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	3.3V
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/mc68e360zq25vlr2

- Parallel Interface Port
  —Centronics[4] Interface Support
  —Supports Fast Connection Between QUICCs
- 240 Pins Defined: 241-Lead Pin Grid Array (PGA) and 240-Lead Plastic Quad Flat Pack (PQFP)

1.2 QUICC ARCHITECTURE OVERVIEW

The QUICC is 32-bit controller that is an extension of other members of the Motorola M68300 family. Like other members of the M68300 family, the QUICC incorporates the inter-module bus (IMB). (The MC68302 is an exception, having an M68000 bus on chip.) The IMB provides a common interface for all modules of the M68300 family, which allows Motorola to develop new devices more quickly by using the library of existing modules. Although the IMB definition always included an option for an on-chip 32-bit bus, the QUICC is the first device to implement this option.

The QUICC is comprised of three modules: the CPU32+ core, the SIM60, and the CPM. Each module utilizes the 32-bit IMB. The MC68360 QUICC block diagram is shown in Figure 1-1.

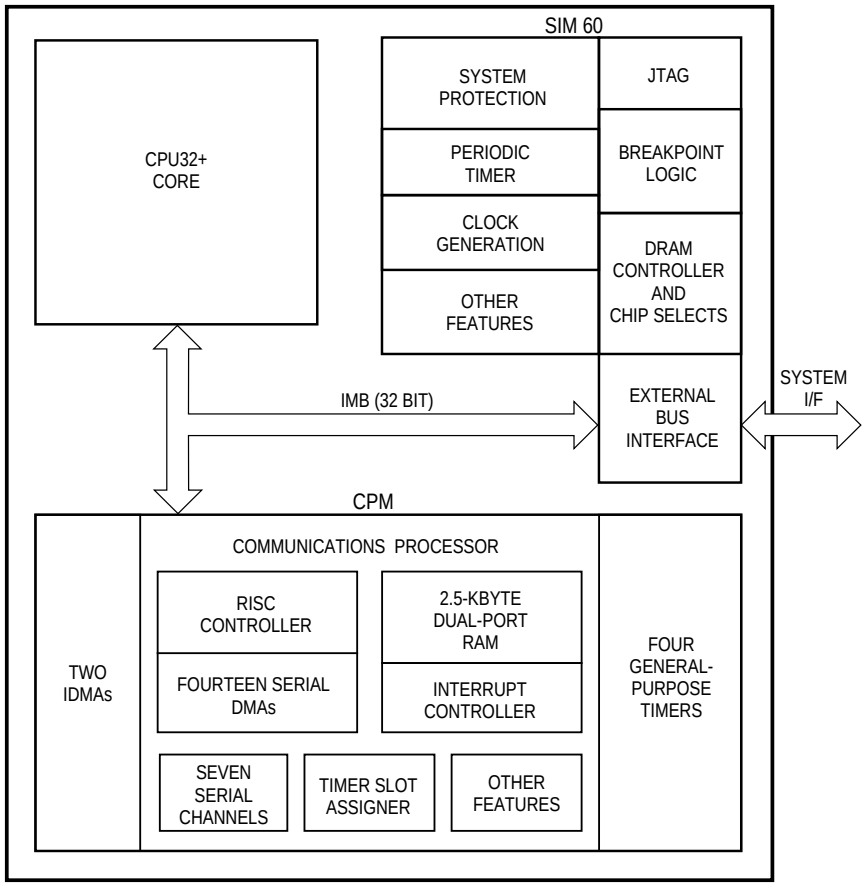


Figure 1-1. QUICC Block Diagram

4. Centronics is a trademark of Centronics, Inc.

pins are the only data pins used. Refer to Section 4 Bus Operation for information on the data bus and its relationship to bus operation.

2.1.3.2 DATA BUS (D15–D0). These pins can function as 16 additional data pins used in long-word and 3-byte transfers. They are three-stated and not used if the QUICC is configured into 16-bit bus mode.

2.1.4 Parity

These three-state bidirectional signals provide parity generation/checking for the data path between the QUICC or external masters and other devices. There are four parity lines—one for every eight data bits. The parity lines consists of two groups. Refer to Section 6 System Integration Module (SIM60) for more information on parity generation/checking.

2.1.4.1 PARITY (PRTY0). This pin is the parity value for data bits 31–24.

2.1.4.2 PARITY (PRTY1). This pin is the parity value for data bits 23–16.

2.1.4.3 PARITY (PRTY2). This pin is the parity value for data bits 15–8.

2.1.4.4 PARITY (PRTY3). This pin has two functions. During total system reset, it is the $\overline{16BM}$ pin to determine whether 16-bit data bus mode is to be enabled. After system reset, it functions as the parity line 3.

PRTY3—This pin is the parity value for data bits 0–7.

$\overline{16BM}$—This pin selects the 16-bit data bus mode. To choose a 32-bit data bus during total system reset, this pin can be left floating (it has an internal pullup resistor) or can be driven/pulled high. To choose a 16-bit data bus during total system reset, this pin should be driven/pulled low.

2.1.5 Memory Controller

The following signals are used to control an external memory device.

2.1.5.1 CHIP SELECT/ROW ADDRESS SELECT ($\overline{CS6}$–$\overline{CS0}$/$\overline{RAS6}$–$\overline{RAS0}$). The chip-select output signals enable peripherals or memory arrays at programmed addresses. CS0 is the global chip select for the boot ROM containing the user's reset vector and initialization program. Refer to Section 6 System Integration Module (SIM60) for more information on chip selects.

NOTE

In addition, $\overline{RAS1}$ can be simultaneously output on the $\overline{RAS1DD}$ pin to increase the $\overline{RAS1}$ line drive capability, and $\overline{RAS2}$ can be simultaneously output on the $\overline{RAS2DD}$ pin to increase the $\overline{RAS2}$ line drive capability.

2.1.5.2 CHIP SELECT/ROW ADDRESS SELECT/INTERRUPT ACKNOWLEDGE ($\overline{CS7}$/$\overline{RAS7}$/$\overline{IACK7}$). This pin can be programmed as a $\overline{CS7}$/$\overline{RAS7}$ pin or as the $\overline{IACK7}$ line. See Section 6 System Integration Module (SIM60) for more information on this selection.

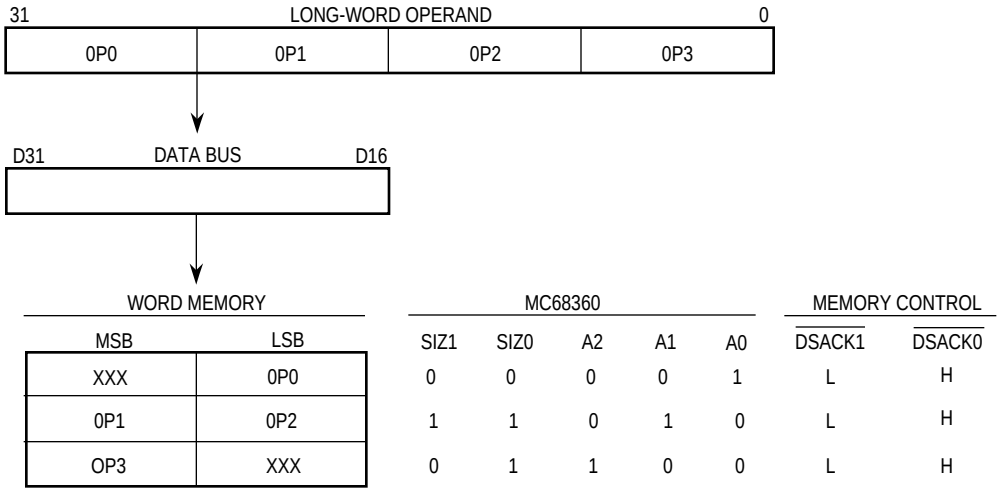


Figure 4-8. Misaligned Long-Word Transfer to Word Port Example

The interrupt acknowledge cycle is a read cycle. It differs from the read cycle described in 4.3.1 Read Cycle in that it accesses the CPU address space. Specifically, the differences are as follows:

1. FC3–FC0 are set to $7 (FC3/FC2/FC1/FC0 = 0111) for CPU address space.
2. A3, A2, and A1 are set to the interrupt request level, and the IACKx strobe corresponding to the current interrupt level is asserted. (Either the function codes and address signals or the IACKx strobes can be monitored to determine that an interrupt acknowledge cycle is in progress and the current interrupt level.)
3. The CPU32+ space type field (A19–A16) is set to $F (interrupt acknowledge).
4. Other address signals (A31–A20, A15–A4, and A0) are set to one.

The responding device places the vector number on the data bus during the interrupt acknowledge cycle. Beyond this, the cycle is terminated normally with DSACKx.

Figure 4-26 is a flowchart of the interrupt acknowledge cycle; Figure 4-27 shows the timing for an interrupt acknowledge cycle terminated with DSACKx.

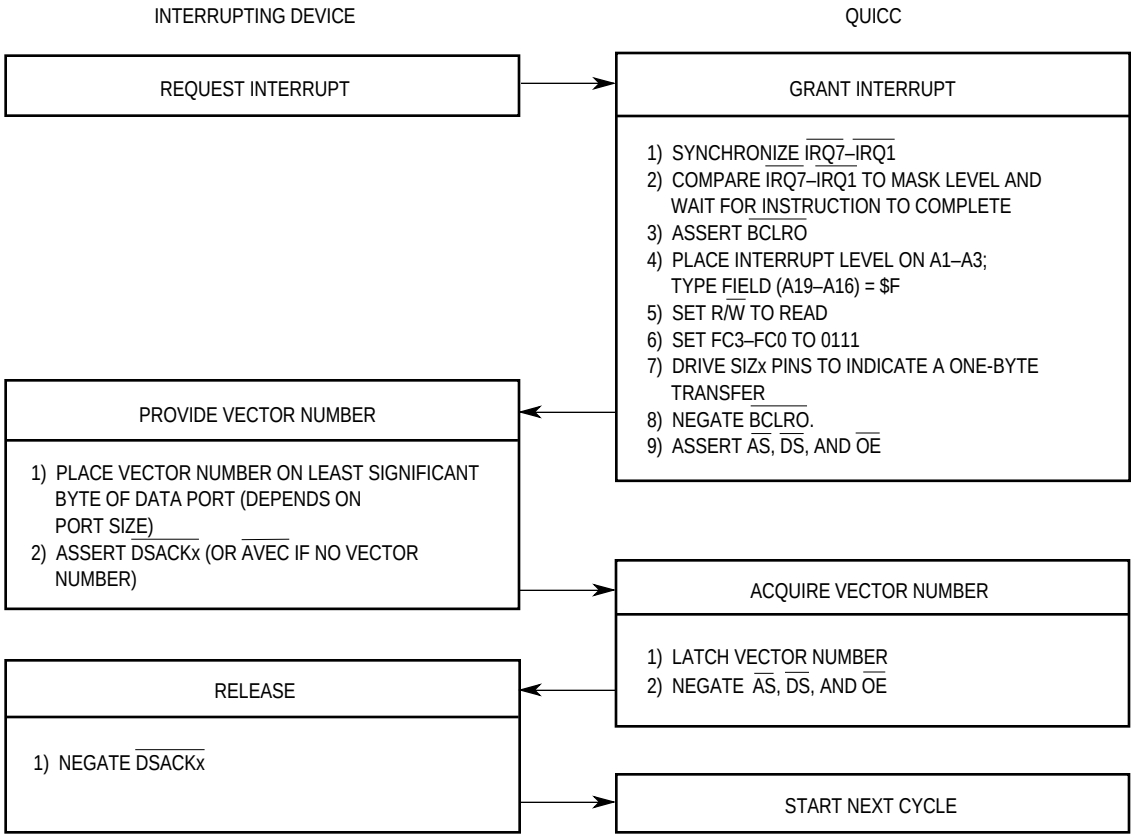


Figure 4-26. Interrupt Acknowledge Cycle Flowchart

5.7.2.2 CALCULATE EFFECTIVE ADDRESS. The calculate EA table indicates the number of clock periods needed for the processor to calculate a specified EA. The timing is equivalent to fetch EA except there is no read cycle. The tail and cycle time are reduced by the amount of time the read would occupy. 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	Notes
Dn	–	–	0(0/0/0)	–
An	–	–	0(0/0/0)	–
(An)	1	0	2(0/0/0)	–
(An)+	1	0	2(0/0/0)	–
–(An)	2	0	2(0/0/0)	–
(d_{16},An) or (d_{16},PC)	1	1	3(0/1/0)	1,3
(xxx).W	1	1	3(0/1/0)	1
(xxx).L	1	3	5(0/2/0)	1
(d_8,An,Xn.Sz × Sc) or (d_8,PC,Xn.Sz × Sc)	4	0	6(0/1/0)	2,3,4
(0) (All Suppressed)	2	0	4(0/1/0)	4
(d_{16})	1	1	5(0/2/0)	1,4
(d_{32})	1	3	7(0/3/0)	1,4
(An)	1	0	4(0/1/0)	4
(Xm.Sz × Sc)	4	0	6(0/1/0)	2,4
(An,Xm.Sz × Sc)	4	0	6(0/1/0)	2,4
(d_{16},An) or (d_{16},PC)	1	1	5(0/2/0)	1,3,4
(d_{32},An) or (d_{32},PC)	1	3	7(0/3/0)	1,3,4
(d_{16},An,Xm) or (d_{16},PC,Xm)	2	0	6(0/2/0)	3,4
(d_{32},An,Xm) or (d_{32},PC,Xm)	1	1	7(0/3/0)	1,3,4
(d_{16},An,Xm.Sz × Sc) or (d_{16},PC,Xm.Sz × Sc)	2	0	6(0/2/0)	2,3,4
(d_{32},An,Xm.Sz × Sc) or (d_{32},PC,Xm.Sz × Sc)	1	1	7(0/3/0)	1,2,3,4

NOTES:
1. Replacement fetches overlap the head of the operation by the amount specified in the tail.
2. Size and scale of the index register do not affect execution time.
3. The PC may be substituted for the base address register An.
4. When adjusting the prefetch time for slower buses, extra clocks may be subtracted from the head until the head reaches zero, at which time additional clocks must be added to both the tail and cycle counts.
5. Timing is calculated with the CPU32+ in 16-bit mode

- General-Purpose Chip Selects (SRAM Banks)
  - —May Be Used with SRAM, EPROM, FEPROM, and Peripherals
  - —Global (Boot) Chip Select Available at System Reset
  - —Two-Clock Accesses to External SRAM
  - —Programmable Port Size of 8, 16, and 32 Bits for Each Chip Select
- DRAM Controller (DRAM Banks)
  - —Supports up to Eight Banks of DRAM of Size 128K × X, 256K × X, 512K × X, 1M × X, 2M × X, 4M × X, 8M × X or 16M × X
  - —Supports a DRAM Port Size of 16 or 32 Bits
  - —Internal Address Multiplexing for 16- and 32-Bit DRAM Systems Available for all On-Chip Bus Masters
  - —Glueless Interface to One Bank of DRAM SIMMs (Only External Buffers Are Required for Additional SIMM Banks)
  - —Four $\overline{\text{CAS}}$ Lines
  - —Two of the Eight $\overline{\text{RAS}}$ Lines May Be Output on Two Pins Each for Double-Drive Capability
  - —Page Mode with Page Switch Detection Logic
  - —Page Mode Supports 128K, 256K, 512K, 1M, 2M, 4M, 8M, and 16M Page Banks
  - —Supports Page Mode Normal, Page Hit, and Page Miss
  - —Burst Support for the MC68040 Accesses to DRAM
- DRAM Controller Also Contains a Refresh Unit with:
  - —$\overline{\text{CAS}}$ Before $\overline{\text{RAS}}$ Refresh Support
  - —A Programmable Refresh Timer
  - —Refresh Active During External Reset
  - —Disable Refresh Mode
  - —Stacking of up to Seven Refresh Cycles
- DRAM Controller Also Supports External Masters
  - —Supports MC68EC040 with 3,2,2,2 Line Fill (60-ns DRAMs)
  - —Supports DRAM for External QUICC or MC68030-Type Accesses (Page Support Available in this Mode)
  - —Supports DRAM Control for System Bus Containing External MC68EC040 and Multiple QUICCs
  - —Synchronous and Asynchronous External Masters Possible
  - —Special Options for External Master to Improve DRAM Performance

6.10.2 Memory Controller Overview

The block diagram of the QUICC memory controller is shown in Figure 6-10. The general-purpose chip selects provide a glueless interface to EPROM, SRAM, flash EPROM (FEPROM), and other peripherals. The general-purpose chip selects are available on lines $\overline{\text{CS0}}$–$\overline{\text{CS7}}$. $\overline{\text{CS0}}$ also functions as the global (boot) chip select for accessing the boot EPROM. The chip selects allow 0 to 15 wait states.

The flexible memory controller allows a glueless DRAM interface to single in-line memory modules (SIMMs) as well as a grid array of DRAMs on a board. The DRAM controller controls the address multiplexing, access mode, refresh operation, and the timing generation

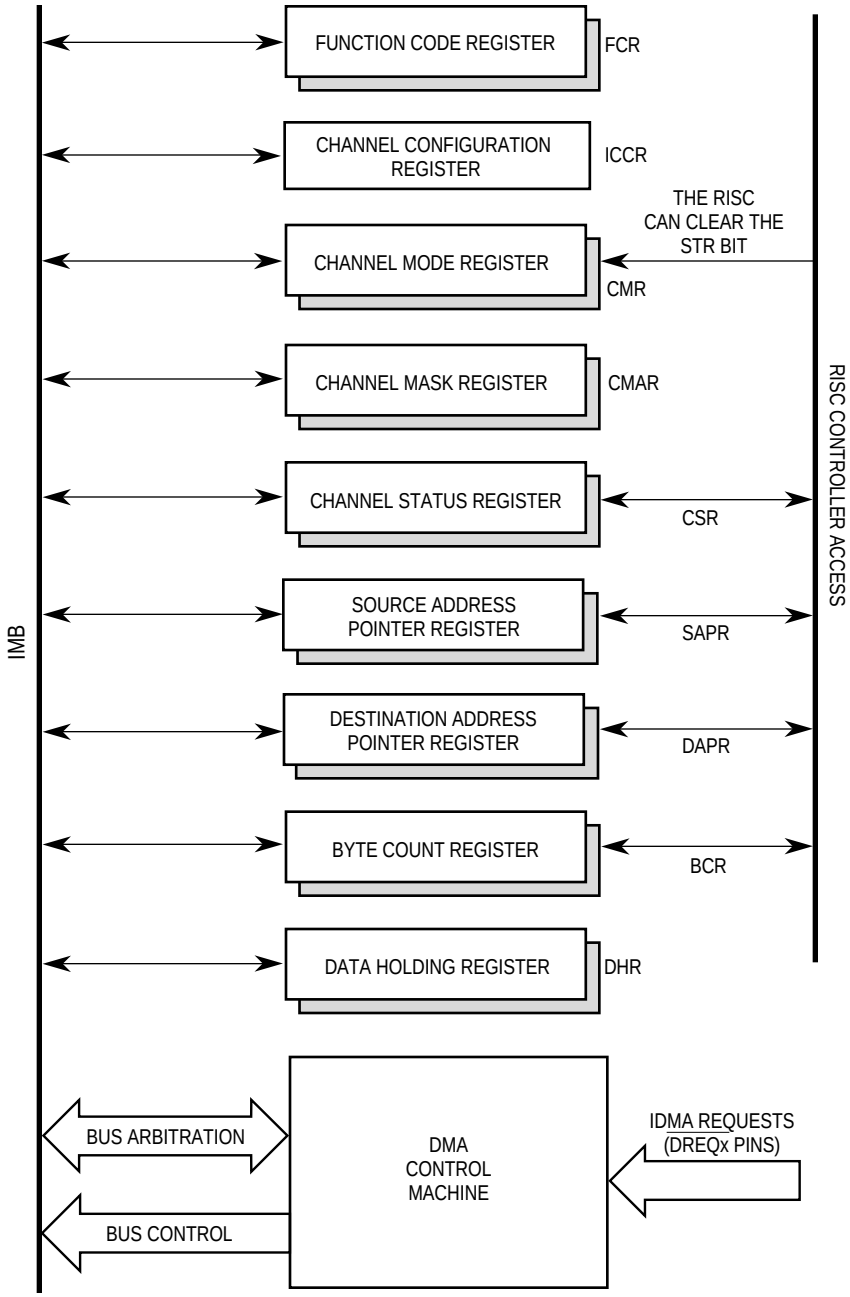


Figure 7-8. IDMA Controller Block Diagram

7.6.1 IDMA Key Features;

The IDMA contains the following features:

- Two Independent, Fully Programmable DMA Channels
- Dual Address or Single Address Transfers with 32-Bit Address and 32-Bit Data Capability

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	**E**	—	**W**	**I**	L	F	—	—	—	—	LG	NO	SH	CR	OV	CL
OFFSET + 2	DATA LENGTH															
OFFSET + 4	**RX DATA BUFFER POINTER**															
OFFSET + 6																

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

E—Empty

0 = The data buffer associated with this Rx BD has been filled with received data, or data reception has been aborted due to an error condition. The CPU32+ core is free to examine or write to any fields of this Rx BD. The CP will not use this BD again while the E-bit remains zero.

1 = The data buffer associated with this Rx 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.

Bits 14, 9–6—Reserved

W—Wrap (Final BD in Table)

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

1 = This is the last BD in the Rx 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 RBASE). The number of Rx BD s 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 used.

1 = The RXB bit or RXF bit in the Ethernet event register will be set when this buffer has been used by the Ethernet controller. These two bits may cause interrupts if they are enabled.

L—Last in Frame

This bit is set by the Ethernet controller when this buffer is the last in a frame. This implies the end of the frame or reception of an error, in which case one or more of the CL, OV, CR, SH, NO, and LG bits are set. The Ethernet controller will write the number of frame octets to the data length field.

0 = The buffer is not the last in a frame.

1 = The buffer is the last in a frame.

F—First in Frame

This bit is set by the Ethernet controller when this buffer is the first in a frame.

0 = The buffer is not the first in a frame.

1 = The buffer is the first in a frame.

M—Miss

This bit is set by the Ethernet controller for frames that were accepted in promiscuous mode, but were flagged as a "miss" by the internal address recognition. Thus, while in pro-

7.11.7.5 SMC UART RECEPTION PROCESSING. When the CPU32+ core enables the SMC receiver in UART mode, it will enter hunt mode, waiting for the first character to arrive. Once the first character arrives, the first Rx BD is checked by the CP to see if it is empty. It then begins storing characters in the associated data buffer.

When the data buffer has been filled or the MAX_IDL timer has expired (assuming it was enabled), the SMC clears the E-bit in the BD and generates an interrupt if the I-bit in the BD is set. If the incoming data exceeds the length of the data buffer, the SMC will fetch the next BD in the table and, if it is empty, will continue to transfer data to this BD's associated data buffer.

If the CM bit is set in the Rx BD, the E-bit will not be cleared, allowing the associated data buffer to be overwritten automatically when the CP next accesses this data buffer.

7.11.7.6 SMC UART PROGRAMMING MODEL. An SMC configured as a UART uses the same data structure as in the other modes. The SMC UART data structure supports multi-buffer operation. The SMC UART allows the user to transmit break and preamble sequences. Overrun, parity, and framing errors are reported via the BDs. In its simplest form, the SMC UART can function in a character-oriented environment. Each character is transmitted with accompanying stop bits and parity (as configured by the user), and received into separate 1-byte buffers. Reception of each buffer may generate a maskable interrupt.

Many applications may want to take advantage of the message-oriented capabilities supported by the SMC UART by using linked buffers (in either receive or transmit). In this case, data is handled in a message-oriented environment; users can work on entire messages rather than operating on a character-by-character basis. A message may span several linked buffers. Each message can be both transmitted and received as a linked list of buffers without any intervention from the CPU32+, which achieves both ease in programming and significant savings in processor overhead.

In the message-oriented environment, the idle sequence is used as the message delimiter. The transmitter is able to generate an idle sequence before starting a new message, and the receiver is able to close a buffer upon detection of idle sequence.

7.11.7.7 SMC UART COMMAND SET. The following transmit and receive commands are issued to the CR.

7.11.7.7.1 Transmit Commands. The following paragraphs describe the SMC UART transmit commands.

STOP TRANSMIT Command. The channel STOP TRANSMIT command disables the transmission of characters on the transmit channel. If this command is received by the SMC UART controller during message transmission, transmission of that message is aborted. The SMC UART completes transmission of any data already transferred to its FIFO and shift register (up to two characters) and then stops transmitting data. The TBPTR is not advanced when this command is issued.

The SMC UART transmitter will transmit a programmable number of break sequences and then start to transmit idles. The number of break sequences (which may be zero) should be

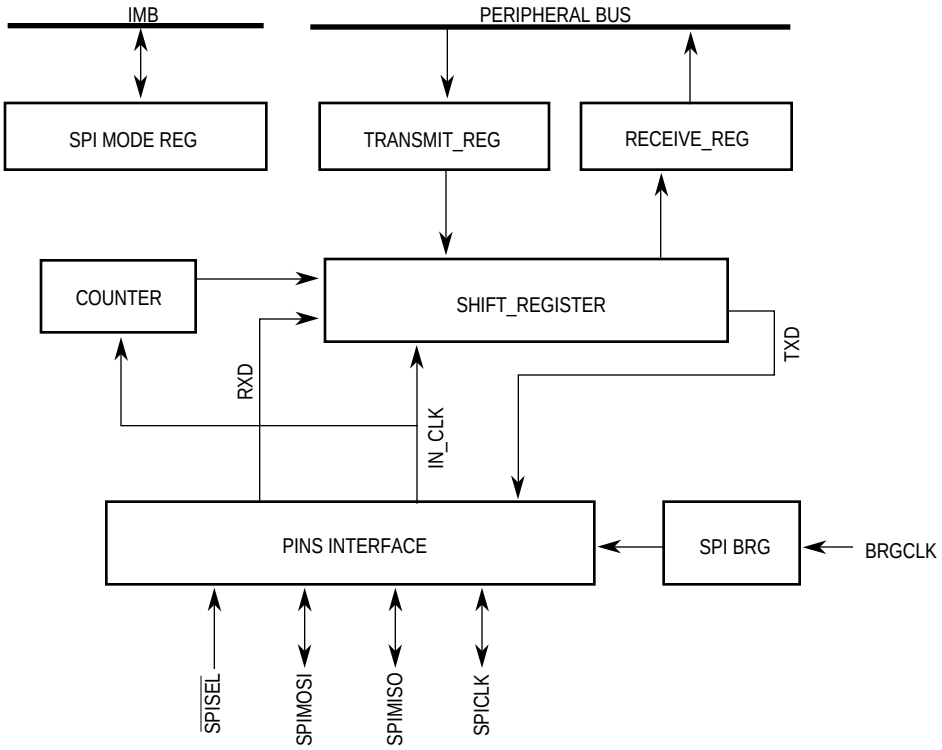


Figure 7-80. SPI Block Diagram

NOTE

The SPI is a superset of the MC68302 serial communications port (SCP).

The SPI receiver and transmitter are double-buffered as shown in the block diagram. This corresponds to an effective FIFO size (latency) of 2 characters.

Note that the LSB of the SPI is labeled as data bit 0 on the serial line; whereas, other devices, such as the MC145554 CODEC, may label the MSB as data bit 0. The QUICC SPI bit 7 (MSB) is shifted out first.

When the SPI is not enabled in the SPMODE, it consumes minimal power.

7.12.2 SPI Key Features

The SPI contains the following key features:

- Four-Wire Interface (SPIMOSI, SPIMISO, SPICLK, and $\overline{\text{SPISEL}}$)
- Full-Duplex Operation
- Works with Data Characters from 4 to 16 bits in length
- Supports Back-to-Back Character Transmission and Reception
- Master or Slave SPI Modes Supported

The PIP shares several registers with the SMC2 serial channel. SMC2 is not available and should not be enabled if the PIP is used. If SMC2 is enabled, erratic behavior will occur.

7.13.3 General-Purpose I/O Pins (Port B)

In this configuration, the PIP is not used, but rather operates as general-purpose parallel I/O port B. See 7.14.7 Port B Registers for more details.

7.13.4 Interlocked Data Transfers

In the interlocked handshake mode, the PIP may be configured as a transmitter or a receiver. This configuration allows a fast connection between QUICCs, and may be used for the P1284-protocol advanced byte transfer mode.

The interlocked handshake mode may be controlled by the RISC or the CPU32+ core. Operation using the RISC requires BDs and parameter RAM initialization very similar to the other serial channels. Data is then stored in the buffers using one of the SDMA channels (one of the available channels from SMC2). Operation by the CPU32+ core is performed by software-controlled reads and writes from/to the PIP data register upon interrupt request.

NOTE

At the time of writing, RISC operation of the PIP has not been fully defined. The user should use the CPU32+ core operation mode, until such time as RISC microcode becomes available or the full PIP microcode is available in the RISC internal ROM. Please contact the local Motorola sales representative to obtain the current status of the PIP RISC microcode. In the following description, the RISC reads and writes of the data register are replaced by CPU32+ core reads and writes.

When configured as a transmitter, the STBO pin (PB16) is used as a strobe output ($\overline{STB}$) handshake control signal, and the STBI pin (PB17) is used as an acknowledge ($\overline{ACK}$) input. When configured as a receiver, the PIP generates the $\overline{ACK}$ signal on the STBO pin and inputs the $\overline{STB}$ signal on the STBI pin.

Bits PB16 and PB17 in the port B data direction register (PBDIR) and the port B data register (PBDAT) corresponding to STBO and STBI are not valid and are ignored by the PIP in the interlocked handshake mode.

When the PIP is in this mode and is configured as a transmitter, the RISC controller loads data into the output latch when it receives a request to begin transfers from the host processor (see Figure 7-85). Once data is loaded, after a programmable setup time, the $\overline{STB}$ signal is asserted (low). Then when $\overline{ACK}$ is sampled as low, the data is transmitted, followed by the $\overline{STB}$ being negated (high). $\overline{STB}$ remains high until new data is loaded into the output latch and $\overline{ACK}$ is negated (high).

When the PIP is configured as a receiver, input data is latched when the $\overline{STB}$ signal is sampled as low. The $\overline{ACK}$ signal is then asserted. $\overline{ACK}$ will be negated (high) when the data has been removed from the input latch.

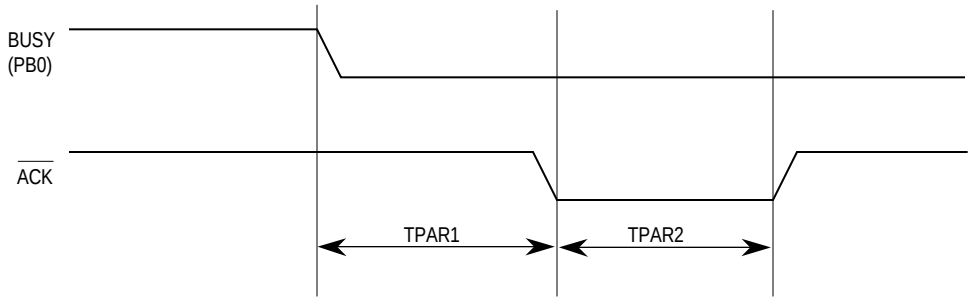


Figure 7-89. Centronics Receiver Timing Mode 0

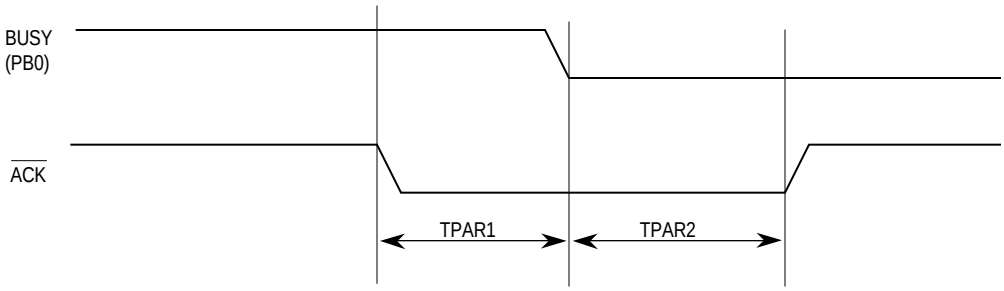


Figure 7-90. Centronics Receiver Timing Mode 1

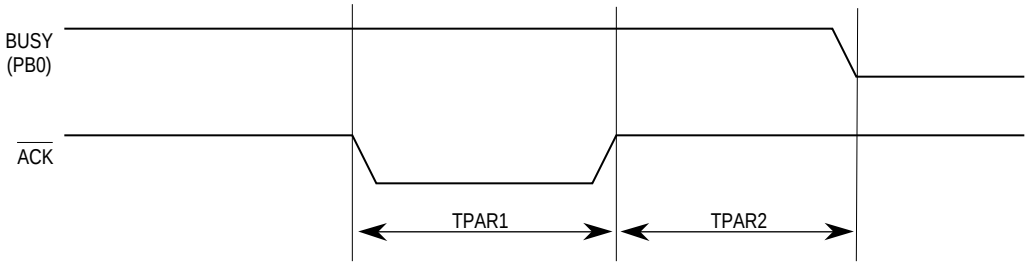


Figure 7-91. Centronics Receiver Timing Mode 2

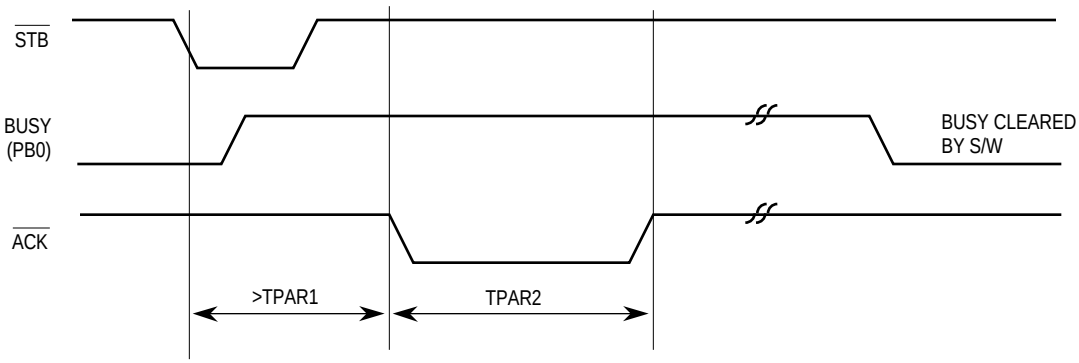


Figure 7-92. Centronics Receiver Timing Mode 3

The SCC status register (SCCS) is also available on the QUICC. The configuration of this register depends on the protocol mode. In all cases, the $\overline{CTS}$ and $\overline{CD}$ bits are available in the PCDAT register, which allows the user to read the current value of these signals directly from the pins. Additionally, the ID bit is available in the QUICC SCCS register in UART and HDLC modes.

NOTE

New status functions are available in the QUICC SCCS, such as whether the HDLC controller is currently receiving flags and the DPLL carrier sense status.

The 16-bit SPMODE register consists of two halves: the 8-bit SMC mode register and the 8-bit SCP mode register.

The SMCs on the QUICC are much more similar to the SCCs on the MC68302. To port an ISDN (GCI or IOM2) application to the QUICC that uses the MC68302 SCCs, the user should use the features provided in the serial interface and SMC.

The QUICC does not contain special support for IDL in the SMCs because IDL no longer uses the A or M bits, but rather uses an out-of-band control bus called the SCP. On the QUICC, the SCP function may be implemented with the SPI.

NOTE

On the QUICC, the two SMCs can also provide a UART or totally transparent control function.

The SCP on the QUICC is enhanced to full serial peripheral interface (SPI) functionality. The SPI is found on many Motorola devices, such as the MC68HC11. The 8-bit SCP mode register is most closely related to the 16-bit SPI mode register (SPMODE) on the QUICC.

The EN bit is in the SPMODE register.

The PM3–PM0 bits are located in the SPMODE register.

The CI bit is located in the SPMODE register.

The LOOP bit is located in the SPMODE register.

The STR bit becomes the R-bit of the SPI transmit buffer descriptor. The buffer descriptor structure of the SPI is like that of the SCCs.

The SIMASK register on the MC68302 is used in GCI and IDL modes to select individual bits of the B-channels to be routed to particular SCCs. On the QUICC, the serial interface has a time slot assigner that is much more flexible and can be used in any time division multiplexed (TDM) interface, not just GCI and IDL. The time slot assigner on the QUICC is programmed using a feature called the serial interface RAM. See 7.8.4 SI RAM.

The SIMODE register on the MC68302 controls the serial interface. On the QUICC, the serial interface control is greatly enhanced and is comprised of the serial interface RAM, the serial interface global mode register (SIGMR), the serial interface mode register (SIMODE),

NOTE

Depending on the assembler used, the acronyms DTT and ITT may have to be used instead of DAC and IAC.

9.6 INTERFACING THE QUICC TO THE 53C90 SCSI CONTROLLER

In the late 1970's, Schugart and Associates introduced a parallel bus called Schugart Associates system interface (SASI). Because of SASI's generic nature and ability to function as a device-independent peripheral or system bus, other manufacturers quickly adopted it. In 1982, ANSI standardized an enhanced version of SASI renaming it the small computer system interface (SCSI). Since its standardization as a bus and due to its diverse potential, SCSI has enjoyed great popularity as an alternative means to network-dissimilar high-performance hosts.

The following paragraphs give a general description of the SCSI bus, including its major signals and functions. The hardware and software interface between the QUICC and the 53C90 SCSI controller is also discussed. This subsection highlights an example of a QUICC IDMA channel and the memory controller features that allow the QUICC to interface to slower peripherals.

9.6.1 SCSI General Overview

SCSI is an 8-bit, parallel I/O bus that provides a host computer with the capability of adding different disk drives, tape drives, printers, and even communication devices without major modifications to the system hardware or software. It uses logical rather than physical addressing for all data blocks.

A maximum of eight devices can be attached to the SCSI bus. Of these eight, only one pair of devices can communicate at one time. Each SCSI device has an ID bit assigned to it that is the bit-significant representation of the SCSI address referring to one of the signal lines DB7–DB0. DB7 has the highest priority. When two devices communicate over the SCSI bus, one acts as an initiator (host), and the other acts as a target (controller). The initiator originates the operation, and the target performs the operation.

9.6.2 Physical Interface

The physical bus interface is composed of a group of characteristics: speed, bus signals, and device count.

Speed, the ability to transfer data, uses two handshaking protocols: synchronous and asynchronous. Synchronous transfers a series of bytes before the handshake occurs; whereas, asynchronous requires a handshake for every byte transferred. Rates up to 6 Mbytes/sec can be accomplished asynchronously; 10 Mbytes/sec is possible on the synchronous protocol.

SCSI devices are daisy-chained using a common cable. This 50-conductor cable is used by the bus signals to interchange data, commands, status, and message information. Table 9-1 and Figure 9-17 describe the SCSI bus signals.

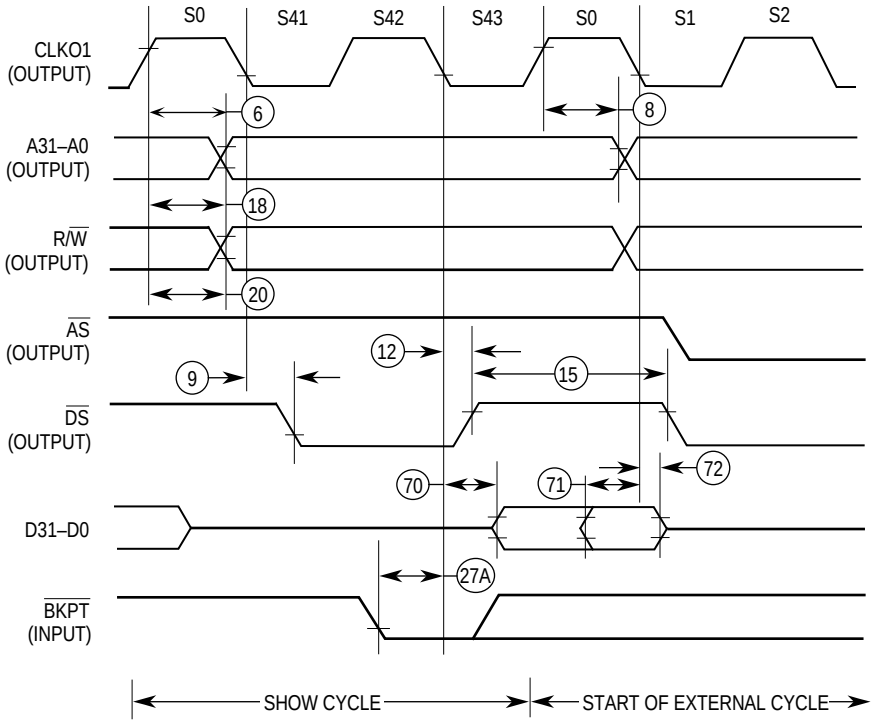


Figure 10-17. Show Cycle

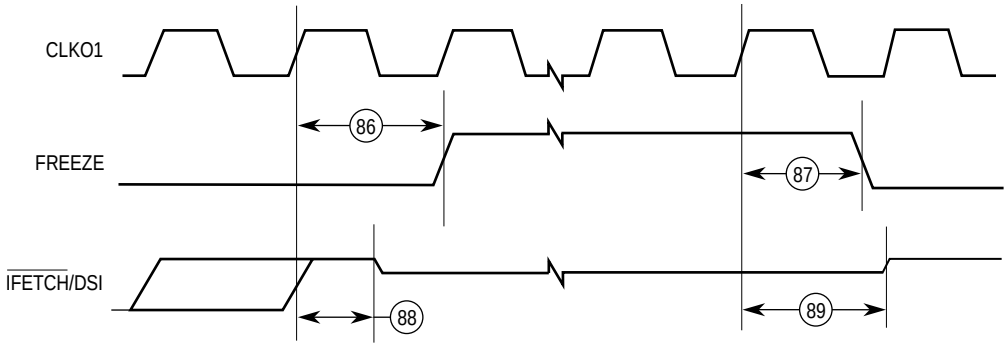
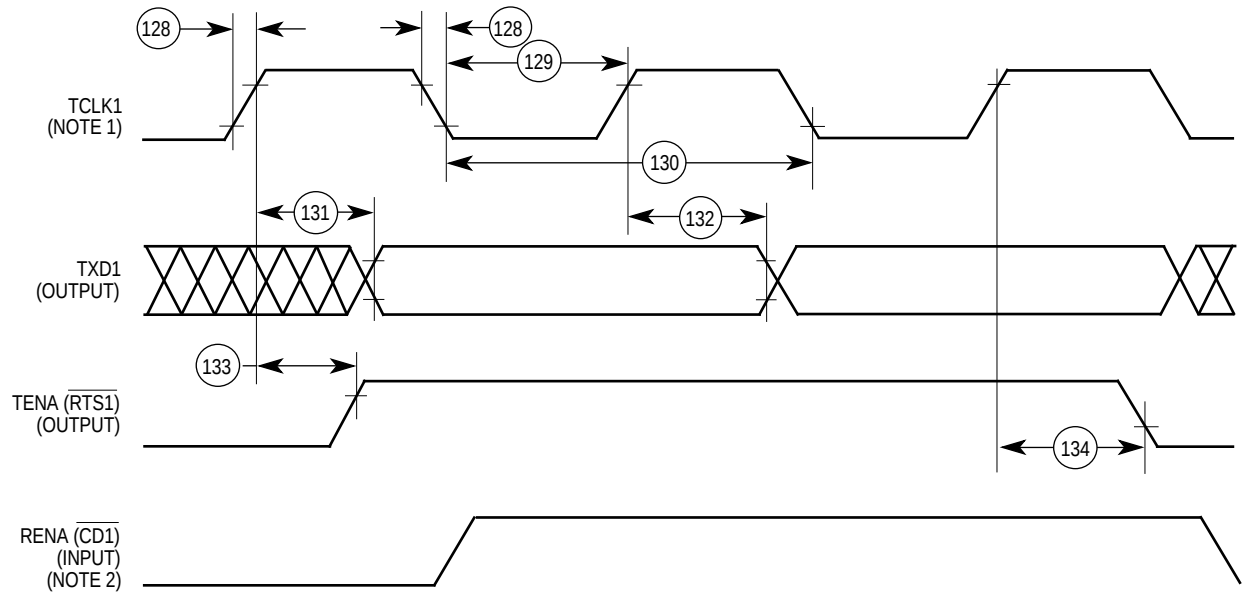
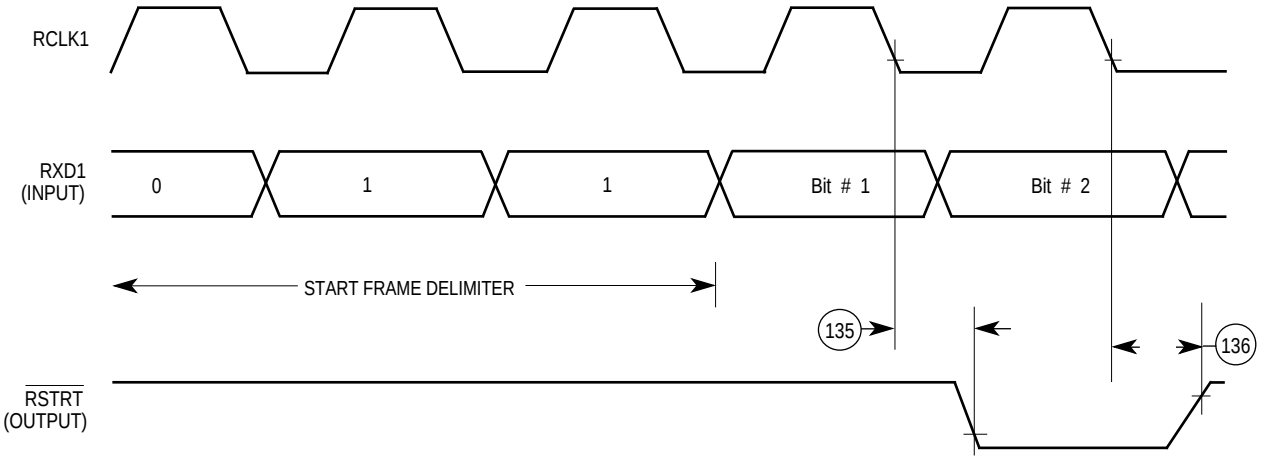


Figure 10-18. Background Debug Mode FREEZE Timing



NOTES:
1. Transmit clock invert (TCI) bit in GSMR is set.
2. If RENA is deasserted before TENA, or RENA is not asserted at all during transit, then CSL bit is set in the buffer descriptor at the end of frame transmission.

Figure 10-70. Ethernet Transmit Timing



NOTE: Valid for the ethernet protocol only.

Figure 10-71. CAM Interface Receive Start Timing

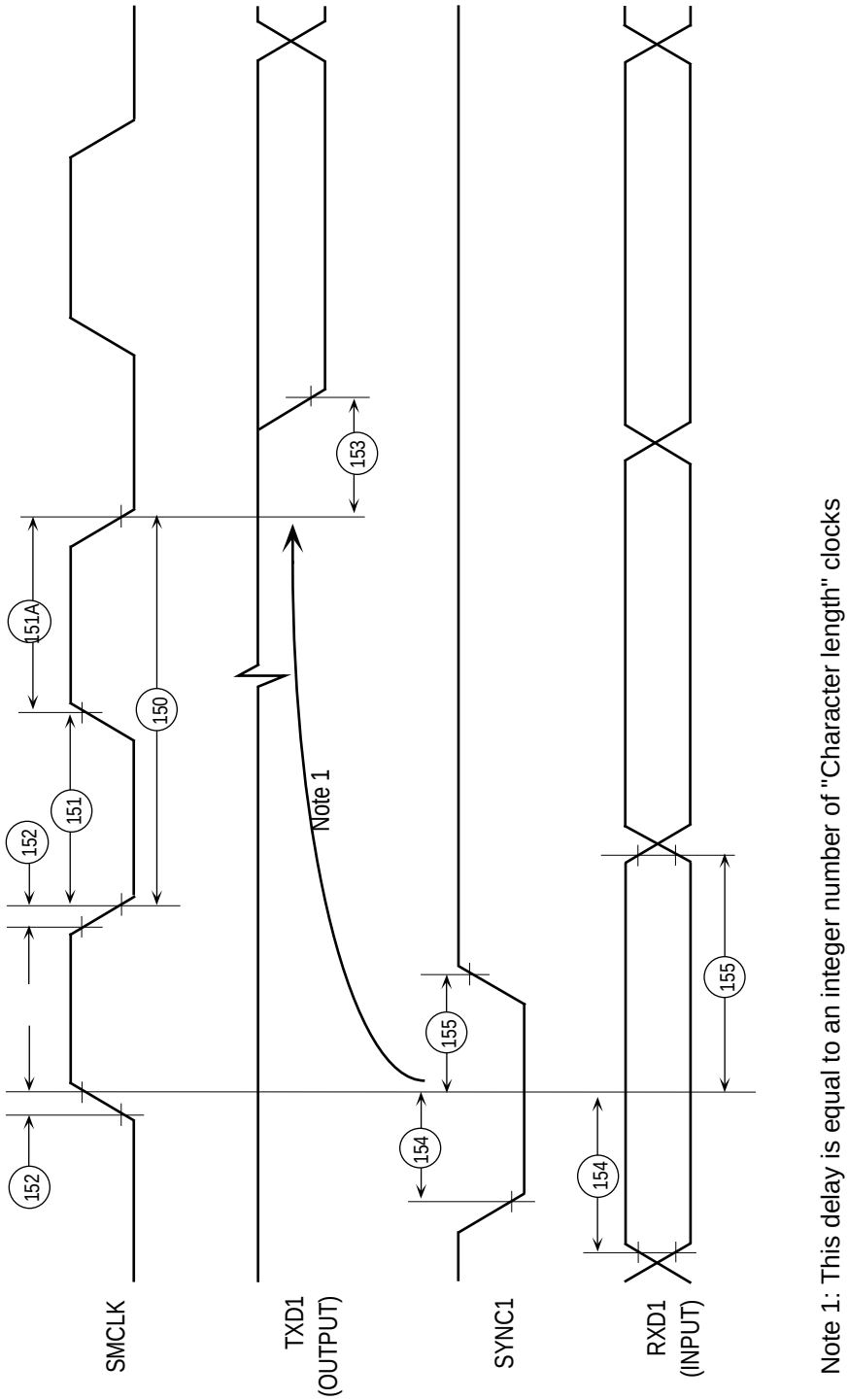


Figure 10-74. SMC Transparent

- All receive data is timestamped.
- Handles 2, 4, 8, or 16 ISDN lines with full C/I and monitor channel functions.
- Handles up to 32 ISDN lines with only C/I channels.
- Simultaneous detection of multiple C/I code changes and transmission changes.
- Maskable interrupts generated for many events.
- Handles GCI monitor messages from 1 to 64 Kilobytes in length.
- Monitor receiver can be locked into a particular channel.
- Monitor receiver and transmitter include timers to prevent lockup due to inactivity.
- Operates independently of user CPU activity.
- Consumes 1280 bytes of the QUICC's internal memory.

C.2.3 Performance

At 25Mhz, an MGCI serial bit rate of 2 Mbps on one SCC consumes 20 - 25% of the processing power of the RISC communications engine. An MGCI SCC operating at a serial bit rate of 2Mbps carries 32 x 64Kbit channels, Table C-2 shows possible QUICC configuration.

Table C-2. MGCI SCC Configuration

MGCI SCCs	Risc Bandwidth Consumed (est)	Possible Configuration of Other Channels
1 x 2 Mbit/s	25%	3 x 2 Mbit HDLC, 2 x 9.6 Kbit SMC UART
2 x 2 Mbit/s	50%	2 x 2 Mbit HDLC, 2 x 9.6 Kbit SMC UART

C.3 ATOM1/ATM CONTROLLER

ATOM1 provides physical layer ATM functions by converting one or more of the QUICC's Serial Communication Controllers (SCCs) into an ATM cell transmitter and receiver. The microcode provides the user with basic cell streaming facilities (cell reception and transmission) and event indications. The primary application of ATOM1 is intended to be plesiochronous digital hierarchy (PDH) and synchronous digital hierarchy (SDH) E1 and DS1 ATM equipment. Such equipment is used for signalling and low rate data transfer.

Figure C-2 shows an ATM / frame relay interworking system as may be used in remote bridging applications. Using the Ethernet channel available on the QUICC, remote LAN bridging equipment can be constructed to link remote Ethernet LANs over E1 telecommunications links.

C.3.1 Key Features

- Cell transmission and reception for all AAL protocols.
- Transmit and receive data buffers located in main memory.
- Microcode constructs the cell header and appends user defined payload on transmit.
- Microcode verifies cell headers and strips HEC before passing cell to the user on receive.

F

G

H