Welcome to E-XFL.COM

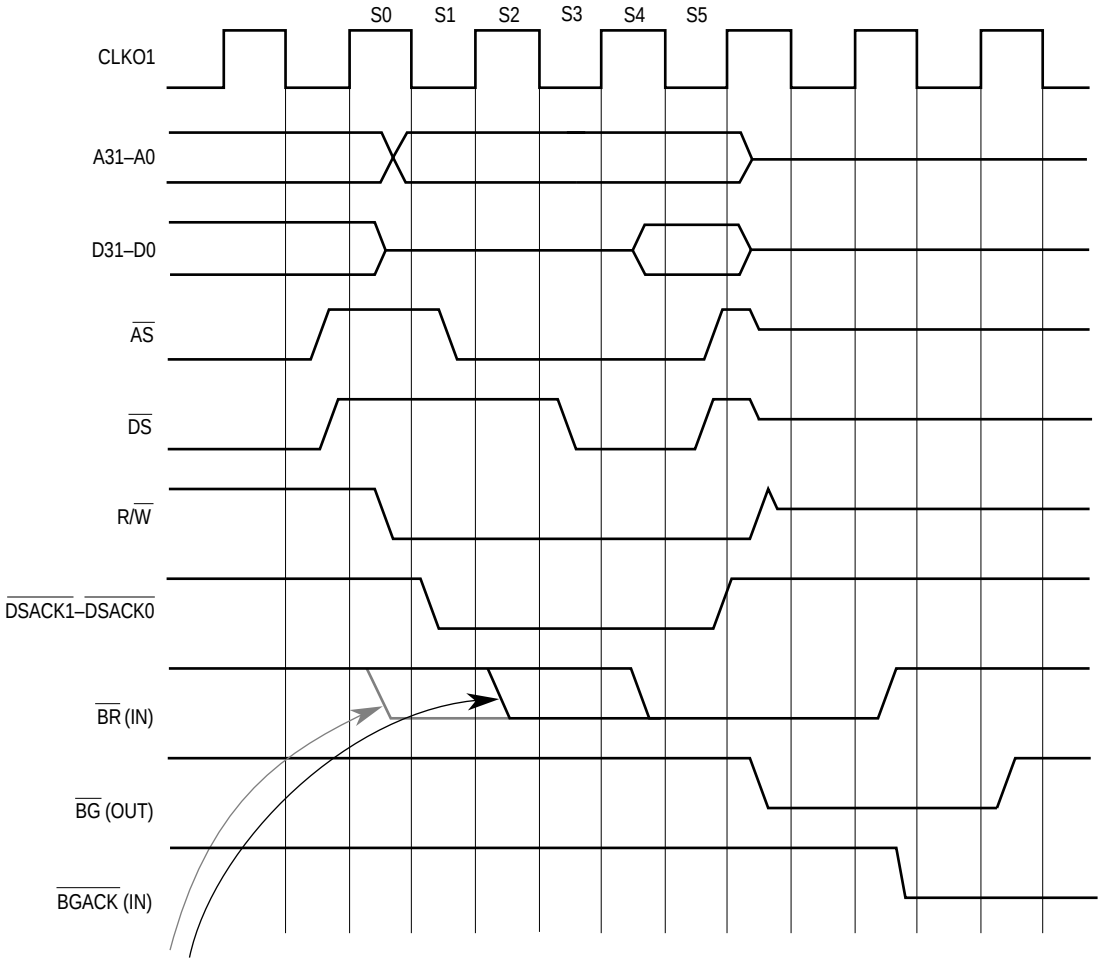
Understanding Embedded - Microprocessors

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	240-BFQFP
Supplier Device Package	240-FQFP (32x32)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68360em25vl



NOTE:
BR has synchronous timing.
BR has synchronous timing.

Figure 4-36. Bus Arbitration Timing Diagram—Active Bus Case

4.6.1 Bus Request

External devices capable of becoming bus masters request the bus by asserting BR. This signal can be wire-ORed to indicate to the QUICC that some external device requires control of the bus. The QUICC is effectively at a lower bus priority level than the external device and relinquishes the bus after it has completed the current bus cycle (if one has started). If no BGACK is received while the BR is active, the QUICC remains bus master once BR is negated. This prevents unnecessary interference with ordinary processing if the arbitration circuitry inadvertently responds to noise or if an external device determines that it no longer requires use of the bus before it has been granted mastership.

5.6 DEVELOPMENT SUPPORT

All M68000 family members have the following special features that facilitate applications development.

Trace on Instruction Execution—All M68000 processors include an instruction-by-instruction tracing facility to aid in program development. The MC68020, MC68030, and CPU32+ can also trace those instructions that change program flow. In trace mode, an exception is generated after each instruction is executed, allowing a debugger program to monitor execution of a program under test. See 5.5.2.10 Tracing for more information.

Breakpoint Instruction—An emulator can insert software breakpoints into target code to indicate when a breakpoint occurs. On the MC68010, MC68020, MC68030, and CPU32+, this function is provided via illegal instructions ($4848–$484F) that serve as breakpoint instructions. See 5.5.2.5 Software Breakpoints for more information.

Unimplemented Instruction Emulation—When an attempt is made to execute an illegal instruction, an illegal instruction exception occurs. Unimplemented instructions (F-line, A-line) utilize separate exception vectors to permit efficient emulation of unimplemented instructions in software. See 5.5.2.8 Illegal or Unimplemented Instructions for more information.

5.6.1 CPU32+ Integrated Development Support

In addition to standard MC68000 family capabilities, the CPU32+ has features to support advanced integrated system development. These features include background debug mode, deterministic opcode tracking, hardware breakpoints, and internal visibility in a single-chip environment.

5.6.1.1 BACKGROUND DEBUG MODE (BDM) OVERVIEW. Microprocessor systems generally provide a debugger, implemented in software, for system analysis at the lowest level. The BDM on the CPU32+ is unique because the debugger is implemented in CPU microcode.

BDM incorporates a full set of debug options—registers can be viewed and/or altered, memory can be read or written, and test features can be invoked.

A resident debugger simplifies implementation of an in-circuit emulator. In a common setup (see Figure 5-18), emulator hardware replaces the target system processor. A complex, expensive pod-and-cable interface provides a communication path between target system and emulator.

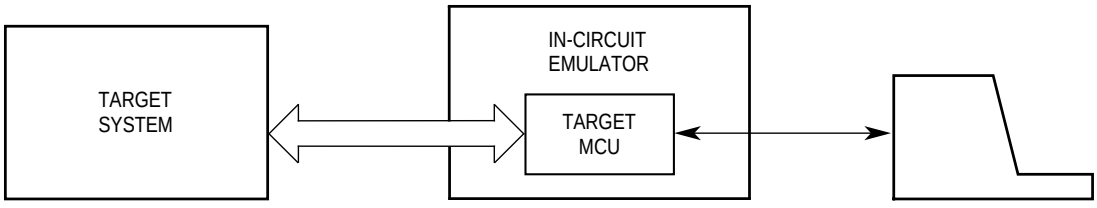


Figure 5-18. In-Circuit Emulator Configuration

"not ready/come again" response. Once the receive data latch has been loaded, the CPU is released to act on the new data. Response data overwrites the "not ready" response when the CPU has completed the current operation.

Data written into the output shift register appears immediately on the DSO signal. In general, this action changes the state of the signal from a high ("not ready" response status bit) to a low (valid data status bit) logic level. However, this level change only occurs if the command completes successfully. Error conditions overwrite the "not ready" response with the appropriate response that also has the status bit set.

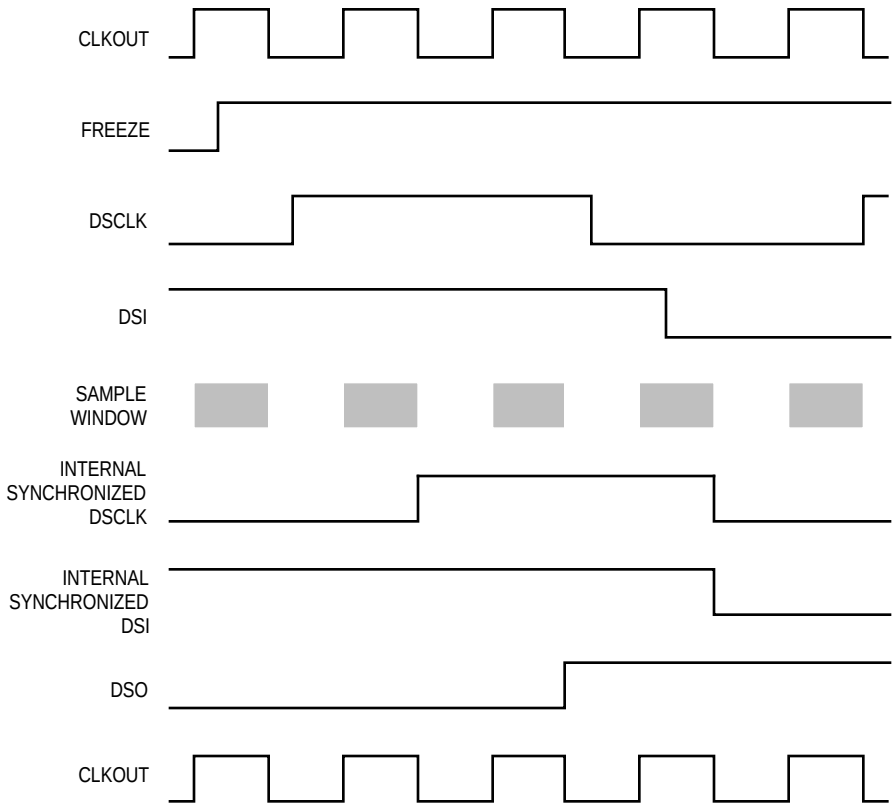


Figure 5-23. Serial InterfaceTiming Diagram

A user can use the state change on DSO to signal hardware that the next serial transfer may begin. A timeout of sufficient length to trap error conditions that do not change the state of DSO should also be incorporated into the design. Hardware interlocks in the CPU prevent result data from corrupting serial transfers in progress.

5.6.2.7.2 Development System Serial Logic. The development system, as the master of the serial data link, must supply the serial clock. However, normal and BDM operations could interact if the clock generator is not properly designed.

Breakpoint requests are made by asserting $\overline{\text{BKPT}}$ to the low state in either of two ways. The primary method is to assert $\overline{\text{BKPT}}$ during a single bus cycle for which an exception is desired. Another method is to assert $\overline{\text{BKPT}}$, then continue to assert it until the CPU32+ responds by asserting FREEZE. This method is useful for forcing a transition into BDM when

The general system clock can switch automatically from low to high frequency whenever one of the following conditions exists:

- The level of the pending or current interrupt is higher than the INTEN bits in CDVCR.
- The CPM RISC controller has a pending request or is currently executing a routine (i.e., it is not idle). This option is maskable by the RRQEN bit in CDVCR.

When neither of these conditions exists, the general system clock automatically switches back to the low frequency.

When the general system clock is divided, its duty cycle is changed. One phase remains the same (e.g., 20 ns @ 25 MHz); the other becomes longer. Note that the CLKO1 and CLKO2 pins no longer have a 50% duty cycle when the general system clock is divided (see Figure 6-8).

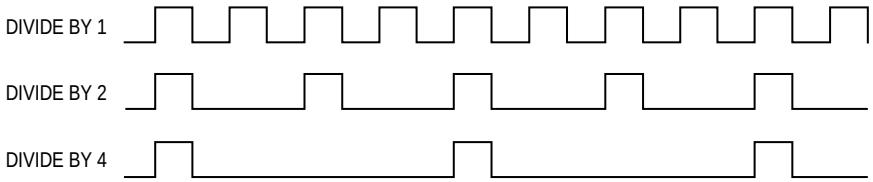


Figure 6-8. Divided Clocks

6.5.5.3 BRGCLK. The BRGCLK is used by the five CPM baud rate generators. There are four SCC/SCM baud rate generators and one SPI baud rate generator. BRGCLK defaults to VCO/2 = 25 MHz (assuming a 25-MHz system frequency). The purpose of BRGCLK is to allow the five baud rate generators to continue to operate at a fixed frequency, even when the rest of the QUICC is operating at a reduced frequency (i.e., the general system clock is divided). See 7.9 Baud Rate Generators (BRGs) for more information on how to save power using the BRGCLK.

NOTES

During early board prototyping, the user should leave BRGCLK at its standard frequency (e.g., 25 MHz) for the sake of simplicity.

Within the four SCC/SMC baud rate generators, the user should not use a baud rate generator divider equal to 1, unless the BRGCLK is at the maximum frequency.

6.5.5.4 SYNCCLK. The SyncCLK is used by the serial synchronization circuitry in the serial ports of the CPM, including the SI, SCCs, and SMCs. The SyncCLK performs the function of synchronizing externally generated clocks before they are used internally. SyncCLK defaults to VCO/2 = 25 MHz (assuming a 25-MHz system frequency).

The purpose of SyncCLK is to allow the SI, SCCs, and SMCs to continue to operate at a fixed frequency, even when the rest of the QUICC is operating at a reduced frequency. Thus, SyncCLK allows the user to maintain the serial synchronization circuitry at the desired rate, while lowering the general system clock to the lowest possible rate. However, the SyncCLK frequency must always be at least as high as the general system clock frequency.

the $\overline{RAS}$ signal of the DRAM bank. This $\overline{RAS}$ signal will remain active until another DRAM bank is accessed. The page size is determined by the PGS bits in the GMR.

If a different bank of DRAM is accessed, followed by an access to a DRAM bank on which page mode is selected, then the DRAM controller negates the $\overline{RAS}$ signal to the other bank and asserts the particular $\overline{RAS}$ line for the page mode bank, followed by the rest of the DRAM access. This is called a page mode normal cycle.

On each access to a DRAM bank in which the page mode is enabled and the previous DRAM cycle was to that bank, the address of the last access to this bank is compared to the current address. If the two addresses fall within the same page, then the access cycle begins immediately with the assertion of the column address and $\overline{CAS}$ signal. This is called a page hit.

In case of a page miss (the address of the last access and current address do not fall within the same page), the $\overline{RAS}$ signal must be negated and held high for a period that matches the value programmed in the WBTQ control field of the current DRAM region, and then a full cycle (including row and column phases) is executed. This is the slowest DRAM access since the $\overline{RAS}$ signal must first be negated, followed by the precharge time.

Since it is difficult to predict the performance impact of page mode, the user may wish to try the application software with and without page mode enabled, and compare the results. The ability to concentrate the code/data accesses into the same page of the DRAM is central to achieving a performance improvement.

Some systems will need an additional wait state to perform write cycles during a page hit. To gain a wait state, set the delay write cycle for the QUICC DWQ bit in the GMR of the DRAM bank.

NOTES

Page mode is supported only for the internal QUICC cycles or external MC68030/QUICC cycles.

If any two DRAM banks overlap each other in their address space, page mode must not be selected for either of those banks.

6.12.3 DRAM Burst Access Support

The DRAM controller supports burst accesses made by an external MC68EC040 (or other MC68040 family member) if the BACK40 bit is set in the BR. The MC68EC040 requests a burst to be performed with a line-fill indication on the SIZx pins ($\overline{SIZ}$ = 11) and the TTx pins. In this case, the DRAM controller performs a normal access ($\overline{RAS}$ and $\overline{CAS}$), followed by requests to the DRAM for the next three sequential long-word operands ($\overline{CAS}$ only). The DRAM controller automatically increments the addresses to the DRAM using the BADDR3–BADDR2 pins.

The length of an MC68EC040 burst cycle can be distinguished from the length of the initial access with the BCYC bits of the OR.

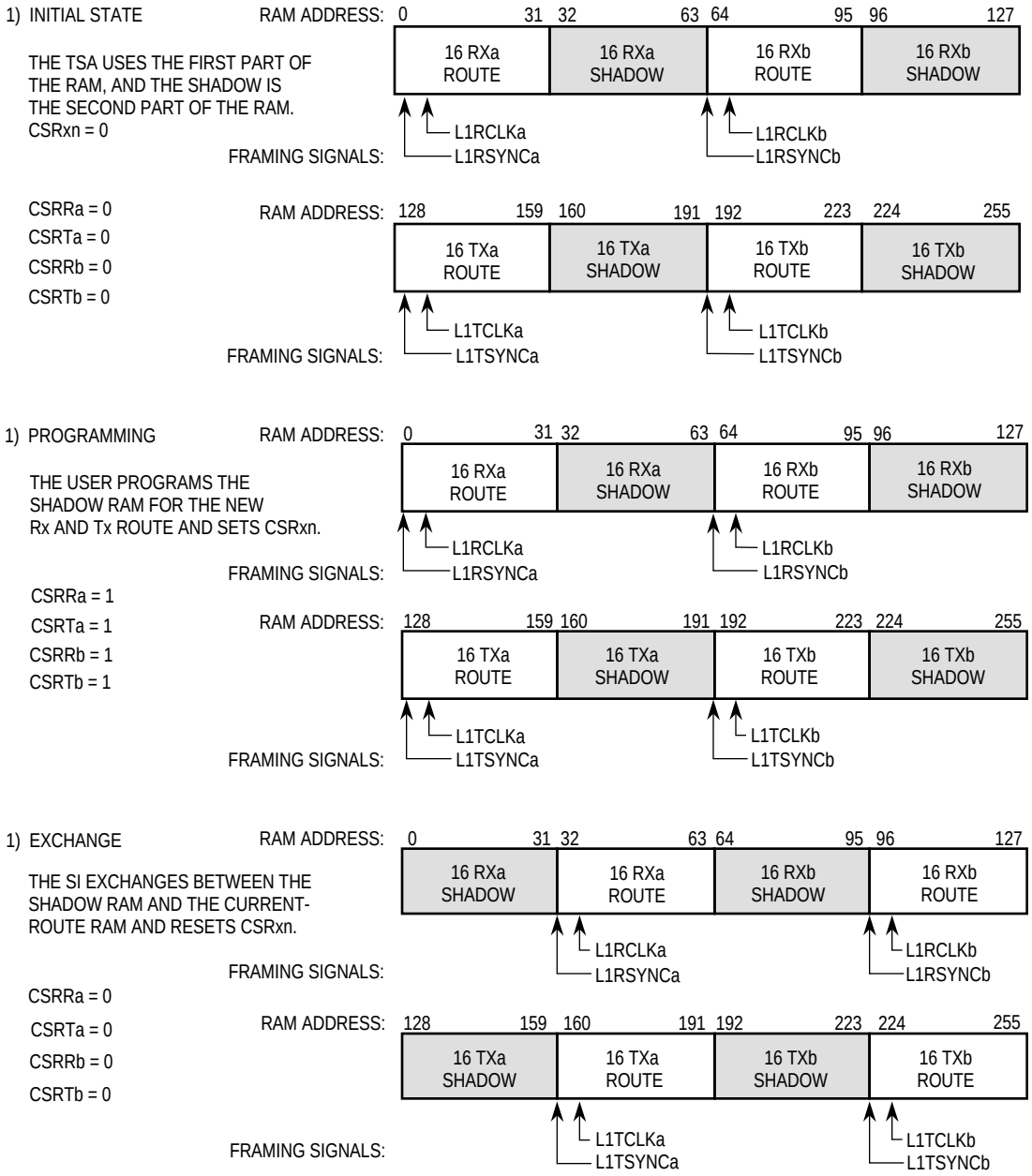


Figure 7-28. SI RAM Dynamic Changes

7.8.5 SI Registers

The following paragraphs describe the SI registers.

7.8.5.1 SI GLOBAL MODE REGISTER (SIGMR). The 8-bit SIGMR defines the RAM division modes. The SIGMR appears to the user as a memory-mapped, read-write register and is cleared at reset.

7	6	5	4	3	2	1	0
—				ENb	ENa	RDM1–RDM0	

Bits 7–4—Reserved

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

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

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

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

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

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

Table 7-8. HDLC-Specific Parameters

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

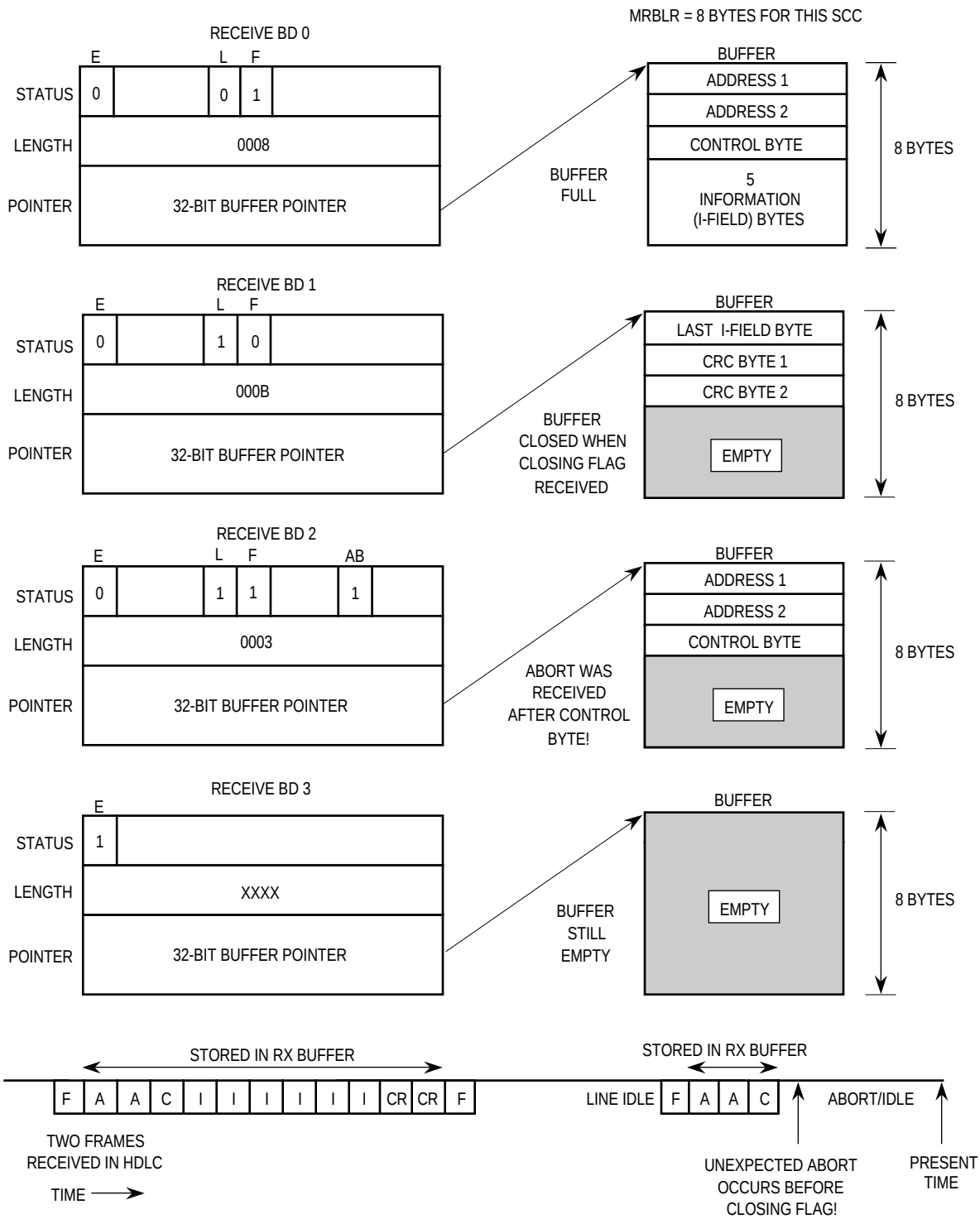


Figure 7-52. HDLC Rx BD Example

sage that was in progress when the command was issued. It will be set immediately if no message was in progress when the command was issued.

TXE—Tx Error

An error (CTS lost or underrun) occurred on the transmitter channel.

RCH—Receive Character

A character has been received and written to the buffer.

BSY—Busy Condition

A character was received and discarded due to lack of buffers. The receiver will resume reception after an ENTER HUNT MODE command.

TX—Tx Buffer

A buffer has been transmitted. This bit is set as the last bit of data or the BCS (if sent) begins transmission.

RX—Rx Buffer

A receive buffer has been closed by the CP on the BISYNC channel.

7.10.20.15 BISYNC MASK REGISTER (SCCM). The SCCM is referred to as the BISYNC mask register when the SCC is operating as a BISYNC controller. It is a 16-bit read-write register that has the same bit format as the BISYNC event register. If a bit in the BISYNC mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared upon reset.

7.10.20.16 SCC STATUS REGISTER (SCCS). The SCCS is an 8-bit read-only register that allows the user to monitor real-time status conditions on the RXD line. The real-time status of the $\overline{\text{CTS}}$ and $\overline{\text{CD}}$ pins are part of the port C parallel I/O.

7	6	5	4	3	2	1	0
—	—	—	—	—	—	CS	—

CS—Carrier Sense (DPLL)

This bit shows the real-time carrier sense of the line as determined by the DPLL if it is used.

0 = The DPLL does not sense a carrier.
1 = The DPLL does sense a carrier.

7.10.20.17 PROGRAMMING THE BISYNC CONTROLLER. There are two general techniques that the software may employ to handle data received by the BISYNC controllers. The simplest way is to allocate single-byte receive buffers, request (in the status word in each BD) an interrupt on reception of each buffer (i.e., byte), and implement the BISYNC protocol entirely in software on a byte-by-byte basis. This simple approach is flexible and may be adapted to any BISYNC implementation. The obvious penalty is the overhead caused by interrupts on each received character.

CRC_P**.** For the 16-bit CRC-CCITT, CRC_P should be initialized with $0000FFFF. For the 32-bit CRC-CCITT, CRC_P should be initialized with $FFFFFFFF. For the CRC-16, CRC_P should be initialized with ones ($0000FFFF) or zeros ($00000000).

CRC_C. For the 16-bit CRC-CCITT, CRC_C should be initialized with $0000F0B8. For the 32-bit CRC-CCITT, CRC_C should be initialized with $DEBB20E3. For the CRC-16 which is normally used with BISYNC, CRC_C should be initialized with $00000000.

NOTE

This value overlaps with the CRC constant (mask) for the HDLC-based protocols. This overlap is not detrimental since the CRC constant (mask) is only used for the receiver; thus, only one entry is required. Therefore, the user may choose HDLC transmitter with a transparent receiver or a transparent transmitter with an HDLC receiver.

7.10.21.6 TRANSPARENT COMMAND SET. The following transmit and receive commands are issued to the CR.

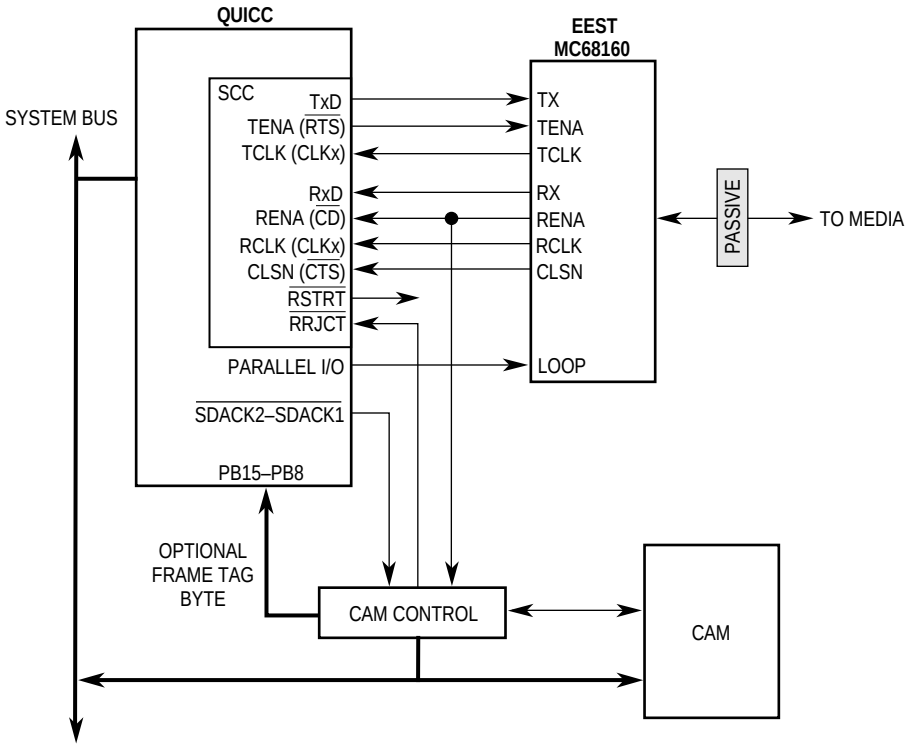
7.10.21.6.1 Transmit Commands. The following paragraphs describe the transparent transmit commands.

STOP TRANSMIT Command**.** After a hardware or software reset and the enabling of the channel in the SCC mode register, the channel is in the transmit enable mode and starts polling the first BD in the table every 64 clocks (immediately if the TOD bit in the TODR is set).

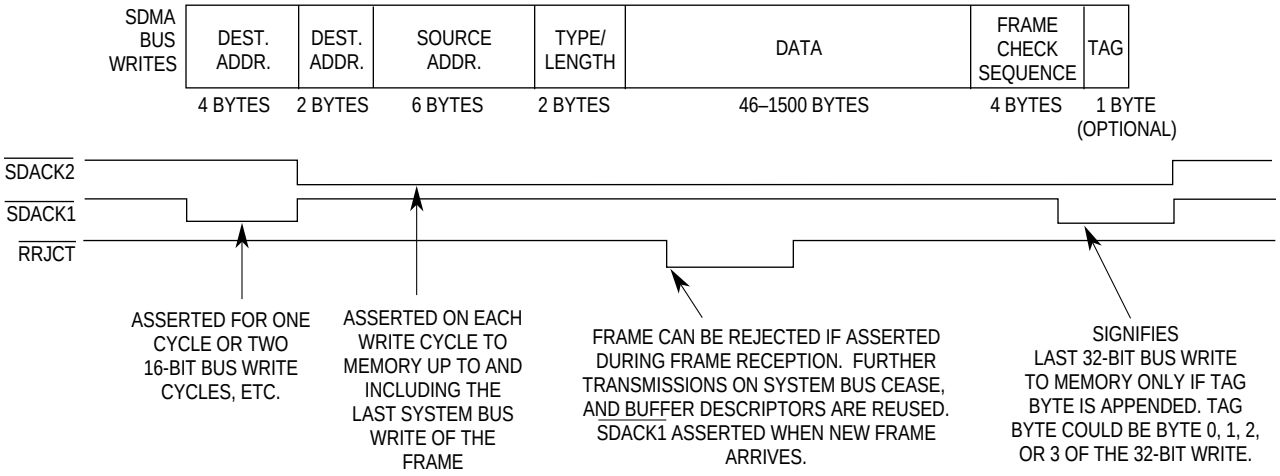
The STOP TRANSMIT command disables the transmission of frames on the transmit channel. If this command is received by the transparent controller during frame transmission, transmission of that buffer is aborted after a maximum of 64 additional bits are transmitted, and the transmit FIFO is flushed. The TBPTR is not advanced, no new BD is accessed, and no new buffers are transmitted for this channel. The transmitter will send idles.

GRACEFUL STOP TRANSMIT Command**.** The GRACEFUL STOP TRANSMIT command is used to stop transmission in an orderly way, rather than abruptly as performed by the regular STOP TRANSMIT command. It stops transmission after the current frame has completed transmission, or immediately if there is no frame being transmitted. (A transparent frame is not complete until a BD with the L-bit set has its associated buffer completely transmitted.) The GRA bit in the SCCE will be set once transmission has stopped. After transmission ceases, the transmit parameters, including BDs, may be modified. The TBPTR will point to the next Tx BD in the table. Transmission will begin once the R-bit of the next BD is set and the RESTART TRANSMIT command is issued.

RESTART TRANSMIT Command**.** The RESTART TRANSMIT command reenables the transmission of characters on the transmit channel. This command is expected by the transparent controller after a STOP TRANSMIT command, after a STOP TRANSMIT command and disabling the channel in its SCC mode register, after a GRACEFUL STOP TRANSMIT



NOTE: The receive data is sent to the CAM as it is written to system memory. The SDACK2–SDACK1 signals are used to identify the destination address and any other frame bytes desired. The RSTRT signal is not required in this configuration, although it is still available.



NOTE: The diagram shows SDMA system bus writes, not data on the RXD pin. Other bus activity may occur between successive 32-bit writes. In such a case, the SDACK2–1 pins would not be asserted for other bus activity.

Figure 7-69. QUICC Ethernet Parallel CAM Interface;

7.10.23.8 ETHERNET MEMORY MAP. When configured to operate in Ethernet mode, the QUICC overlays the structure described in Table 7-5 onto the protocol-specific area of the SCC parameter RAM described in Table 7-11.

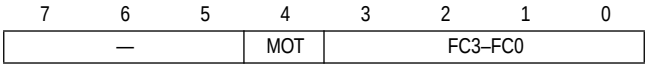
sponding channel. Furthermore, the user should not configure BD tables of two enabled SMCs to overlap, or erratic operation will occur.

NOTE

RBASE and TBASE should contain a value that is divisible by 8.

7.11.4.2 SMC FUNCTION CODE REGISTERS (RFCR, TFCR). There are four separate function code registers for the two SMC channels: two for receive data buffers (RFCRx) and two for transmit data buffers (TFCRx). The FC entry contains the value that the user would like to appear on the function code pins FC3–FC0 when the associated SDMA channel accesses memory. It also controls the byte-ordering convention to be used in the transfers.

Receive Function Code Register



Bits 7–5—Reserved

MOT—Motorola

This bit should be set by the user to achieve normal operation. MOT *must be set* if the data buffer is located in external memory and has a 16-bit wide memory port size.

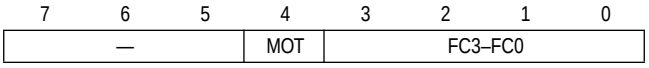
0 = DEC (and Intel) convention is used for byte ordering—swapped operation. It is also called little-endian byte ordering. The bytes stored in each buffer word are reversed as compared to the Motorola mode.

1 = Motorola byte ordering—normal operation. It is also called big-endian byte ordering. As data is received from the serial line and put into the buffer, the most significant byte of the buffer word contains data received earlier than the least significant byte of the same buffer word.

FC3–FC0—Function Code 3–0

These bits contain the function code value used during this SDMA channel's memory accesses. The user should write bit FC3 with a one to identify this SDMA channel access as a DMA-type access. Example: FC3–FC0 = 1000 (binary). Do not write the value 0111 (binary) to these bits.

Transmit Function Code Register



Bits 7–5—Reserved

C/I DATA—Command/Indication Data Bits

C/I DATA is a 4-bit data field for C/I channel 0 and a 6-bit data field for C/I channel 1. It contains the data received from the C/I channel. For C/I channel 0, bits 5-2 contain the 4-bit data field, and bits 7 and 6 are always written with zeros. For C/I channel 1, bits 7-2 contain the 6-bit data field..

7.11.14.8 SMC C/I CHANNEL TRANSMIT BUFFER DESCRIPTOR (TX BD). The CP reports information about the C/I channel transmit byte using the BD.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	—							C/I DATA						—	

R—Ready

0 = This bit is cleared by the CP after transmission to indicate that the BD is now available to the CPU32+ core.

1 = This bit is set by the CPU32+ core to indicate that the data associated with this BD is ready for transmission.

Bits 14–6—Reserved

These bits should be cleared by the user.

C/I DATA—Command/Indication Data Bits

C/I DATA is a 4-bit data field for C/I channel 0 and a 6-bit data field for C/I channel 1. It contains the data to be transmitted onto the C/I channel. For C/I channel 0, bits 5-2 contain the 4-bit data field, and bits 7 and 6 are always written with zeros. For C/I channel 1, bits 7-2 contain the 6-bit data field.

7.11.14.9 SMC EVENT REGISTER (SMCE). The SMCE is an 8-bit register used to report events recognized by the SMC channel and to generate interrupts. On recognition of event, the SMC sets its corresponding bit in this register. Interrupts generated by this register may be masked in the SMC mode register.

The SMCE is a memory-mapped register that may be read at any time. A bit is cleared by writing a one (writing a zero does not affect a bit's value). More than one bit may be cleared at a time. All unmasked bits must be cleared before the CP will clear the internal interrupt request to the CPM interrupt controller. This register is cleared at reset.

7	6	5	4	3	2	1	0				
—	—	—	—	CTXB	CRXB	MTXB	MRXB				
INITIAL VALUE:						0	0	0	0		

Bits 7–4—Reserved

CTXB—C/I Channel Buffer Transmitted

The C/I transmit buffer became empty.

It is important for the designer to distinguish the difference between an MC68EC030 accessing memory synchronously and the QUICC memory controller operating in synchronous mode. The MC68EC030 has the ability to access memory in standard asynchronous bus cycles (i.e., three clocks or longer) and synchronous bus cycles (two clocks). For MC68EC030 asynchronous accesses, the QUICC memory controller can generate chip selects, $\overline{\text{DSACK}}$x, etc. However, the QUICC does not support MC68EC030 synchronous bus cycles (nor does it support MC68EC030 bursting). This does not mean that the MC68EC030 cannot perform two-clock accesses in a QUICC system—only that the QUICC will not assist (i.e., generate chip selects, etc.) during these accesses.

The QUICC memory controller can operate asynchronously or synchronously (defined by the BSTM bit in the MCR and the SYNC bit in the GMR). When the QUICC memory controller is operating synchronously, the external signals being monitored are not synchronized internally by the QUICC, and must be provided with the proper setup and hold times. When the synchronous function is not enabled, externally generated bus signals are latched on the negative edge of the QUICC clock before being recognized, permitting external signals to be completely asynchronous to the QUICC clock. The QUICC memory controller is normally used in synchronous mode with the MC68EC030, even if the MC68EC030 is generating only asynchronous bus cycles.

The QUICC clocking section allows for the clock oscillator to be kept running through the VDDSYN pin in a power-down situation, if desired. Low-power issues are not addressed.

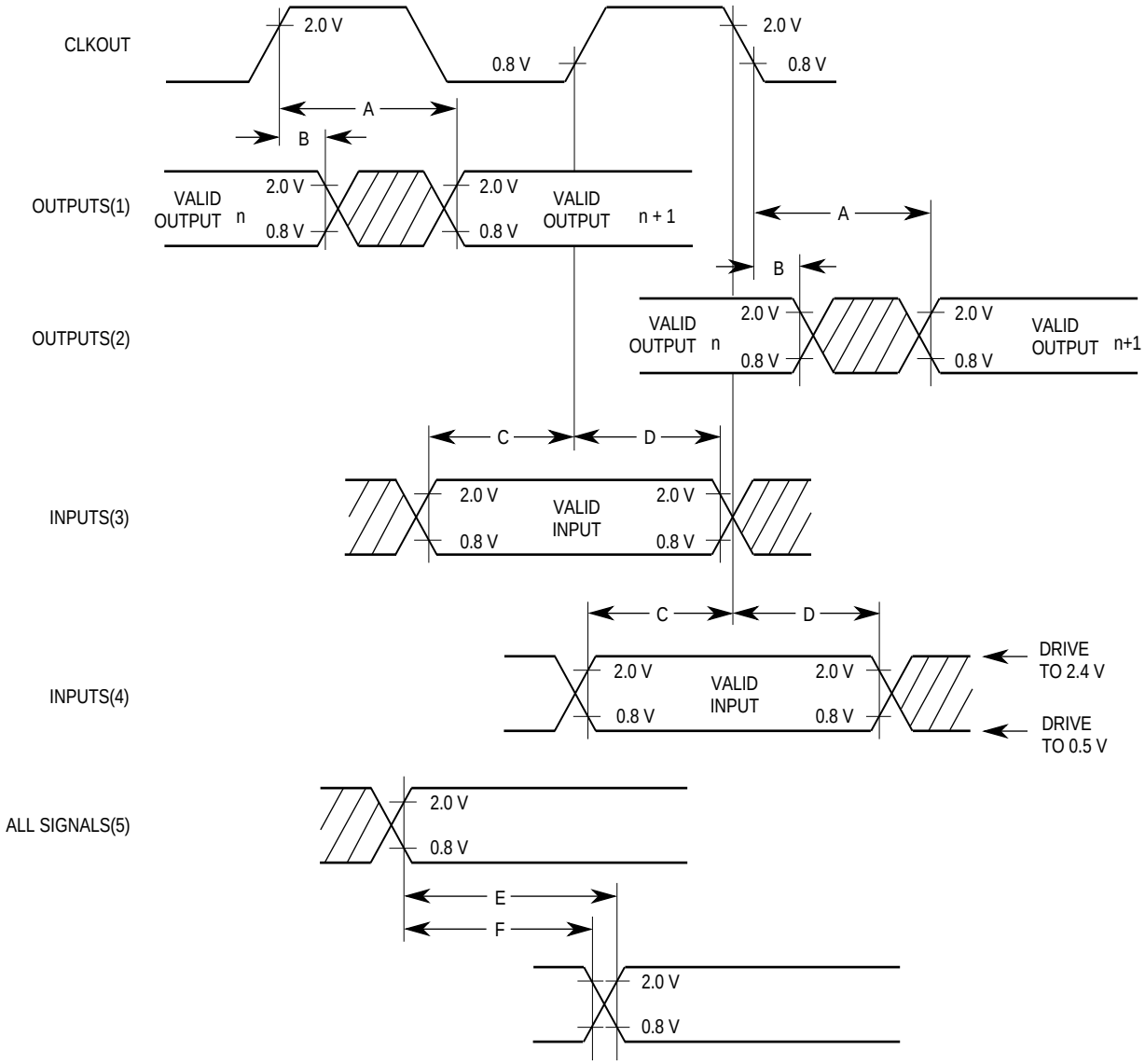
9.8.1.3 RESET STRATEGY. If a QUICC is configured to provide the global chip select, it will also provide an internal power-on reset generation. Thus, the $\overline{\text{RESETH}}$ pin of the QUICC just needs to be connected to the $\overline{\text{RESET}}$ pin on the MC68EC030. If a pushbutton switch is needed, it can be connected by an open-drain buffer to the $\overline{\text{RESET}}$ line, once debounced.

9.8.1.4 INTERRUPTS. External interrupts may be brought into the QUICC through either the $\overline{\text{IRQ}}$x pins or parallel I/O pins. The QUICC prioritizes these interrupts with its own internally generated interrupts (e.g., timers) to obtain the current highest pending request. In slave mode, the QUICC can output this request to another processor in the system.

The request can take the form of a single request pin or three request pins ($\overline{\text{IOUT2}}$–$\overline{\text{IOUT0}}$) that encode the priority of the request. Since the MC68EC030 uses encoded inputs ($\overline{\text{IPL2}}$–$\overline{\text{IPL0}}$), the $\overline{\text{IOUT2}}$–$\overline{\text{IOUT0}}$ pins are chosen.

In addition, the QUICC allows the $\overline{\text{IOUT2}}$–$\overline{\text{IOUT0}}$ pins to be generated in two different ways: at the expense of parity pins or at the expense of some interrupt requests. In this application, parity is not used in the system; thus, the parity pins are chosen for this function, leaving more interrupt pins available.

Once the MC68EC030 recognizes the interrupt, it responds with an interrupt acknowledge cycle. The QUICC recognizes the MC68EC030 interrupt acknowledge cycle using the address and function code pins (FC3–FC0). The QUICC then responds by placing the vector on the bus or by outputting the $\overline{\text{AVECO}}$ signal to the MC68EC030, depending on what was programmed into the autovector register in the SIM60.



NOTES:
1. This output timing is applicable to all parameters specified relative to the rising edge of the clock.
2. This output timing is applicable to all parameters specified relative to the falling edge of the clock.
3. This input timing is applicable to all parameters specified relative to the rising edge of the clock.
4. This input timing is applicable to all parameters specified relative to the falling edge of the clock.
5. This timing is applicable to all parameters specified relative to the assertion/negation of another signal.

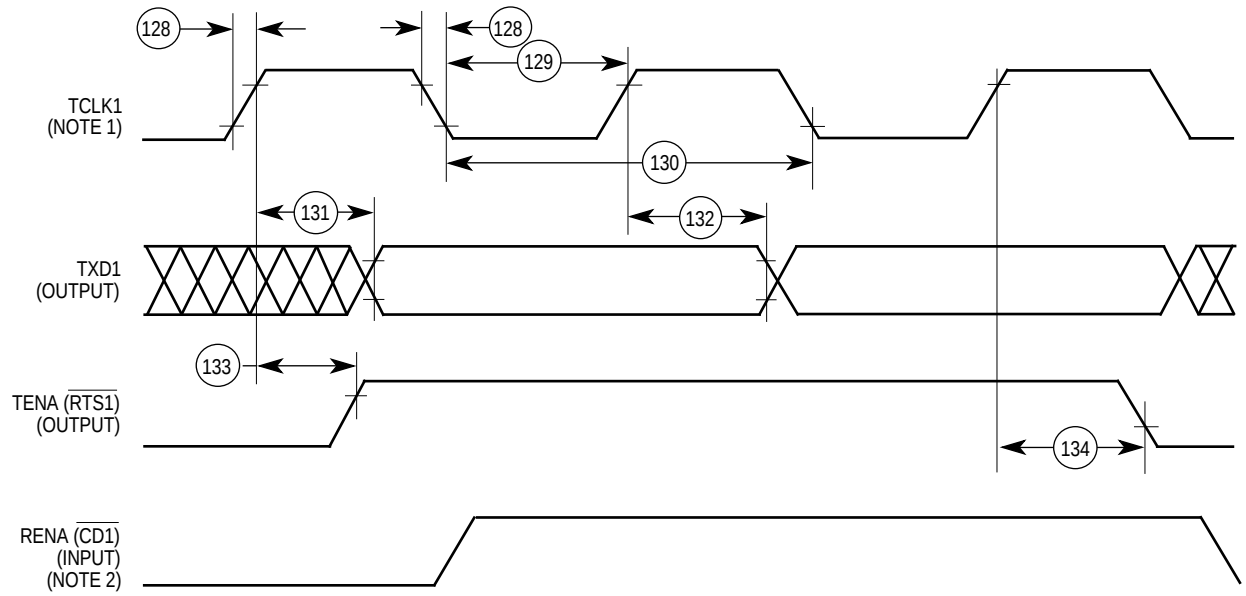
LEGEND:
A. Maximum output delay specification.
B. Minimum output hold time.
C. Minimum input setup time specification.
D. Minimum input hold time specification.
E. Signal valid to signal valid specification (maximum or minimum).
F. Signal valid to signal invalid specification (maximum or minimum).

Figure 10-1. Drive Levels and Test Points for AC Specifications

10.9BUS OPERATION AC TIMING SPECIFICATIONS (CONTINUED

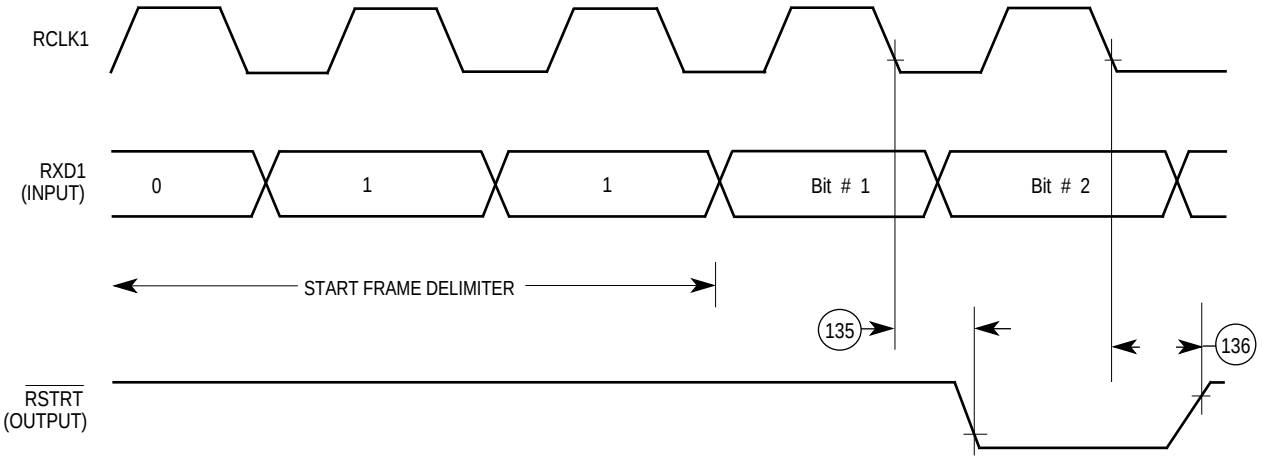
Num.	Characteristic	Symbol	3.3 V/5.0 V 25.0 MHz		5.0V 33.34MHz		Unit
			Min	Max	Min	Max	
88	CLKO1 High to IFETCH High Impedance	t_{IFZ}	0	35	0	26.25	ns
89	CLKO1 High to IFETCH Valid	t_{IF}	0	35	0	26.25	ns
90	CLKO1 High to PERR Asserted	t_{CHPA}	0	20	0	15	ns
91	CLKO1 High to PERR Negated	t_{CHPN}	0	20	0	15	ns
92	Minimum Vcc Ramp-Up Time At Power-On Reset	t_{RMIN}	5	-	5	-	ms

NOTES:
1. All AC timing is shown with respect to 0.8 V and 2.0 V levels unless otherwise noted.
2. This number can be reduced to 5 ns if strobes have equal loads.
3. If multiple chip selects are used, the CS width negated (#15) applies to the time from the negation of a heavily loaded chip select to the assertion of a lightly loaded chip select.
4. These hold times are specified with respect to DS or CS on asynchronous reads and with respect to CLKO1 on fast termination reads. The user is free to use either hold time for fast termination reads.
5. If the asynchronous setup time (#47) requirements are satisfied, the DSACK low to data setup time (#31) and DSACK low to BERR low setup time (#48) can be ignored. The data must only satisfy the data-in to CLKO1 low setup time (#27) for the following clock cycle: BERR must only satisfy the late BERR low to CLKO1 low setup time (#27A) for the following clock cycle.
6. To ensure coherency during every operand transfer, BG will not be asserted in response to BR until after cycles of the current operand transfer are complete and RMC is negated.
7. In the absence of DSACK, BERR is an asynchronous input using the asynchronous setup time (#47).
8. During interrupt acknowledge cycles, the processor may insert up to two wait states between states S0 and S1.
9. These specs are for Synchronous Arbitration only. ASTM=1.
10. These CS specs are for TRLX=0. If RAS and RAS DD are connected together, reduce max value of RAS specification by 1.5ns.
11. These CS specs are for TRLX=1. If RAS and RAS DD are connected together, reduce max value of RAS specification by 1.5ns.
12. These CS specs are for CSNTQ=0.
13. These CS specs are for CSNTQ=1; or RAS specs for DRAM accesses.
14. These specs are read cycles with parity check and PBEE=1.
15. These specs are read cycles with parity check and PBEE=0,PAREN=1.
16. These RAS specs are for page miss case.
17. These specifications only apply to CS/RAS pins.
18. This specification applies to non fast termination cycles. In fast termination cycles, the $\overline{\text{BERR}}$ signal must be negated by 20ns after negation of $\overline{\text{AS}}$, $\overline{\text{DS}}$.



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

INDEX