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	357-BBGA
Supplier Device Package	357-PBGA (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68360zq25l

NOTE

Write enable does not have the capability to follow dynamic bus sizing with external assertion of DSACK. Write enable will always follow the port size that is programed in GMR and the OR. For more information see 6.10 Memory Controller.

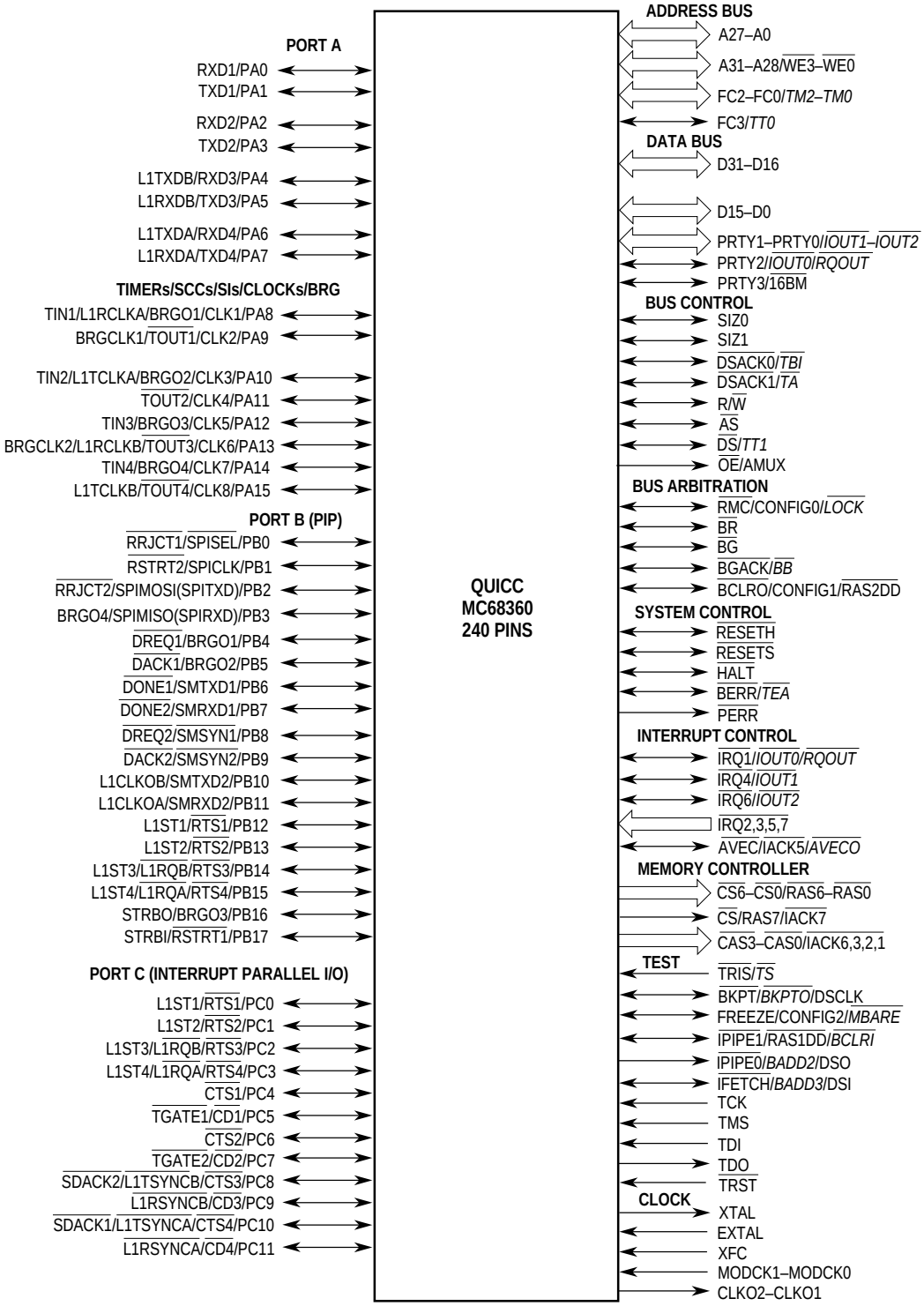

Figure 2-1. QUICC Functional Signal Groups

When the CPU32+ acknowledges hardware breakpoint ($\overline{\text{BKPT}}$ pin assertion or internal breakpoint logic) with background mode disabled, the CPU32+ performs a word read from CPU space, type 0, at an address corresponding to all ones on A4–A2 (BKPT#7), and the T-bit (A1) is set. If this bus cycle is terminated by $\overline{\text{BERR}}$, the QUICC performs hardware breakpoint exception processing. If this bus cycle is terminated by $\overline{\text{DSACKx}}$, the QUICC ignores data on the data bus and continues execution of the next instruction.

NOTE

The $\overline{\text{BKPT}}$ pin is sampled on the same clock phase as data and is latched with data as it enters the CPU32+ pipeline. If $\overline{\text{BKPT}}$ is asserted for only one bus cycle and a pipeline flush occurs before $\overline{\text{BKPT}}$ is detected by the CPU32+, $\overline{\text{BKPT}}$ is ignored. To ensure detection of $\overline{\text{BKPT}}$ by the CPU32+, $\overline{\text{BKPT}}$ can be asserted until a breakpoint acknowledge cycle is recognized.

When the QUICC is configured for a 32-bit bus, the CPU32+ can fetch two instructions simultaneously. Since there is only one $\overline{\text{BKPT}}$ pin, the external user cannot break individually on those instructions, but rather must break on both, causing the BKPT exception to be taken after the first instruction and before the second instruction. The internal breakpoint logic, however, can individually assert a breakpoint for either instruction. (See the BKAR and BKCR discussion in Section 6 System Integration Module (SIM60) for details).

The breakpoint operation flowchart is shown in Figure 4-23. Figure 4-24 and Figure 4-25 show the timing diagrams for the breakpoint acknowledge cycle with instruction opcodes supplied on the cycle and with an exception signaled, respectively.

able operation of the QUICC. If BERR remains asserted into the next bus cycle, it may cause incorrect operation of that cycle. When BERR is issued to terminate a bus cycle, the QUICC may enter exception processing immediately following the bus cycle, or it may defer processing the exception.

The instruction prefetch mechanism requests instruction words from the bus controller before it is ready to execute them. If a bus error occurs on an instruction fetch, the QUICC does not take the exception until it attempts to use that instruction word. Should an intervening instruction cause a branch or should a task switch occur, the bus error exception does not occur. The bus error condition is recognized during a bus cycle in any of the following cases:

1. DSACKx and HALT are negated, and BERR is asserted.
2. HALT and BERR are negated, and DSACKx is asserted. BERR is then asserted within one clock cycle (HALT remains negated).

When the QUICC recognizes a bus error condition, it terminates the current bus cycle in the normal way. Figure 4-29 shows the timing of a bus error for the case in which DSACKx is not asserted. Figure 4-30 shows the timing for a bus error that is asserted after DSACKx. Exceptions are taken in both cases. (Refer to Section 5 CPU32+ for details of bus error exception processing.)

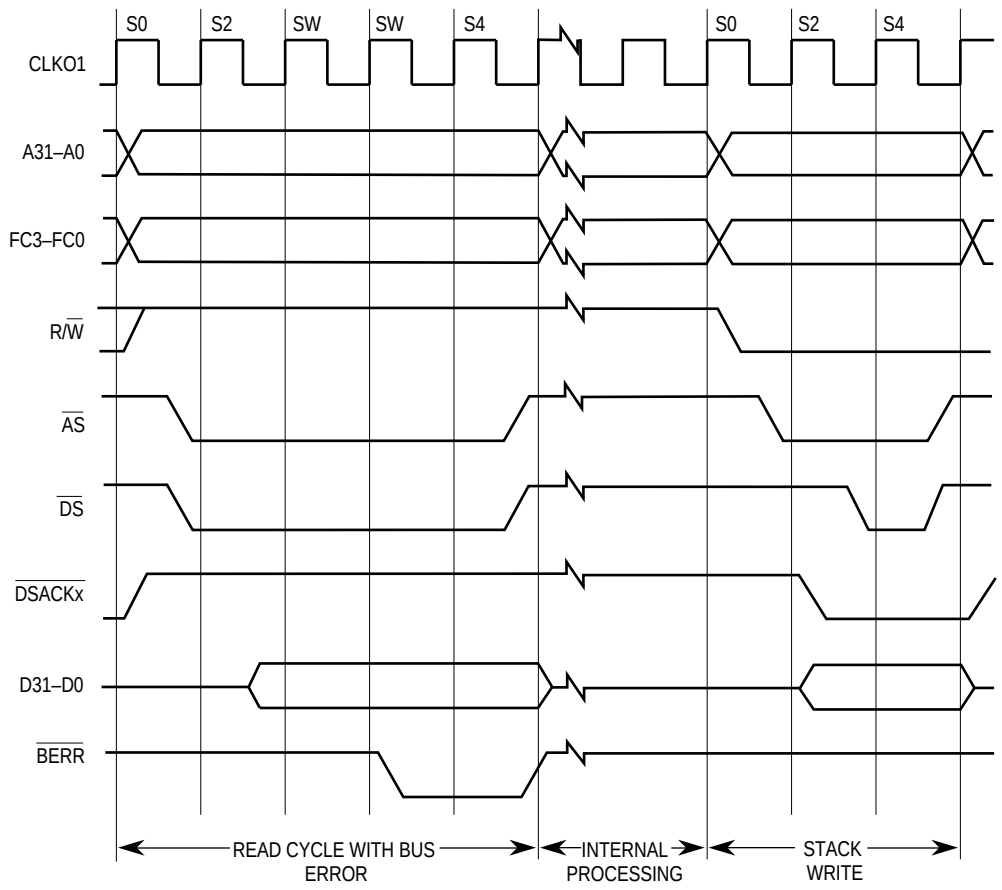

Figure 4-29. Bus Error without DSACKx

Table 5-6. Logic Operations

Instruction	Operand Syntax	Operand Size	Operation
AND	⟨ea⟩, Dn Dn, ⟨ea⟩	8, 16, 32 8, 16, 32	Source Λ Destination ⇒ Destination
ANDI	#⟨data⟩, ⟨ea⟩	8, 16, 32	Immediate Data Λ Destination ⇒ Destination
EOR	Dn, ⟨ea⟩	8, 16, 32	Source ⊕ Destination ⇒ Destination
EORI	#⟨data⟩, ⟨ea⟩	8, 16, 32	Immediate Data ⊕ Destination ⇒ Destination
NOT	⟨ea⟩	8, 16, 32	$\overline{\text{Destination}}$ ⇒ Destination
OR	⟨ea⟩, Dn Dn, ⟨ea⟩	8, 16, 32 8, 16, 32	Source V Destination ⇒ Destination
ORI	#⟨data⟩, ⟨ea⟩	8, 16, 32	Immediate Data V Destination ⇒ Destination
TST	⟨ea⟩	8, 16, 32	Source – 0, to set condition codes

5.3.3.5 SHIFT AND ROTATE INSTRUCTIONS. The arithmetic shift instructions, ASR and ASL, and logical shift instructions, LSR and LSL, provide shift operations in both directions. The ROR, ROL, ROXR, and ROXL instructions perform rotate (circular shift) operations, with and without the extend bit. All shift and rotate operations can be performed on either registers or memory.

Register shift and rotate operations shift all operand sizes. The shift count may be specified in the instruction operation word (to shift from 1 to 8 places) or in a register (modulo 64 shift count).

Memory shift and rotate operations shift word-length operands one bit position only. The SWAP instruction exchanges the 16-bit halves of a register. Performance of shift/rotate instructions is enhanced so that use of the ROR and ROL instructions with a shift count of eight allows fast byte swapping. Table 5-7 is a summary of the shift and rotate operations.

5.3.4.1 TABLE EXAMPLE 1: STANDARD USAGE. The table consists of 257 word entries. As shown in Figure 5-7, the function is linear within the range 32768 ≤ X ≤ 49152. Table entries within this range are as given in Table 5-13 .

Table 5-13. Standard Usage Entries

Entry Number	X-Value	Y-Value
128*	32768	1311
162	41472	1659
163	41728	1669
164	41984	1679
165	42240	1690
192*	49152	1966

*These values are the end points of the range.
All entries between these points fall on the line.

Figure 5-7. Table Example 1

The table instruction is executed with the following bit pattern in Dx:

31 … 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0

Table Entry Offset ⇒ Dx [8:15] = $A3 = 163

Interpolation Fraction ⇒ Dx [0:7] = $80 = 128

Using this information, the table instruction calculates dependent variable Y:

Y = 1669 + (128 (1679 – 1669)) / 256 = 1674

Bit 5—PWW

This read-only bit is used to indicate if the $\overline{WE}$/ADDR and the PRTY lines have been programmed by the user or are still in the three-state condition because the PEPAR register has not been written.

0 = PEPAR has not been written. The $\overline{WE}$/ADDR and the PRTY lines are still being three-stated.
1 = PEPAR was written. The $\overline{WE}$/ADDR and the PRTY lines have been programmed in the PEPAR, so the configuration choices of these pins in the PEPAR are valid.

Bit 4—$\overline{CAS2}$, $\overline{CAS3}$/$\overline{IACK3}$, $\overline{IACK6}$

0 = The $\overline{CAS2}$ and $\overline{CAS3}$ output functions are selected.
1 = The $\overline{IACK3}$ and $\overline{IACK6}$ output functions are selected.

Bit 2—$\overline{CAS0}$, $\overline{CAS1}$/$\overline{IACK1}$, $\overline{IACK2}$

0 = The $\overline{CAS0}$ and $\overline{CAS1}$ output functions are selected.
1 = The $\overline{IACK1}$ and $\overline{IACK2}$ output functions are selected.

Bit 1—$\overline{CS7}$/$\overline{IACK7}$

0 = The $\overline{CS7}$ output function is selected.
1 = The $\overline{IACK7}$ output function is selected.

Bit 0—AVEC ($\overline{AVECO}$)/$\overline{IACK5}$

0 = The $\overline{AVEC}$ input function is selected in normal operation, or $\overline{AVECO}$ is selected in slave mode.
1 = The $\overline{IACK5}$ output function is selected.

6.10 MEMORY CONTROLLER

The memory controller is a sub-block of the SIM60 that is responsible for up to eight general-purpose chip-select lines and the DRAM controller. The DRAM controller itself can control up to eight memory banks.

6.10.1 Memory Controller Key Features

The key features of the memory controller are as follows:

- All Eight Memory Banks Support the Following:
  - —32-Bit Address Decode with 17 Bits of Address Masking
  - —Various Block Sizes—2 Kbytes up to 256 Mbytes
  - —From 0 to 15 Wait States Programmable with $\overline{DSACK}$ Generation
  - —Memory Bank Can Be Used by an External Master
  - —Supports Burst Accesses of the MC68040
  - —Byte Parity Generation/Checking
  - —Write-Protect Capability
  - —Four Byte-Write Enable ($\overline{WE}$) Signals
  - —Output Enable ($\overline{OE}$) Signal
  - —Special Options for Interfacing to Slow Peripherals
  - —Function Code Match with Mask Can Qualify Memory Bank Accesses

2. In buffer chaining or auto buffer modes, the BCR has decremented to zero, the L-bit in the BD has been set, and no errors have occurred during any IDMA transfer bus cycle.
3. An external peripheral has asserted $\overline{\text{DONE}}$x during an access by the IDMA to that peripheral and no errors have occurred during any IDMA transfer bus cycle.

DONE will not be set if the channel terminates due to an error. DONE is cleared by writing a one or by setting RST in the CMR. Writing a zero has no effect on DONE.

7.6.2.8 CHANNEL MASK REGISTER (CMAR). The CMAR is an 8-bit, memory-mapped, read-write register that has the same bit format as the CSR. If a bit in the CMAR is a one, the corresponding interrupt in the CSR will be enabled. If the bit is a zero, the corresponding interrupt in the CSR will be masked. CMAR is cleared at reset.

7.6.2.9 DATA HOLDING REGISTER (DHR). This 7-byte register serves as a buffer register for the data being transferred during dual address IDMA cycles. No address for DHR is given since this register cannot be addressed by the programmer. The DHR allows the data to be packed and unpacked by the IDMA during the transfer. For example, if the source operand size is byte and the destination operand size is word, then two-byte read cycles occur, followed by a one-word write cycle. The two bytes of data are buffered in the DHR until the word write cycle occurs. The DHR allows for packing and unpacking of operands for all possible combinations: bytes to words, bytes to long words, words to long words, words to bytes, long words to bytes, and long words to words.

7.6.3 Interface Signals

The IDMA has three dedicated control signals per channel: DMA request ($\overline{\text{DREQ}}$x), DMA acknowledge ($\overline{\text{DACK}}$x), and end of IDMA transfer ($\overline{\text{DONE}}$x). The peripheral used with these signals may be either a source or a destination of the IDMA transfers.

NOTE

$\overline{\text{DREQ}}$ must be level sensitive if IDMA uses buffer chaining mode.

7.6.3.1 $\overline{\text{DREQ}}$ AND $\overline{\text{DACK}}$. These are the handshake signals between the peripheral requiring service and the QUICC. When the peripheral requires IDMA service, it asserts $\overline{\text{DREQ}}$x, and the QUICC begins the IDMA process. When the IDMA service is in progress, $\overline{\text{DACK}}$x is asserted during accesses to the device. $\overline{\text{DREQ}}$x is ignored when the IDMA is programmed to one of the internal request modes.

7.6.3.2 $\overline{\text{DONE}}$X. This bidirectional open-drain signal is used to indicate the last IDMA transfer. $\overline{\text{DONE}}$x is always an output of the IDMA if the transfer count is exhausted.

$\overline{\text{DONE}}$x may also operate as an input. If $\overline{\text{DONE}}$x is externally asserted during internal request modes, the IDMA transfer is terminated. With external request modes, $\overline{\text{DONE}}$x may be used as an input to the IDMA controller to indicate that the device being serviced requires no more transfers and the transmission is to be terminated.

The user can terminate the transfer by setting the RST bit in the CMR and then issuing the INIT_IDMA command.

The user can terminate the transfer with an "out of buffers" error if the V-bit of one of the BDs is cleared by the user. When the RISC reaches this IDMA BD, it will terminate activity. This technique is useful when the IDMA is required to stop transfers after fully completing a BD transfer.

If the BCR is decremented to zero, the transfer from this BD completes, but the RISC controller reloads the IDMA registers with the values from the next IDMA BD, and the IDMA transfer continues. Thus, the fact that the BCR is decremented to zero does not terminate a transfer in auto buffer mode; it only terminates the current BD transfer.

If DONEx is asserted externally, the transmission from this BD is terminated and the following actions are performed by the RISC controller:

1. Sets the Done Bit in the status register
2. Sets the DA bit in the BD
3. Clears the Valid bit in the BD
4. Resets the start bit in the CMR

Thus the current buffer is closed immediately and all IDMA operation ceases.

7.6.4.8.3 Buffer Chaining Mode Termination. The user can suspend a transfer in auto buffer mode by clearing the STR bit in the CMR. When STR is set once again, the transfer will continue.

The user can terminate the transfer by setting the RST bit in the CMR and then issuing the INIT_IDMA command.

The user can also terminate the transfer by setting the L-bit in the IDMA BD. When processing of this BD has completed, the transmission will terminate with the DONE bit being set in the CSR. This can cause an interrupt if the corresponding bit in the CMAR is set.

If the BCR is decremented to zero, the transfer from this BD completes, but the RISC controller reloads the IDMA registers with the values from the next IDMA BD, and the IDMA transfer continues. Thus, the fact that the BCR is decremented to zero does not terminate a transfer in buffer chaining mode; it only terminates the current BD transfer.

If $\overline{\text{DONEx}}$ is asserted externally, the transmission from this BD is terminated and the following actions are performed by the RISC controller.

1. Sets the Done Bit in the status register
2. Sets the Abort bit in the BD
3. Clears the Ready bit in the BD
4. Resets the start bit in the CMR
5. Sets the Reset bit in the CMR

NOTES

Each strobe is changed with the corresponding RAM clock and will be output only if the corresponding parallel I/O is configured as a dedicated pin.

If a strobe is programmed to be asserted in more than one set of entries (e.g., the SI Rx route for the TDMa entries and the SI Tx route for TDMb entries both select the same strobe), then the assertion of the strobe corresponds to the logical OR of all possible sources. This use of the strobes is not useful for most applications. It is recommended that a given strobe be selected in only one set of SI RAM entries.

CSEL—Channel Select

000 = The bit/byte group is not supported within the QUICC. The transmit data pin is three-stated, and the receive data pin is ignored.
001 = The bit/byte group is routed to SCC1.
010 = The bit/byte group is routed to SCC2.
011 = The bit/byte group is routed to SCC3.
100 = The bit/byte group is routed to SCC4.
101 = The bit/byte group is routed to SMC1.
110 = The bit/byte group is routed to SMC2.
111 = The bit/byte group is not supported within the QUICC. This code is also used in SCIT mode as the D channel grant (refer to 7.8.7.2.2 SCIT Programming.)

CNT—Count

This value indicates the number of bits/bytes (according to the BYT bit) that the routing and strobe select of this entry controls. If CNT = 0000, then 1 bit/byte is chosen; if CNT = 1111, then 16 bits/bytes are selected.

BYT—Byte Resolution

0 = Bit resolution—the CNT value indicates the number of bits in this group.
1 = Byte resolution—the CNT value indicates the number of bytes in this group.

LST—Last Entry in the RAM

Whenever the SI RAM is used, this bit must be set in one of the Tx or Rx entries of each group that is used. Even if all entries of a group are used, this bit must still be set in the last entry.

0 = This is not the last entry in this section of the route RAM.
1 = This is the last entry in this RAM. After this entry, the SI will wait for the sync signal to start the next frame.

NOTE

If a second sync signal is received before the end of a frame (as defined by the last SI RAM entry), an error occurs. The SI will terminate SI RAM processing, and cease transmitting or receiving data until a third sync signal is received.

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

The status and control bits are prepared by the user before transmission and are set by the CP after the buffer has been transmitted.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	**R**	—	**W**	**I**	**L**	**TB**	**CM**	**BR**	**TD**	**TR**	**B**	—	—	—	UN	CT
OFFSET + 2	DATA LENGTH															
OFFSET + 4	**TX DATA BUFFER POINTER**															
OFFSET + 6																

NOTE: Entries in boldface must be initialized by the user.

R—Ready

0 = The data buffer associated with this BD is not ready for transmission. The user is free to manipulate this BD or its associated data buffer. The CP clears this bit after the buffer has been transmitted or after an error condition is encountered.

1 = The data buffer, which has been prepared for transmission by the user, has not been transmitted or is currently being transmitted. No fields of this BD may be written by the user once this bit is set.

W—Wrap (Final BD in Table)

0 = This is not the last BD in the Tx BD table.

1 = This is the last BD in the Tx BD table. After this buffer has been used, the CP will receive incoming data into the first BD in the table (the BD pointed to by TBASE). The number of Tx BDs in this table is programmable, and is determined only by the W-bit and the overall space constraints of the dual-port RAM.

I—Interrupt

0 = No interrupt is generated after this buffer has been serviced.

1 = Either TX or TXE in the BISYNC event register will be set when this buffer has been serviced by the CP, which can cause an interrupt.

L—Last in Message

0 = The last character in the buffer is not the last character in the current block.

1 = The last character in the buffer is the last character in the current block. The transmitter will enter (remain in) normal mode after sending the last character in the buffer and the BCS (if enabled).

TB—Transmit BCS

This bit is valid only when the L-bit is set.

0 = Transmit the SYN1–SYN2 sequence or idle (according to the RTSM bit in the GSMR) after the last character in the buffer.

1 = Transmit the BCS sequence after the last character. The BISYNC controller will also reset the BCS generator after transmitting the BCS.

CM—Continuous Mode

0 = Normal operation.

1 = The R-bit is not cleared by the CP after this BD is closed, allowing the associated data buffer to be retransmitted automatically when the CP next accesses this BD.

—Physical—One 48-Bit Address Recognized or 64-Bin Hash Table for Physical Addresses
—Logical—64-Bin Group Address Hash Table plus Broadcast Address Checking
—Promiscuous—Receives All Addresses, but Discards Frame If Reject Pin Asserted

- External CAM Support on Both Serial and System Bus Interfaces
- Up to Eight Parallel I/O Pins May Be Sampled and Appended to Any Frame
- Heartbeat Indication
- Transmitter Network Management and Diagnostics

—Lost Carrier Sense
—Underrun
—Number of Collisions Exceeded the Maximum Allowed
—Number of Retries per Frame
—Deferred Frame Indication
—Late Collision

- Receiver Network Management and Diagnostics

—CRC Error Indication
—Nonoctet Alignment Error
—Frame Too Short
—Frame Too Long
—Overrun
—Busy (Out of Buffers)

- Error Counters

—Discarded Frames (Out of Buffers or Overrun Occurred)
—CRC Errors
—Alignment Errors

- Internal and External Loopback Mode

7.10.23.3 LEARNING ETHERNET ON THE QUICC. The following paragraphs detail the Ethernet functionality on the QUICC. However, they show the additions made to the standard SCC functionality to implement Ethernet. Therefore, the reader is encouraged to learn the basics of the SCCs and the overall architecture of the CPM before attempting to learn this section in great detail.

A first-time user of the QUICC who plans to use Ethernet on the QUICC should first read the following sections of this user manual.

1. 7.1 RISC Controller**,** 7.2 Command Set, and 7.3 Dual-Port RAM. The RISC controller is used to issue special commands to the Ethernet channel. The dual-port RAM is used to load Ethernet parameters and initialize BDs for use by the Ethernet channel.
2. 7.7 SDMA Channels discusses how SDMA channels are used to transfer data to/from the Ethernet channel and system memory.
3. 7.8.9 NMSI Configuration explains how clocks are routed to the SCCs through the bank of clocks.
4. 7.10.1 SCC Overview contains more detailed information on the SCCs that are appli-

Table 8-1. Boundary Scan Control Bits

Name	Bit Number	Name	Bit Number	Name	Bit Number
g1.cntl	2	g10.cntl	32	pbhl.ctl	155
g2.cntl	5	add.cntl	59	pblh.ctl	159
g3.cntl	8	addh.cntl	83	pchh.ctl	170
g4.cntl	13	g11.cnt	89	pchl.ctl	174
g5.cntl	16	db.ctl	110	pclh.ctl	179
g6.cntl	18	pahh.ctl	131	g12.cntl	188
g7.cntl	23	pahl.ctl	136	g13.cntl	191
g8.cntl	27	palh.ctl	141	g14.cntl	194
g9.cntl	29	pbhh.ctl	150		

The boundary scan bit definitions are listed in Table 8-2.

The first column in the table defines the bit's ordinal position in the boundary scan register. The shift register cell nearest TDO (i.e., first to be shifted out) is defined as bit 0; the last bit to be shifted out is 195.

The second column references one of the four QUICC cell types depicted in Figure 8-3 through , which describe the cell structure for each type.

The third column lists the pin name for all pin-related cells or defines the name of bidirectional control register bits.

The fourth column lists the pin type for convenience, where TS-Output indicates a three-stateable output pin, I/O indicates a bidirectional pin, and OD-I/O denotes an open-drain bidirectional pin.

The last column indicates the associated boundary scan register control bit for bidirectional, three-state, and open-drain output pins.

Bidirectional pins include a single scan cell for data (IO.Cell) as depicted in Figure 8-6. These bits are controlled by the cell shown in Figure 8-5. The value of the control bit determines whether the bidirectional pin is an input or an output. One or more bidirectional data cells can be serially connected to a control cell as shown in Figure 8-7. Note that, when sampling the bidirectional data cells, the cell data can be interpreted only after examining the IO control cell to determine pin direction, and also note that the control cell captures the value of the following cell.

The RBASE and TBASE values are new to the QUICC. They point to the start of the buffer descriptor tables, must be long-word aligned, and must point to the dual-port RAM area.

The RFCR and TFCR registers have one additional purpose and a different bit placement on the QUICC. If RFCR and TFCR are cleared on the MC68302 or not used with the MC68302, then the equivalent function on the QUICC can be implemented by writing $18 to the QUICC RFCR and TFCR. Note that on the QUICC there is an additional function code pin to signify DMA operation, and the suggestion of $18 uses this capability.

The MRBLR value has the same function on both devices. Additionally, if the receive FIFO is set to 32-bit-wide mode, the MRBLR value must be aligned to a long word.

The internal state parameter has the same function on both devices.

The MC68302 RBD# and TBD# have the same concept on the QUICC, except the parameters are called RBPTR and TBPTR and are now 16-bit values. They point to the current buffer descriptor; however, they are now offsets from the beginning location of the QUICC dual-port RAM. For example, the current transmit buffer descriptor location may be found by MBAR + TBPTR.

The internal data pointer parameter has the same function on both devices.

The internal byte count parameter has the same function on both devices.

9.3.4.2.3 Protocol-Dependent Parameter RAM Values. These values are very similar between devices, although new functions are often added on the QUICC. Where possible, any parameter for a given protocol that is the same for the MC68302 and the QUICC carries the same name on both devices.

The following points note changes between the MC68302 and the QUICC protocol-dependent parameter RAM for a given protocol:

In the UART mode, if the MAX_IDL entry is programmed to $0000 on the QUICC, the function will be disabled.

In the UART mode, the ability to send special control characters, such as XOFF, no longer requires the CHARACTER8 entry. Rather, a new entry, called the TOSEQ entry, has been created. The programming of the TOSEQ entry, however, is the same as the old MC68302 CHARACTER8 entry.

In UART mode, note the new parameters for a receive character mask (RCCM) and a function that times the length of a receive break (BRKLN).

The HDLC parameter RAM is the same, except for two new entries: RFTHR and RFCNT. They allow the user to reduce the number of received frame interrupts generated, and should be used in higher data rate applications.

The BISYNC parameter RAM is unchanged.

DDCMP is a microcode RAM product on the QUICC. The port of DDCMP from the MC68302 is not discussed.

V.110 is not supported on the QUICC.

This design also uses the RAS1 double-drive capability, whereby the RAS1DD signal is output by the QUICC on the BCLRI pin to increase the effective drive capability of the RAS1 signal. The RAS1 line should be programmed to respond to a 4-Mbyte address space.

After power-on reset, the software must wait the required time before accessing the DRAM. The required eight read cycles must then be performed either in software or by waiting for the refresh controller to perform these accesses.

9.8.2.6 DRAM DEVICES. Figure 9-33 shows the interface to a standalone DRAM device. In this case the MCM54260 256K × 16 DRAM device is chosen. This allows a full 32-bit wide DRAM solution using only two DRAM devices, with byte writes still supported using the upper and lower CAS pins. Both the MC68EC030 and the QUICC can access the DRAM array. The RAS1 line should be programmed to respond to a 1-Mbyte address space.

The address multiplexing scheme shown is the same as that for the DRAM SIMM. No parity support is provided in this case. The RAS1DD signal is not used in this case, since only two devices are supported.

After power-on reset, the software must wait the required time before accessing the DRAM, and then perform the required eight read cycles, either in software or by waiting for the refresh controller to perform these accesses.

9.8.3 Software Configuration

The following paragraphs discuss a number of key points for a software engineer desiring to initialize the system. The only items discussed are those that are required to allow the previously discussed hardware configuration.

9.8.3.1 BASIC INITIALIZATION. The following register initializations are basic to all types of applications.

The module base address register (MBAR) should be set as desired. However, the QUICC 8-Kbyte block should not overlap any memory array.

The module base address register enable (MBARE) should not be accessed.

In the module configuration register (MCR), ASTM and BSTM should be set to indicate synchronous operation. SHEN1–SHEN0 should be cleared.

In the system protection control register (SYPCR), DBFE should be cleared. BME should be set. If the software watchdog is used, the SWRI bit should be set.

The periodic interrupt control register (PICR) may be set as desired.

The port E pin assignment register (PEPAR) should be set to $51C0. This configures three IOUTx lines to go out on the unused parity pins, the RAS1DD pin, WE lines instead of the A31–A28 lines, the AMUX pin (assuming DRAM is used in the system; otherwise, the OE function should be programmed), four CASx lines, CS7, and AVECO.

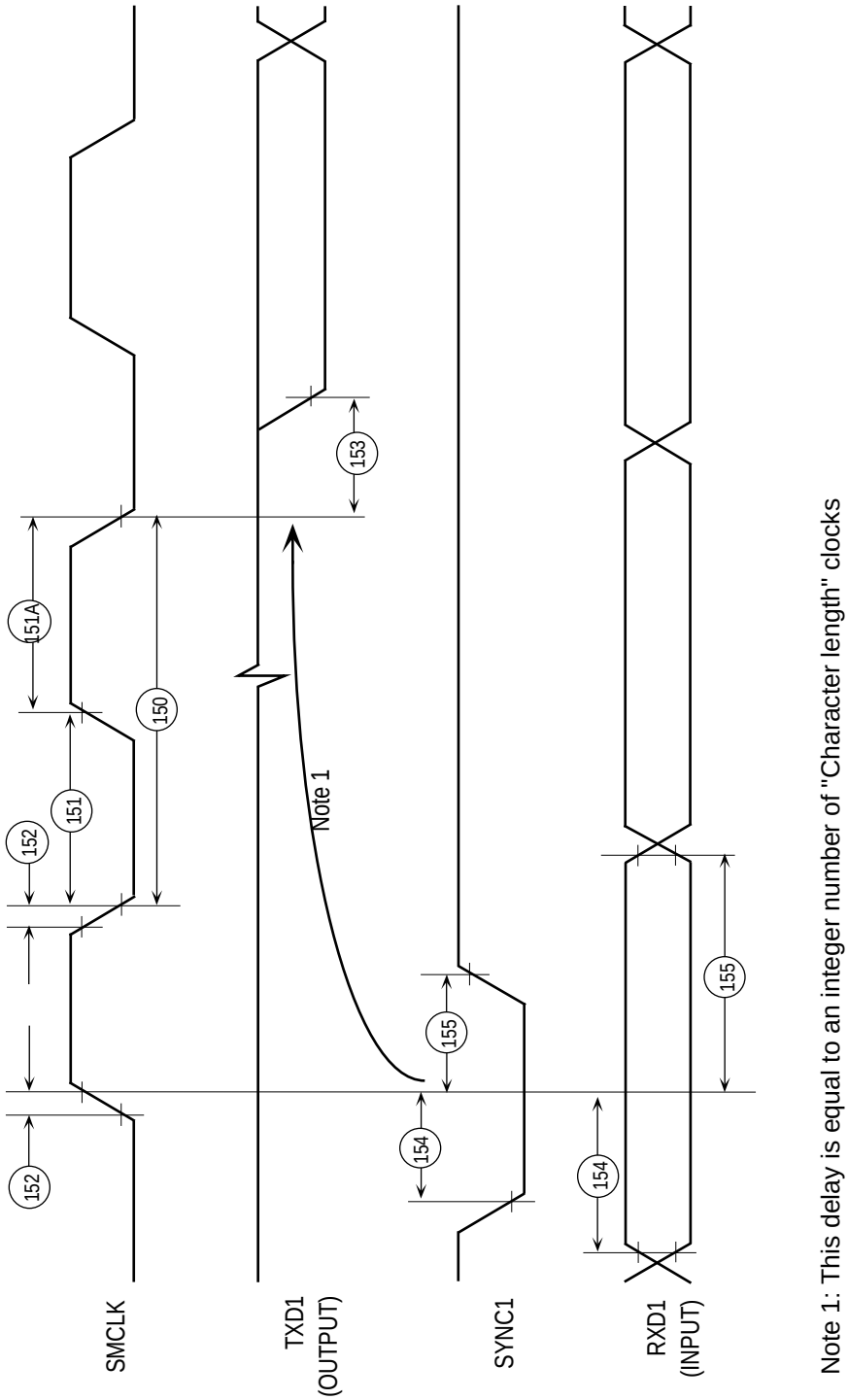

Figure 10-74. SMC Transparent

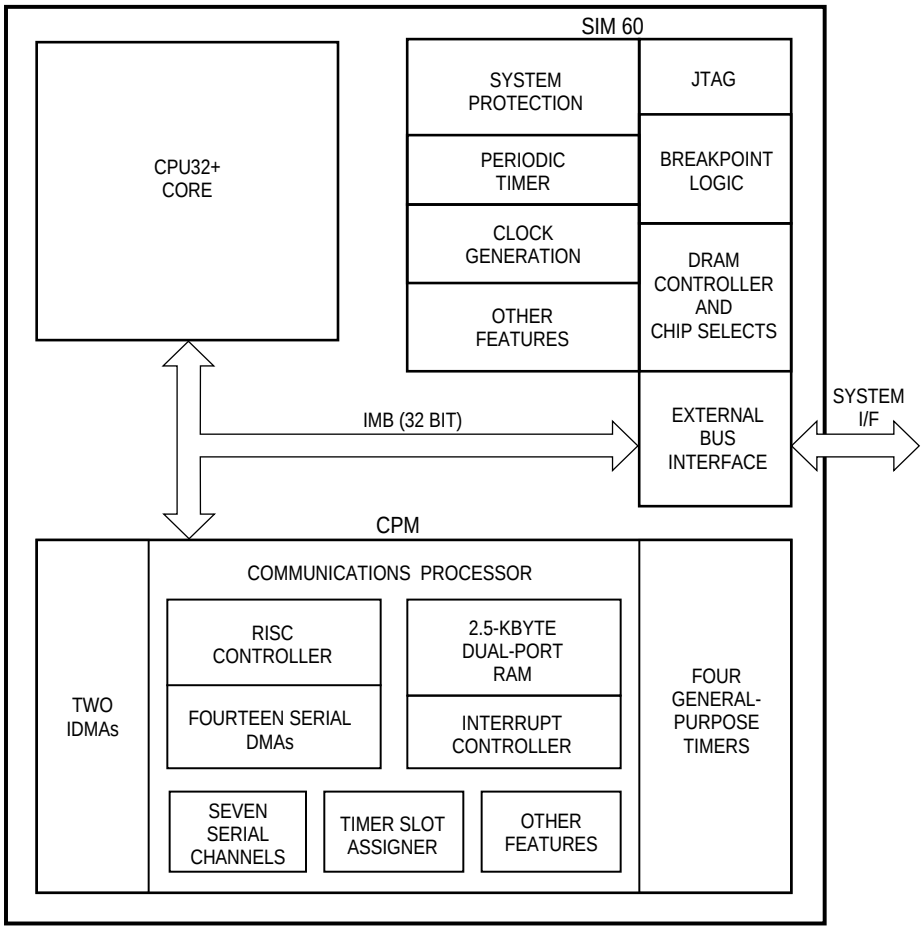

Figure D-1. QUICC Block Diagram

D.2.1 CPU32+ Core

The CPU32+ core is a CPU32 that has been modified to connect directly to the 32-bit IMB and apply the larger bus width. Although the original CPU32 core had a 32-bit internal data path and 32-bit arithmetic hardware, its interface to the IMB was 16 bits. The CPU32+ core can operate on 32-bit external operands with one bus cycle. This allows the CPU32+ core to fetch a long-word instruction in one bus cycle and to fetch two word-length instructions in one bus cycle, filling the internal instruction queue more quickly. The CPU32+ core can also read and write 32-bits of data in one bus cycle.

Although the CPU32+ instruction timings are improved, its instruction set is identical to that of the CPU32. It will also execute the entire M68000 instruction set. It contains the same background debug mode (BDM) features as the CPU32. No new compilers, assemblers, or other software support tools need be implemented for the CPU32+; standard CPU32 tools can be used.

The CPU32+ delivers approximately 4.5 MIPS at 25 MHz, based on the standard (accepted) assumption that a 10-MHz M68000 delivers 1 VAX MIPS. If an application requires more

INDEX