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	-40°C ~ 85°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/mc68360cem25l

The equations of the byte write enables for 32-bit port (16BM = 1) are as follows:

```
WE0 = R/W + AS + A0 + A1
WE1 = R/W + AS + not {(A1 * SIZ0) + (A0 * A1) + (A1 * SIZ1)}
WE2 = R/W + AS + not {(A0 * A1) + (A1 * SIZ0 * SIZ1) + (A1 * SIZ0 * SIZ1) +
       (A0 * A1 * SIZ0)}
WE3 = R/W + AS + not {(A0 * SIZ0 * SIZ1) + (SIZ0 * SIZ1) + (A0 * A1) + (A1 *
       SIZ1)}
```

These signals have the same timing as AS. The equations are valid only for a 32-bit port.

The equations of the byte write enables for 16-bit port (B16M = 0) are as follows:

```
WE0 = R/W + AS + A0
WE1 = R/W + AS + (A0 * SIZ0 * SIZ1)
```

These signals have the same timing as AS. The equations are valid only for a 16-bit port.

WEx signals are not shown in the diagrams in this section. Use AS timing instead during write cycles. The particular WEx signals that are active in a given bus cycle depend on which bytes are being written.

NOTE

Note that the WE signals are not affected by dynamic bus sizing. External assertion of DSACKx will have no effect on which WEx signal gets asserted.

When 16-bit mode is selected and Bit 7 of PEPAR is set, WE2 and WE3 are used as address lines A29 and A28 respectively.

4.1.9 Bus Cycle Termination Signals

The following signals can terminate a bus cycle.

4.1.9.1 DATA TRANSFER AND SIZE ACKNOWLEDGE (DSACK1 AND DSACK0). During bus cycles, external devices assert DSACK1 and/or DSACK0 as part of the bus protocol. During a read cycle, this signals the QUICC to terminate the bus cycle and to latch the data. During a write cycle, this indicates that the external device has successfully stored the data and that the cycle may terminate. These signals also indicate to the QUICC the size of the port for the bus cycle just completed (see Table 4-3). Refer to 4.3.1 Read Cycle for timing relationships of DSACK1 and DSACK0.

Additionally, the system integration module (SIM60) can be programmed to internally generate DSACK1 and DSACK0 for external accesses, eliminating logic required to generate these signals. The SIM60 can alternatively be programmed to generate a fast termination, providing a two-cycle external access. Refer to 4.2.6 Fast Termination Cycles for additional information on these cycles.

4.1.9.2 BUS ERROR (BERR). This signal is also a bus cycle termination indicator and can be used in the absence of DSACKx to indicate a bus error condition. BERR can also be asserted in conjunction with DSACKx to indicate a bus error condition, provided it meets the

Bits 14–12, 3–1—Reserved

OPCODE—Operation Code

The opcodes are listed in Table 7-1.

Table 7-1. Opcodes

Opcode	SCC	SMC (UART/Trans)	SMC (GCI)	SPI	IDMA	Timer
0000	INIT RX & TX PARAMS	INIT RX & TX PARAMS	INIT RX & TX PARAMS	INIT RX & TX PARAMS		
0001	INIT RX PARAMS	INIT RX PARAMS		INIT RX PARAMS		
0010	INIT TX PARAMS	INIT TX PARAMS		INIT TX PARAMS		
0011	ENTER HUNT MODE	ENTER HUNT MODE				
0100	STOP TX[1]	STOP TX				
0101	GR STOP TX[2]				INIT IDMA	
0110	RESTART TX	RESTART TX				
0111	CLOSE RX BD	CLOSE RX BD		CLOSE RX BD		
1000	SET GROUP ADDR					SET TIMER
1001			GCI TIMEOUT			
1010	RESET BCS		GCI ABORT REQ			
1011						
1100	U	U	U	U	U	U
1101	U	U	U	U	U	U
1110	U	U	U	U	U	U
1111	U	U	U	U	U	U

NOTES:
1.STOP TX = MC68302 original STOP TRANSMIT command.
2.GR STOP TX = GRACEFUL STOP TRANSMIT command.

INIT TX and RX PARAMETERS. This command initializes the transmit and receive parameters in the parameter RAM to the values that they had after the last reset of the CP. This command is especially useful when switching protocols on a given serial channel.

INIT RX PARAMETERS. This command initializes the receive parameters of the serial channel.

INIT TX PARAMETERS. This command initializes the transmit parameters of the serial channel.

ENTER HUNT MODE. This command causes the receiver to stop receiving and begin looking for a new frame. The exact operation of this command may vary depending on the protocol used.

STOP TX. This command aborts the transmission from this channel as soon as the transmit FIFO has been emptied. It should be used in cases where transmission needs to be stopped as quickly as possible. Transmission will proceed when the RESTART command is issued.

For synchronous communication, the internal clock is identical to the baud rate output. To get the desired rate, the user can select the appropriate system clock according to the following equation:

sync baud rate = (BRGCLK or CLK2 or CLK6) ÷ (clock divider + 1) ÷ (1 or 16 according to the DIV16 bit)

For example, to get the rate of 64 kbps, the system clock can be 24.96 MHz, DIV16 = 0, and the clock divider = 389.

7.10 SERIAL COMMUNICATION CONTROLLERS (SCCS)

The SCC key features are as follows:

- Implements HDLC/SDLC, HDLC Bus, BISYNC, Synchronous Start/Stop, Asynchronous Start/Stop (UART), AppleTalk (LocalTalk), and Totally Transparent Protocols
- Ethernet Version of QUICC Supports Full 10 Mbps Ethernet/IEEE 802.3 on SCC1
- Additional Protocols Supported Through Motorola-Supplied RAM Microcodes: Profibus, Signaling System#7 (SS7), Async HDLC, DDCMP, V.14, and X.21 (see Appendix C RISC Microcode from RAM).
- 2 Mbps HDLC, HDLC Bus, and/or Transparent Data Rates Supported on All Four SCCs Simultaneously (Full Duplex).
- 10 Mbps Ethernet (Half Duplex) on SCC1 and 2 Mbps on the Other SCCs Supported Simultaneously (Full Duplex)
- A Single HDLC or Transparent Channel Can Be Supported at 8 Mbps (Full Duplex)
- SCC Clocking Rates up to 12.5 MHz at 25 MHz.
- DPLL Circuitry for Clock Recovery with NRZ, NRZI, FM0, FM1, Manchester, and Differential Manchester (Also Known as Differential Biphase-L)
- SCC Clocks May Be Derived from a Baud Rate Generator, an External Pin, or DPLL. Data Clock May Be as High as 3.125 MHz with a 25-MHz Clock
- Supports Automatic Control of the $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{CD}}$ Modem Signals
- Multibuffer Data Structure for Receive and Transmit (up to 224 BDs May Be Partitioned in Any Way Desired)
- Deep FIFOs (SCC1 Has 32-Byte Rx and Tx FIFOs; SCC2, SCC3, and SCC4 Have 16-Byte Rx and Tx FIFOs)
- Transmit-On-Demand Feature Decreases Time to Frame Transmission
- Low FIFO Latency Option for Transmit and Receive in Character-Oriented and Totally Transparent Protocols
- Frame Preamble Options
- Full-Duplex Operation
- Fully Transparent Option for Receiver/Transmitter While Another Protocol Executes on the Transmitter/Receiver
- Echo and Local Loopback Modes for Testing

TCI—Transmit Clock Invert

0 = Normal operation.

1 = The internal transmit clock (TCLK) is inverted by the SCC before it is used. This option allows the SCC to clock data out one-half clock earlier, on the rising edge of TCLK rather than the falling edge. In this mode, the SCC offers a minimum and maximum "rising clock edge to data" specification. Data output by the SCC after the rising edge of an external transmit clock can be latched by the external receiver one clock cycle later on the next rising edge of the same transmit clock. This option is recommended for Ethernet, HDLC, or transparent operation when the clock rates are high (e.g., above 8 MHz) to improve data setup time for the external receiver.

TSNC—Transmit Sense

This bit indicates the amount of time the internal sense signal will stay active after the last transition on the RXD pin, indicating that the line is free. For instance, these bits can be used in the AppleTalk protocol to avoid the spurious $\overline{CS}$-changed interrupt that would otherwise occur during the frame sync sequence that precedes the opening flags.

If RDCR is configured to 1× mode, the delay is the greater of the two numbers listed. If RDCR is configured to 8×, 16×, or 32× mode, the delay is the lesser of the two numbers listed.

00 = Infinite—carrier sense is always active (default)

01 = 14 or 6.5 bit times as determined by the RDCR bits

10 = 4 or 1.5 bit times as determined by the RDCR bits (normally chosen for AppleTalk)

11 = 3 or 1 bit times as determined by the RDCR bits

RINV—DPLL Receive Input Invert Data

0 = No invert

1 = Invert the data before it is sent to the on-chip DPLL for reception. This setting is used to produce FM1 from FM0, NRZI space from NRZI mark, etc. It may also be used in regular NRZ mode to invert the data stream.

NOTE

This bit must be 0 in HDLC BUS mode.

TINV—DPLL Transmit Input Invert Data

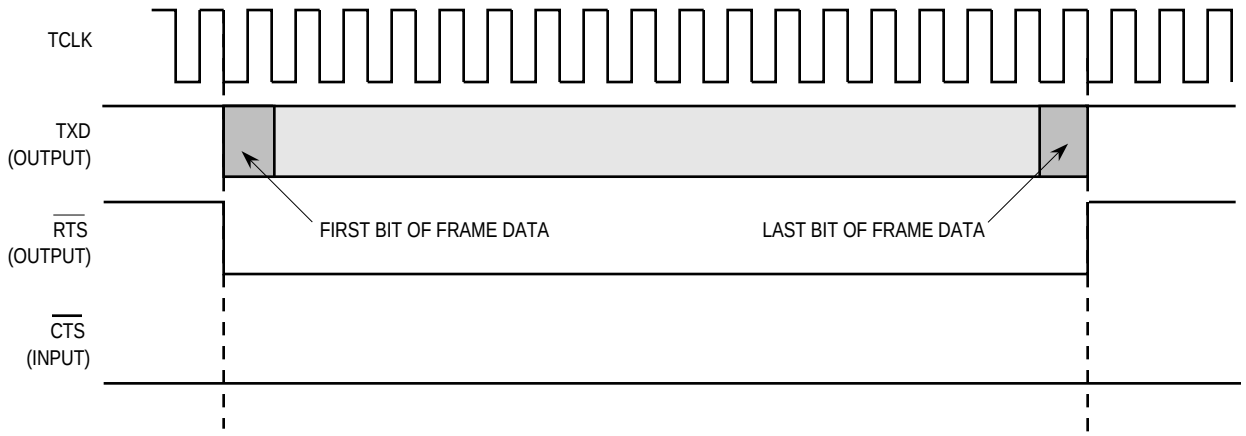
0 = No invert

1 = Invert the data before it is sent to the on-chip DPLL for transmission. This setting is used to produce FM1 from FM0, NRZI space from NRZI mark, etc. It may also be used in regular NRZ mode to invert the data stream.

NOTE

This bit must be 0 in HDLC BUS mode.

In T1 applications, setting TINV and TEND creates a continuously inverted HDLC data stream.



NOTES:
1. A frame includes opening and closing flags and syncs, if present in the protocol.

Figure 7-39. Output Delays from $\overline{\text{RTS}}$ Asserted for Synchronous Protocols

If the $\overline{\text{CTS}}$ pin is not already asserted when the $\overline{\text{RTS}}$ pin is asserted, then the delays to the first bit of data depend on when $\overline{\text{CTS}}$ is asserted. Figure 7-40 shows the delay between $\overline{\text{CTS}}$ and the data can be approximately 0.5 to 1 bit time or 0 bit times, depending on the CTSS bit in the GSMR.

states that dynamic changes are allowed, the following sequences are not required, and the register or bit may be changed immediately. In all other cases, the sequence should be used. For instance, the internal baud rate generators allow on-the-fly changes, but the DPLL-related bits in the GSMR do not.

NOTE

The modification of parameter RAM does not require a full disabling of the SCC. See the parameter RAM description for details on when parameter RAM values may be modified.

If the user desires to disable all SCCs, SMCs, and the SPI, then the CR may be used to reset the entire CP with a single command.

7.10.14.1 SCC TRANSMITTER FULL SEQUENCE. •For the SCC transmitter, the full disable and enable sequence is as follows:

1. STOP TRANSMIT command. This command is recommended if the SCC is currently in the process of transmitting data since it stops transmission in an orderly way. If the SCC is not transmitting (e.g., no Tx BDs are ready or the GRACEFUL STOP TRANSMIT command has been issued and has completed), then the STOP TRANSMIT command is not required. Furthermore, if the TBPTR will be overwritten by the user or the INIT TX PARAMETERS command will be executed, this command is not required.
2. Clear the ENT bit in the GSMR, which disables the SCC transmitter and puts it in a reset state.
3. Make modifications. The user may make modifications to the SCC transmit parameters including the parameter RAM. If the user desires to switch protocols or restore the SCC transmit parameters to their initial state, the INIT TX PARAMETERS command may now be issued.
4. RESTART TRANSMIT command. This command is required if the INIT TX PARAMETERS command was not issued in step 3.
5. Set the ENT bit in the GSMR. Transmission will now begin using the Tx BD pointed to by the TBPTR as soon as the Tx BD R-bit is set.

7.10.14.2 SCC TRANSMITTER SHORTCUT SEQUENCE. •A shorter sequence is possible if the user desires to reinitialize the transmit parameters to the state they had after reset. This sequence is as follows:

1. Clear the ENT bit in the GSMR.
2. INIT TX PARAMETERS command. Any additional modifications may now be made.
3. Set the ENT bit in the GSMR.

7.10.14.3 SCC RECEIVER FULL SEQUENCE. •The full disable and enable sequence for the receiver is as follows:

1. Clear the ENR bit in the GSMR. Reception will be aborted immediately. This disables the receiver of the SCC and puts it in a reset state.
2. Make modifications. The user may make modifications to the SCC receive parameters

A—Address

This bit is valid only in multidrop mode (either automatic or non-automatic).

0 = This buffer contains data only.
1 = Set by the CPU32+ core, this bit indicates that this buffer contains address character(s). All the buffer's data will be transmitted as address characters.

CM—Continuous Mode

0 = Normal operation.
1 = The R-bit is not cleared by the CP after this BD is closed, allowing the associated data buffer to be retransmitted automatically when the CP next accesses this BD. However, the R-bit will be cleared if an error occurs during transmission, regardless of the CM bit.

P—Preamble

0 = No preamble sequence is sent.
1 = The UART sends one character of all ones before sending the data so that the other end will detect an idle line before the data. If this bit is set and the data length of this BD is zero, only a preamble will be sent.

NS—No Stop Bit Transmitted

0 = Normal operation. Stop bits are sent with all characters in this buffer.
1 = The data in this buffer will be sent without stop bits if the SYNC mode is selected by setting the SYN bit in the PSMR. If ASYNC is selected the stop bit is SHAVED according to the value of the DSR register.

The following bit is written by the CP after it has finished transmitting the associated data buffer.

CT—CTS Lost

0 = The $\overline{\text{CTS}}$ signal remained asserted during transmission.
1 = The $\overline{\text{CTS}}$ signal was negated during transmission.

Data Length

The data length is the number of octets that the CP should transmit from this BD's data buffer. It is never modified by the CP. Normally, this value should be greater than zero. The data length may be equal to zero with the P-bit set, and only a preamble will be sent.

Tx Data Buffer Pointer

The transmit buffer pointer, which always points to the first location of the associated data buffer, may be even or odd. The buffer may reside in either internal or external memory.

7.10.16.18 UART EVENT REGISTER (SCCE). The SCCE is called the UART event register when the SCC is operating as a UART. It is a 16-bit register used to report events recognized by the UART channel and to generate interrupts. On recognition of an event, the UART controller will set the corresponding bit in the UART event register. Interrupts generated by this register may be masked in the UART mask register. An example of interrupts that may be generated by the UART is shown in Figure 7-49.

when the transmitter is disabled. Note that the INIT TX AND RX PARAMETERS command may also be used to reset both transmit and receive parameters.

7.10.17.6.2 Receive Commands. The following paragraphs describe the HDLC receive commands.

ENTER HUNT MODE Command**.** After a hardware or software reset and the enabling of the channel in the SCC mode register, the channel is in the receive enable mode and will use the first BD in the table.

The ENTER HUNT MODE command is generally used to force the HDLC receiver to abort reception of the current frame and enter the hunt mode. In the hunt mode, the HDLC controller continually scans the input data stream for the flag sequence. After receiving the command, the current receive buffer is closed, and the CRC is reset. Further frame reception will use the next BD.

CLOSE Rx BD Command**.** This command should not be used in the HDLC protocol.

INIT RX PARAMETERS Command**.** This command initializes all the receive parameters in this serial channel's parameter RAM to their reset state. This command should only be issued when the receiver is disabled. Note that the INIT TX AND RX PARAMETERS command may also be used to reset both receive and transmit parameters.

7.10.17.7 HDLC ERROR-HANDLING PROCEDURE. The HDLC controller reports frame reception and transmission error conditions using the channel BDs, the error counters, and the HDLC event register.

7.10.17.7.1 Transmission Errors. The following paragraphs describe various types of HDLC transmission errors.

Transmitter Underrun**.** When this error occurs, the channel terminates buffer transmission, closes the buffer, sets the underrun (U) bit in the BD, and generates the TXE interrupt if it is enabled. The channel will resume transmission after reception of the RESTART TRANSMIT command. The transmit FIFO size is 32 bytes on SCC1 and 16 bytes on SCC2, SCC3, and SCC4.

CTS Lost During Frame Transmission**.** When this error occurs, the channel terminates buffer transmission, closes the buffer, sets the CT bit in the BD, and generates the TXE interrupt if it is enabled. The channel will resume transmission after reception of the RESTART TRANSMIT command.

If this error occurs on the first or second buffer of the frame and the RTE bit in the HDLC mode register is set, the channel will retransmit the frame when the CTS line becomes active again. When working in an HDLC mode with collision possibility, to ensure the retransmission method functions properly, the first and second data buffers should contain more than 36 bytes of data (SCC1), and 20 bytes of data (SCC2, SCC3, and SCC4) if multiple buffers per frame are used. (Small frames consisting of a single buffer are not subject to this requirement.) The channel will also increment the retransmission counter.

receiver is not in hunt mode and a SYNC character has been received, the receiver will discard this character if the valid bit is set.

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

NOTE

When using 7-bit characters with parity, the parity bit should be included in the SYNC register value.

7.10.20.8 BDLE-BISYNC DLE REGISTER. The 16-bit, memory-mapped, read-write BDLE register is used to define the BISYNC stripping and insertion of the DLE character. When the BISYNC controller is in transparent mode and an underrun occurs during message transmission, the BISYNC controller inserts DLE-SYNC pairs until the next data buffer is available for transmission.

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

When the BISYNC receiver is in transparent mode and a DLE character is received, the receiver discards this character and excludes it from the BCS if the valid bit is set. If the second (next) character is a SYNC character, the BISYNC controller discards it and excludes it from the BCS. If the second character is a DLE, the BISYNC controller will write it to the buffer and include it in the BCS. If the character is not a DLE or SYNC, the BISYNC controller will examine the control characters table and act accordingly. If the character is not in the table, the buffer will be closed with the DLE follow character error (DLE) bit set. If the valid bit is not set, the receiver will treat the character as a normal character.

NOTE

When using 7-bit characters with parity, the parity bit should be included in the DLE register value.

7.10.20.9 TRANSMITTING AND RECEIVING THE SYNCHRONIZATION SEQUENCE. The BISYNC channel can be programmed to transmit and receive a synchronization pattern. The pattern is defined in the DSR. The length of the SYNC pattern is defined in the SYNL bits in the GSMR. The receiver synchronizes on the synchronization pattern that is located in the DSR. If the SYNL bits specify a non-zero synchronization pattern, then the transmitter sends the entire contents of the DSR prior to each frame, starting with the LSB first. Thus, the user may wish to repeat the desired SYNC pattern in the other DSR bits as well.

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-BIT SYNC															

BD pointed to by the RBPTR if the Rx BD E-bit is set.

7.11.5.4 SMC RECEIVER SHORTCUT SEQUENCE. A shorter sequence is possible if the user desires to reinitialize the receive parameters to the state they had after reset. This sequence is as follows:

1. Clear the REN bit in the SMCMR.
2. INIT RX PARAMETERS command. Any additional modifications may now be made.
3. Set the REN bit in the SMCMR.

7.11.5.5 SWITCHING PROTOCOLS. Sometimes the user desires to switch the protocol that the SMC is executing (for instance, UART to Transparent) without resetting the board or affecting any other SMC. This can be accomplished using only one command and a short number of steps:

1. Clear the TEN and REN bits in the SMCMR.
2. INIT TX AND RX PARAMETERS command. This one command initializes both transmit and receive parameters. Any additional modifications may now be made in the SMCMR to change the protocol, etc.
3. Set the SMCMR TEN and REN bits. The SMC is enabled with the new protocol.

7.11.6 Saving Power

When the TEN and REN bits of an SMC are cleared, that SMC consumes a minimal amount of power.

7.11.7 SMC as a UART

The following paragraphs describe the use of the SMC as a UART.

7.11.7.1 SMC UART KEY FEATURES. The SMC UART contains the following key features:

- Flexible Message-Oriented Data Structure
- Programmable Data Length (5–14 Bits)
- Programmable 1 or 2 Stop Bits
- Even/Odd/No Parity Generation and Checking
- Frame Error, Break, and IDLE Detection
- Transmit Preamble and Break Sequences
- Received Break Character Length Indication
- Continuous Receive and Transmit Modes

7.11.7.2 SMC UART COMPARISON. As compared to the UART modes supported by the SCCs, the SMCs generally offer less functionality and performance. This fits with their purpose of providing simple debug/monitor ports rather than full-featured UARTs. The SMC UARTs do not support the following features:

- $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{CD}}$ Pins
- Receive and Transmit Sections Being Clocked at Different Rates

- Multi-Master Environment Support
- Continuous Transfer Mode for auto scanning of a peripheral
- Supports Clock Rates up to 6.25 MHz in Master Mode and up to 12.5 MHz in Slave Mode (assuming a 25-MHz system clock)
- Independent Programmable Baud Rate Generator
- Programmable Clock Phase and Polarity
- Open-Drain Output Pins support multi-master configuration
- Local Loopback Capability for Testing

7.12.3 SPI Clocking and Pin Functions

The SPI can be configured as a master for the serial channel, meaning that it generates both the enable and clock signals, or as slave, meaning that the enable and clock signals are inputs to the SPI. The SPI also supports operation in a multi-master environment.

When the SPI is a master, the SPI baud rate generator is used to generate the SPI transmit and receive clocks. The SPI baud rate generator takes its input from the BRGCLK.

The BRGCLK is generated in the clock synthesizer of the QUICC specifically for the SPI baud rate generator and the other four baud rate generators in the CPM. BRGCLK defaults to the system frequency (25 MHz). However, the clock synthesizer in the SIM60 has an option to divide the BRGCLK by 1, 4, 16, or 64 before it leaves the clock synthesizer. Whatever the resulting frequency of BRGCLK, the user may use that BRGCLK frequency as the input to the SPI baud rate generator.

NOTE

User should note the maximum clock rate dose not equal maximum data rate. See Appendix A Serial Performance for more detail.

The ability to reduce the frequency of BRGCLK before it leaves the clock synthesizer is useful for two reasons. First, in a low-power mode, the baud rate generator clocking could be a significant factor in overall QUICC power consumption. Thus, if none of the QUICC baud rate generators need to generate high frequencies nor require a high resolution in the user application, a lower frequency BRGCLK may be input to the baud rate generators. Secondly, the user may wish to dynamically change the general system clock frequency in the clock synthesizer (SLOW GO mode), while still having the baud rate generator run at the original frequency. The BRGCLK allows this option also.

The SPI master-in slave-out (SPIMISO) pin is an input in master mode and an output in slave mode. The SPI master-out slave-in (SPIMOSI) pin is an output in master mode and an input in slave mode. The reason the pins names SPIMOSI and SPIMISO change functionality between master and slave mode is to support a multi-master configuration that allows communication from any SPI to any other SPI with the same hardware configuration.

RAM values will not normally need to be accessed by user software. They should only be modified when no SPI activity is in progress.

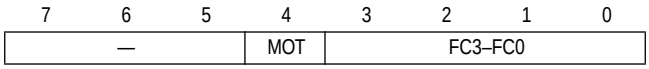
7.12.5.3.1 BD Table Pointer (RBASE, TBASE). The RBASE and TBASE entries define the starting location in the dual-port RAM for the set of BDs for receive and transmit functions of the SPI. This provides a great deal of flexibility in how BDs for an SPI are partitioned. By setting the W-bit in the last BD in each BD list, the user may select how many BDs to allocate for the transmit and receive side of the SPI. The user must initialize these entries before enabling the SPI. Furthermore, the user should not configure BD tables of the SPI to overlap any other serial channel's BDs, or erratic operation will occur.

NOTE

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

7.12.5.3.2 SPI Function Code Registers (RFCR, TFCR). 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

These bits should be set to zero by the user.

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.

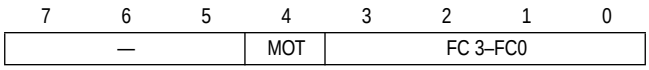
0 = DEC and Intel convention is used for byte ordering—swapped operation. It is also called little-endian byte ordering. The bytes stored in each buffer word are reversed as compared to the Motorola mode.

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. To keep interrupt acknowledge cycles unique in the system, do not write the value 0111 binary to these bits.

Transmit Function Code Register



7.14.6 Port B Pin Functions

Refer to Table 7-20 for the description of all port B pin options. All port B pins except PB17 and PB16 may be open-drain. Port B pins are independently configured as a general-purpose I/O pins if the corresponding bit in the port B pin assignment register (PBPAR) is cleared. They are configured as dedicated on-chip peripheral pins if the corresponding PBPAR bit is set.

When acting as a general-purpose I/O pin, the signal direction for that pin is determined by the corresponding control bit in the port B data direction register (PBDIR). The port I/O pin is configured as an input if the corresponding PBDIR bit is cleared; it is configured as an output if the corresponding PBDIR bit is set. All PBPAR bits and PBDIR bits are cleared on total system reset, configuring all port B pins as general-purpose input pins.

If a port B pin is selected as a general-purpose I/O pin, it may be accessed through the port B data register (PBDAT). Data written to the PBDAT is stored in an output latch. If a port B pin is configured as an output, the output latch data is gated onto the port pin. In this case, when PBDAT is read, the port pin itself is read. If a port B pin is configured as an input, data written to PBDAT is still stored in the output latch but is prevented from reaching the port pin. In this case, when PBDAT is read, the state of the port pin is read.

All of the port B pins have more than one option. These options include on-chip peripheral functions relating to the IDMA, Ethernet CAM interface, SPI, SMC1, SMC2, TDMa, and TDMb.

Port B is also multiplexed with the PIP. The PIP is a CPM parallel port that can implement fast parallel interfaces, such as Centronics. For a functional description of the dedicated pin functions of the PIP, refer to 7.13 Parallel Interface Port (PIP).

NOTES

If the user does not use the PIP, the description in this section is sufficient to describe the features of port B, and the PIP description does not need to be studied.

The PIP STRBI and STRBO pins are not listed in Table 7-20. See 7.13 Parallel Interface Port (PIP) for instructions on how to enable them.

PB3–PB5 and PB16 have an unusual property in that their on-chip peripheral functions (BRGO4, BRGO3, BRGO2, and BRGO1) are repeated in port A. This gives an alternate way to output the BRGO pins if other functions are used on port A. PB12–PB15 have an unusual property in that their on-chip peripheral functions (such as $\overline{\text{RTSx}}$ or L1ST1) are repeated in port C. This gives an alternate location to output these pins if other functions on port C are used.

dled differently. See 9.4.2 Memory Interfaces.) To provide dynamic bus sizing, the MC68150 device may be added to the design shown here.

Parity is supported for both SRAM and DRAM arrays using the four-byte parity lines PRTY3–PRTY0. When a parity error occurs, the error indication on the $\overline{\text{PERR}}$ pin causes the QUICC to generate a level 5 interrupt to the MC68EC040. (Level 7 has already been used for the breakpoint generation interrupt.) The parity error timing is not fast enough to allow an MC68EC040 bus error to be generated on the bus cycle that generated a parity error.

The QUICC supports MC68EC040 bursting using the BADD3–BADD2 pins. These pins normally reflect the values on A3–A2, but, in the case of a burst, are used to increment the address to the memory array. If the memory devices already support MC68EC040 bursting internally, the BADD3–BADD2 pins are not required.

The DRAM arrays require the four $\overline{\text{CAS3}}$–$\overline{\text{CAS0}}$ pins. Also, since an external address multiplexer is used, the AMUX pin is required to select between rows and columns. If, however, the user's configuration does not require DRAM, the AMUX pin can be used as an $\overline{\text{OE}}$ pin instead. This would save an inverter in a number of memory arrays.

NOTE

Many memory arrays show an inverter on the R/$\overline{\text{W}}$ pin to create the $\overline{\text{OE}}$ signal. When using multiple memory arrays, it is possible to share one inverter between multiple memory arrays; however, this configuration is not shown.

The QUICC also provides four write enable ($\overline{\text{WEx}}$) pins to select the correct byte during write operations.

9.4.2.2 REGULAR EPROM. Figure 9-9 shows the glueless interface to standard EPROM in the system. The assumption is made that only the MC68EC040 will access this array. No bursting capability is used. The CONFIG2–CONFIG0 pins are configured to initialize the system to slave mode, $\overline{\text{CS0}}$ operating on a 32-bit port at reset, the MBAR at its normal location, and MC68040 companion mode.

It would have been possible to use 16-bit-wide EPROM to reduce the chip count, if desired. (See 9.4.2.3 Burst EPROM. for an example.)

9.4.2.3 BURST EPROM. Figure 9-10 shows the glueless interface to two burst EPROMs available from National Semiconductor. These devices support a glueless interface to the MC68040. In this design, the assumption is made that only the MC68EC040 will access this array.

QUICC supports internally (e.g., the level of the SIM60 and the level of the CPM). If such a condition occurs, $\overline{\text{BERR}}$ will be asserted by the QUICC.

9.8.1.9 SOFTWARE WATCHDOG. If desired, the MC68EC030 can program the QUICC software watchdog to generate a level 7 interrupt or a system reset. In this application, the software watchdog is configured in software to generate a reset so that the breakpoint logic can use level 7 interrupts. No additional hardware is required because the connection between the reset pins of the QUICC and the MC68EC030 is already made.

9.8.1.10 PERIODIC INTERVAL TIMER. If desired, the MC68EC030 can use the periodic interval timer on the QUICC to generate a system interrupt, such as for a real-time kernel. No additional hardware is required for this function.

9.8.1.11 MC68EC030 CACHING CONFIGURATION. The MC68EC030 can cache or not cache data and program memory as desired. However, it is strongly advisable not to cache the data that is accessed by the QUICC serial channels because of the overhead incurred every time the cached data area is written.

9.8.1.12 DOUBLE BUS FAULT. In slave mode, the QUICC double bus fault monitor is not operational.

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

When the QUICC is in master mode, it provides a $\overline{\text{TRIS}}$ pin that allows all outputs on the device to be three-stated. In slave mode, this feature is not available since the QUICC is a peripheral of the system.

9.8.1.14 QUICC SERIAL PORTS. The functions on QUICC parallel I/O ports A, B, and C may be used as desired in this application and have no bearing on the MC68EC030 interface. However, any unused parallel I/O pins should be configured as outputs, so they are not left floating.

9.8.2 Memory Interfaces

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

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

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

9.8.2.4 EEPROM. Figure 9-31 shows the interface to an EEPROM device to give a small amount of nonvolatile storage. In this case a byte-wide EEPROM bank is defined to minimize cost. If the port size of the chip select is selected to be 8-bits, then each byte of the EEPROM may be accessed in succession. The $\overline{CS4}$ pin should be programmed to respond to an 8K byte area in this design.

Only one byte should be written at a time. After a write is made, software is responsible for waiting the appropriate time (e.g. 10 ms) or for doing data polling to see if the newly written data byte is correct.

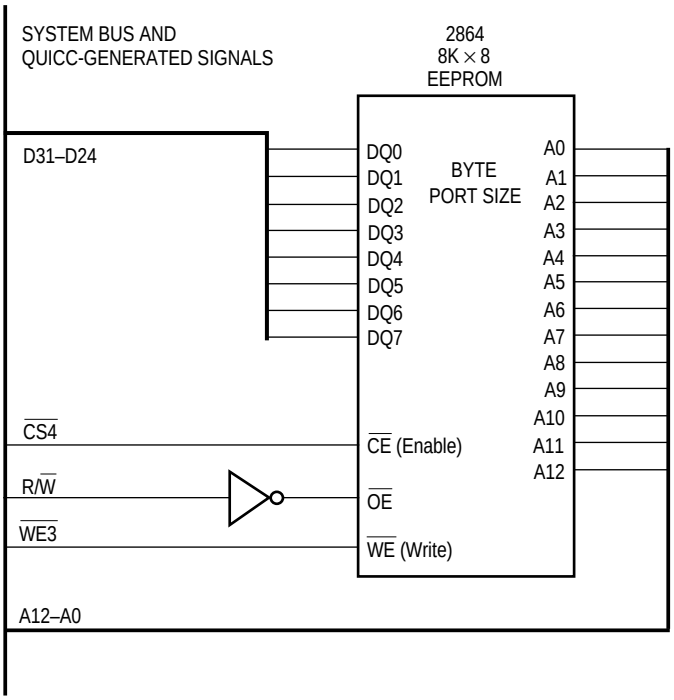


Figure 9-31. 8-Kbyte EEPROM Bank—8 Bits Wide

9.8.2.5 DRAM SIMM. Figure 9-32 shows the interface to an MCM32100S DRAM single in-line memory module (SIMM). Both the MC68EC030 and the QUICC can access the DRAM.

When the QUICC is a slave to an external MC68EC030, the address multiplexing for the DRAM must be done externally to the QUICC, which is accomplished in the three F157 multiplexers. The external address multiplexing scheme is very simple and allows page mode operation to be provided for the MC68EC030, if desired. This multiplexing scheme externally provides, the same multiplexing method that the QUICC implements internally.

NOTE

This multiplexing scheme allows the use of page mode, but requires hardware modification if larger SIMMs are to be used on the board. If the user is interested in the latter, rather than the

former, see the DRAM multiplexing scheme in 9.4 Using the QUICC MC68040 Companion Mode.

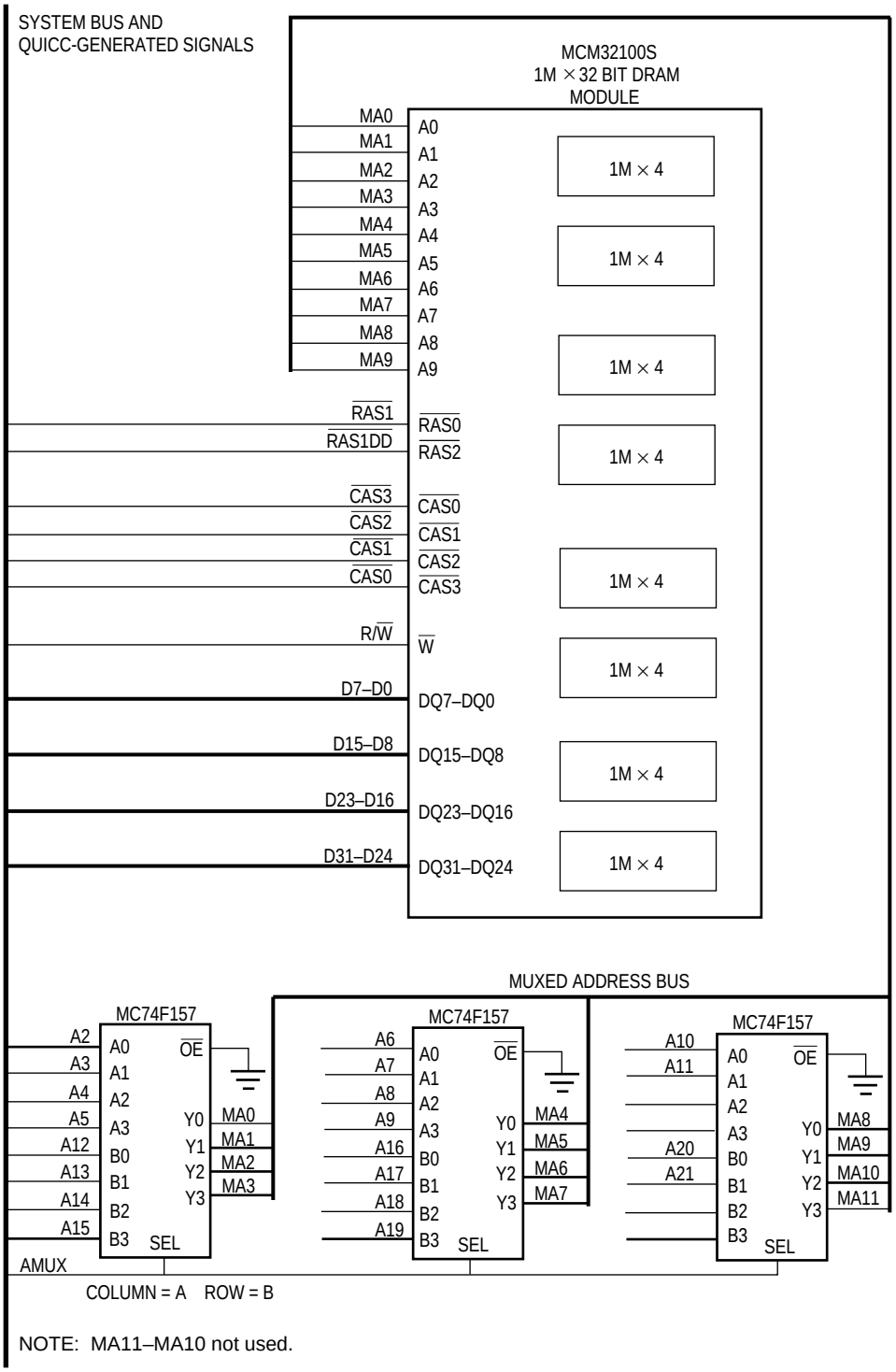


Figure 9-32. 4-Mbyte DRAM Bank—32 Bits Wide

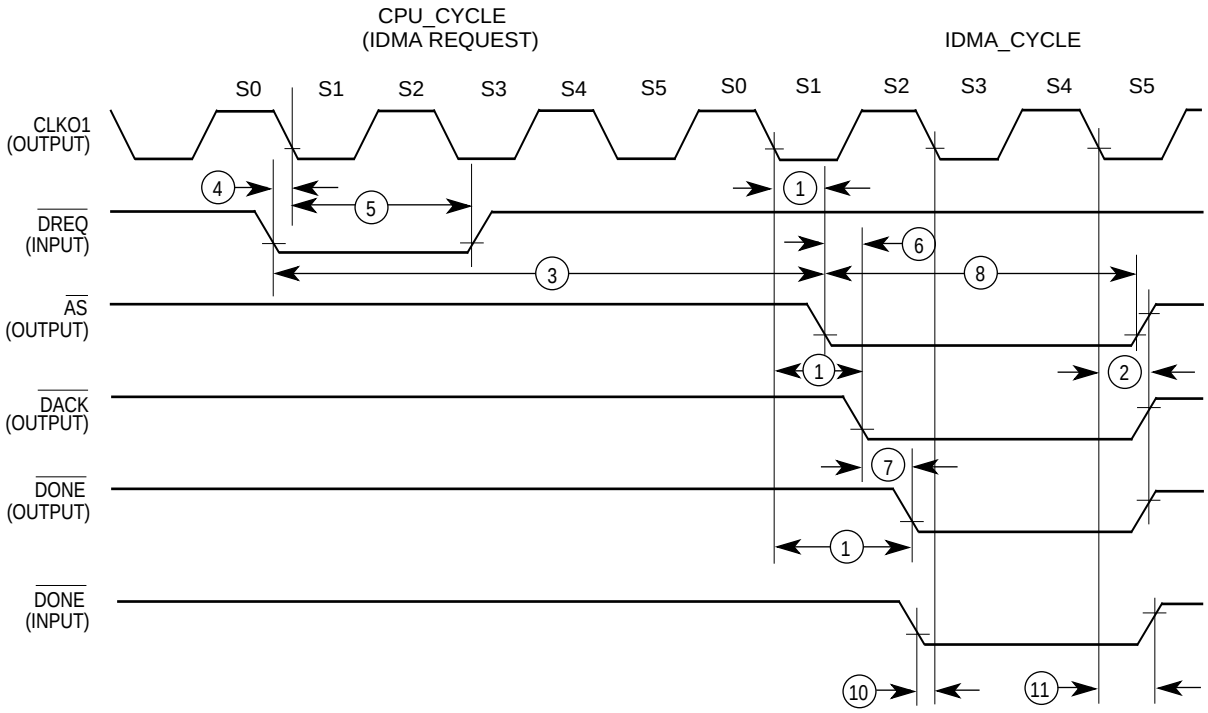


Figure 10-49. IDMA Signal Asynchronous Timing diagram

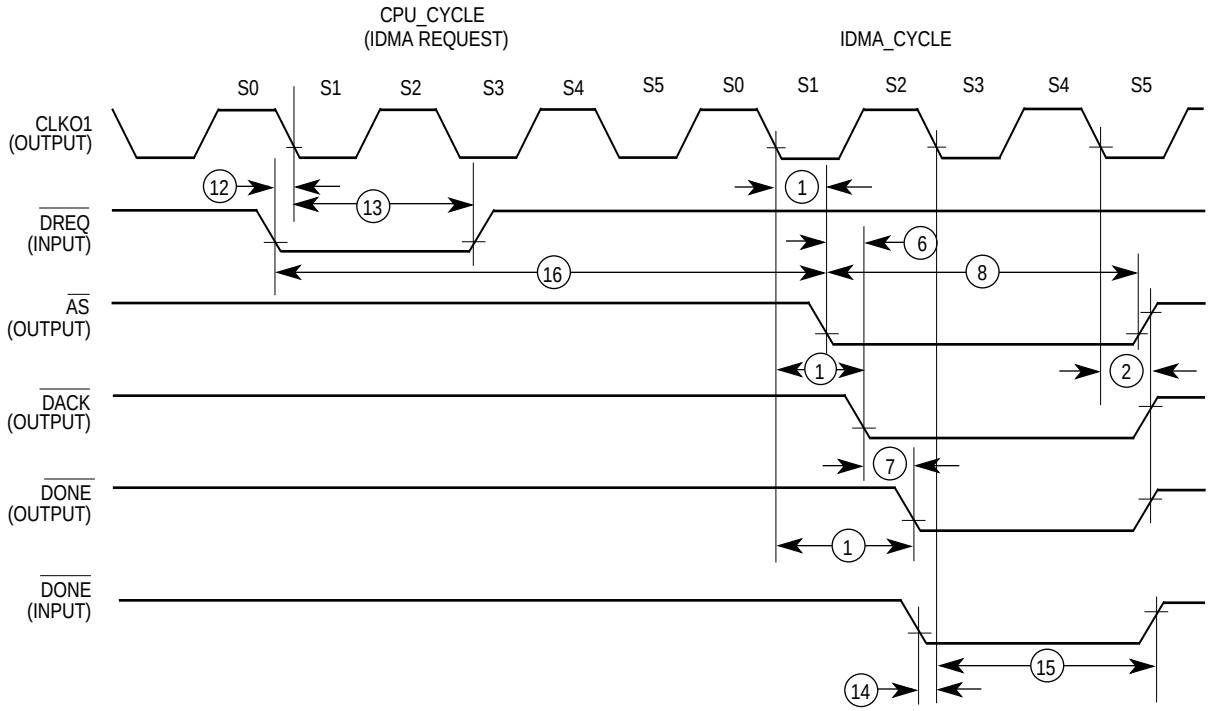


Figure 10-50. IDMA Signal Synchronous Timing diagram

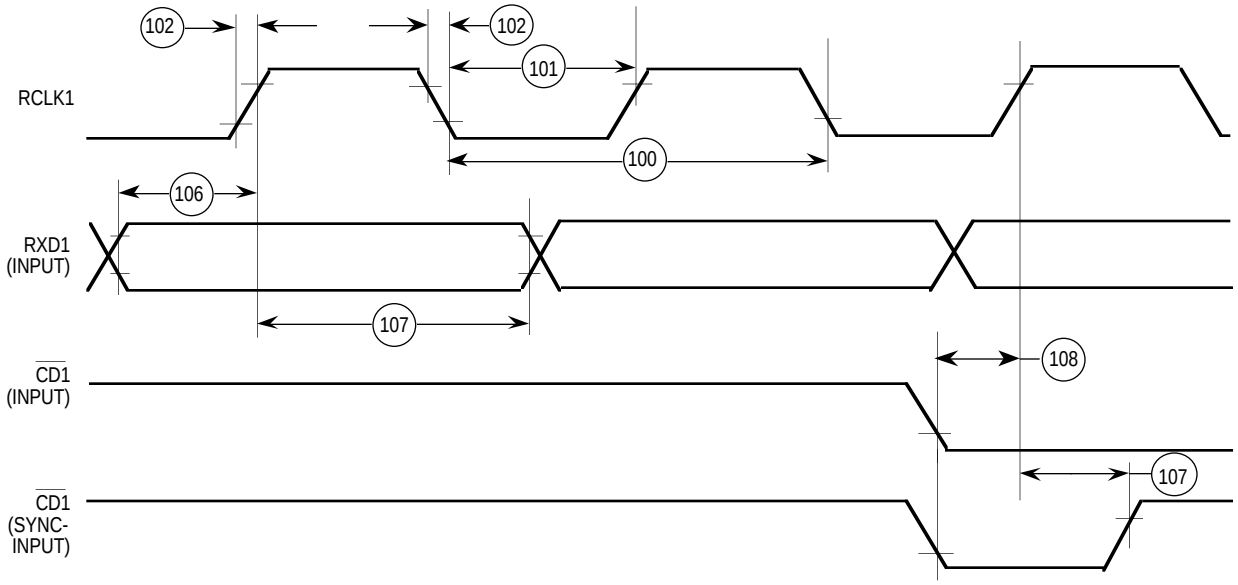


Figure 10-65. SCC NMSI Receive

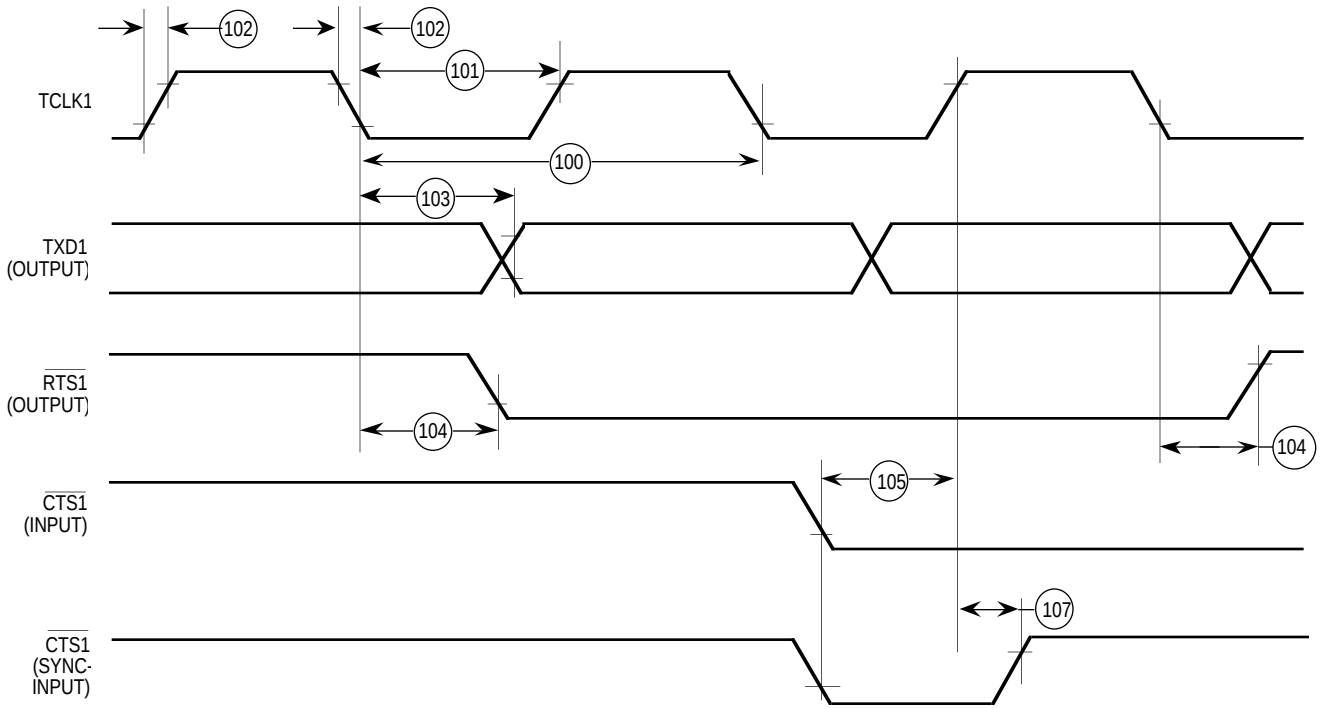


Figure 10-66. SCC NMSI Transmit