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-BGA
Supplier Device Package	357-PBGA (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68360zp25l

—Independent (Can Be Connected to Any SCC or SMC)
—Allows Changes During Operation
—Autobaud Support Option

- Four SCCs

—Ethernet/IEEE 802.3 Optional on SCC1 (Full 10-Mbps Support)
—HDLC/SDLC[1] (All Four Channels Supported at 2 Mbps)
—HDLC Bus (Implements an HDLC-Based Local Area Network (LAN))
—AppleTalk[2]
—Signaling System #7
—Universal Asynchronous Receiver Transmitter (UART)
—Synchronous UART
—Binary Synchronous Communication (BISYNC)
—Totally Transparent (Bit Streams)
—Totally Transparent (Frame Based with Optional Cyclic Redundancy Check (CRC))
—Profibus (RAM Microcode Option)
—Asynchronous HDLC (RAM Microcode Option)
—DCMP[3] (RAM Microcode Option)
—V.14 (RAM Microcode Option)
—X.21 (RAM Microcode Option)

- Two SMCs

—UART
—Transparent
—General Circuit Interface (GCI) Controller
—Can Be Connected to the Time-Division Multiplexed (TDM) Channels

- One SPI

—Superset of the MC68302 SCP
—Supports Master and Slave Modes
—Supports Multimaster Operation on the Same Bus

- Time-Slot Assigner
- Supports Two TDM Channels

—Each TDM Channel Can Be T1, CEPT, PCM Highway, ISDN Basic Rate, ISDN Primary Rate, User Defined
—1- or 8-Bit Resolution
—Allows Independent Transmit and Receive Routing, Frame Syncs, Clocking
—Allows Dynamic Changes
—Can Be internally Connected to Six Serial Channels (Four SCCs and Two SMCs)

1. SDLC is a trademark of International Business Machines.
2. AppleTalk is a registered trademark of Apple Computer, Inc.
3. DDCMP is a trademark of Digital Equipment Corporation.

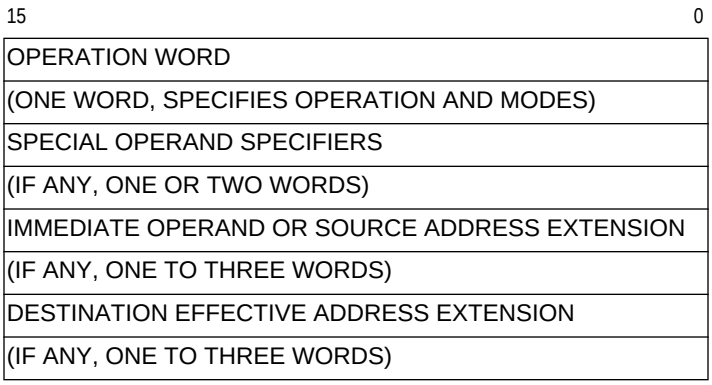


Figure 5-6. Instruction Word General Format

Besides the operation code, which specifies the function to be performed, an instruction defines the location of every operand for the function. Instructions specify an operand location in one of three ways:

- Register Specification — A register field of the instruction contains the number of the register.
- Effective Address — An effective address field of the instruction contains address mode information.
- Implicit Reference — The definition of an instruction implies the use of specific registers.

The register field within an instruction specifies the register to be used. Other fields within the instruction specify whether the register is an address or data register and how it is to be used. The M68000PM/AD, *M68000 Family Programmer's Reference Manual*, contains detailed register information.

Except where noted, the following notation is used in this section:

Data	Immediate data from an instruction
Destination	Destination contents
Source	Source contents
Vector	Location of exception vector
An	Any address register (A7–A0)
Ax, Ay	Address registers used in computation
Dn	Any data register (D7–D0)
Rc	Control register (VBR, SFC, DFC)
Rn	Any address or data register
Dh, Dl	Data registers, high- and low-order 32 bits of product
Dr, Dq	Data registers, division remainder, division quotient
Dx, Dy	Data registers, used in computation
Dym, Dyn	Data registers, table interpolation values

Xn	Index register
[An]	Address extension
cc	Condition code
d#	Displacement
	Example: d_{16} is a 16-bit displacement
⟨ea⟩	Effective address
#⟨data⟩	Immediate data; a literal integer
label	Assembly program label
list	List of registers
	Example: D3–D0
[...]	Bits of an operand
	Examples: [7] is bit 7; [31:24] are bits 31–24
(...)	Contents of a referenced location
	Example: (Rn) refers to the contents of Rn
CCR	Condition code register (lower byte of SR)
	X—extend bit
	N—negative bit
	Z—zero bit
	V—overflow bit
	C—carry bit
PC	Program counter
SP	Active stack pointer
SR	Status register
SSP	Supervisor stack pointer
USP	User stack pointer
FC	Function code
DFC	Destination function code register
SFC	Source function code register
+	Arithmetic addition or postincrement
–	Arithmetic subtraction or predecrement
/	Arithmetic division or conjunction symbol
×	Arithmetic multiplication
=	Equal to

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.

NOTES

At system reset, the SIM60 has a higher priority (at the same interrupt level) than the CPM. This priority can be changed if the SIM60 value is written to be less than 8, the level of the CPM.

No two modules are allowed to have the same interrupt arbitration value.

Assuming that the PIT wins the arbitration process, the SIM60 places the PIT 8-bit vector on the bus. The SWT also has a user-defined interrupt vector. The PIT and SWT interrupts do not allow autovectors. The IRQx lines can be vectored (externally supplied) or autovectored.

Arbitration for servicing interrupts is controlled by the value programmed into the interrupt arbitration (IARB) field of the MCR. Because no two modules are allowed to share the same IARB value and the only other module that generates interrupts (the CPM) has a fixed IARB value (IARB = 8), the SIM60 IARB value should be programmed to a value between 1 and 7 or between 9 and 15.

The autovector register (AVR) contains bits that correspond to external interrupt levels that require an autovector response. The SIM60 supports up to seven discrete external interrupt requests. If the bit corresponding to an interrupt level is set in the AVR, the SIM60 returns an internal autovector in response to the interrupt acknowledge cycle servicing that external interrupt request. Otherwise, external circuitry must either return an interrupt vector or externally assert the external $\overline{\text{AVEC}}$ signal.

See 6.8.4 Interrupts in Slave Mode for more information.

6.3.1.2 SIMULTANEOUS SIM60 INTERRUPT SOURCES. If the possible level 7 interrupt sources in the SIM60 are simultaneously asserted, the SIM60 will prioritize and service the interrupts in the following order: 1) SWT, 2) PIT, and 3) external interrupts. At level 6 or less, the PIT is higher than an external interrupt request asserted at the same level as the PIT.

6.3.1.2.1 Bus Monitor. The bus monitor ensures that each bus cycle is terminated within a reasonable period of time. It continually checks the duration of the internal/external $\overline{\text{AS}}$ line ($\overline{\text{TS}}$ for 68040). $\overline{\text{AS}}$ is normally negated by $\overline{\text{DSACK}}$x, $\overline{\text{BERR}}$, ($\overline{\text{TA}}$ or $\overline{\text{TEA}}$ for 68040).or $\overline{\text{HALT}}$ (or $\overline{\text{AVEC}}$ during an interrupt acknowledge cycle) The bus monitor asserts $\overline{\text{BERR}}$ if the response time is excessive on any bus cycle including interrupt acknowledge cycles. The BME bit in the SYPCR enables the bus monitor.

The bus cycle termination response time is measured in clock cycles, and the maximum-allowable response time is programmable. The bus monitor response time period ranges from 128 to 1K system clocks (see Table 6-5). The value chosen by the user should be larger than the longest possible response time of the slowest peripheral in the system.

6.3.1.2.2 Spurious Interrupt Monitor. In normal interrupt handling, one or more internal sub-modules recognize the CPU32+ interrupt acknowledge cycle as a signal that the CPU32+ is responding to their interrupt requests. The sub-modules then arbitrate for the privilege of returning a vector or asserting $\overline{\text{AVEC}}$ to the CPU32+. (The SIM60 also performs

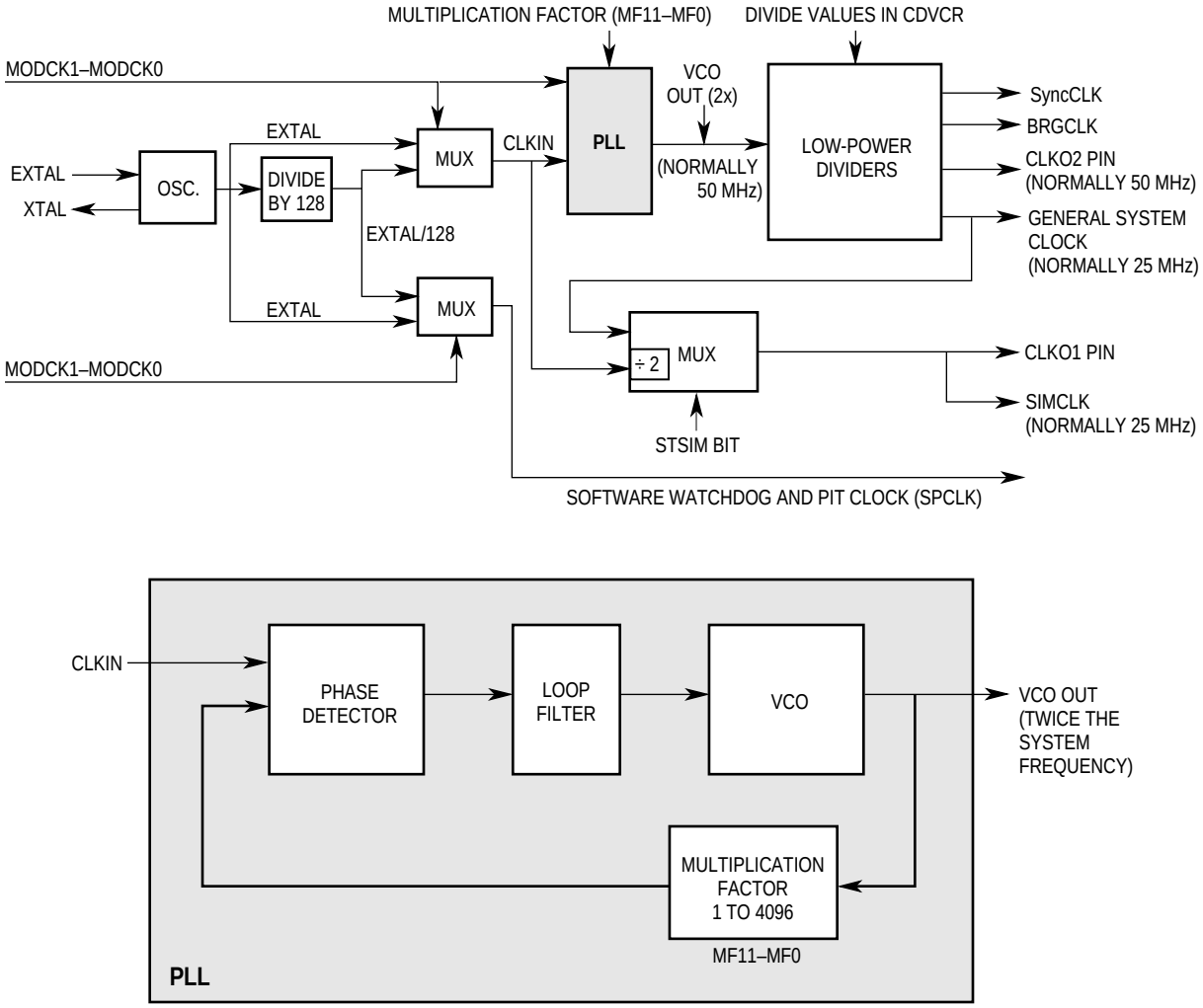


Figure 6-5. System Clocks Schematic

NOTE

User must select the proper MODCK configuration as well as correct XFC capacitor value when selecting the input clock frequency and desired system frequency.

For more information on MODCK see 6.5.8 Configuration Pins (MODCK1–MODCK0). For more information on XFC see 10.8 External Capacitor for PLL.

6.5.2 Oscillator Prescaler (Divide by 128)

In some applications, the use of a ~ 32-kHz crystal is attractive because of cost and low power, but is not attractive due to the extra startup time required for such a slow frequency crystal. Therefore, the SIM60 has an option for the user to provide a higher frequency crystal (for instance, in the ~ 4-MHz range) and divide it by 128 (back down to the 32-kHz range) before it is used by the QUICC. This results in much faster startup time than a 32-kHz crystal, plus low cost (if a common 4.192-MHz frequency is chosen), with only a small impact on power consumption during low-power modes (since the ~4-MHz frequency is immediately

NOTES

The 16-bit data bus is available on the D16–D31 pins. Dynamic bus sizing for 8- and 16-bit ports is possible with a 16-bit data bus.

PRTY3 has a small internal pullup to pull a floating PRTY3 signal high. Thus, the default condition of the QUICC provides a full 32-bit data bus, with 8-, 16-, and 32-bit dynamic bus sizing possible.

6.7.3 Port E

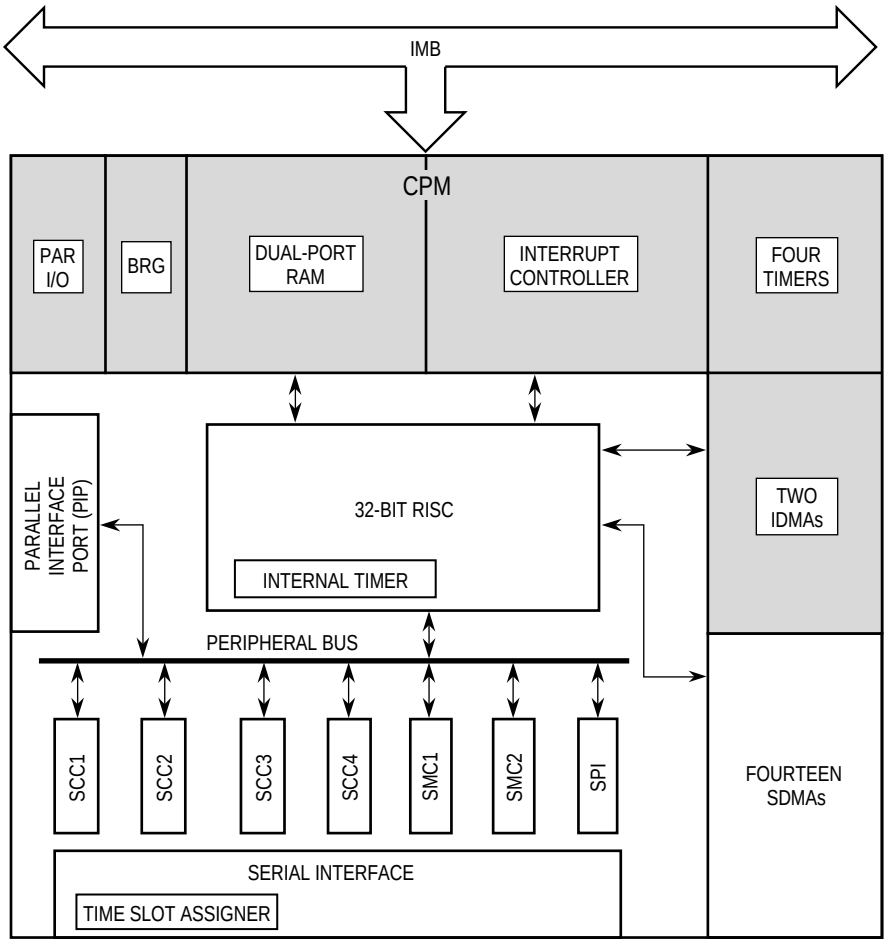
Port E pins can be independently programmed to select a number of system bus signal alternatives, including $\overline{CAS}$ lines, $\overline{WE}$ lines, $\overline{OE}$ lines, $\overline{IACK}$ lines, etc. The port E pin assignment register controls the function of the port E pins. See 6.9.4 Port E Pin Assignment Register (PEPAR).

6.8 SLAVE (DISABLE CPU32+) MODE

In this mode, the CPU32+ core on the QUICC is disabled, and the QUICC functions as an intelligent peripheral. Slave mode is enabled during system (or power-up) reset by the configuration of the CONFIG pins. In slave mode, the IDMAs and SDMAs on a QUICC can still obtain ownership of the system bus, even through the CPU32+ core is disabled. The slave mode has several common uses:

1. A multiple QUICC system. One QUICC in the system works normally with its CPU32+ enabled. It is called the system master. The rest of the QUICCs are used in slave mode as peripherals, with their CPUs disabled. The slaves would have their CONFIG pins configured to 110.
2. MC68040 companion mode. The QUICC operates solely as a peripheral to the MC68EC040 processor (or other M68040 family member). In this case, the QUICC provides a two-chip MC68EC040 system solution. One benefit of this configuration is that the QUICC memory controller can provide DRAM control for the MC68EC040 that includes MC68EC040 bursting support. In an MC68EC040+QUICC system, the QUICC's CONFIG pins would normally be set to 011 for a 32-bit boot ROM.
3. MC68040 companion mode with multiple QUICCs. In this case, multiple QUICCs can be slaves to a single MC68EC040. The user then chooses one of the QUICCs to provide the DRAM control for the MC68EC040 as well as the other QUICCs. In this case, the MC68EC040 access to the DRAM is not slowed down by the relatively slower QUICC accesses. The first QUICC in the system would have its CONFIG pins set to 011, and the other QUICCs added to that system would have their CONFIG pins set to 110.
4. QUICC is a slave to the MC68030. The QUICC operates as a peripheral to the MC68EC030 processor (or other MC68030 family member). The QUICC's standard slave mode is used since its bus is an MC68030-type bus. The QUICC does not support MC68030 burst accesses. In an MC68EC030+QUICC system, the QUICC's CONFIG pins could be set to 000, 001, or 010, depending on the boot ROM size.

- Serial Peripheral Interface (SPI) for Synchronous Interchip Communication
- Fourteen Serial Direct Memory Access (SDMA) Channels Support the SCC, SMCs, and SPI
- Two Independent Direct Memory Access (IDMA) Channels Support External Memory and Peripherals
- A Command Set Register Supports the RISC, IDMA, SCCs, SMCs, and SPI
- Four General-Purpose 16-Bit Timers or Two 32-Bit Timers
- Internal Timers to Implement Up to 16 Additional Timers
- General-Purpose Parallel Port for Parallel Protocols such as Centronics (Can Also Be Used as Standard Parallel I/O)
- CPM Interrupt Controller
- 2.5-kbyte Dual-Port RAM
- Twelve Parallel I/O Lines with Interrupt Capability



NOTE: The term "CP" refers to the nonshaded portion of the CPM.

Figure 7-1. CPM Block Diagram

8. CC3 Rx
9. SCC3 Tx
10. SCC4 Rx
11. SCC4 Tx
12. SMC1 Rx
13. SMC1 Tx
14. SMC2 Rx
15. SMC2 Tx
16. SPI Rx
17. SPI Tx
18. PIP
19. RISC Timer Tables

The RISC controller has an option to execute microcode from a portion of user RAM, located in the on-chip dual-port RAM. In this mode, either 512 bytes or 1024 bytes of the user RAM cannot be accessed by the host or another bus master and are used exclusively by the RISC. In this mode, the RISC controller can fetch instructions from both the dual-port RAM and its private ROM. This mode allows Motorola to add new protocols or enhancements to the QUICC in the form of Motorola-supplied RAM microcodes. The binary microcode is obtained from Motorola and then loaded by the user into the dual-port RAM.

The RISC controller contains one configuration register described in the following paragraph.

7.1.1 RISC Controller Configuration Register (RCCR)

The 16-bit, memory-mapped, read-write RCCR is used to configure the RISC processor and controls the RISC internal timer. This register is initialized to zero at reset. Bits 0-7 should not be modified unless the user is downloading a Motorola-supplied RAM microcode package..

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME	—	TIMEP						RESERVED							

TIME—Timer Enable

This bit enables the RISC controller internal timer. The timer will generate a tick to the RISC based on the value programmed into the TIMEP bit. TIME may be modified at any time to start or stop the scanning of the RISC timer tables.

Bit 14—Reserved

TIMEP—Timer Period

This field controls the RISC controller timer tick. The RISC timer tables are scanned on each timer tick. The input to this timer tick generator is the general system clock divided by 1024. The formula is (TIMEP + 1) × 1024 = (general system clock period). Thus, a val-

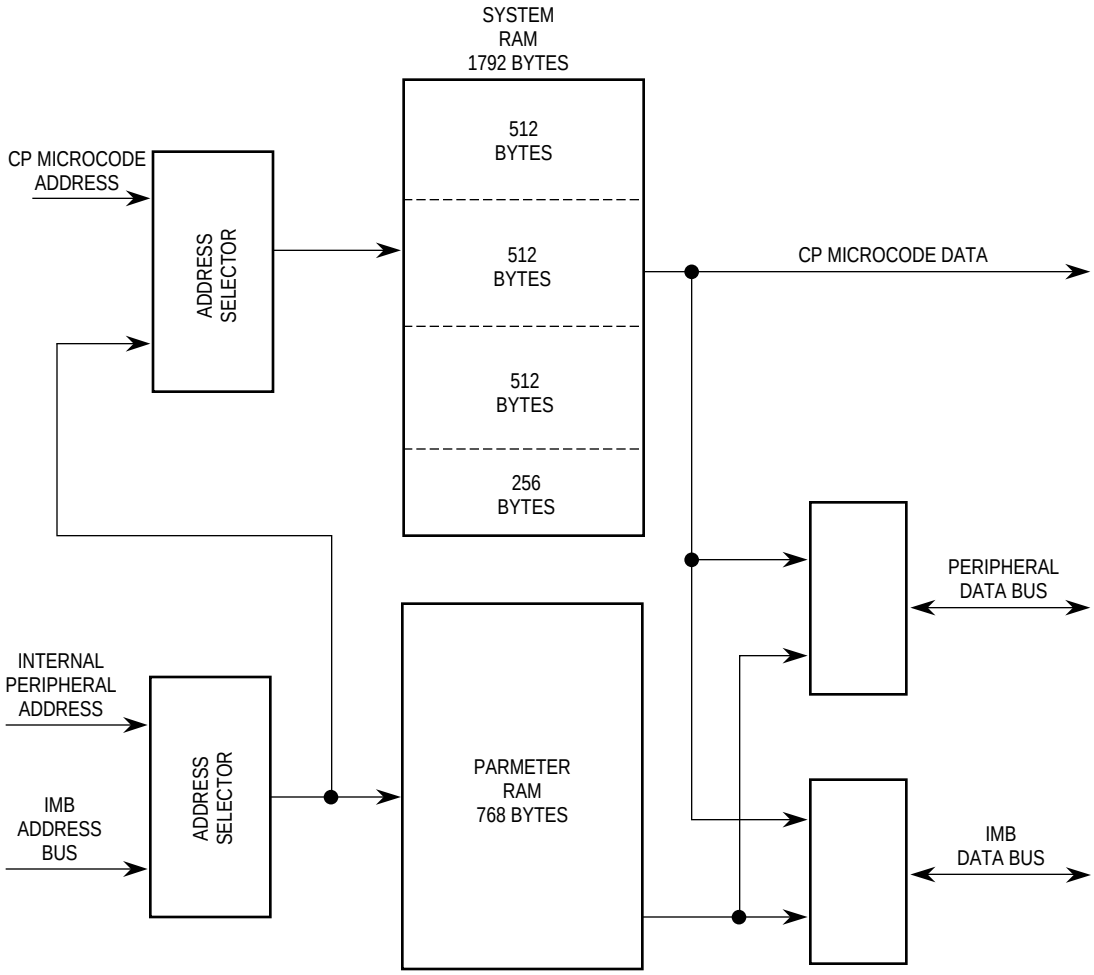


Figure 7-3. Dual-Port RAM Block Diagram

The dual-port RAM can be accessed by the RISC or one of four bus masters: CPU32+ core, IDMAs, SDMAs, or external bus master. When the dual-port RAM is accessed by an external bus master, CPU32+ core, IDMA, or SDMA channel, it is accessed in three clocks. When the dual-port RAM is accessed by the RISC, it is accessed in one clock. In the case of simultaneous access (with at least one write operation), the RISC is delayed by one clock.

When the dual-port RAM is accessed by the CPU32+ core, IDMAs, SDMAs, or external bus master, the data and address are taken from the IMB. The data is then presented on the IMB data bus. The RISC has access to the entire dual-port RAM for data fetches and portions of the system RAM for microcode instruction fetches.

The dual-port RAM is used for five possible tasks; any two tasks can occur simultaneously. The first use is to store parameters associated with the SCCs, SMCs, SPI, and IDMAs in the 768-byte parameter RAM. The second use is to store the buffer descriptors that describe where data is to be received and transmitted from. The third use is to store data from the serial channels. This usage is optional since data may also be stored externally in the system memory. The fourth use is to store RAM microcode for the RISC processor. This feature allows additional protocols to be added by Motorola in the future. The fifth use is for additional scratchpad RAM space for the user program.

FEx—Frame Sync Edge for TDM A or B

The L1RSYNCx and L1TSYNCx pulses are sampled with the falling/rising edge of the channel clock according to this bit.

0 = Falling edge (Use for IDL and GCI.)
1 = Rising edge

GMx—Grant Mode for TDM A or B

0 = GCI/SCIT mode. The GCI/SCIT D channel grant mechanism for transmission is internally supported. The grant is one bit from the receive channel. This bit is marked by programming the channel select bits of the SI RAM with 111 to assert an internal strobe on it. Refer to 7.8.7.2.2 SCIT Programming.
1 = IDL mode. A GRANT mechanism is supported if the corresponding GR1–GR4 bits in the SIMODE register are set. The grant is a sample of the L1GRx pin while L1TSYNCx is asserted. This GRANT mechanism implies the IDL access controls for transmission on the D channel. Refer to 7.8.6.2 IDL Interface Programming.

TFSDx—Transmit Frame Sync Delay for TDM A or B

These two bits determine the number of clock delays between the transmit sync and the first bit of the transmit frame. If the CRTx bit is set (recommended with IDL or GCI), then the transmit sync is not used, and these bits are ignored.

00 = No bit delay (The first bit of the frame is transmitted/received on the same clock as the sync.)
01 = 1 bit delay
10 = 2 bit delay
11 = 3 bit delay

Refer to Figure 7-29 and Figure 7-30 for an example of the use of these bits.

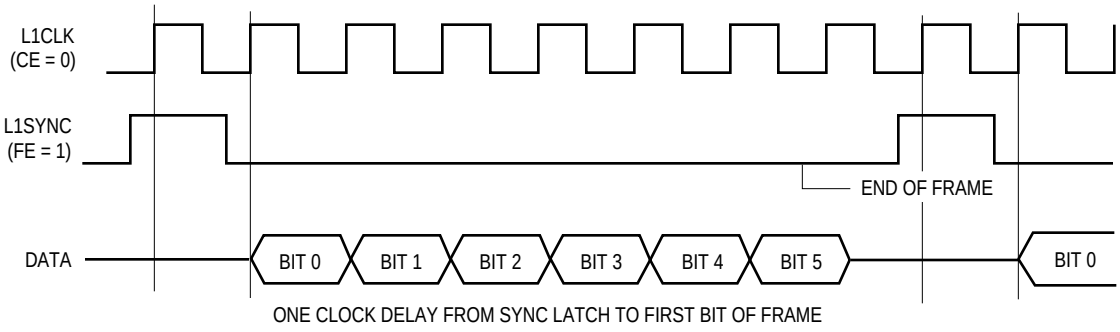


Figure 7-29. One Clock Delay from Sync to Data (RFSD = 01)

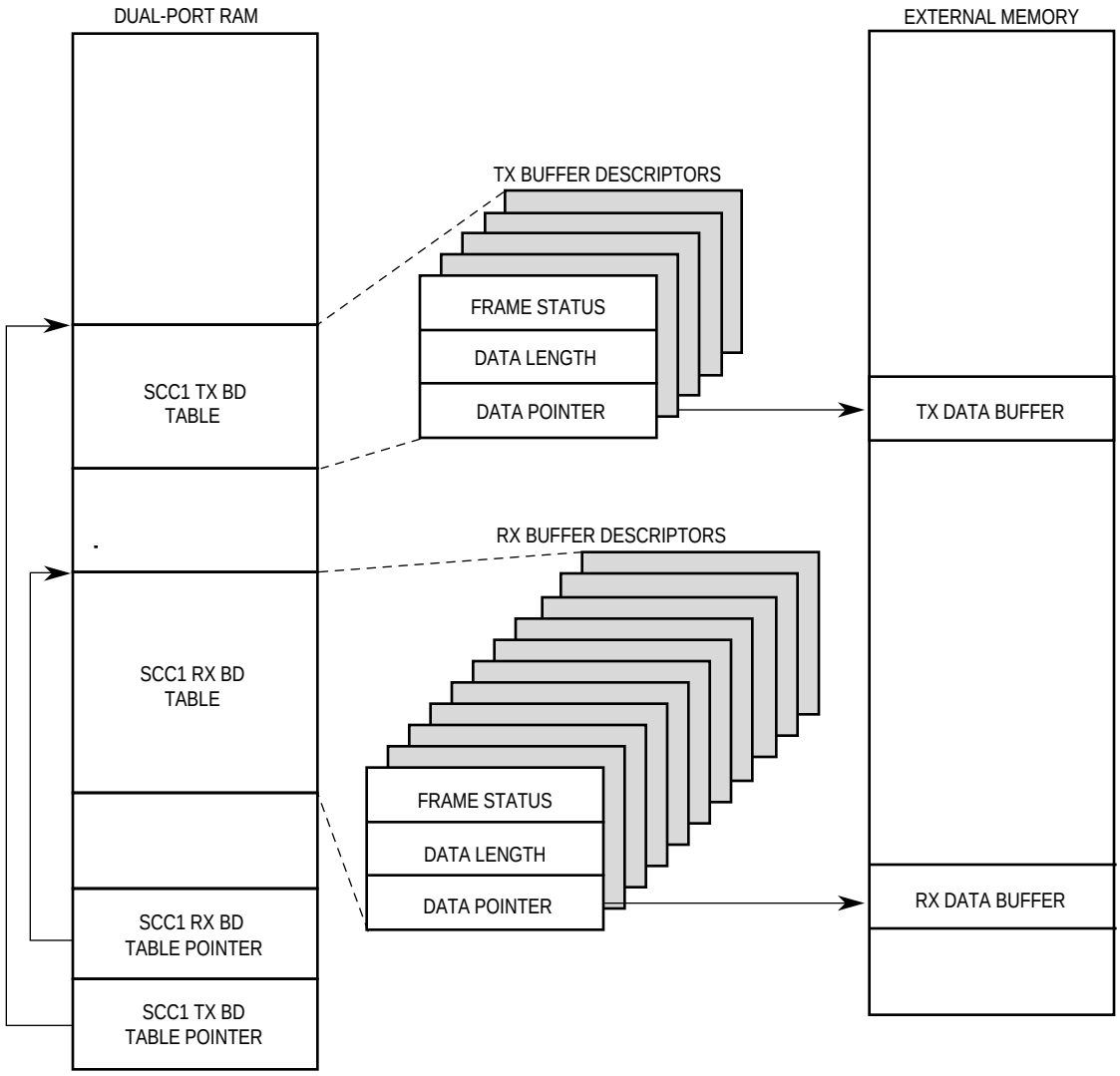


Figure 7-38. SCC Memory Structure

All protocols can have their buffer descriptors point to data buffers that are located in internal dual-port RAM. Typically, however, due to the internal RAM being used for buffer descriptors, it is customary for the data buffers to be located in external RAM, especially if the data buffers are large in size. In all cases, the IMB is used to transfer the data to the data buffer.

The CP processes the Tx BDs in a straightforward fashion. Once the transmit side of an SCC is enabled, it starts with the first BD in that SCC's transmit table. Once the CP detects that the Tx BD R-bit was set, it will begin processing the buffer. (The CP will detect that the BD is ready either by polling the R-bit periodically or by the user writing to the transmit-on-demand register (TODR).) Once the data from the BD has been placed in the transmit FIFO, the CP moves on to the next BD, again waiting for that BD's R-bit to be set. Thus, the CP does no look-ahead BD processing, nor does it skip over BDs that are not ready. When the CP sees the wrap (W) bit set in a BD, it goes back to the beginning of the BD table after processing of the BD is complete. After using a BD, the CP normally sets the R-bit to not-

If $\overline{CTS}$ is already asserted when $\overline{RTS}$ is asserted, transmission begins in two additional bit times. If $\overline{CTS}$ is not already asserted when $\overline{RTS}$ is asserted and CTSS = 0, then transmission begins in three additional bit times. If $\overline{CTS}$ is not already asserted when $\overline{RTS}$ is asserted and CTSS = 1, then transmission begins in two additional bit times.

7.10.12 Digital Phase-Locked Loop (DPLL)

DPLL data encoding and DPLL operations are discussed in the following paragraphs.

7.10.12.1 DATA ENCODING. Each SCC contains a DPLL unit that may be programmed to encode and decode the SCC data as NRZ, NRZI Mark, NRZI Space, FM0, FM1, Manchester, and Differential Manchester. Examples of the different encoding methods are shown in Figure 7-43.

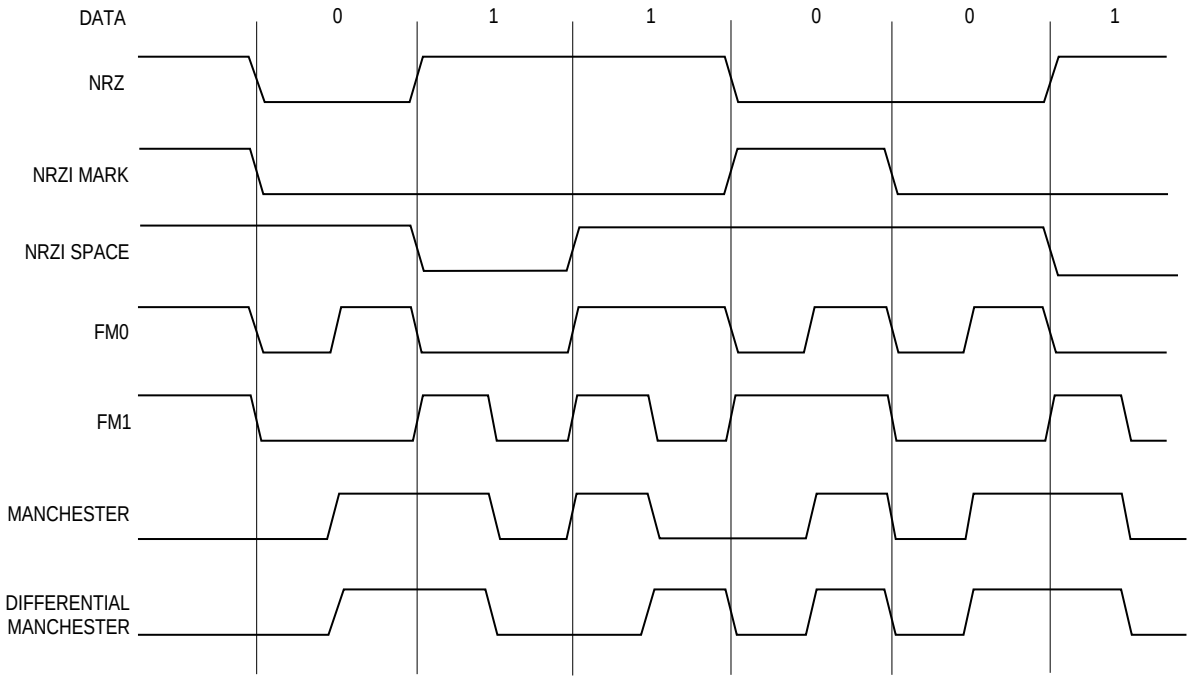


Figure 7-43. DPLL Encoding Examples

If it is not desired to use the DPLL, the NRZ coding may be chosen by the user in the GSMR. The definition of the encodings are as follows:

```
    NRZ             A 1 is represented by a high level for the
duration of the bit.
                    A 0 is represented by a low level for the duration
of the bit.
    NRZI Mark       A 1 is represented by no transition at all.
                    A 0 is represented by a transition at the
beginning of the bit
                    (i.e., the level present in the preceding bit is
reversed).
```

NRZI SpaceA 1 is represented by a transition at the beginning of the bit
(i.e., the level present in the preceding bit is reversed).

Bits 10–9—Should be written with zeros.

A—Address

When working in a multidrop configuration, the user should include the address bit in this position.

CHARSEND

This value contains the character to be transmitted. Any 5-, 6-, 7-, or 8-bit character value may be transmitted in accordance with the UART's configuration. The character should comprise the LSBs of CHARSEND. This value may be modified only while the REA bit is cleared.

7.10.16.11 SEND BREAK (TRANSMITTER). A break is an all-zeros character without a stop bit(s). A break is sent by issuing the STOP TRANSMIT command. The UART completes transmission of any outstanding data, sends a programmable number of break characters according to the break count register, and then reverts to idle or sends data if the RESTART TRANSMIT command was given before completion. At the completion of the break code, the transmitter sends at least one high bit before transmitting any data to guarantee recognition of a valid start bit.

The break characters do not preempt characters already in the transmit FIFO. This means that the break character may not be transmitted for 8 character times (SCC1) or 4 character times (SCC2, SCC3, and SCC4). To reduce this latency, the TFL bit in the GSMR should be set to decrease the FIFO size to one character prior to enabling the SCC transmitter.

7.10.16.12 SENDING A PREAMBLE (TRANSMITTER). A preamble sequence gives the programmer a convenient way of ensuring that the line goes idle before starting a new message. The preamble sequence length is constructed of consecutive ones of one character length. If the preamble bit in a BD is set, the SCC will send a preamble sequence before transmitting that data buffer. Example: for 8 data bits, no parity, 1 stop bit, and 1 start bit, a preamble of 10 ones would be sent before the first character in the buffer.

7.10.16.13 FRACTIONAL STOP BITS (TRANSMITTER). The asynchronous UART transmitter can be programmed to transmit fractional stop bits. Four bits in the SCC data synchronization register (DSR) are used to program the length of the last stop bit transmitted. These DSR bits may be modified at any time. If two stop bits are transmitted, only the second one is affected. Idle characters are always transmitted as full-length characters.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	FSB				1	1	0	0	1	1	1	1	1	1	0

NOTE

The definition of what constitutes a late collision is made in the LCW bit in the PSMR.

Heartbeat. Some transceivers have a self-test feature called “heartbeat” or “signal quality error.” To signify a good self-test, the transceiver indicates a collision to the QUICC within 20 clocks after completion of a frame transmitted by the Ethernet controller. This indication of a collision does not imply a real collision error on the network, but is rather an indication that the transceiver still seems to be functioning properly. This is called the heartbeat condition.

If the HBC bit is set in the Ethernet mode register and the heartbeat condition is not detected by the QUICC after a frame transmission, then a heartbeat error occurs. When this error occurs, the channel closes the buffer, sets the HB bit in the Tx BD, and generates the TXE interrupt if it is enabled.

7.10.23.16.2 Reception Errors. The following paragraphs describe various types of Ethernet reception errors.

Overrun Error. The Ethernet controller maintains an internal FIFO for receiving data. If a receiver FIFO overrun occurs, the channel writes the received data byte to the internal FIFO over the previously received byte. The previous data byte and the frame status are lost. The channel closes the buffer, sets the OV bit in the Rx BD, sets RXF in the Ethernet event register, and increments the discarded frame counter (DISFC). The receiver then enters the hunt mode.

Busy Error. A frame was received and discarded due to lack of buffers. The channel sets BSY in the Ethernet event register and increments the discarded frame counter (DISFC).

Nonoctet Error (Dribbling Bits). The Ethernet controller can handle up to seven dribbling bits when the receive frame terminates nonoctet aligned. The Ethernet controller checks the CRC of the frame on the last octet boundary. If there is a CRC error, then the frame nonoctet aligned (NO) error is reported, the RXF bit is set, and the alignment error counter (ALEC) is incremented. If there is no CRC error, then no error is reported.

CRC Error. When a CRC error occurs, the channel closes the buffer, sets the CR bit in the Rx BD, and sets the RXF bit. The channel also increments the CRC error counter (CRCEC). After receiving a frame with a CRC error, the receiver enters hunt mode. CRC checking cannot be disabled, but the CRC error may be ignored if checking is not required.

7.10.23.17 ETHERNET MODE REGISTER (PSMR). The Ethernet mode register is a 16-bit, memory-mapped, read-write register that controls the SCC operation. The term Ethernet mode register refers to the PSMR of the SCC when that SCC is configured for Ethernet. This register is cleared at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBC	FC	RSH	IAM	CRC		PRO	BRO	SBT	LPB	SIP	LCW	NIB			FDE

20. Clear GADDR1–GADDR4. The group hash table is not used.
21. Write PADDR1_H with $0000, PADDR1_M with $0000, and PADDR1_L with $0040 to configure the physical address.
22. Write P_Per with $0000. It is not used.
23. Clear IADDR1–IADDR4. The individual hash table is not used.
24. Clear TADDR_H, TADDR_M, and TADDR_L for the sake of clarity.
25. Initialize the Rx BD. Assume the Rx data buffer is at $00001000 in main memory. Write $B000 to Rx_BD_Status. Write $0000 to Rx_BD_Length (not required—done for instructional purposes only). Write $00001000 to Rx_BD_Pointer.
26. Initialize the Tx BD. Assume the Tx data frame is at $00002000 in main memory and contains fourteen 8-bit characters (destination and source addresses plus the type field). Write $FC00 to Tx_BD_Status. Add PAD to the frame and generate a CRC. Write $000D to Tx_BD_Length. Write $00002000 to Tx_BD_Pointer.
27. Write $FFFF to the SCCE to clear any previous events.
28. Write $001A to the SCCM to enable the TXE, RXF, and TXB interrupts.
29. Write $40000000 to the CIMR to allow SCC1 to generate a system interrupt. (The CICR should also be initialized.)
30. Write $00000000 to GSMR_H1 to enable normal operation of all modes.
31. Write $1088000C to GSMR_L1 to configure the $\overline{\text{CTS}}$ (CLSN) and $\overline{\text{CD}}$ (RENA) pins to automatically control transmission and reception (DIAG bits) and the Ethernet mode. TCI is set to allow more setup time for the EEST to receive the QUICC's transmit data. TPL and TPP are set as required for Ethernet. The DPLL is not used with Ethernet. Notice that the transmitter (ENT) and receiver (ENR) have not been enabled yet.
32. Write $D555 to DSR
33. Set the PSMR1 to $0A0A to configure 32-bit CRC, promiscuous mode (receive all frames), and begin searching for the start frame delimiter 22 bits after RENA.
34. Enable the TENA pin ($\overline{\text{RTS}}$). Since the MODE bits in GSMR have been written to Ethernet, the TENA signal is low. Write PCPAR bit 0 with a one. Write PCDIR bit 0 with a zero.
35. Write $1088003C to GSMR_L1 to enable the SCC1 transmitter and receiver. This additional write ensures that the ENT and ENR bits will be enabled last.

NOTE

After 14 bytes and the 46 bytes of automatic pad (plus the 4 bytes of CRC) have been transmitted, the Tx BD is closed. Additionally, the receive buffer is closed after a frame is received. Any additional receive data beyond 1520 bytes or a single frame will cause a busy (out-of-buffers) condition since only one Rx BD was prepared.

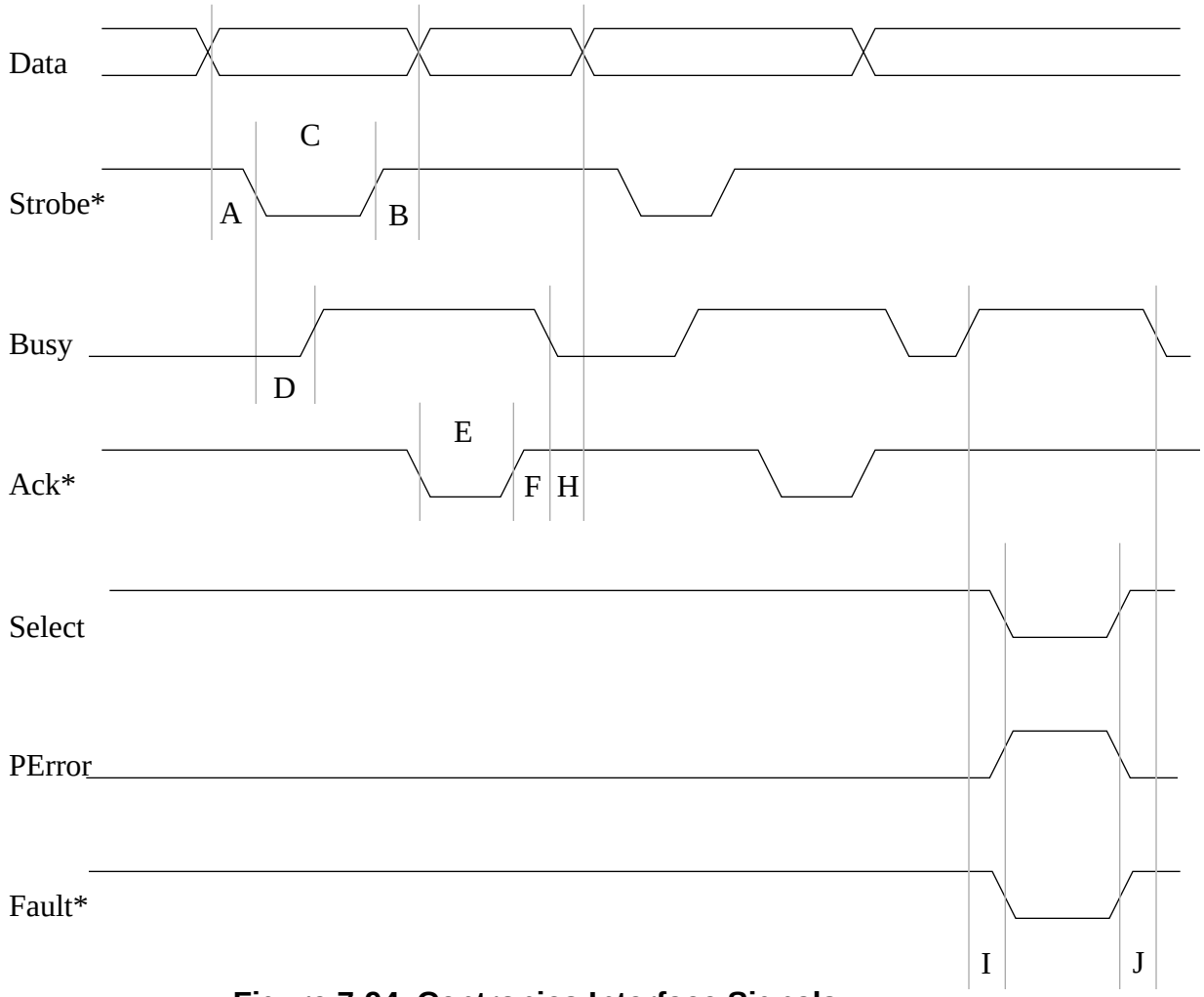


Figure 7-94. Centronics Interface Signals

The Centronics controller can be operated as a host port (transmitter), peripheral port (receiver), or can support bidirectional data transfer using some s/w support and a combination of the two modes.

Table 7-23. Encoding the Interrupt Vector

Interrupt Number	Interrupt Source Description	Lower 5 Bits of Vector
1F	Parallel I/O—PC0	11111
1E	SCC1	11110
1D	SCC2	11101
1C	SCC3	11100
1B	SCC4	11011
1A	Parallel I/O—PC1	11010
19	Timer 1	11001
18	Parallel I/O—PC2	11000
17	Parallel I/O—PC3	10111
16	SDMA Channel Bus Error	10110
15	IDMA1	10101
14	IDMA2	10100
13	Reserved	10011
12	Timer 2	10010
11	RISC Timer Table	10001
10	Reserved	10000
F	Parallel I/O—PC4	01111
E	Parallel I/O—PC5	01110
D	Reserved	01101
C	Timer 3	01100
B	Parallel I/O—PC6	01011
A	Parallel I/O—PC7	01010
9	Parallel I/O—PC8	01001
8	Reserved	01000
7	Timer 4	00111
6	Parallel I/O—PC9	00110
5	SPI	00101
4	SMC1	00100
3	SMC2 / PIP	00011
2	Parallel I/O—PC10	00010
1	Parallel I/O—PC11	00001
0	Error	00000

vector exists as the last entry in this table. The error vector is issued by the CPM if an interrupt was requested by the CPM but was masked by the user prior to being serviced by CPU32+ core and if no other pending interrupts for the CPM are present. The user should provide an error interrupt service routine, even if it is simply an RTE instruction.

follow. (In embedded control applications, most users leave the CPU in supervisor mode permanently.)

Step 3: Write the VBR

Many users initialize the ROM to contain an initial exception vector table. If so, the user should write the CPU vector base register (VBR) to point to the starting location of the table. (The exception vector table defines where to find the routines that handle interrupts, bus errors, traps, etc.) Note, however, that exceptions will not be ready to be handled properly until the stack pointer points to addressable RAM.

Step 4: Write the MBAR

The module base address register (MBAR), which always exists at a fixed address, determines where the 8-Kbyte block of QUICC internal RAM and internal peripherals are to be mapped. To put the 8-Kbyte block at $700000, write $00700001 to the MBAR. The MBAR must be accessed before any other QUICC internal peripheral or RAM location. Remember that the MBAR must be written in a special way as described in the Section 6 System Integration Module (SIM60). (If multiple QUICCs exist in the system, it may be necessary to write the MBARE before writing MBAR.)

Step 5: Verify a Dual-Port RAM Location

First, verify that the MBAR address was programmed correctly. This can be done by testing one of the dual-port RAM locations. Write $33 and $CC to location $700000 and verify that these values can be correctly read. Location $700000 is the beginning of the QUICC internal RAM.

Step 6: Is This a Power-Up Reset?

Next, determine the cause of the reset from the reset status register (RSR). The RSR is normally cleared by the user after it is read (ones are written to RSR). If this is not a power-up reset, the user may wish to take actions other than the ones listed or notify the system of the cause of an unexpected reset to aid in debugging. Many of the following steps are only necessary after a power-on reset. See Section 4 Bus Operation for more details on the effects of the different types of resets.

Step 7: Deal with the Clock Synthesizer

The next step is to set up the clock synthesizer, which is located in the SIM60. The three registers that control the clock synthesizer are the CLKOCR, the PLLCR, and the CDVCR.

If a low-speed external crystal is used (such as 32 kHz or 4 MHz), multiply the QUICC clock frequency up to the desired speed (e.g., 25 MHz). If an external oscillator is used to provide the exact system frequency, then this step is not needed.

The clock synthesizer has other options, such as SyncCLK and BRGCLK dividers, as well as the ability to divide the general system clock frequency. These are used in low-power applications. At this time, leave all these options in their default conditions. These options should only be enabled when the rest of the application code is in a more stable state.

Table C-3. ATOM1 Configuration

ATOM1 Channels	Risc Bandwidth Consumed (est)	Possible Configuration of Other Channels
1 x 6 Mbit/s with scrambling	90%	3 x 64Kbit HDLC or Transparent
1 x 10 Mbit/s non-scrambling	90%	2 x 64Kbit HDLC or Transparent
2 x 2 Mbit/s with scrambling	40%	2 x 2.5 Mbit HDLC or Transparent, 9.6 Kbit SMC UART
2 x 2 Mbit/s non-scrambling	40%	1 x 10 Mbit Ethernet, 9.6 Kbit SMC UART

C.4 ASYNCHRONOUS HDLC FOR PPP

Asynchronous HDLC is a frame-based protocol (see Figure 3), defined by the Internet Engineering Task Force (IETF) "Request For Comments #1549" which uses HDLC framing techniques in conjunction with UART-type characters. This protocol is typically used as the physical layer for the Point-to-Point (PPP) protocol. While this protocol can be implemented by the UART controller on the QUICC in conjunction with the CPU32+, it is more efficient and less compute-intensive for the CPU to allow the Communications Processor Module (CPM) of the QUICC to perform the framing and transparency functions of the protocol.

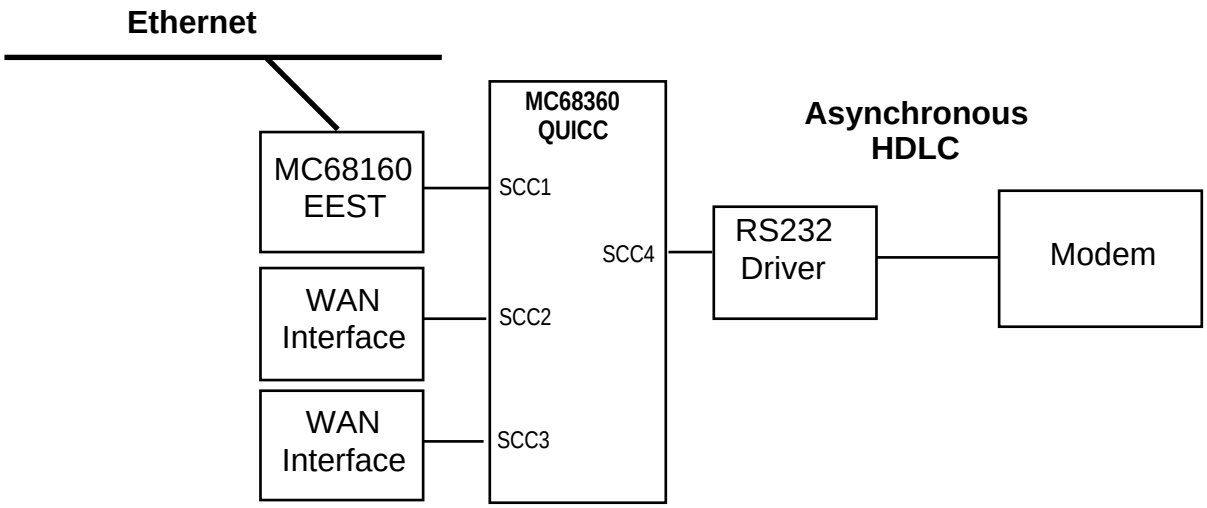


Figure C-3. Asynchronous HDLC Block Diagram

C.4.1 Key Features

- Flexible data buffer structure which allows an entire frame or a section of a frame to be transmitted and received.
- Separate interrupts for received frames and transmitted buffers.
- Automatic 16-bit CRC generation and checking (CRC-CCITT).
- Automatic generation of opening and closing flags.