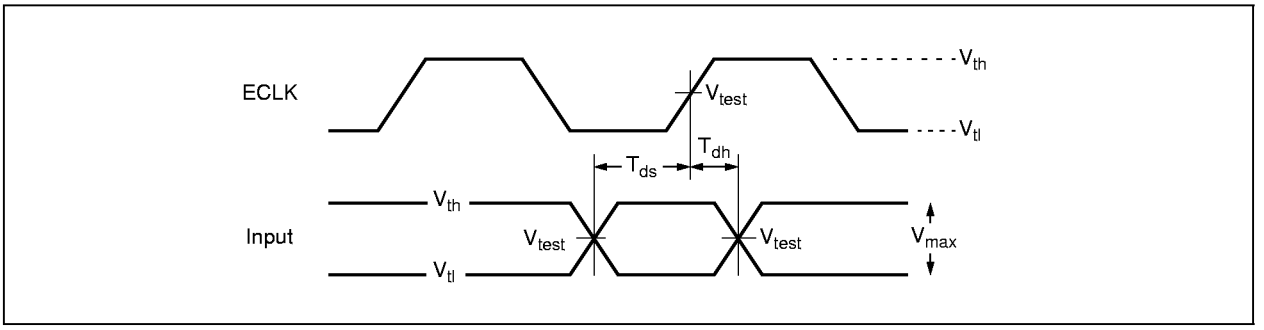
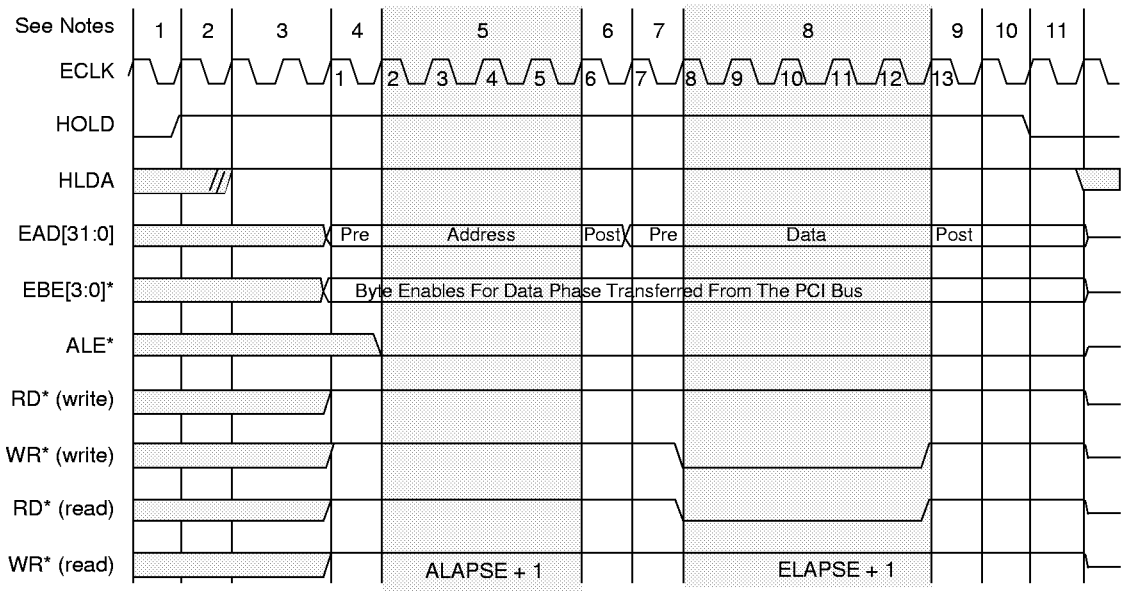


Figure 33. EBUS Input Timing Waveform



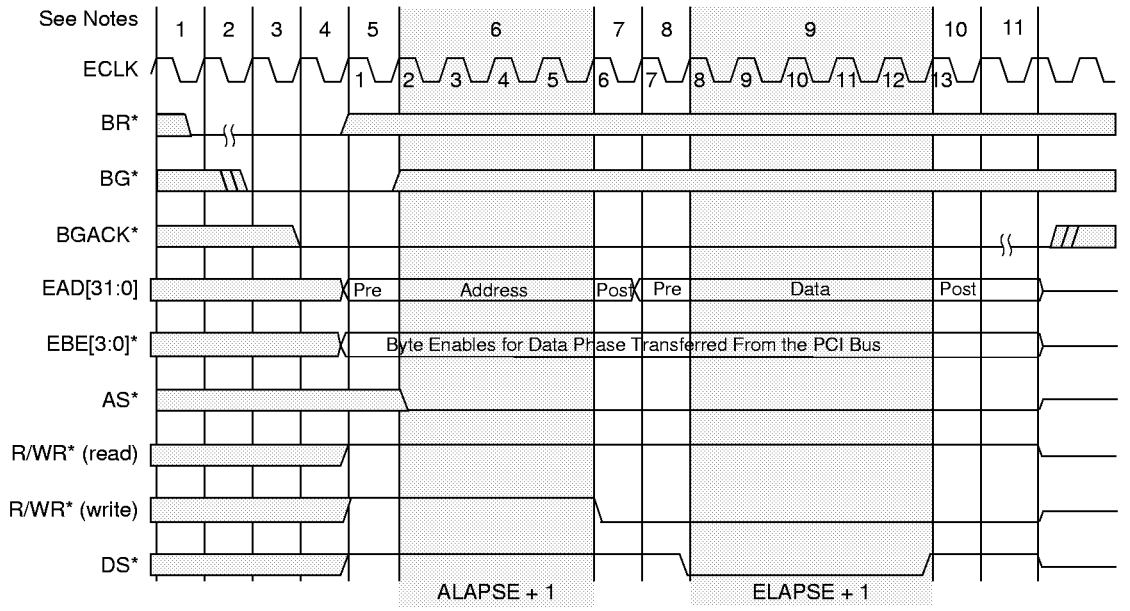
EBUS Arbitration Timing Specification

Figure 34. EBUS Write/Read Cycle, Intel-Style



Notes:
1. MUSYCC waits BLAPSE+1 ECLK periods between two sequential accesses of the EBUS.
2. The time between HOLD assertion and HLDA assertion is dependent on MPU bus arbitration circuitry.
3. The time between HLDA assertion and MUSYCC starting the EBUS address-data cycle (t.b.d. minimum 1 ECLK).
4. PRE-address is the address setup time and is always 1 ECLK period in length.
5. MUSYCC keeps address bits valid for ALAPSE+1 ECLK periods past the setup time.
6. POST-address is the address hold time where ALE* is asserted and valid address bits can be read.
7. PRE-data is the data setup time and is always 1 ECLK period in length.
8. MUSYCC keeps data bits valid for ELAPSE+1 ECLK periods past the setup time. During this time MUSYCC strobes either the RD* or WR* signal to indicate the data bits are valid.
9. POST-data keeps data bits valid for 1 ECLK period past the deassertion of RD* or WR*.
10. The time between end of EBUS address-data cycle and HOLD deassertion (t.b.d. minimum 1 ECLK).
11. The time between HOLD deassertion and HLDA deassertion is dependent on MPU bus arbitration circuitry.

Figure 35. EBUS Write/Read Cycle, Motorola-Style



Notes:
1. MUSYCC waits BLAPSE+1 ECLK periods between two sequential accesses of the EBUS.
2. The time between BR* assertion and BG* assertion is dependent on MPU bus arbitration circuitry.
3. The time between BG* assertion and BGACK* assertion by MUSYCC (t.b.d. minimum 1 ECLK).
4. The time between BGACK* assertion and MUSYCC starting the address-data cycle (t.b.d. minimum 1 ECLK).
5. PRE-address is the address setup time and is always 1 ECLK period in length. Also, the time between BR* deassertion and BG* deassertion is dependent on MPU bus arbitration circuitry.
6. MUSYCC keeps address bits valid for ALAPSE+1 ECLK periods past the setup time.
7. POST-address is the address hold time where AS* is asserted and valid address bits can be read.
8. PRE-data is the data setup time and is always 1 ECLK period in length.
9. MUSYCC keeps data bits valid for ELAPSE+1 ECLK periods past the setup time. During this time MUSYCC strobes either the R/WR* or DS* signal to indicate the data bits are valid for write or read.
10. POST-data keeps data bits valid for 1 ECLK period past the deassertion of DS*.
11. The time between end of EBUS address-data cycle and BGACK* deassertion (t.b.d. minimum 1 ECLK).

Serial Interface Timing and Switching Characteristics

Table 71. Serial Interface Clock (RCLK, TCLK) Parameters

Symbol	Parameter	Min	Max	Units
F_c	Clock Frequency	DC	8.192 ±10%	MHz
T_r	Clock Rise Time	–	2	ns
T_f	Clock Fall Time	–	2	ns

Figure 36. Serial Interface Clock (RCLK,TCLK) Waveform

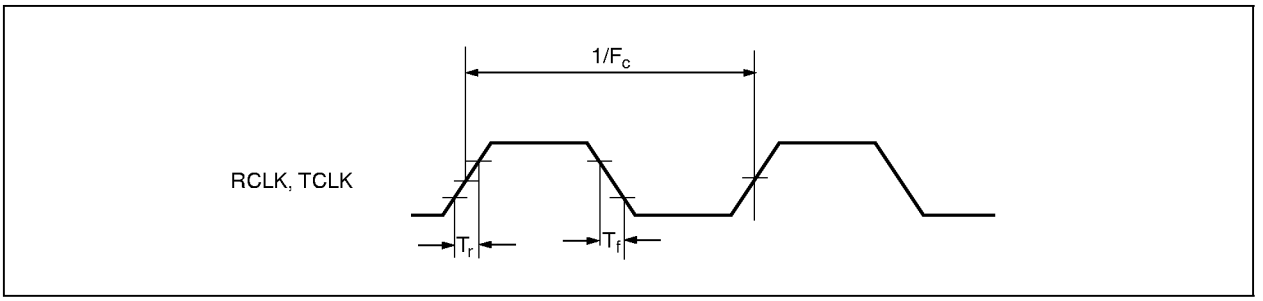


Table 72. Serial Interface Input/Output Timing Parameters

Symbol	Parameter	Min	Max	Units
T_{val}	Clock to Signal Valid Delay	2	12	ns
T_{ds}	Data Setup Time	10	–	ns
T_{dh}	Data Hold Time	10	–	ns

Table 73. Serial Interface Input/Output Measure Conditions

Symbol	Parameter	Value	Units
V_{th}	Voltage Threshold High(1)	2.4	V
V_{tl}	Voltage Threshold Low(1)	0.4	V
V_{test}	Voltage Test Point	1.5	V
V_{max}	Maximum Peak-to-Peak(2)	2.0	V
–	Input Signal Edge Rate	1	V/ns

Notes: (1). The input test for the 5 V environment is done with 400 mV of overdrive (over V_{ih} and V_{il}). Timing parameters must be met with no more overdrive than this. Production testing may use different voltage values, but must correlate results back to these parameters.
(2). V_{max} specifies the maximum peak-to-peak voltage waveform allowed for measuring input timing. Production testing may use different voltage values, but must correlate results back to these parameters.

Figure 37. Serial Interface Data Input Waveform

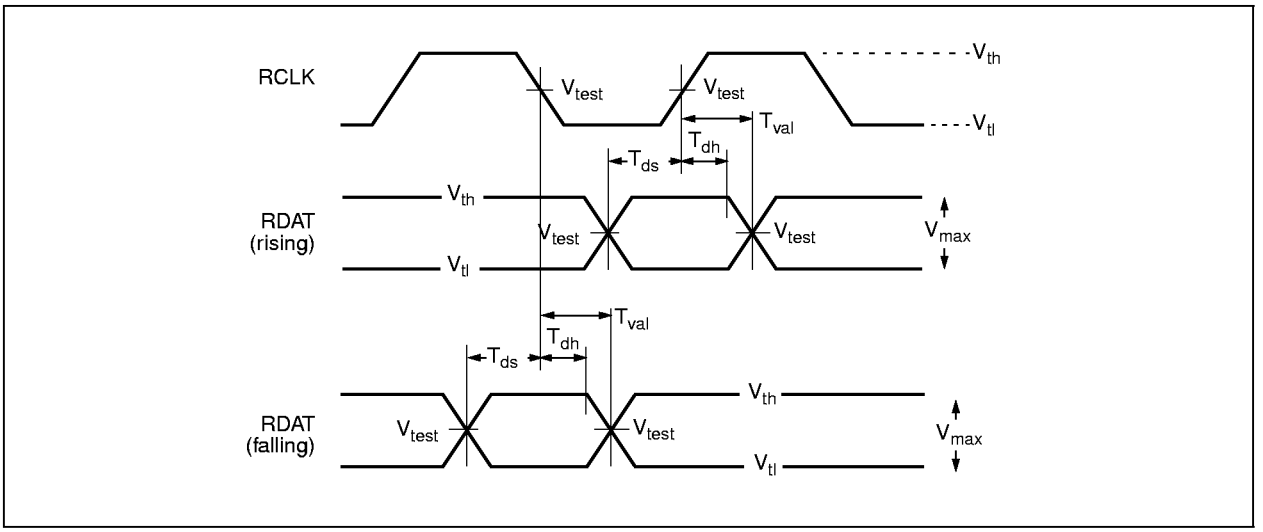


Figure 38. Serial Interface Data Delay Output Waveform

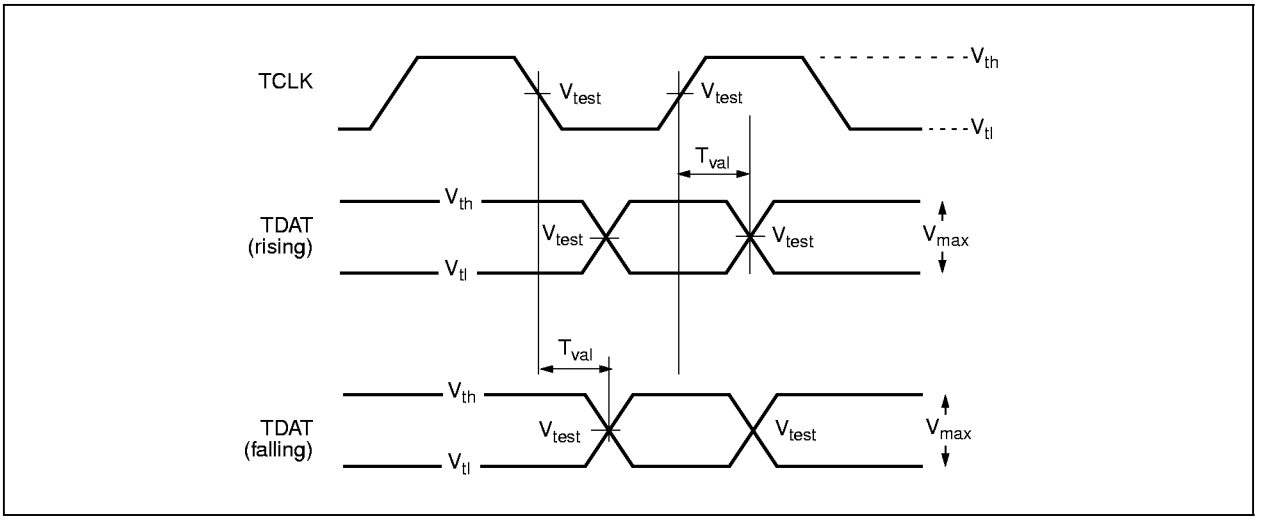
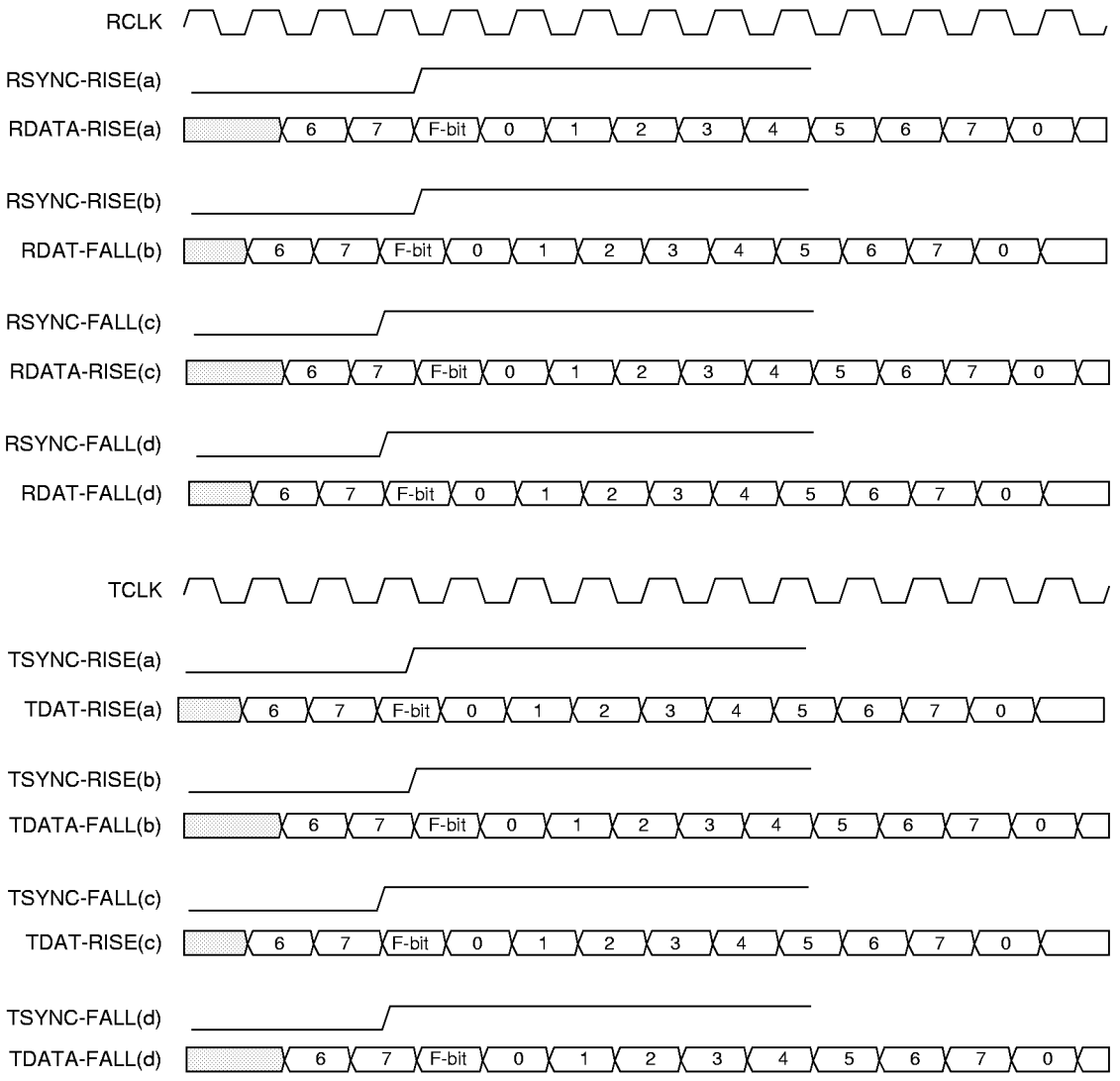
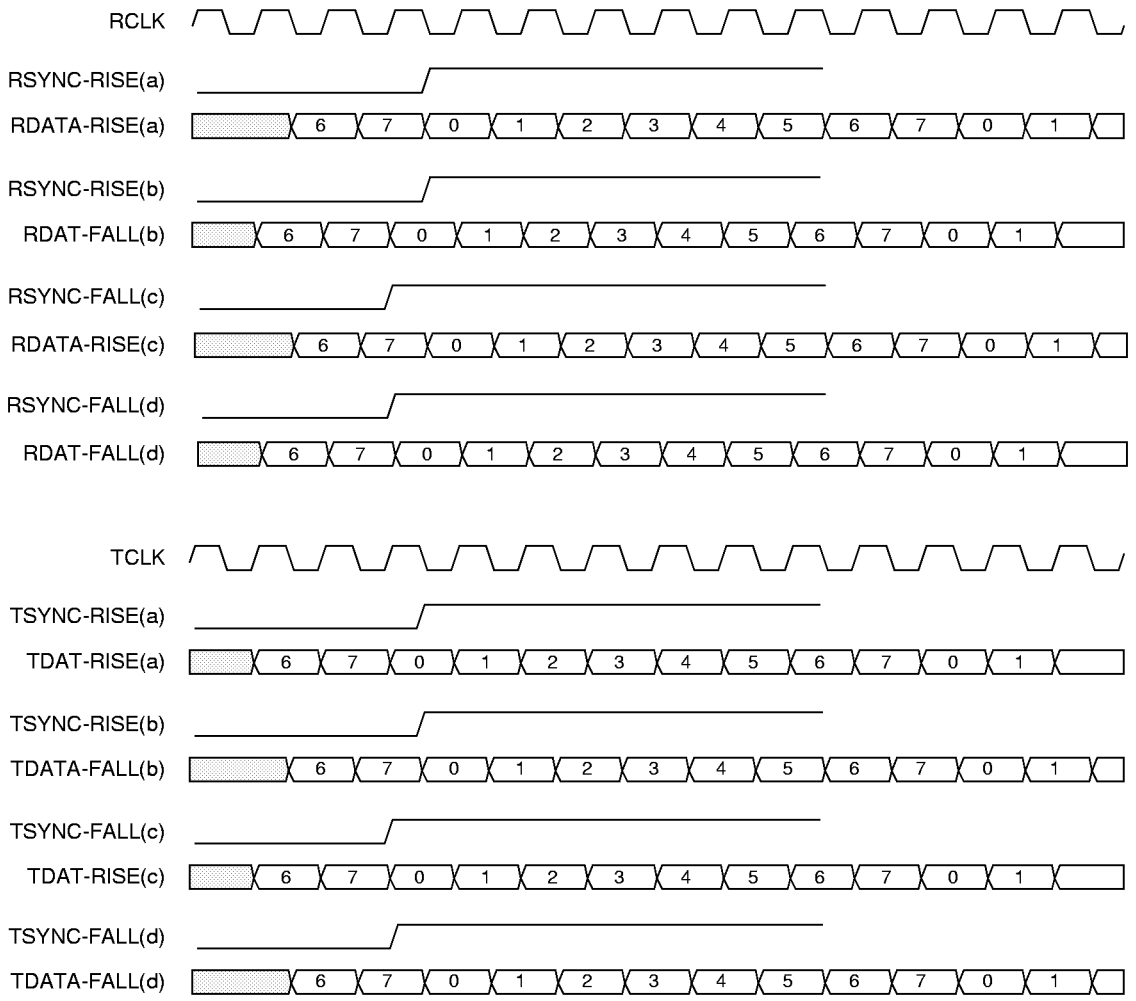


Figure 39. Transmit and Receive T1 Mode



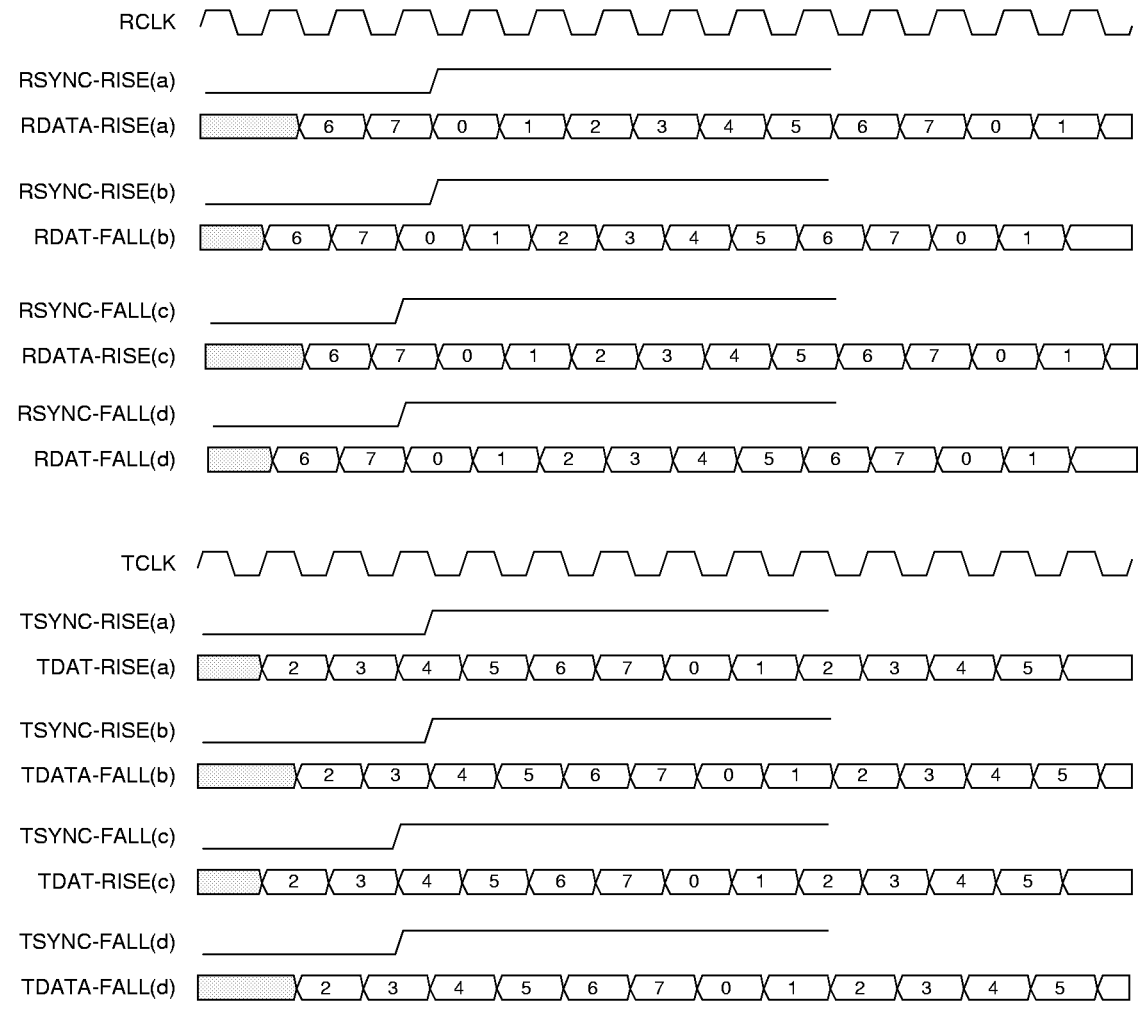
Notes:
1. T1 Mode employs 24 timeslots (0–23) with 8 bits per timeslot (0–7) and 1 Frame-bit every 193 clock periods. One frame of 193 bits occurs every 125 µs—1.544 MHz.
2. RSYNC and TSYNC must be asserted for a minimum of 1 CLK period.
3. MUSYCC can be configured to sample RSYNC, TSYNC, RDAT, and TDAT on either a rising or falling clock edge independently of any other signal sampling configuration.
4. Relationships between the various configurations of active edges for the synchronization signal and the data signal are shown using a common clock signal for receive and transmit operations. Note the relationship between the frame bit (within RDAT, TDAT) and the frame synchronization signal (e.g. RSYNC, TSYNC).
5. All received signals (e.g. RSYNC, RDAT, TSYNC) are “sampled” in on the specified clock edge (e.g. RCLK, TCLK). All transmit data signals (TDAT) are “latched” on the specified clock edge.
6. In configuration (a), synchronization and data signals are sampled/latched on a rising clock edge.
7. In configuration (b), synchronization signal is sampled on a rising clock edge and the data signal is sampled/latched on a falling clock edge.
8. In configuration (c), synchronization signal is sampled on a falling clock edge and the data signal is sampled/latched on a rising clock edge.
9. In configuration (d), synchronization and data signals are sampled/latched on a falling clock edge.

Figure 40. Transmit and Receive E1 (also 2xE1, 4xE1) Mode



Notes:
1. E1 Mode employs 32 timeslots (0–31) with 8 bits per timeslot (0–7) and 256 bits per frame and one frame every 125 µs — 2.048 MHz.
2. 2xE1 Mode employs 64 timeslots (0–63) with 8 bits per timeslot (0–7) and 512 bits per frame and one frame every 125 µs — 4.096 MHz.
3. 4xE1 Mode employs 128 timeslots (0–127) with 8 bits per timeslot (0–7) and 1024 bits per frame and one frame every 125 µs — 8.192 MHz.
4. RSYNC and TSYNC must be asserted for a minimum of 1 CLK period.
5. MUSYCC can be configured to sample RSYNC, TSYNC, RDAT, and TDAT on either a rising or falling clock edge independently of any other signal sampling configuration.
6. Relationships between the various configurations of active edges for the synchronization signal and the data signal are shown using a common clock signal for receive and transmit operations. Note the relationship between the frame bit (within RDAT, TDAT) and the frame synchronization signal (e.g. RSYNC, TSYNC).
7. All received signals (e.g. RSYNC, RDAT, TSYNC) are “sampled” in on the specified clock edge (e.g. RCLK, TCLK). All transmit data signals (TDAT) are “latched” on the specified clock edge.
8. In configuration (a), synchronization and data signals are sampled/latched on a rising clock edge.
9. In configuration (b), synchronization signal is sampled on a rising clock edge and the data signal is sampled/latched on a falling clock edge.
10. In configuration (c), synchronization signal is sampled on a falling clock edge and the data signal is sampled/latched on a rising clock edge.
11. In configuration (d), synchronization and data signals are sampled/latched on a falling clock edge.

Figure 41. Transmit and Receive Nx64 Mode



Notes:
1. Nx64 Mode employs N timeslots with 8 bits (0–7) per timeslot.
2. Assertion of TSYNC must precede transmission of bit-0 of a frame by exactly 4 line clock periods due to the internal buffer scheme used for transmitting of Nx64 Mode data bits.
3. RSYNC and TSYNC signals must be provided for every received and transmitted frame in Nx64 Mode.
4. If N= 1, the minimum, then 8 bits/frame = 64 KHz. If N = 128, the maximum, then 1024 bits/frame = 8.192 MHz.
5. Relationships between the various configurations of active edges for the synchronization signal and the data signal are shown using a common clock signal for receive and transmit operations. Note the relationship between the frame bit (within RDAT, TDAT) and the frame synchronization signal (e.g. RSYNC, TSYNC).
6. All received signals (e.g. RSYNC, RDAT, TSYNC) are “sampled” in on the specified clock edge (e.g. RCLK, TCLK). All transmit data signals (TDAT) are “latched” on the specified clock edge.
7. In configuration (a), synchronization and data signals are sampled/latched on a rising clock edge.
8. In configuration (b), synchronization signal is sampled on a rising clock edge and the data signal is sampled/latched on a falling clock edge.
9. In configuration (c), synchronization signal is sampled on a falling clock edge and the data signal is sampled/latched on a rising clock edge.
10. In configuration (d), synchronization and data signals are sampled/latched on a falling clock edge.

Test and Diagnostic Interface Timing

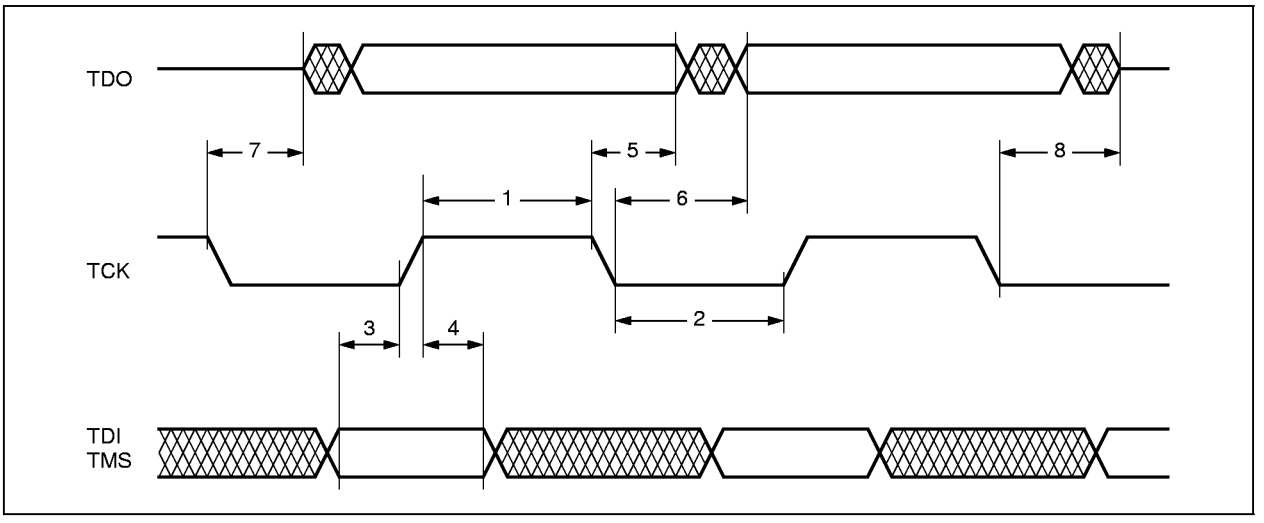
Table 74. Test and Diagnostic Interface Timing Requirements

Symbol	Parameter	Minimum	Maximum	Units
1	TCK Pulse-Width High	80		ns
2	TCK Pulse-Width Low	80		ns
3	TMS, TDI Setup Prior to TCK Rising Edge(1)	20		ns
4	TMS, TDI Hold after TCK High(1)	20		ns
Notes: (1). Also applies to functional inputs for SAMPLE/PRELOAD and EXTEST instructions.				

Table 75. Test and Diagnostic Interface Switching Characteristics

Symbol	Parameter	Minimum	Maximum	Units
5	TDO Hold after TCK Falling Edge(1)	0		ns
6	TDO Delay after TCK Low(1)		50	ns
7	TDO Enable (Low Z) after TCK Falling Edge(1)	2		ns
8	TDO Disable (High Z) after TCK Low(1)		25	ns
Notes: (1). Also applies to functional outputs for the EXTEST instruction.				

Figure 42. JTAG Interface Timing



Package Thermal Specification

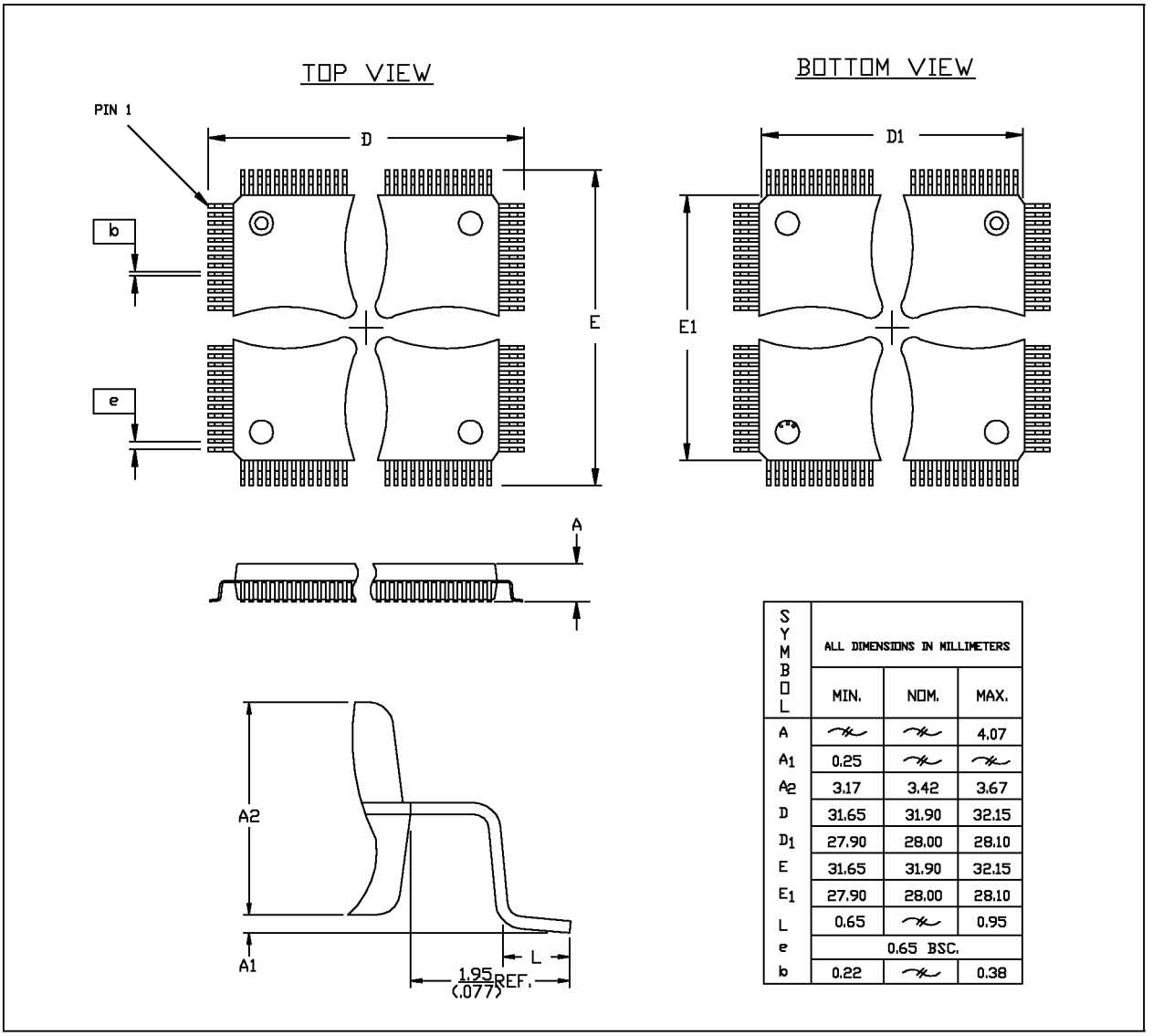
Table 76. MUSYCC 160-Pin PQFP Package Thermal Resistance Characteristics

PQFP Package	Mounting Conditions	Airflow–LFM (LMS)				
		0 (0.000)	50 (0.256)	100 (0.505)	200 (1.01)	400 (2.03)
Thermal Resistance (junction to ambient) °C/W						
Standard	Surface–Board	28	26	25	23	21
Standard	Socket	36	35	33	30	25

Notes: 1. LFM–linear feet per minute.
2. LMS–linear meters per second.
3. Junction to case temperature (°C): $T_{jc} = T_{ac} + (\theta_{ja} \times P_d)$.

Mechanical Specifications

Figure 43. 160-Pin PQFP Package Mechanical Drawing



SYMBOL	ALL DIMENSIONS IN MILLIMETERS		
	MIN.	NOM.	MAX.
A			4.07
A_1	0.25		
A_2	3.17	3.42	3.67
D	31.65	31.90	32.15
D_1	27.90	28.00	28.10
E	31.65	31.90	32.15
E_1	27.90	28.00	28.10
L	0.65		0.95
e	0.65 BSC.		
b	0.22		0.38

Revision History

Table 77. Bt8474/2 Datasheet Revisions

Revision	Date	Change	Description
Rev. A	01/10/97		Initial Release

Terms, Definitions, and Conventions

This document assumes the reader is familiar with the design and development of PCI systems software and hardware, and with the message layout used in the High-Level Data Link Control (HDLC) protocol. Most of the PCI reference material is located in the PCI Local Bus Specification.

Applicable Specifications

The following documents were used as reference material for MUSYCC and this document.

- PCI Local Bus Specification
  Revision 2.1, Production Version, June 1, 1995
- ITU-T Recommendation Q.921 (03/93)
  Digital Subscriber Signaling System No. 1
- ITU-T Recommendation Q.703 (03/93)
  Specification of Signaling System No. 7
- ANSI T1.408-1990
- ITU-T Recommendation G.704
- IEEE Standard 1149.1-1990
- Brooktree Bt8370 Specification

Numeric Notation

The general representation for numbers is shown in Table 78. The suffix may be dropped for clarity when the context is felt to make the intended radix obvious. The suffix convention requires letters within hexadecimal numbers to be capitalized [ABCDEF].

Table 78. Number Representation

Type	Suffix	Example
Binary	b	01b, 1010b
Decimal	d	01d, 999d
Octal	o	01o, 174o
Hexadecimal	h	01h, 08E002FCh

Bit Stream Transmission Convention

Digitized voice transmission represented by octets (8-bit-fields) generally number bits left to right, 0 to 7, respectively. Data is transmitted serially starting at the most significant bit (left most bit numbered bit 0) and proceeding to the right-most bit (least significant bit numbered bit 8) of the sample. The receiver receives the most significant bit first. Table 79 illustrates this sequence.

Table 79. Digitized Voice Transmission Convention

Bit	0 (msb)	1	2	3	4	5	6	7 (lsb)
Data	1	0	1	0	1	1	1	1
Transmission Order	bit stream = 10101111..... MSB is transmitted first							

Digital data transmission uses n-bit words and generally numbers bits right to left, 0 to [n-1], respectively. Data is transmitted serially starting with the least significant bit (right most bit or bit 0) and proceeding to the most significant bit (left most bit or bit [n–1]). The receiver receives the least significant bit first. Table 80 illustrates this sequence.

Table 80. Digital Data Transmission Convention

Bit	7 (msb)	6	5	4	3	2	1	0 (lsb)
Data	1	0	1	0	1	1	1	1
Transmission Order	bit stream = 11110101..... LSB is transmitted first							

MUSYCC employs the digital data transmission convention. MUSYCC extensively uses 32-bit wide ‘dwords’ or double-words data transfers. The data is transmitted and received as shown in Table 81.

Table 81. MUSYCC Byte Transmission Convention

Byte	3 msb bits 7 <- 0	2	1	0 lsb bits 7 <- 0
Transmission and Reception Order	byte stream = byte 0, byte 1, byte 2, byte 3.....			

Bit Stream Storage Convention

MUSYCC stores and retrieves bit stream data to and from memory using little-endian-style byte ordering. The little-endianness causes the least significant byte to be stored in and retrieved from the lowest memory address.

In Table 82 and Table 83, little-and big-endian byte ordering within a dword is demonstrated using the 32-bit dword 7654321h.

Table 82. Little-Endian Storage Convention (Intel-style)

address	x	x+1	x+2	x+3
data	10h	32h	54h	76h

Table 83. Big-Endian Storage Convention (Motorola-style)

address	x	x+1	x+2	x+3
data	76h	54h	32h	10h

Acronyms

BLP	Bit Level Processor
BPS (bps)	Bits Per Second
COFA	Change Of Frame Alignment
CMOS	Complementary Metal Oxide Semiconductor
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
DMI	Digital Multiplexed Interface
DXI	Data Exchange Interface
EOB	End Of Buffer
EBUS	Expansion Bus
EOM	End Of Message
EOP	End Of Pad Fill
FCS	Frame Check Sequence
FIFO	First In First Out
FREC	Frame Recovery
HDLC	High-Level Data Link Control
IC	Idle Code, Integrated Circuit
INTC	Interrupt Controller
ISDN	Integrated Service Digital Network
ISO	International Standard for Organization
JTAG	Joint Test Action Group
KBPS (Kbps)	Kilobits Per Second
LSB (lsb)	Least Significant Bit or Byte
MBPS (Mbps)	Megabits Per Second
MSB (msb)	Most Significant Bit or Byte
MUSYCC	Multichannel Synchronous Communication Controller
OOF	Out Of Frame
OSI	Open System Interconnection
PCI	Peripheral Component Interface
PCM	Pulse Code Modulated

PQFP Plastic Quad Flat Pack

ROOF Receiver Out of Framet

RX (Rx) Receive/Receiver

SERI Serial Port Interfaces

SS7 Signaling System 7

SUERM Signal Unit Error Rate Monitor

SUET Signal Unit Error Threshold

TS Timeslot

TX (Tx) Transmit/Transmitter

Definitions

bit-field Any group of associated information bits that must always be viewed together to provide the desired information. For example, in a 3-bit-field, the 3 bits can represent 8 related values and hence must always be viewed together.

byte A field made up of 8 binary bits.

channel A logical bit stream through MUSYCC. A channel has an associated transmit and receive direction. The transmit direction is for the bit stream flowing from shared memory towards the serial port. The receive direction is for the bit stream flowing from the serial port to the shared memory. A channel within MUSYCC is bi-directional. The rate of data flow is configurable and is specified in bits per second.

channelized A serial port configuration whereby a higher speed bit stream is partitioned into lower speed bit streams or timeslots. A frame synchronization signal is required and allows mapping of individual bits within the timeslots into logical channels. This term is synonymous with PCM Highway.

channel group MUSYCC is designed around four independent and full-duplexed sets of channels. Each channel group supports up to 32 logical channels.

data buffer A block of shared memory where data messages are stored. As messages are received from the serial port, MUSYCC writes the message to shared memory data buffers. As messages are sent out on the serial port, MUSYCC takes messages from shared memory data buffers.

descriptor A data structure used to specify attributes of a separate block of data.

dword A field made up of 32 binary bits or 2 words concatenated or 4 bytes concatenated.

fifo A region of memory space designed to facilitate the movement of bits of information in a first-in-first-out manner.

flag As defined by HDLC protocol, an octet with the value 7Eh (01111110b).

frame In the context of an HDLC bit stream, this term is synonymous with message and packet. In terms of a serial interface, a frame is a grouping of synchronous bits relative to a serial line clock and delimited by a synchronization signal. The frame structure is defined by the physical interface providing the framed data.

hyperchannel Concatenation of timeslots into a single logical channel. The available bandwidth for such a logical channel is the sum of bandwidth of each timeslot.

idle code An octet pattern used to fill the time between the closing flag of one message and the opening flag of the subsequent message. The following idles codes are supported: 7Eh, FFh, and 00h.

message In the context of an HDLC protocol, a data message consists of a header field, an address field, a control field, a payload field and an FCS field delimited by an opening and a closing flag—7Eh (01111110h). This term is synonymous with frame and packet.

octet A field made up of 8 binary bits. Synonymous with byte.

pointer A 32-bit-field containing the address of another bit-field, descriptor, dword, word, or byte.

subchannel When a 64 Kbps timeslot (or channel) consists of lower rate bit streams (in multiples of 8 Kbps), then each bit stream is said to be a subchannel of the original channel.

timeslot An 8-bit portion of a channelized T1 or E1 frame which repeats every 125 µs and represents a 64 Kbps signal. In channelized T1 and E1 frames, 24 and 32 timeslots operate at 64 Kbps.

word A field made up of 16 binary bits or 2 bytes concatenated.