Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	CPU32+
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	25MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	240-BFQFP
Supplier Device Package	240-FQFP (32x32)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68360ai25l

Once an external device receives the bus and asserts $\overline{\text{BGACK}}$, it should negate $\overline{\text{BR}}$. If $\overline{\text{BR}}$ remains asserted after $\overline{\text{BGACK}}$ is asserted, the QUICC assumes that another device is requesting the bus and prepares to issue another $\overline{\text{BG}}$.

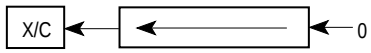
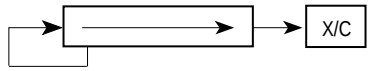
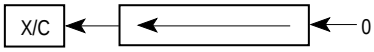
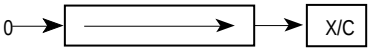
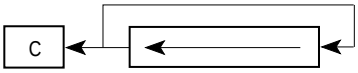
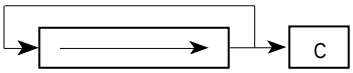
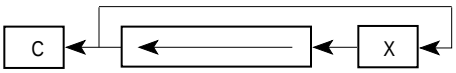
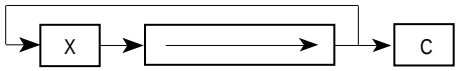
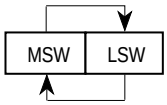
4.6.4 Bus Arbitration Control

The bus arbitration control unit in the QUICC is implemented with a finite state machine. As discussed previously, all asynchronous inputs to the QUICC are internally synchronized in a maximum of two cycles of the clock. As shown in Figure 4-37, input signals labeled R and A are internally synchronized versions of $\overline{\text{BR}}$ and $\overline{\text{BGACK}}$, respectively. The $\overline{\text{BG}}$ output is labeled G, and the internal high-impedance control signal is labeled T. If T is true, the address, data, and control buses are placed in the high-impedance state after the next rising edge following the negation of $\overline{\text{AS}}$ and $\overline{\text{RMC}}$. All signals are shown in positive logic (active high), regardless of their true active voltage level. The state machine shown in Figure 4-37 does not have a state 1 or state 4.

State changes occur on the next rising edge of the clock after the internal signal is valid. The $\overline{\text{BG}}$ signal transitions on the rising edge of the clock after a state is reached during which G changes. The bus control signals (controlled by T) are driven by the QUICC immediately following a state change, when bus mastership is returned to the QUICC. State 0, in which G and T are both negated, is the state of the bus arbiter while the QUICC is bus master. R and A keep the arbiter in state 0 as long as they are both negated.

The QUICC does not allow arbitration of the external bus during the $\overline{\text{RMC}}$ sequence. For the duration of this sequence, the QUICC ignores the $\overline{\text{BR}}$ input. If mastership of the bus is required during an $\overline{\text{RMC}}$ operation, $\overline{\text{BERR}}$ must be used to abort the $\overline{\text{RMC}}$ sequence.

Table 5-7. Shift and Rotate Operations

Instruction	Operand Syntax	Operand Size	Operation
ASL	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
ASR	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
LSL	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
LSR	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
ROL	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
ROR	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
ROXL	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
ROXR	Dn, Dn #⟨data⟩, Dn ⟨ea⟩	8, 16, 32 8, 16, 32 16	
SWAP	Dn	16	

5.3.3.6 BIT MANIPULATION INSTRUCTIONS. Bit manipulation operations are accomplished using the following instructions: bit test (BTST), bit test and set (BSET), bit test and clear (BCLR), and bit test and change (BCHG). All bit manipulation operations can be performed on either registers or memory. The bit number is specified as immediate data or in a data register. Register operands are 32 bits, and memory operands are 8 bits. Table 5-8 is a summary of bit manipulation instructions.

frames of other M68000 family members. The only internal machine state required in the CPU32+ stack frame is the bus controller state at the time of the error and a single register.

Bus operation in progress at the time of a fault is conveyed by the SSW.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP	MV	SZC1	TR	B1	B0	RR	RM	IN	RW	SZC0	SIZ		FUNC		

The bus error stack frame is 12 words in length. There are three variations of the frame, each distinguished by different values in the SSW TP and MV fields.

An internal transfer count register appears at location SP + $14 in all bus error stack frames. The register contains an 8-bit microcode revision number and, for type III faults, an 8-bit transfer count. Register format is shown in Figure 5-14.

15	8	7	0
MICROCODE REVISION NUMBER		TRANSFER COUNT	

Figure 5-14. Internal Transfer Count Register

The microcode revision number is checked before a bus error stack frame is restored via RTE. In a multiprocessor system, this check ensures that a processor using stacked information is at the same revision level as the processor that created it.

The transfer count is ignored unless the MV bit in the stacked SSW is set. If the MV bit is set, the least significant byte of the internal register is reloaded into the MOVEM transfer counter during RTE execution.

For faults occurring during normal instruction execution (both prefetches and non-MOVEM operand accesses), SSW TP,MV = 00. Stack frame format is shown in Figure 5-15.

Faults that occur during the operand portion of the MOVEM instruction are identified by SSW TP,MV = 01. Stack frame format is shown in Figure 5-16.

When a bus error occurs during exception processing, SSW TP,MV = 10. The frame shown in Figure 5-17 is written below the faulting frame. Stacking begins at the address pointed to by SP – 6 (SP value is the value before initial stacking on the faulted frame).

The frame can have either four or six words, depending on the type of error. Four-word stack frames do not include the faulted instruction PC. (The internal transfer count register is located at SP + $10 and the SSW is located at SP + $12.)

The fault address of a dynamically sized bus cycle is the address of the upper byte, regardless of the byte that caused the error.

5.7.2.3 MOVE INSTRUCTION. The MOVE instruction table indicates the number of clock periods needed for the processor to calculate the destination EA and to perform a MOVE or MOVEA instruction. For entries with CEA or FEA, refer to the appropriate table to calculate that portion of the instruction time.

Destination EAs are divided by their formats (see CPU32 Reference Manual). The total number of clock cycles is outside the parentheses. The numbers inside parentheses (r/p/w) are included in the total clock cycle number. All timing data assumes two-clock reads and writes.

When using this table, begin at the top and move downward. Use the first entry that matches both source and destination addressing modes.

Instruction	Head	Tail	Cycles
MOVE Rn, Rn	0	0	2(0/1/0)
MOVE ⟨FEA⟩, Rn	0	0	2(0/1/0)
MOVE Rn, (Am)	0	2	4(0/1/X)
MOVE Rn, (Am)+	1	1	5(0/1/X)
MOVE Rn, –(Am)	2	2	6(0/1/X
MOVE Rn, ⟨CEA⟩	1	3	5(0/1/X)
MOVE ⟨FEA⟩, (An)	2	2	6(0/1/X
MOVE ⟨FEA⟩, (An)+	2	2	6(0/1/X)
MOVE ⟨FEA⟩, –(An)	2	2	6(0/1/X)
MOVE #, ⟨CEA⟩	2	2	6(0/1/X∗
MOVE ⟨CEA⟩, ⟨FEA⟩	2	2	6(0/1/X)

X = There is one bus cycle for byte and word operands and two bus cycles for long-word operands. For long-word bus cycles, add two clocks to the tail and to the number of cycles. Timing is calculated with the CPU32+ in 16-bit mode.

∗ = An # fetch EA time must be added for this instruction: ⟨FEA⟩ +⟨CEA⟩ + ⟨OPER⟩

NOTE: For instructions not explicitly listed, use the MOVE ⟨CEA⟩, ⟨FEA⟩ entry. The source EA is calculated by the calculate EA table, and the destination EA is calculated by the fetch EA table, even though the bus cycle is for the source EA.

5.7.2.4 SPECIAL-PURPOSE MOVE INSTRUCTION. The special-purpose MOVE instruction table indicates the number of clock periods needed for the processor to fetch, calculate, and perform the special-purpose MOVE operation on control registers or a specified EA. Footnotes indicate when to account for the appropriate EA times. The total number of clock cycles is outside the parentheses. The numbers inside parentheses (r/p/w) are included in the total clock cycle number. All timing data assumes two-clock reads and writes.

X = There is one bus cycle for byte and word operands and two bus cycles for long operands. For long bus cycles, add two clocks to the tail and to the number of cycles. Timing is calculated with the CPU32+ in 16-bit mode.
< = Maximum time (certain data or mode combinations may execute faster).
su = The execution time is identical for signed or unsigned operands.

* = These instructions have an additional save operation that other instructions do not have. To calculate total instruction time, calculate save, ⟨ea⟩, and operation execution times, then combine in the order listed, using equations in 5.7.1.6 Instruction Execution Time Calculation. A save operation is not run for long-word divide and multiply instructions when ⟨FEA⟩ = Dn.

5.7.2.6 IMMEDIATE ARITHMETIC/LOGIC INSTRUCTIONS. The immediate arithmetic/logic instruction table indicates the number of clock periods needed for the processor to fetch the source immediate data value and to perform the specified arithmetic/logic instruction using the specified addressing mode. Footnotes indicate when to account for the appropriate fetch effective or fetch immediate EA times. The total number of clock cycles is outside the parentheses. The numbers inside parentheses (r/p/w) are included in the total clock cycle number. All timing data assumes two-clock reads and writes.

Instruction	Head	Tail	Cycles
MOVEQ#, Dn	0	0	2(0/1/0)
ADDQ #, Rn	0	0	2(0/1/0)
ADDQ #, ⟨FEA⟩	0	3	5(0/1/x)
SUBQ #, Rn	0	0	2(0/1/0)
SUBQ #, ⟨FEA⟩	0	3	5(0/1/x)
ADDI #, Rn	0	0	2(0/1/0)∗
ADDI #, ⟨FEA⟩	0	3	5(0/1/x)∗
ANDI #, Rn	0	0	2(0/1/0)∗
ANDI #, ⟨FEA⟩	0	3	5(0/1/x)∗
EORI #, Rn	0	0	2(0/1/0)∗
EORI #, ⟨FEA⟩	0	3	5(0/1/x)∗
ORI #, Rn	0	0	2(0/1/0)∗
ORI #, ⟨FEA⟩	0	3	5(0/1/x)∗
SUBI #, Rn	0	0	2(0/1/0)∗
SUBI #, ⟨FEA⟩	0	3	5(0/1/x)∗
CMPI #, Rn	0	0	2(0/1/0)∗
CMPI #, ⟨FEA⟩	0	3	5(0/1/x)∗

X = There is one bus cycle for byte and word operands and two bus cycles for long-word operands. For long-word bus cycles, add two clocks to the tail and to the number of cycles. Timing is calculated with the CPU32+ in 16-bit mode.
∗ = An # fetch EA time must be added for this instruction: ⟨FEA⟩ + ⟨FEA⟩ + ⟨OPER⟩

5.7.2.7 BINARY-CODED DECIMAL AND EXTENDED INSTRUCTIONS. The BCD and extended instruction table indicates the number of clock periods needed for the processor to perform the specified operation using the specified addressing mode. No additional tables are needed to calculate total effective execution time for these instructions. The total number of clock cycles is outside the parentheses. The numbers inside parentheses (r/p/w) are included in the total clock cycle number. All timing data assumes two-clock reads and writes.

5.7.2.13 EXCEPTION-RELATED INSTRUCTIONS AND OPERATIONS. The exception-related instructions and operations table indicates the number of clock periods needed for the processor to perform the specified exception-related actions. No additional tables are needed to calculate total effective execution time for these instructions. The total number of clock cycles is outside the parentheses. The numbers inside parentheses (r/p/w) are included in the total clock cycle number. All timing data assumes two-clock reads and writes.

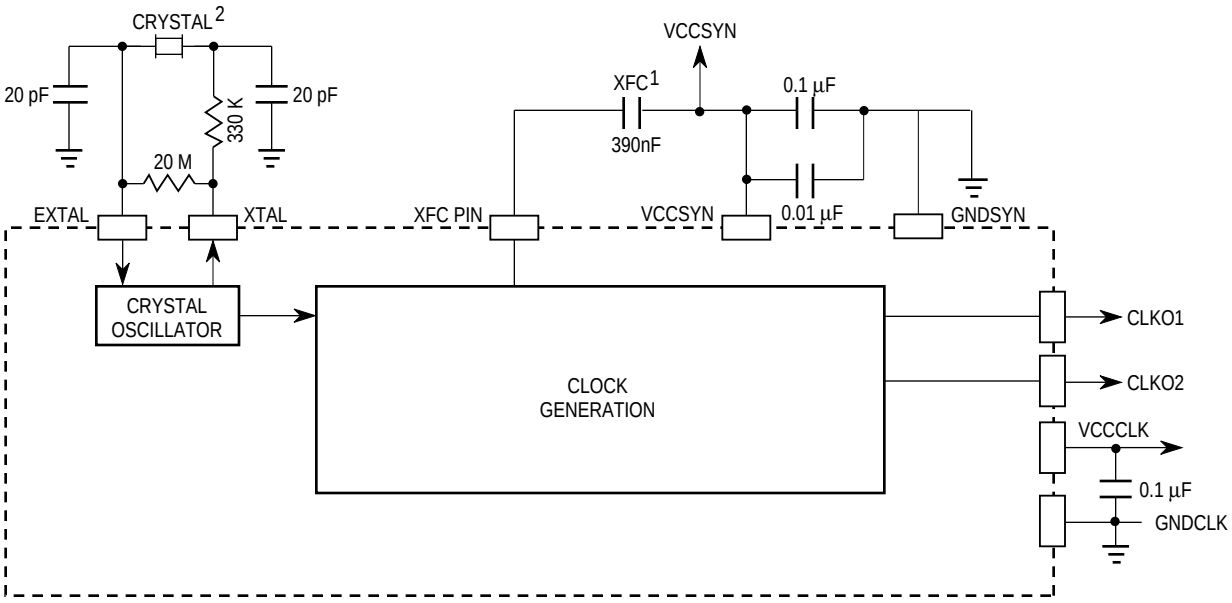
Instruction	Head	Tail	Cycles
BKPT (Acknowledged)	0	0	14(1/0/0)
BKPT (Bus Error)	0	–2	35(3/2/4)
Breakpoint (Acknowledged)	0	0	10(1/0/0)
Breakpoint (Bus Error)	0	–2	42(3/2/6)
Interrupt	0	–2	30(3/2/4)∗
RESET	0	0	518(0/1/0)
STOP	2	0	12(0/1/0)
LPSTOP	3	–2	25(0/3/1)
Divide-by-Zero	0	–2	36(2/2/6)
Trace	0	–2	36(2/2/6)
TRAP #	4	–2	29(2/2/4)
ILLEGAL	0	–2	25(2/2/4)
A-line	0	–2	25(2/2/4)
F-line (First word illegal)	0	–2	25(2/2/4)
F-line (Second word illegal) ea = Rn	1	–2	31(2/3/4)
F-line (Second word illegal) ea ≠ Rn (Save)	1	1	3(0/1/0)
F-line (Second word illegal) ea ≠ Rn (Op)	4	–2	29(2/2/4)
Privileged	0	–2	25(2/2/4)
TRAPcc (trap)	2	–2	38(2/2/6)
TRAPcc (no trap)	2	0	4(0/1/0)
TRAPcc.W (trap)	2	–2	38(2/2/6)
TRAPcc.W (no trap)	0	0	4(0/2/0)
TRAPcc.L (trap)	0	–2	38(2/2/6)
TRAPcc.L (no trap)	0	0	6(0/3/0)
TRAPV (trap)	2	–2	38(2/2/6)
TRAPV (no trap)	2	0	4(0/1/0)

∗ = Minimum interrupt acknowledge cycle time is assumed to be three clocks.
Timing is calculated with the CPU32+ in 16-bit mode.

NOTE: The F-line (second word illegal) operation involves a save step which other operations do not have. To calculate the total operation time, calculate the save, the calculate EA, and the operation execution times, and combine in the order listed, using the equations given in 5.7.1.6 Instruction Execution Time Calculation.

divided prior to being used by any QUICC on-chip module). Furthermore, the divide-by-128 function allows the value of the final system frequency to be chosen with much greater precision, since it is a multiple of ~32 kHz rather than a multiple of ~4 MHz.

The choice of whether to use the divide-by-128 function is made with the MODCK1–MODCK0 pins. This resulting frequency is called CLKIN.



NOTE:
1. Must be low-leakage capacitor. See Section 10 Electrical Characteristics for recommended values.
2. Values are for 32 kHz crystal and may vary due to capacitance on PCB.

Figure 6-6. External Components

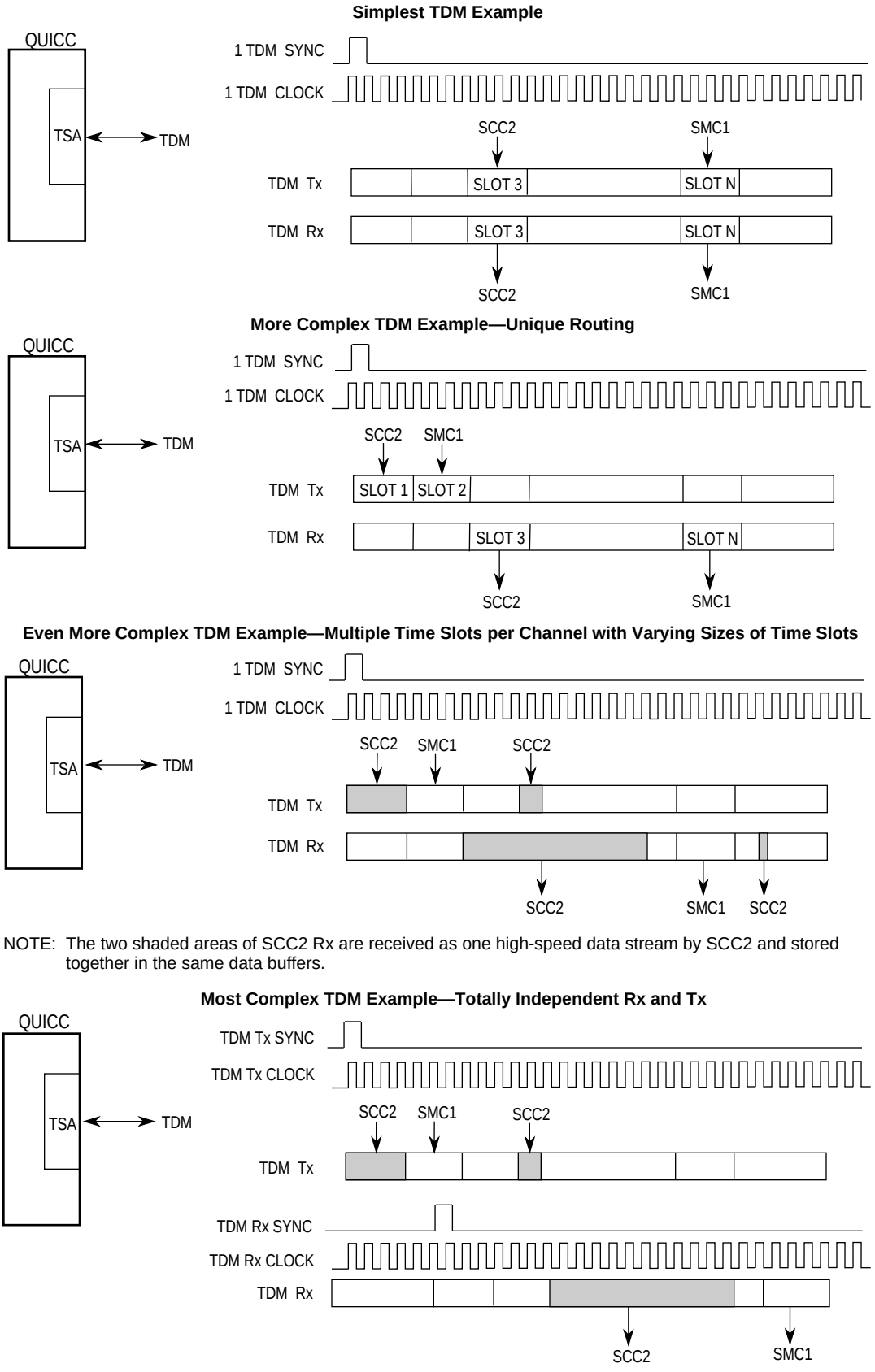
6.5.3 Phase-Locked Loop (PLL)

The PLL takes the CLKIN frequency and outputs a high-frequency source used to derive the general system frequency of the QUICC. The PLL is comprised of a phase detector, loop filter, voltage-controlled oscillator (VCO), and multiplication block. The VCO output can be as high as 50 MHz for a 25-MHz QUICC.

The PLL's main functions are frequency multiplication and skew elimination.

6.5.3.1 FREQUENCY MULTIPLICATION. The PLL can multiply the CLKIN input frequency by any integer between 1 and 4096. The output of the VCO is twice the QUICC system frequency after reset.

If a low frequency crystal is chosen (e.g., ~32 kHz), the multiplier defaults to 401, giving a 2× VCO output of ~26 MHz and an initial general system clock of ~13 MHz. The multiplication factor may then be changed to the desired value by writing the MF11–MF0 bits in the PLLCR. When the PLL multiplier is modified in software, the PLL will lose lock, and the clocking to the QUICC will stop until lock is regained (worst case is 2500 clocks; typical case is 500 clocks). See 6.5.4 Low-Power Divider for methods of reducing clock rates without losing lock.



NOTE: The two shaded areas of SCC2 Rx are received as one high-speed data stream by SCC2 and stored together in the same data buffers.

Figure 7-20. Various Configurations of a Single TDM Channel

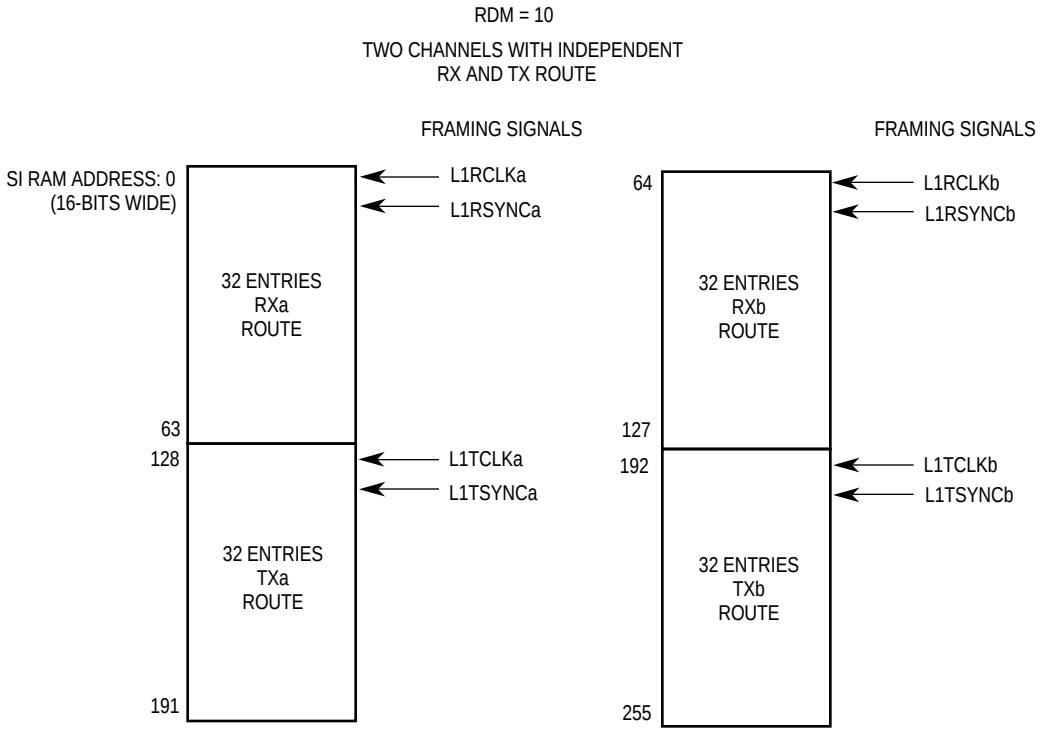
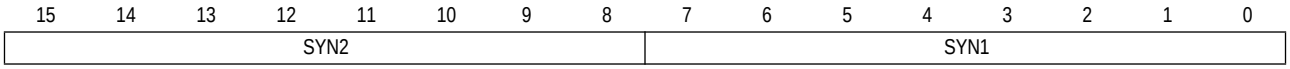


Figure 7-25. SI RAM: Two TDMs with Static Frames

7.8.4.4 TWO MULTIPLEXED CHANNELS WITH DYNAMIC FRAMES. With this configuration (see Figure 7-26), there are two multiplexed channels. Each channel has 16 entries for transmit data and strobe routing and 16 entries for receive data and strobe routing. In each RAM, one of the partitions is the current-route RAM, and the other is a shadow RAM used to allow the user to change the serial routing. After programming the shadow RAM, the user sets the CSRx bit of the associated channel in the SI CR. When the next frame sync arrives, the SI will automatically exchange the current-route RAM for the shadow RAM. Refer to 7.8.4.7 SI RAM Dynamic Changes for more details on how to dynamically change the channel's route. This configuration should be chosen when two TDMs are required and the routing on each TDM may need to be changed dynamically.

7.10.4 SCC Data Synchronization Register (DSR)

Each of the four SCC has a 16-bit, memory-mapped, read-write DSR. The DSR specifies the pattern used in the frame synchronization procedure in the synchronous protocols. In the UART protocol, it is used to configure fractional stop bit transmission. In the BISYNC and totally transparent protocol, it should be programmed with the desired SYNC pattern. In the Ethernet protocol, it should be programmed with $D555. At reset, it defaults to $7E7E (two HDLC flags), so it does not need to be written for HDLC mode. When DSR is used to send out SYNCs (such as in BISYNC or transparent mode), the contents of the DSR are always transmitted LSB first.



7.10.5 SCC Transmit on Demand Register (TODR)

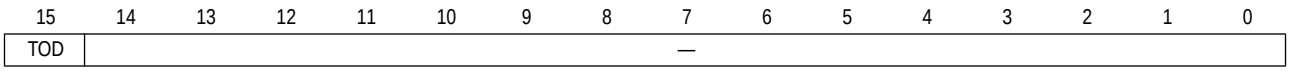
If no frame is currently being transmitted by an SCC, the RISC controller periodically polls the R-bit of the next Tx BD to see if the user has requested a new frame/buffer to be transmitted. This polling algorithm depends on the SCC configuration, but occurs every 8 to 32 serial transmit clocks. The user, however, has an option to request that the RISC begin the processing of the new frame/buffer immediately, without waiting until the normal polling time. To obtain immediate processing, the TOD bit in the transmit-on-demand register is set by the user once the user has set the R-bit in the Tx BD.

This feature, which decreases the transmission latency of the transmit buffer/frame, is particularly useful in LAN-type protocols where maximum interframe GAP times are limited by the protocol specification. Since the transmit-on-demand feature gives a high priority to the specified Tx BD, it can conceivably affect the servicing of the other SCC FIFOs. Therefore, it is recommended that the transmit-on-demand feature only be used when a high-priority Tx BD has been prepared and transmission on this SCC has not occurred for a period of time.

The TOD bit does not need to be set if a new Tx BD is added to the circular queue but other Tx BDs in that queue have not fully completed transmission. In that case, the new Tx BD will be processed immediately following the completion of the older Tx BD s.

The first bit of the frame will typically be clocked out 5-6 bit times after TOD has been written to a 1.

TOD—Transmit on Demand



0 = Normal operation
1 = The RISC will give a high priority to the current Tx BD and will not wait for the normal polling time to check that the Tx BD’s R-bit has been set. It will begin transmitting the frame. This bit will be cleared automaticaly after one serial clock.

7.10.17.3 HDLC CHANNEL FRAME RECEPTION PROCESSING. The HDLC receiver is also designed to work with almost no intervention from the CPU32+ core. The HDLC receiver can perform address recognition, CRC checking, and maximum frame length checking. The received frame is available to the user for performing any HDLC-based protocol.

When the CPU32+ core enables one of the receivers, the receiver waits for an opening flag character. When the receiver detects the first byte of the frame, the HDLC controller will compare the frame address against the user-programmable addresses. The user has four 16-bit address registers and an address mask available for address matching. The HDLC controller will compare the received address field to the user-defined values after masking with the address mask. The HDLC controller can also detect broadcast (all ones) address frames, if one address register is written with all ones.

If a match is detected, the HDLC controller will fetch the next BD and, if it is empty, will start to transfer the incoming frame to the BD's associated data buffer. When the data buffer has been filled, the HDLC controller clears the E-bit in the BD and generates an interrupt if the I-bit in the BD is set. If the incoming frame exceeds the length of the data buffer, the HDLC controller will fetch the next BD in the table and, if it is empty, will continue to transfer the rest of the frame to this BD's associated data buffer.

During this process, the HDLC controller will check for a frame that is too long. When the frame ends, the CRC field is checked against the recalculated value and is written to the data buffer. The data length written to the last BD in the HDLC frame is the length of the entire frame. This enables HDLC protocols that "lose" frames to correctly recognize the frame-too-long condition. The HDLC controller then sets the last buffer in frame bit, writes the frame status bits into the BD, and clears the E-bit. The HDLC controller next generates a maskable interrupt, indicating that a frame has been received and is in memory. The HDLC controller then waits for a new frame. Back-to-back frames may be received with only a single shared flag between frames.

The user can configure the HDLC controller not to interrupt the CPU32+ core until a certain number of frames has been received. This is configured in the received frames threshold (RFTHR) location of the parameter RAM. The user can combine this function with a timer to implement a timeout if less than the threshold number of frames is received.

7.10.17.4 HDLC MEMORY MAP. When configured to operate in HDLC mode, the QUICC overlays the structure listed in Table 7-5 with the HDLC-specific parameters described in Table 7-8.

Table 7-8. HDLC-Specific Parameters

Address	Name	Width	Description
SCC Base + 30	RES	Long	Reserved
SCC Base + 34	C_MASK	Long	CRC Constant
SCC Base + 38	C_PRES	Long	CRC Preset
SCC Base + 3C	DISFC	Word	Discard Frame Counter
SCC Base + 3E	CRCEC	Word	CRC Error Counter

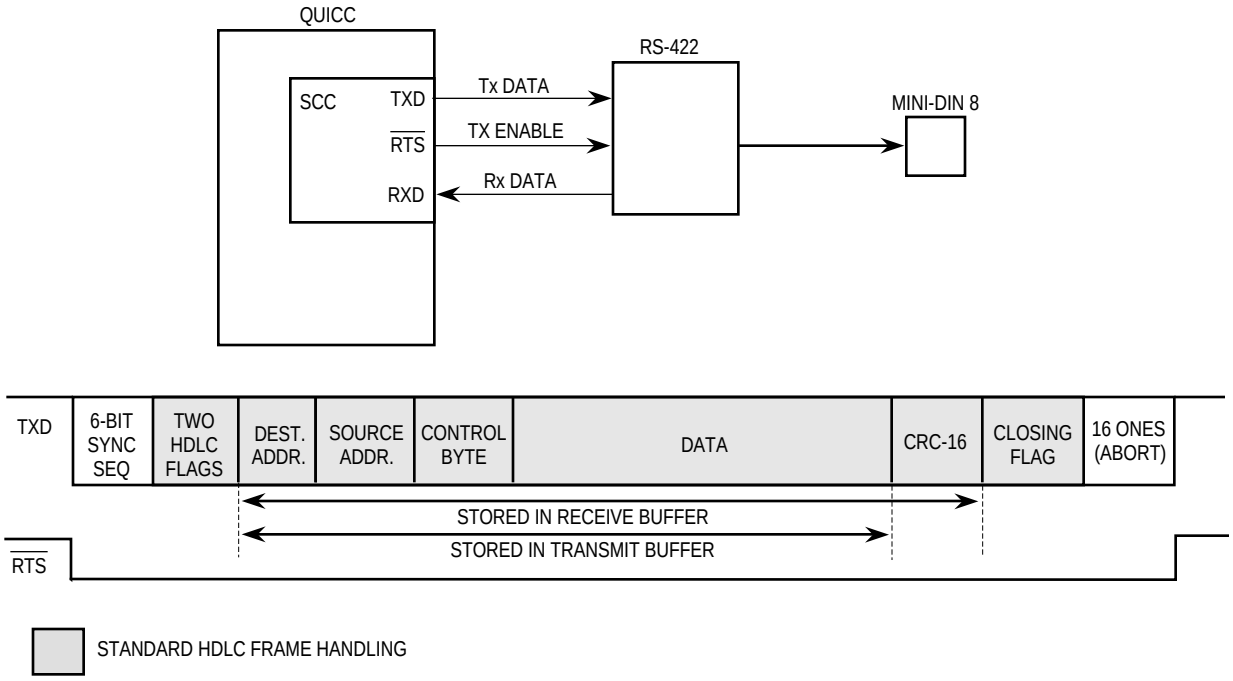


Figure 7-62. Connecting the QUICC to LocalTalk

7.10.19.4.1 GSMR Programming. The GSMR programming sequence is as follows:

1. The MODE bits should be set to AppleTalk.
2. The ENT and ENR bits should be set.
3. The DIG bits should be set for normal operation, with the $\overline{CD}$ and $\overline{CTS}$ pins grounded or with the $\overline{CD}$ and $\overline{CTS}$ pins configured for parallel I/O, which causes $\overline{CD}$ and $\overline{CTS}$ to be internally asserted to the SCC.
4. The RDCR and TDCR bits should usually be set to 16x clock.
5. The TENC and RENC bits should be set for FM0.
6. The Tend bit should be zero.
7. The TPP bits should be 11.
8. The TPL bits should be set to 000 to transmit the next frame with no synchronization sequence and to 001 to transmit the next frame with the LocalTalk synchronization sequence. For example, data frames do not require a preceding synchronization sequence. These bits may be modified on the fly if the AppleTalk protocol is selected.
9. The TINV and RINV bits should be zero.
10. The TSNC bits should be set to 1.5 bit times 10.
11. The EDGE bits should be zero.
12. RTSM should be zero.
13. All other bits should be set to zero or to their default condition.

7	6	5	4	3	2	1	0
–	–	–	TXE	–	–	CHR	TX

TXE—Transmit Error

An error condition was detected. This error status is reported in the buffer descriptor.

CHR—Character Transmitted

Acknowledgment that the last character was strobed into the receiver input latch (STB was asserted by the transmitter) and a new character was written to the data register.

TX—Tx Buffer

A buffer has been transmitted over the Centronics channel. This bit is set only after the last character of the buffer was strobed into the receiver input latch (STB was asserted by the transmitter).

7.13.8.12 CENTRONICS CHANNEL RECEPTION. The Centronics receiver supports the same general data structure that is used by the SCCs for other protocols. Upon receiving a character from the Centronics interface, the receiver will check if the current buffer descriptor (BD) in the Centronics receiver BD table is ready for use. If the BD is ready, the Centronics receiver will compare the character against a user defined control character table. If no match was found, the character will be written to the BD's associated buffer. If a match was found, the character will be either written to the receive buffer (upon which the buffer is closed and a new receive buffer taken) or rejected, depending on the R bit in the Control Character Table. If rejected, the character is written to the Received Control Character Register (RCCR) in internal RAM and a maskable interrupt is generated. A maskable interrupt will be generated at the completion of the BD processing. A single received data frame may span several BDs.

For each transfer, the Centronics controller will generate ACK and BUSY handshake signals on the Centronics interface. The ACK pulse width and the timing of BUSY with respect to the ACK signal are determined by the setting in the PIP Timing Parameter Register (PTPR).

7.13.8.13 CENTRONICS RECEIVER MEMORY MAP. When configured to operate in Centronics receive mode, the QUICC overlays the structure illustrated in Table 7-17 with the SMC2 parameter RAM area.

Table 7-18. Centronics Receiver Parameter RAM

Address	Name	Width	Description
PIP Base+00	RBASE	Word	Rx Buffer Descriptors Base Address
PIP Base+02	Res	Word	Reserved
PIP Base+04	CFCR	Byte	Centronics Function Code
PIP Base+05	Res	Byte	Reserved
PIP Base+06	MRBLR	Word	Maximum Receive Buffer Length
PIP Base+08	RSTATE	Long	Rx Internal State
PIP Base+0C	R_PTR	Long	Rx Internal Data Pointer
PIP Base+10	RBPTR	Word	Rx Buffer Descriptor Pointer

RCCR—Received Control Character Register

Upon a control character match for which the Reject bit is set, the Centronics will write the control character into the RCCR and generate a maskable interrupt. The core must process the interrupt and read the RCCR before a second control character arrives. Failure to do so will result in the Centronics overwriting the first control character.

7.13.8.19 CENTRONICS SILENCE PERIOD. The Centronics controller may be programmed to close the receive data buffer after a programmable silence period. The length of the silence period is determined by the MAX_SL register value. The centronics controller will decrement the MAX_SL value every 1024 system clocks. If it reaches zero before any data received, the receive buffer will be closed automatically. Setting MAX_SL value to zero disables this function.

7.13.8.20 CENTRONICS RECEIVER COMMAND SET.

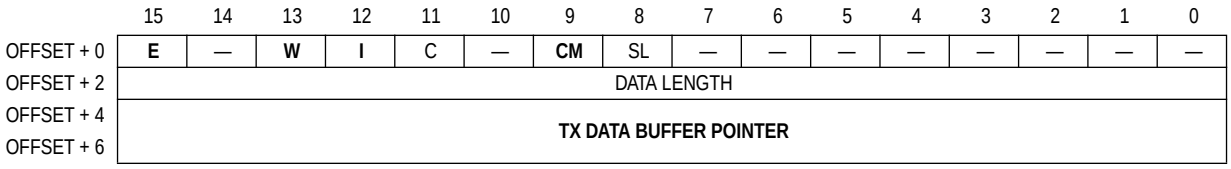
7.13.8.20.1 INIT RX PARAMETERS Command. Initializes all the receive parameters in the Centronics parameter RAM to their reset state. This command should only be issued when the receiver is disabled.

7.13.8.20.2 CLOSE RX BD Command. The CLOSE RX BD command is used to force the Centronics controller to close the current receive BD if it is currently being used and to use the next BD in the list for any subsequent data that is received. If the Centronics controller is not in the process of receiving data, no action is taken by this command.

7.13.8.21 RECEIVER ERRORS.

7.13.8.21.1 Buffer Descriptor Busy. This error occurs if a character was received from the Centronics interface and the current BD that should be processed by the Centronics controller is not empty (E bit in the BD = 0). The channel will resume reception after the s/w prepares the BD.

7.13.8.22 CENTRONICS RECEIVE BUFFER DESCRIPTOR. The CP confirms transmission (or indicates error conditions) via the buffer descriptors to inform the processor that the buffers have been serviced.



E—Empty

0 = The data buffer associated with this BD has been filled with received data, or data reception has been aborted due to an error condition. The core is free to examine or write to any fields of this Rx BD. The CP will not use this BD again while the empty bit remains zero.

1 = The data buffer associated with this BD is empty, or reception is currently in progress. This Rx BD and its associated receive buffer are owned by the CP. Once the E bit is set, the CPU32+ core should not write any fields of this Rx BD.

9.1.1.10 DOUBLE BUS FAULT. The QUICC double bus fault monitor may be used in the design. No additional hardware is required.

9.1.1.11 JTAG AND THREE-STATE. The QUICC provides a JTAG test access port, commonly known as JTAG. This interface uses five pins: TMS, TDI, TDO, TCK, and $\overline{\text{TRST}}$. TMS and TDI are left unconnected because they have internal pullups. The JTAG port is disabled in this application; however, the capability could be easily added.

When the QUICC is in master mode, it provides a pin ($\overline{\text{TRIS}}$) that allows all outputs on the device to be three-stated. This pin is simply pulled high in this application.

9.1.1.12 QUICC SERIAL PORTS. The functions on QUICC parallel I/O ports A, B, and C may be used as desired in this application, and have no bearing on the design as shown. However, any unused parallel I/O pins should be configured as outputs so they are not left floating.

9.1.2 Memory Interfaces

In this application, a number of memory arrays have been developed for EPROM, flash EPROM, SRAM, EEPROM, and DRAM. Each memory interface can be attached to the system bus as desired.

One issue not discussed is the decision of whether external buffers are needed on the system bus. This issue depends on the number of memory arrays used in the design and the layout (i.e., capacitance) of the system bus. This issue is left to the user for his particular design.

Another issue left to the user is the number of wait states used with each memory system. This depends on the memory speed, whether external buffers are used, and the loading on the system bus pins. (The QUICC provides capacitance de-rating figures to calculate the effect of additional or less capacitance on the AC Timing Specifications.)

9.1.2.1 QUICC MEMORY INTERFACE PINS. In this design, a number of QUICC pins are made available to the memory arrays (see Figure 9-1). Eight chip select or $\overline{\text{RAS}}$ pins are available in the system. In this design, $\overline{\text{CS0}}$ is used for any of the EPROM arrays since this is the global (boot) chip select. $\overline{\text{RAS1}}$ is used for the DRAM arrays because of its double-drive capability. $\overline{\text{CS2}}$/$\overline{\text{RAS2}}$ is not used in the design and is available for other purposes, such as a second DRAM bank. $\overline{\text{CS3}}$ is for SRAM; $\overline{\text{CS4}}$ is for EEPROM. $\overline{\text{CS5}}$, $\overline{\text{CS6}}$, and $\overline{\text{CS7}}$ are unused.

Parity may be supported for both SRAM and DRAM arrays using the 4-byte parity lines PRTY3–PRTY0. In this design, it is shown with only a DRAM. The QUICC is configured in software to generate a bus error when a parity error occurs.

The QUICC provides the address multiplexing for the DRAM arrays internally, which is configured in software. Therefore, the address multiplex pin is not needed, and it can be used as its other function—an output enable ($\overline{\text{OE}}$) pin. The DRAM arrays require the four $\overline{\text{CAS3}}$–$\overline{\text{CAS0}}$ pins provided by the QUICC. The QUICC also provides four write enable ($\overline{\text{WE}}$) pins to select the correct byte during write operations.

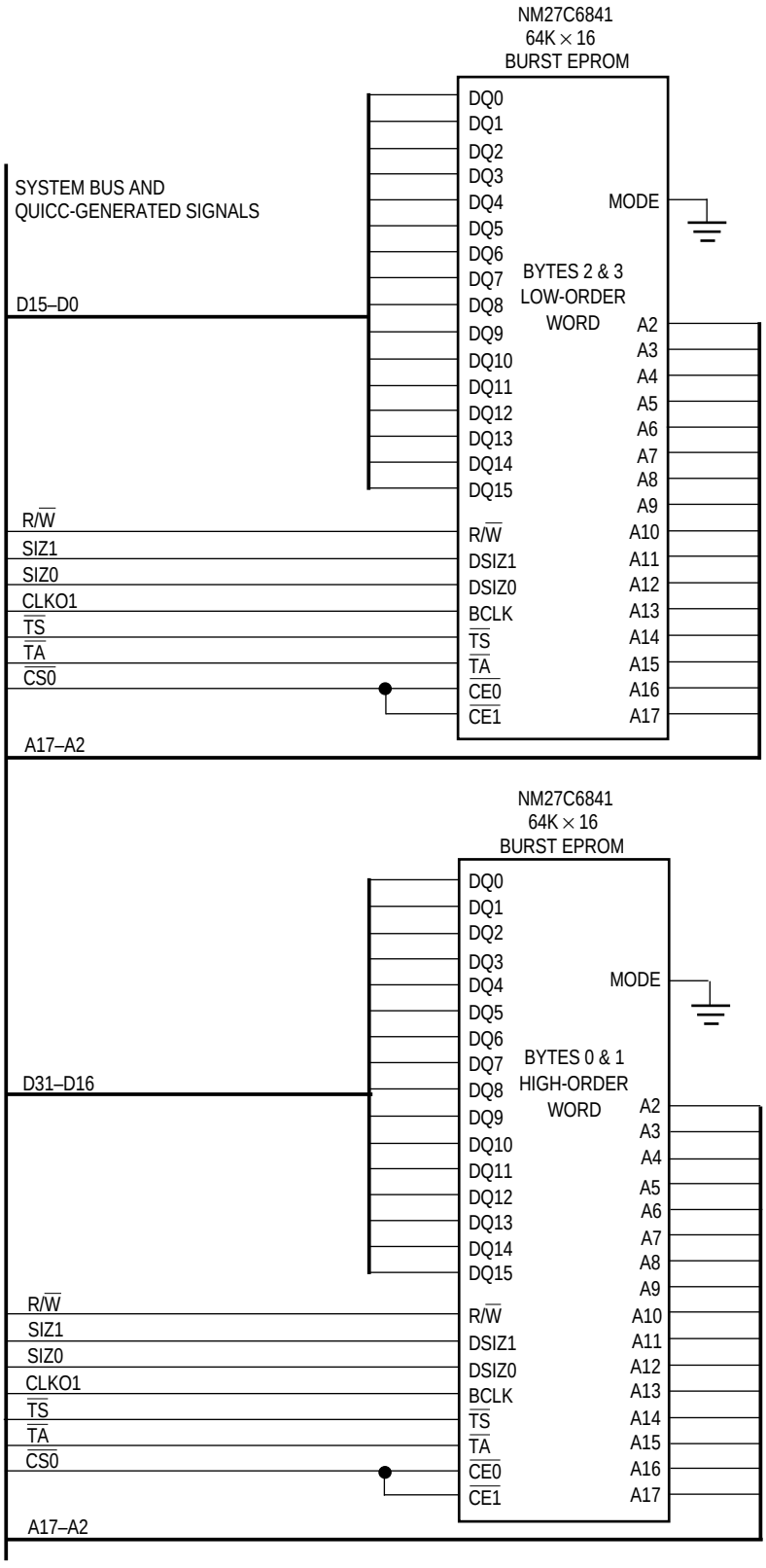
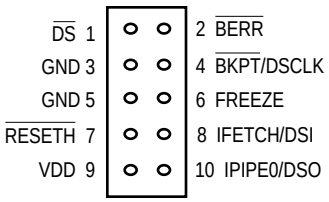


Figure 9-10. 256-Kbyte Burst EPROM Bank—32 Bits Wide

ing needs (such as in-field debugging). In the second and third cases, the target board needs to provide the BDM pins exposed so that an external monitor/debugger can connect to the target board. Figure 9-34 simply shows a standard connector for use with BDM equipment.

The standard connector originally an 8-pin connector, has been expanded to 10 pins as shown in Figure 9-34. This connector is sometimes referred to as the Berg connector. It has the standard 0.1-inch spacing between pins. The original 8 pins on the connector are the lower 8 pins (pins 3–10). If a debug monitor provides an 8-pin BDM connector, it should be plugged onto the original 8 pins. The additional 2 pins were added to allow hardware breakpoints to be implemented using the BDM connector. Since the QUICC contains an on-chip breakpoint address register with masking, this function can be implemented on-chip. To do this, the user can load the breakpoint address register in the SIM60 using the debugger commands (or under normal program control). Once execution of the program resumes, an address breakpoint then causes the QUICC to reenter BDM.



NOTES:
1. On other M68300 family devices RESETH is simply RESET, and IPIPE0 is simply IPIPE.
2. The original 8-pin connector consists of pins 3 through 10.

Figure 9-34. BDM Connector

10.18 IDMA AC ELECTRICAL SPECIFICATIONS

(The electrical specifications in this document are preliminary. See Figure 10-49 and Figure 10-50.)

Num.	Characteristic	3.3 V or 5.0 V 25.0 MHz		5.0V 33.34MHz		Unit
		Min	Max	Min	Max	
1	CLKO1 Low to DACK , DONE Asserted	—	24	—	18	ns
2	CLKO1 Low to DACK , DONE Negated	—	24	—	18	ns
3[1]	DREQ Asserted to AS Asserted (for DMA Bus Cycle)	$3t_{cyc} + t_{AIST} + t_{CLSA}$				
4[1]	Asynchronous Input Setup Time to CLKO1 Low	12	—	9	—	ns
5[1]	Asynchronous Input Hold Time from CLKO1 Low	0	—	0	—	ns
6	AS to DACK Assertion Skew	0	20	0	15	ns
7	DACK to DONE Assertion Skew	–8	8	-6	6	ns
8	AS, DACK , DONE Width Asserted	70	—	52.5	—	ns
8A	AS, DACK , DONE Width Asserted (Fast Termination Cycle)	28	—	20.5	—	ns
10[1]	Asynchronous Input Setup Time to CLKO1 Low	5	—	4	—	ns
11[1]	Asynchronous Input Hold Time from CLKO1 Low	10	—	7.5	—	ns
12[2]	DREQ Input Setup Time to CLKO1 Low	20	—	15	—	ns
13[2]	DREQ Input Hold Time from CLKO1 Low	5	—	3.75	—	ns
14[2]	DONE Input Setup Time to CLKO1 Low	20	—	15	—	ns
15[2]	DONE Input Hold Time from CLKO1 Low	5	—	3.75	—	ns
16[2]	DREQ Asserted to AS Asserted	2	—	2	—	clk

NOTES:
1.These specifications are for asynchronous mode.
2.These specifications are for synchronous mode.

10.28 SPI MASTER ELECTRICAL SPECIFICATIONS

(The electrical specifications in this document are preliminary. See Figure 10-75 and Figure 10-76)

Num.	Characteristic	3.3 V or 5.0 V		5.0 V		Unit
		25.0 MHz		33.34 MHz		
		Min	Max	Min	Max	
160	Master Cycle Time	4	1024	4	1024	tcyc
161	Master Clock (SPICLK) High or Low Time	2	512	2	512	tcyc
162	Master Data Setup Time (Inputs)	50	—	50	—	ns
163	Master Data Hold Time (Inputs)	0	—	0	—	ns
164	Master Data Valid (after SPICLK Edge)	—	20	—	20	ns
165	Master Data Hold Time (Outputs)	0	—	0	—	ns
166	Rise Time: Output		15		15	s
167	Fall Time: Output		15		15	ns

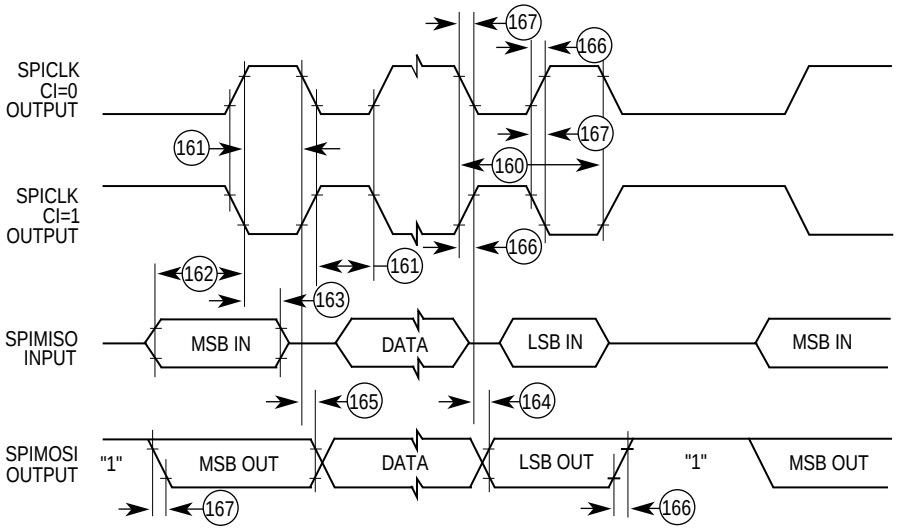


Figure 10-75. SPI Master (CP = 0)

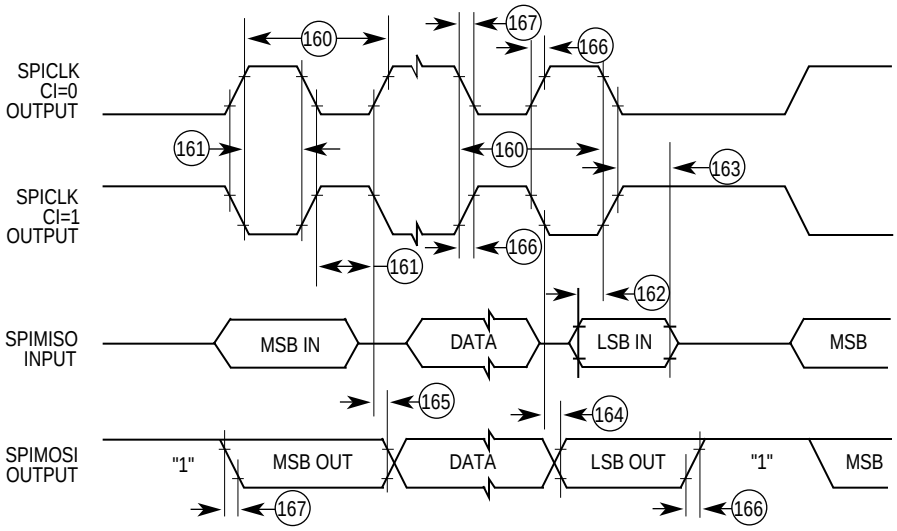


Figure 10-76. SPI Master (CP = 1)