Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	CPU32+
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	25MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	3.3V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	357-BGA
Supplier Device Package	357-PBGA (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68360zp25vl

aligned on word or long-word boundaries, respectively. The QUICC IDMAs, when used, reduce the misalignment overhead to a minimum.

4.1 BUS TRANSFER SIGNALS

The bus transfers information between the QUICC and external memory or a peripheral device. External devices can accept or provide 8, 16, or 32 bits in parallel and must follow the handshake protocol described in this section. The maximum number of bits accepted or provided during a bus transfer is defined as the port width. The QUICC contains an address bus that specifies the address for the transfer and a data bus that transfers the data. Control signals indicate the beginning and type of the cycle as well as the address space and size of the transfer. The selected device then controls the length of the cycle with the signal(s) used to terminate the cycle. Strobe signals, one for the address bus and another for the data bus, indicate the validity of the address and provide timing information for the data.

Both asynchronous and synchronous operation is possible for any port width. In asynchronous operation, the bus and control input signals are internally synchronized to the QUICC clock, introducing a delay. This delay is the time required for the QUICC to sample an input signal, synchronize the input to the internal clocks, and determine whether it is high or low. In synchronous mode, the bus and control input signals must be timed to setup and hold times. Since no synchronization is needed, bus cycles can be completed in three clock cycles in this mode. Additionally, using the fast-termination option of the chip-select signals, two-clock operation is possible.

Furthermore, for all inputs, the QUICC latches the level of the input during a sample window around the falling edge of the clock signal. This window is illustrated in Figure 4-1, where t_{su} and t_h are the input setup and hold times, respectively. To ensure that an input signal is recognized on a specific falling edge of the clock, that input must be stable during the sample window. If an input makes a transition during the window time period, the level recognized by the QUICC is not predictable; however, the QUICC always resolves the latched level to either a logic high or low before using it. In addition to meeting input setup and hold times for deterministic operation, all input signals must obey the protocols described in this section.

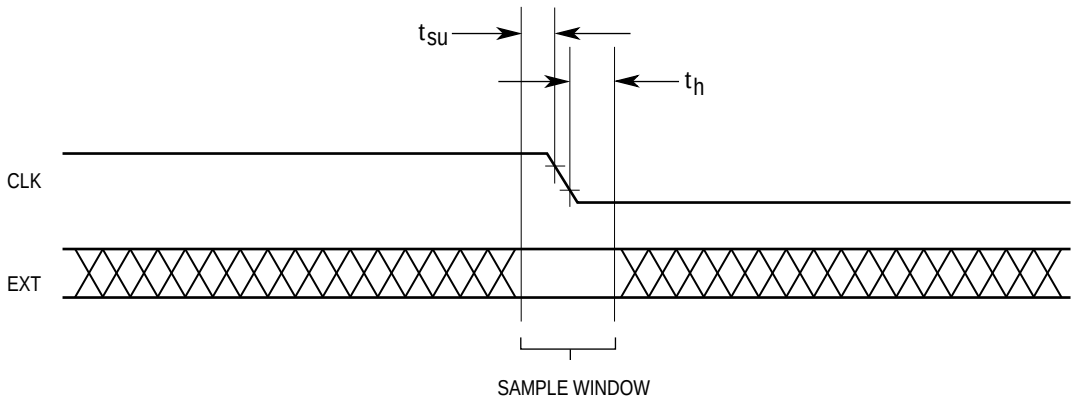


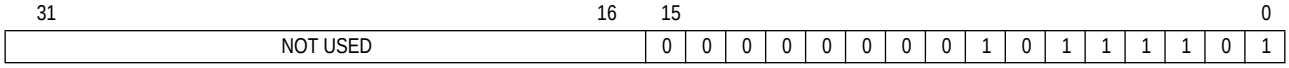
Figure 4-1. Input Sample Window

Table 5-15. T8-Bit Independent Variable Entries

X (Subroutine)	X (Instruction)	Y
0	0	0
1	256	16
2	512	32
3	768	48
4	1024	64
5	1280	80
6	1536	96
7	1792	112
8	2048	128
9	2304	112
10	2560	96
11	2816	80
12	3072	64
13	3328	48
14	3584	32
15	3840	16
16	4096	0

The first column is the value passed to the subroutine, the second column is the value expected by the table instruction, and the third column is the result returned by the subroutine.

The following value has been calculated for independent variable X:



Since X is an 8-bit value, the upper four bits are used as a table offset, and the lower four bits are used as an interpolation fraction. The following results are obtained from the subroutine:

Table Entry Offset ⇒ Dx [4:7] = $B = 11

Interpolation Fraction ⇒ Dx [0:3] = $D = 13

Thus, Y is calculated as follows:

Y = 80 + (13 (64 – 80)) / 16 = 67

If the 8-bit value for X were used directly by the table instruction, interpolation would be incorrectly performed between entries 0 and 1. Data must be shifted to the left four places before use:

LSL.W #4, Dx

TP—BERR Frame Type

The TP field defines the class of the faulted bus operation. Two bus error exception frame types are defined. One is for faults on prefetch and operand accesses, and the other is for faults during exception frame stacking.

0 = Operand or prefetch bus fault
1 = Exception processing bus fault

MV—MOVEM in Progress

MV is set when the operand transfer portion of the MOVEM instruction is in progress at the time of a bus fault. If a prefetch bus fault occurs while prefetching the MOVEM opcode and extension word, both the MV and IN bits will be set.

0 = MOVEM was not in progress when fault occurred
1 = MOVEM was in progress when fault occurred

SZC1,SCZ0—Original Operand Size

The SZC1,SZC0 field specifies the size of the original bus cycle (i.e., the size bits of the first cycle, when a transaction is divided into two or three cycles due to bus size or operand address).

00 = Original operand size was long word
01 = Original operand size was byte
10 = Original operand size was word
11 = Unused, reserved

TR—Trace Pending

TR indicates that a trace exception was pending when a bus error exception was processed. The instruction that generated the trace will not be restarted upon return from the exception handler. This includes MOVEM and released write bus errors indicated by the assertion of either MV or RR in the SSW.

0 = Trace not pending
1 = Trace pending

B1—Breakpoint Channel 1 Pending

B1 indicates that a breakpoint exception was pending on channel 1 (external breakpoint source) when a bus error exception was processed. Pending breakpoint status is stacked, regardless of the type of bus error exception.

0 = Breakpoint not pending
1 = Breakpoint pending

B0—Breakpoint Channel 0 Pending

B0 indicates that a breakpoint exception was pending on channel 0 (internal breakpoint source) when the bus error exception was processed. Pending breakpoint status is stacked, regardless of the type of bus error exception.

0 = Breakpoint not pending
1 = Breakpoint pending

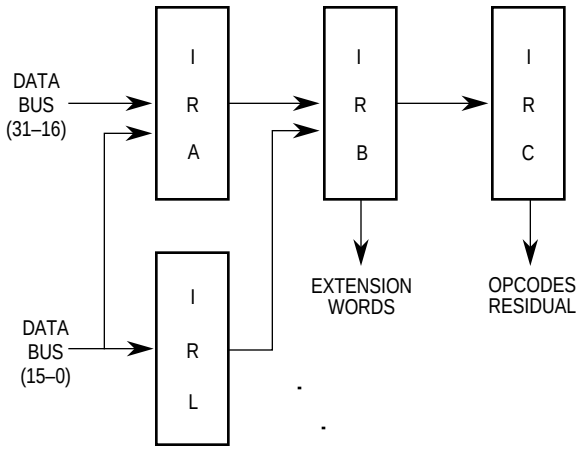


Figure 5-28. Functional Model of Instruction Pipeline

When $\overline{\text{IPIPE1}}$ is low during a clock cycle, it indicates the use of data from IRB on that clock cycle. $\overline{\text{IPIPE1}}$ should be sampled by the user on the falling edge of CLKO1. Regardless of the presence of valid data in IRA or IRL, the contents of IRB are invalidated when $\overline{\text{IPIPE1}}$ is asserted. If IRA or IRL contain valid data, the data is copied into IRB (IRA/IRL ⇒ IRB), and the IRB stage is revalidated.

When $\overline{\text{IPIPE0}}$ is low during a clock cycle, it indicates the start of a new instruction and subsequent replacement of data in IRC. This action causes a full advance of the pipeline (IRB ⇒ IRC and IRA/IRL ⇒ IRB). IRA and/or IRL is refilled during the next instruction fetch bus cycle.

Data loaded into IRA and IRL propagates automatically through subsequent empty pipeline stages. Signals that show the progress of instructions through IRB and IRC are necessary to accurately monitor pipeline operation. These signals are provided by IRA, IRL and IRB validity bits. When a pipeline advance occurs, the validity bit of the stage being loaded is set, and the validity bit of the stage supplying the data is negated.

Because instruction execution is not timed to bus activity, $\overline{\text{IPIPE1}}$–$\overline{\text{IPIPE0}}$ are synchronized with the system clock and not the bus. Figure 5-29 illustrates the timing in relation to the system clock.

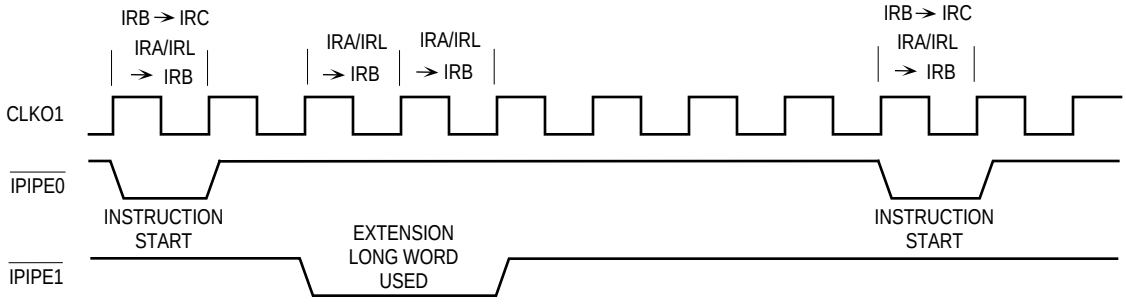


Figure 5-29. Instruction Pipeline Timing Diagram

bers inside parentheses (r/p/w) are included in the total clock cycle number. All timing data assumes two-clock reads and writes.

Instruction	Head	Tail	Cycles
ADD(A) Rn, Rm	0	0	2(0/1/0)
ADD(A) ⟨FEA⟩, Rn	0	0	2(0/1/0)
ADD Dn, ⟨FEA⟩	0	3	5(0/1/x)
AND Dn, Dm	0	0	2(0/1/0)
AND ⟨FEA⟩, Dn	0	0	2(0/1/0)
AND Dn, ⟨FEA⟩	0	3	5(0/1/x)
EOR Dn, Dm	0	0	2(0/1/0)
EOR Dn, ⟨FEA⟩	0	3	5(0/1/x)
OR Dn, Dm	0	0	2(0/1/0)
OR ⟨FEA⟩, Dn	0	0	2(0/1/0)
OR Dn, ⟨FEA⟩	0	3	5(0/1/x)
SUB(A) Rn, Rm	0	0	2(0/1/0)
SUB(A) ⟨FEA⟩, Rn	0	0	2(0/1/0)
SUB Dn, ⟨FEA⟩	0	3	5(0/1/x)
CMP(A) Rn, Rm	0	0	2(0/1/0)
CMP(A) ⟨FEA⟩, Rn	0	0	2(0/1/0)
CMP2 (Save)*⟨FEA⟩, Rn	1	1	3(0/1/0)
CMP2 (Op)⟨FEA⟩, Rn	2	0	16-18(X/1/0)
MUL(su).W⟨FEA⟩, Dn	0	0	26(0/1/0)
MUL(su).L (Save)*⟨FEA⟩, Dn	1	1	3(0/1/0)
MUL(su).L (Op)⟨FEA⟩, Dl	2	0	46-52(0/1/0)
MUL(su).L (Op)⟨FEA⟩, Dn:Dl	2	0	46(0/1/0)
DIVU.W ⟨FEA⟩, Dn	0	0	32(0/1/0)
DIVS.W ⟨FEA⟩, Dn	0	0	42(0/1/0)
DIVU.L (Save)*⟨FEA⟩, Dn	1	1	3(0/1/0)
DIVU.L (Op)⟨FEA⟩, Dn	2	0	<46(0/1/0)
DIVS.L (Save)*⟨FEA⟩, Dn	1	1	3(0/1/0)
DIVS.L (Op)⟨FEA⟩, Dn	2	0	<62(0/1/0)
TBL(su) Dn:Dm, Dp	26	0	28-30(0/2/0)
TBL(su) (Save)*⟨CEA⟩, Dn	1	1	3(0/1/0)
TBL(su) (Op)⟨CEA⟩, Dn	6	0	33-35(2X/1/0)
TBLSN Dn:Dm, Dp	30	0	30-34(0/2/0)
TBLSN (Save)*⟨CEA⟩, Dn	1	1	3(0/1/0)
TBLSN (Op)⟨CEA⟩, Dn	6	0	35-39(2X/1/0)
TBLUN Dn:Dm, Dp	30	0	34-40(0/2/0)
TBLUN (Save)*⟨CEA⟩, Dn	1	1	3(0/1/0)
TBLUN (Op)⟨CEA⟩, Dn	6	0	39-45(2X/1/0)

0 = The CP is ready to receive a new command.
1 = The CR contains a command that the CP is currently processing. The CP clears this bit at the end of the command execution or after reset.

7.2.1 Command Register Examples

To perform a complete reset of the CP, the value $8001 should be written to the CR. Following this command, the CR will return the value $0000 in two clocks.

To execute an ENTER HUNT MODE command to SCC3, the value $0381 should be written to the CR. While the command is executing, the CR will return the value $0381. When the command has been completely executed, the CR will return the value $0380.

7.2.2 Command Execution Latency

The worst-case command execution latency is 120 clocks. The typical command execution latency is about 40 clocks.

7.3 DUAL-PORT RAM

The CPM has 2560 bytes of static RAM configured as dual-port memory. The dual-port RAM memory map is shown in Figure 7-2, and a block diagram is shown in Figure 7-3.

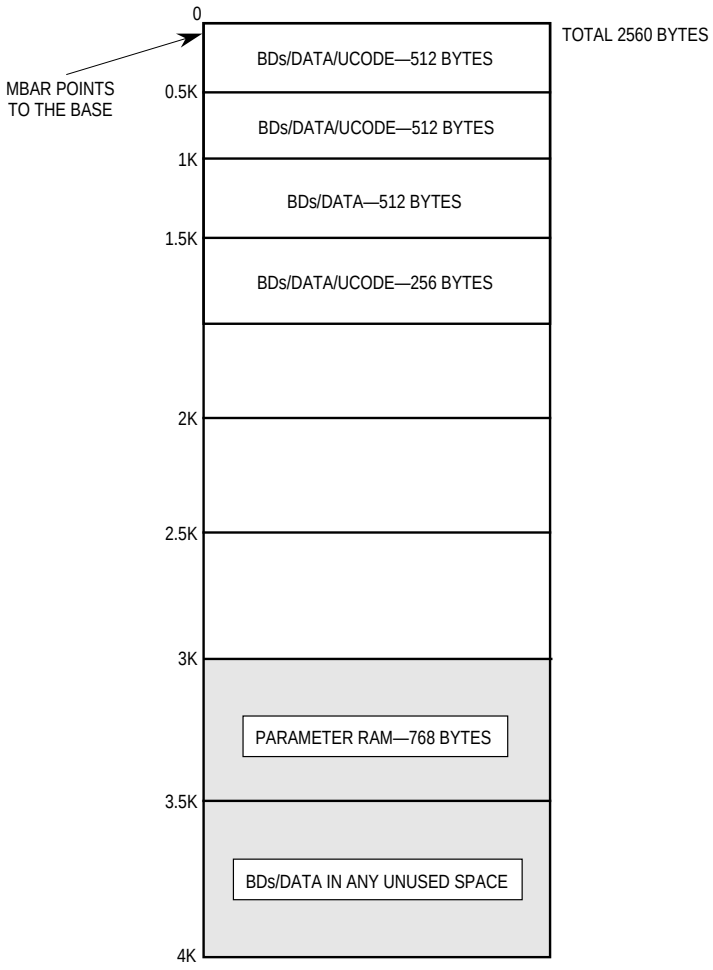


Figure 7-2. Dual-Port RAM Memory Map

The pointers provided by this register indicate the SI RAM entry word offset that is currently in progress. The register is cleared at reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
—	—	V	RbPTR					—	—	V	RaPTR				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	V	TbPTR					—	—	V	TaPTR				

In all cases, the value in the TxPTR or RxPTR increments by one for each entry (i.e., 16-bit SI RAM word) that is processed by the SI. Since each TxPTR and RxPTR is 5 bits each, the values in each TxPTR and RxPTR can range from 0 to 31, corresponding to 32 different SI RAM entries.

The full pointer range may not necessarily be used. For instance, if the last bit is set in the fifth SI RAM entry, then the pointer will only reflect values from 0 to 4. Once the fifth entry is processed by the SI, the pointer is reset to 0.

The V-bit in each entry shows that the entry is valid. This information is particularly useful if the PTR value happens to be zero. Additionally, the V-bits save the user from having to read both the SIRP and the SISTR to obtain the needed information.

The pointer values are described based on the four possible ways the SI RAM can be configured.

7.8.5.6.1 SIRP When RDM = 00 (One Static TDM). •In this case, since 64 entries cannot be signified with a single 5-bit pointer, two 5-bit pointers are used—one for the first 32 entries and one for the second 32 entries.

RaPTR and RbPTR contain the address of the RAM entry currently active. When the SI services entries 1–32, RaPTR will be incremented, and RbPTR will be continuously cleared. When the SI services entries 33–64, RaPTR will be continuously cleared, and RbPTR will be incremented.

TaPTR and TbPTR contain the address of the Tx entry currently active. When the SI services entries 1–32, TaPTR will be incremented, and TbPTR will be continuously cleared. When the SI services entries 33–64, TaPTR will be continuously cleared, and TbPTR will be incremented.

7.8.5.6.2 SIRP When RDM = 01 (One Dynamic TDM). •For the receiver, either RaPTR or RbPTR is used, depending on which portion of the SI Rx RAM is currently active. For the transmitter, either TaPTR or TbPTR is used, depending on which portion of the SI Tx RAM is currently active.

If its V-bit is set, RaPTR contains the address of the Rx entry currently active. The SI RAM receive address block in use is 0–63, and CRORa = 0 in SISTR.

If its V-bit is set, RbPTR contains the address of the Rx entry currently active. The SI RAM receive address block in use is 64–127, and CRORa = 1 in SISTR.

If its V-bit is set, TaPTR contains the address of the Tx entry currently active. The SI RAM transmit address block in use is 128–191, and CROTa = 0 in SISTR.

7. If the 1× GCI data clock is required, set PBPAR bit 11 and PBDIR bit 11, which configures L1CLKOa as an output.
8. PCPAR bit 11 = 1. Configures L1RSYNCa.
9. SIGMR = $04. Enable TDMa (one static TDM).
10. 1SICMR is not used.
11. 1SISTR and SIRP do not need to be read, but can be used for debugging information once the channels are enabled.
12. 1Enable the SCC1 for HDLC operation (to handle the LAPD protocol of the D channel), set SCC2 and SCC4 as desired, and enable SMC1 for SCIT operation.

7.8.8 Serial Interface Synchronization

On rev A and B of the QUICC, the SI would reset itself if an unexpected sync pulse was seen during the middle of a time frame. This would cause the SI to sync again on the following sync pulse but it would also lead to an unresolved loss of synchronization of an SCC or SMC operating in transparent or GCI modes (assuming that SCC or SMC was receiving data from the SI).

In revision C.1 and later of the QUICC, the SI will ignore this unexpected sync pulse and synchronize on the next sync pulse (it will not reset itself). This may lead to a reception of one or two “bad” slots but the SCC or SMC will remain synchronized.

NOTE

Rev A mask is C63T

Rev B mask are C69T, and F35G

Current Rev C mask are E63C, E68C and F15W

7.8.9 NMSI Configuration

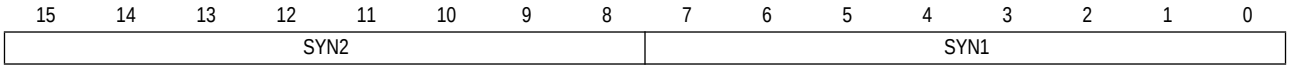
The SI supports an NMSI mode for each of the SCCs and SMCs. The decision of whether to connect a given SCC to the NMSI is made in the SICR. The decision of whether to connect a given SMC to the NMSI is made in the SIMODE register.

An SCC or SMC may be connected to the NMSI, regardless of which other channels are connected to a TDM channel. The user should note, however, that NMSI pins may be multiplexed with other functions at the parallel I/O lines. Therefore, if a combination of TDM and NMSI channels is used, the decision of which SCCs and SMCs to connect and where to connect them should be made consulting the QUICC pinout. Generally speaking, the TDMa channel is multiplexed with many of the SCC4 pins; whereas, the TDMb channel is multiplexed with many of the SCC3 pins.

The clocks that are provided to the SCCs and SMCs are derived from twelve sources: four internal baud rate generators and eight external CLK pins (see Figure 7-35). There are two main advantages to the bank-of-clocks approach. First, an SCC or SMC is not forced to

7.10.4 SCC Data Synchronization Register (DSR)

Each of the four SCC has a 16-bit, memory-mapped, read-write DSR. The DSR specifies the pattern used in the frame synchronization procedure in the synchronous protocols. In the UART protocol, it is used to configure fractional stop bit transmission. In the BISYNC and totally transparent protocol, it should be programmed with the desired SYNC pattern. In the Ethernet protocol, it should be programmed with $D555. At reset, it defaults to $7E7E (two HDLC flags), so it does not need to be written for HDLC mode. When DSR is used to send out SYNCs (such as in BISYNC or transparent mode), the contents of the DSR are always transmitted LSB first.



7.10.5 SCC Transmit on Demand Register (TODR)

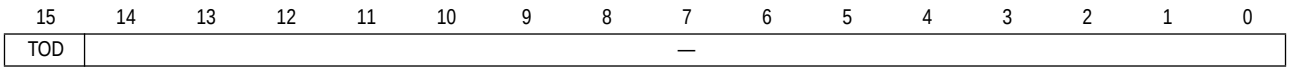
If no frame is currently being transmitted by an SCC, the RISC controller periodically polls the R-bit of the next Tx BD to see if the user has requested a new frame/buffer to be transmitted. This polling algorithm depends on the SCC configuration, but occurs every 8 to 32 serial transmit clocks. The user, however, has an option to request that the RISC begin the processing of the new frame/buffer immediately, without waiting until the normal polling time. To obtain immediate processing, the TOD bit in the transmit-on-demand register is set by the user once the user has set the R-bit in the Tx BD.

This feature, which decreases the transmission latency of the transmit buffer/frame, is particularly useful in LAN-type protocols where maximum interframe GAP times are limited by the protocol specification. Since the transmit-on-demand feature gives a high priority to the specified Tx BD, it can conceivably affect the servicing of the other SCC FIFOs. Therefore, it is recommended that the transmit-on-demand feature only be used when a high-priority Tx BD has been prepared and transmission on this SCC has not occurred for a period of time.

The TOD bit does not need to be set if a new Tx BD is added to the circular queue but other Tx BDs in that queue have not fully completed transmission. In that case, the new Tx BD will be processed immediately following the completion of the older Tx BD s.

The first bit of the frame will typically be clocked out 5-6 bit times after TOD has been written to a 1.

TOD—Transmit on Demand



0 = Normal operation
1 = The RISC will give a high priority to the current Tx BD and will not wait for the normal polling time to check that the Tx BD’s R-bit has been set. It will begin transmitting the frame. This bit will be cleared automaticaly after one serial clock.

7.10.17.7.2 Reception Errors. The following paragraphs describe various types of HDLC reception errors.

Overrun Error*.* The HDLC controller maintains an internal FIFO; for receiving data. The CP begins programming the SDMA channel (if the data buffer is in external memory) and updating the CRC when 8 or 32 bits (according to the RFW bit in the GSMR) are received in the FIFO. When a receive 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 with the overrun (OV) bit in the BD set and generates the RXF interrupt if it is enabled. The receiver then enters the hunt mode.

Even if the overrun occurs during a frame whose address is not matched in the address recognition logic, an Rx BD with data length two will be opened to report the overrun, and the RXF interrupt will be generated if it is enabled.

CD Lost During Frame Reception*.* When this error occurs, the channel terminates frame reception, closes the buffer, sets the CD bit in the Rx BD, and generates the RXF interrupt if it is enabled. This error has the highest priority. The rest of the frame is lost, and other errors are not checked in that frame. The receiver then enters the hunt mode.

Abort Sequence*.* An abort sequence is detected by the HDLC controller when seven or more consecutive ones are received. When this error occurs and the HDLC controller is currently receiving a frame, the channel closes the buffer by setting the AB bit in the Rx BD and generates the RXF interrupt (if enabled). The channel also increments the abort sequence counter. The CRC and nonoctet error status conditions are not checked on aborted frames. The receiver then enters hunt mode.

If the HDLC controller is not currently receiving a frame when an abort is received, no indication is given to the user.

Nonoctet Aligned Frame*.* When this error occurs, the channel writes the received data to the data buffer, closes the buffer, sets the Rx nonoctet aligned frame (NO) bit in the Rx BD, and generates the RXF interrupt (if enabled). The CRC error status should be disregarded on nonoctet frames. After a nonoctet aligned frame is received, the receiver enters hunt mode. (An immediately following back-to-back frame will still be received.) The nonoctet data may be derived from the last word in the data buffer as follows:

MSB											LSB
						1	0				0
<--------VALID DATA------->						<-------NONVALID DATA------->					

NOTE

If the data buffer swapping option is used (MOT bit cleared in the RFCR), then the above diagram refers to the last byte of the data buffer, not the last word. In HDLC, the LSB of each octet is transmitted first, and the MSB of the CRC is transmitted first.

7.10.19.4.2 PSMR Programming. The PSMR programming sequence is as follows:

1. The NOF bits should be set to 0001 (binary) giving two flags before frames (one opening flag, plus one additional flag).
2. The CRC should be set to 16-bit CRC-CCITT.
3. The DRT bit should be set.
4. All other bits should be set to zero or to their default condition.

7.10.19.4.3 TODR Programming. To expedite a transmit frame, the transmit on demand register (TODR) may be used.

7.10.19.4.4 AppleTalk Controller Example. Except for the previously discussed register programming, the HDLC Example #1 may be followed.

7.10.20 BISYNC Controller

The byte-oriented binary synchronous communication (BISYNC) protocol was originated by IBM for use in networking products. The three classes of BISYNC frames are transparent, non-transparent with header, and non-transparent without header (see Figure 7-63). The transparent mode in BISYNC allows full binary data to be transmitted with any possible character pattern. Each class of frame starts with a standard two-octet synchronization pattern and ends with a block check code (BCC). The end of text character (ETX) is used to separate the text and BCC fields.

NOTE

The transparent frame type in BISYNC is not related to the totally transparent protocol supported by the QUICC. See 7.10.21 Transparent Controller for details.

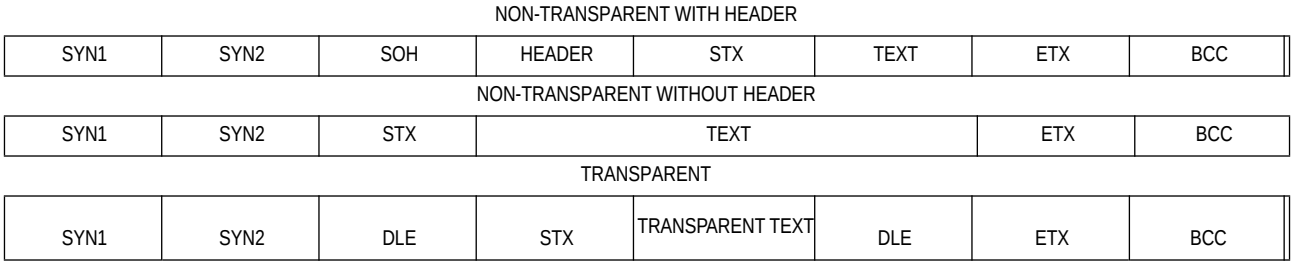


Figure 7-63. Typical BISYNC Frames

The bulk of the frame is divided into fields whose meaning depends on the frame type. The BCC is a 16-bit CRC (CRC16) format if 8-bit characters are used; it is a longitudinal check (a sum check) in combination with vertical redundancy check (parity) if 7-bit characters are used. In transparent operation, to allow the BISYNC control characters to be present in the frame as valid text data, a special character (DLE) is defined, which informs the receiver that the character following the DLE is a text character, not a control character. If a DLE is transmitted as valid data, it must be preceded by a DLE character. This technique is sometimes called byte-stuffing.

the buffer. Thus, MAX_IDL provides a convenient way to demarcate frames in the UART mode. If the MAX_IDL functionality is not desired, the user should program MAX_IDL to $0000, and the buffer will never be closed, regardless of the number of idle characters received. A character of idle is calculated as the following number of bit times: 1 + data length (5 to 14) + 1 (if parity bit is used) + number of stop bits (1 or 2). Example: for 8 data bits, no parity, and 1 stop bit, the character length is 10 bits.

IDLC. This value is used by the RISC to store the current idle counter value in the MAX_IDL timeout process. IDLC is a down-counter; it does not need to be initialized or accessed by the user.

BRKLN. This value is used to store the length of the last break character received. This value is the length in bits of that character. Example: If the receive pin is low for 257 bit times, BRKLN will show the value $0101. BRKLN is accurate to within one character unit of bits. For example, for 8 data bits, no parity, 1 stop bit, and 1 start bit, BRKLN is accurate to within 10 bits.

BRKEC. This counter counts the number of break conditions that occurred on the line. Note that one break condition may last for hundreds of bit times, yet this counter is incremented only once during that period.

BRKCR. The SMC UART controller will send an a break character sequence whenever a STOP TRANSMIT command is given. The number of break characters sent by the UART controller is determined by the value in BRKCR. In the case of 8 data bits, no parity, 1 stop bit, and 1 start bit, each break character is 10 bits in length and consists of all zeros.

7.11.7.4 SMC UART TRANSMISSION PROCESSING. The UART transmitter is designed to work with almost no intervention from the CPU32+ core. When the CPU32+ core enables the SMC transmitter, it will start transmitting idles. The SMC immediately polls the first BD in the transmit channel's BD ring, and thereafter once every character time, depending on the character length (i.e., every 7 to 16 serial clocks). When there is a message to transmit, the SMC will fetch the data from memory and start transmitting the message.

When a BD's data has been completely written to the transmit FIFO, the SMC writes the message status bits into the BD and clears the R-bit. An interrupt is issued if the I-bit in the BD is set. If the next Tx BD is ready, the data from its data buffer will be appended to the previous data and transmitted out on the transmit pin, with no gaps occurring between buffers. If the next Tx BD is not ready, the SMC will start transmitting idles and wait for the next Tx BD to become ready.

By appropriately setting the I-bit in each BD, interrupts can be generated after the transmission of each buffer, a specific buffer, or each block. The SMC will then proceed to the next BD in the table.

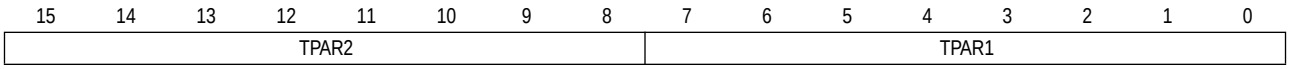
If the CM bit is set in the Tx BD, the R-bit will not be cleared, allowing the associated data buffer to be retransmitted automatically when the CP next accesses this data buffer. For instance, if a single Tx BD is initialized with the CM bit set and the W-bit set, the data buffer will be continuously transmitted until the user clears the R-bit of the BD.

T/R—Transmit/Receive Select

This bit selects transmitter or receiver operation for the PIP when it is using the interlocked, pulsed, or transparent handshake modes.

0 = Data is input to the PIP.
1 = Data is output from the PIP.

7.13.7.3 PIP TIMING PARAMETERS REGISTER (PTPR). The PTPR is a 16-bit read-write register that is cleared at reset. The PTPR holds two timing parameters, TPAR1 and TPAR2, which are used in the pulsed handshake modes for both a PIP transmitter and a receiver.



TPAR1—Timing Parameter 1

This 8-bit value defines the number of system clocks for TPAR1 in the transmitter or receiver pulsed handshake mode. The value $00 corresponds to 1 QUICC general system clock, and the value $FF corresponds to 256 QUICC general system clocks. A general system clock defaults to 40 ns, assuming a 25-MHz QUICC system.

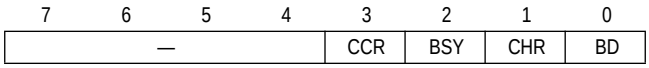
TPAR2—Timing Parameter 2

This 8-bit value defines the number of system clocks for TPAR2 in the transmitter or receiver pulsed handshake mode. The value $00 corresponds to 1 QUICC general system clock, and the value $FF corresponds to 256 QUICC general system clocks. A general system clock defaults to 40 ns, assuming a 25-MHz QUICC system.

7.13.7.4 PIP BUFFER DESCRIPTORS. BDs for the receiver and transmitter that support PIP operation were still in preparation at the time of writing.

7.13.7.5 PIP EVENT REGISTER (PIPE). The PIPE is an 8-bit register used to report events recognized by the PIP and to generate interrupts. It shares the same address as the SMC2 event register; thus, SMC2 cannot be used simultaneously with the PIP. Upon recognition of an event, the PIP sets its corresponding bit in the PIPE. Interrupts generated by this register may be masked in the PIP mask register.

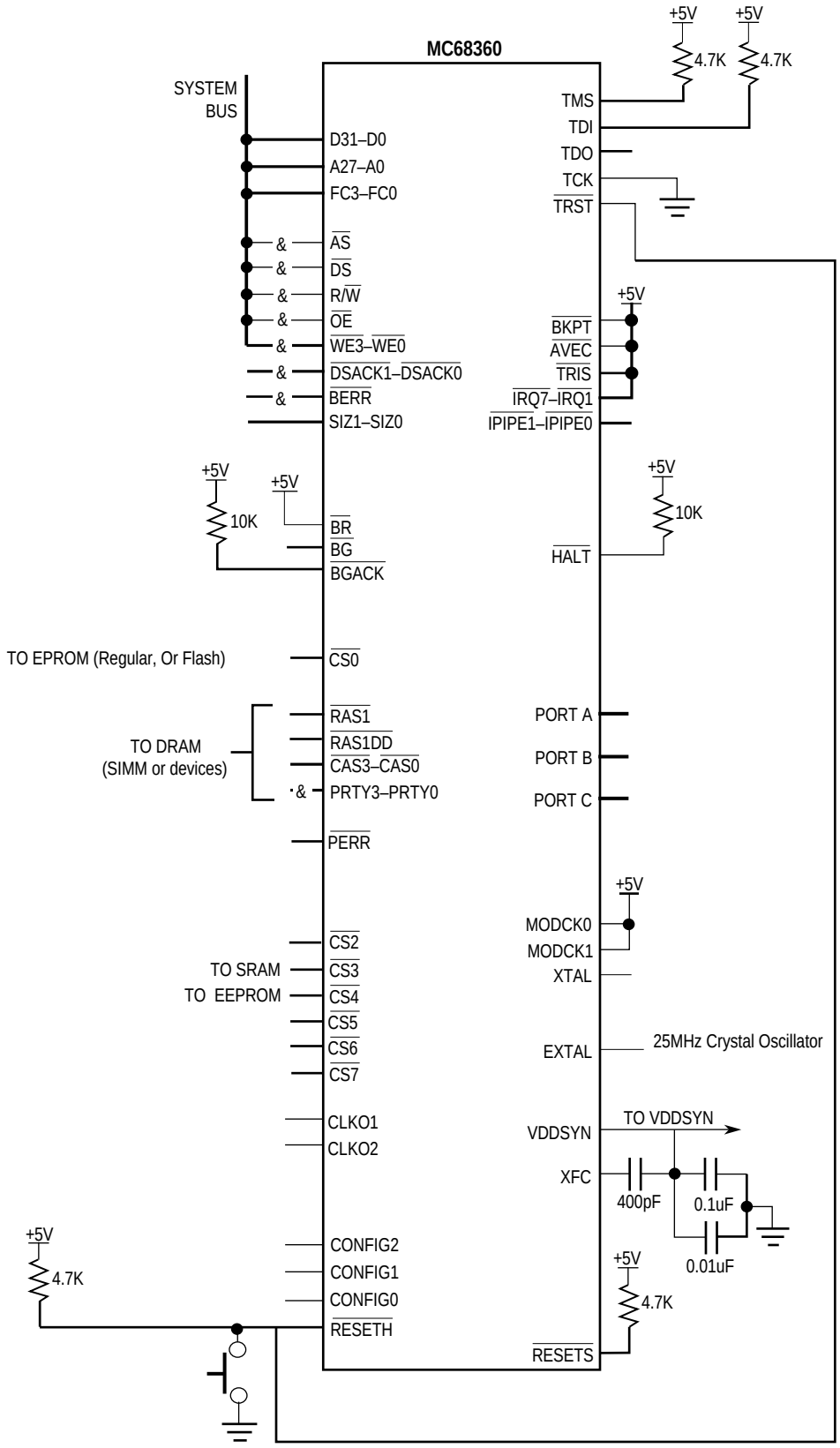
The PIPE is a memory-mapped register that may be read at any time. A bit is cleared by writing a one (writing a zero does not affect a bit's value). More than one bit may be cleared at a time. All unmasked bits must be cleared before the CP will clear the internal interrupt request. This register is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



Bits 7–4—Reserved

CCR—Control Character Received

A control character was received (with reject (R) = 1) and stored in the receive control character register.



LEGEND:
& = Pullups recommended (<= 10K)

Figure 9-1. MC68360 Minimum System Configuration

For the transparent protocol, neither the MC68302 nor the QUICC requires parameter RAM for its implementation. The QUICC, however, offers the new feature of allowing CRCs to be generated and checked on transparent frames. In that case, the QUICC's transparent parameter RAM must be initialized.

9.3.4.3 INTERNAL REGISTERS (SYSTEM INTEGRATION BLOCK). Both the MC68302 and the QUICC have a number of internal registers. The following paragraphs detail the registers according to their ascending order in the MC68302 memory map. Note that the address offsets of these registers on the QUICC are different from the address offsets on the MC68302.

IDMA

The MC68302 CMR is closely related to the QUICC CMR.

The STR, RST, and BT bits are the same on both parts.

The DSIZE bits are compatible, but on the QUICC, an encoding exists for long-word size.

The DAPI, SAPI, and REQG bits are the same.

INTE and INTN are, in effect, moved to the CMAR, but more control is given to the user.

ECO is in bit 15 of the QUICC CMR.

The QUICC CMR contains new functions in the RCI, S/D, and SRM bits.

SAPR on the MC68302 is identical to SAPR on the QUICC.

DAPR on the MC68302 is identical to DAPR on the QUICC.

BCR on the MC68302 is increased to 32 bits on the QUICC.

CSR on the MC68302 is extended on the QUICC.

The DONE, BED, and BES bits are the same on both parts.

The DNS bit is not needed on the QUICC.

The QUICC additionally contains the OB and BRKP bits.

The QUICC contains a CMAR that is not available on the MC68302.

The FCR is the same on both parts, except that the QUICC has a provision for four function code lines; whereas, the MC68302 only has three.

INTERRUPTS

The GIMR on the MC68302 is most closely related to the QUICC CICR and AVR.

The V7–V5 bits become the VBA2–VBA0 bits.

The ET1, ET6, and ET7 bits control the edge/level sensitivity of the $\overline{\text{IRQ1}}$, $\overline{\text{IRQ6}}$, and $\overline{\text{IRQ7}}$ pins on the MC68302. On the QUICC, these pins are handled by the SIM60. On the SIM60, all $\overline{\text{IRQx}}$ pins except $\overline{\text{IRQ7}}$ are level sensitive. These bits do not exist on the QUICC.

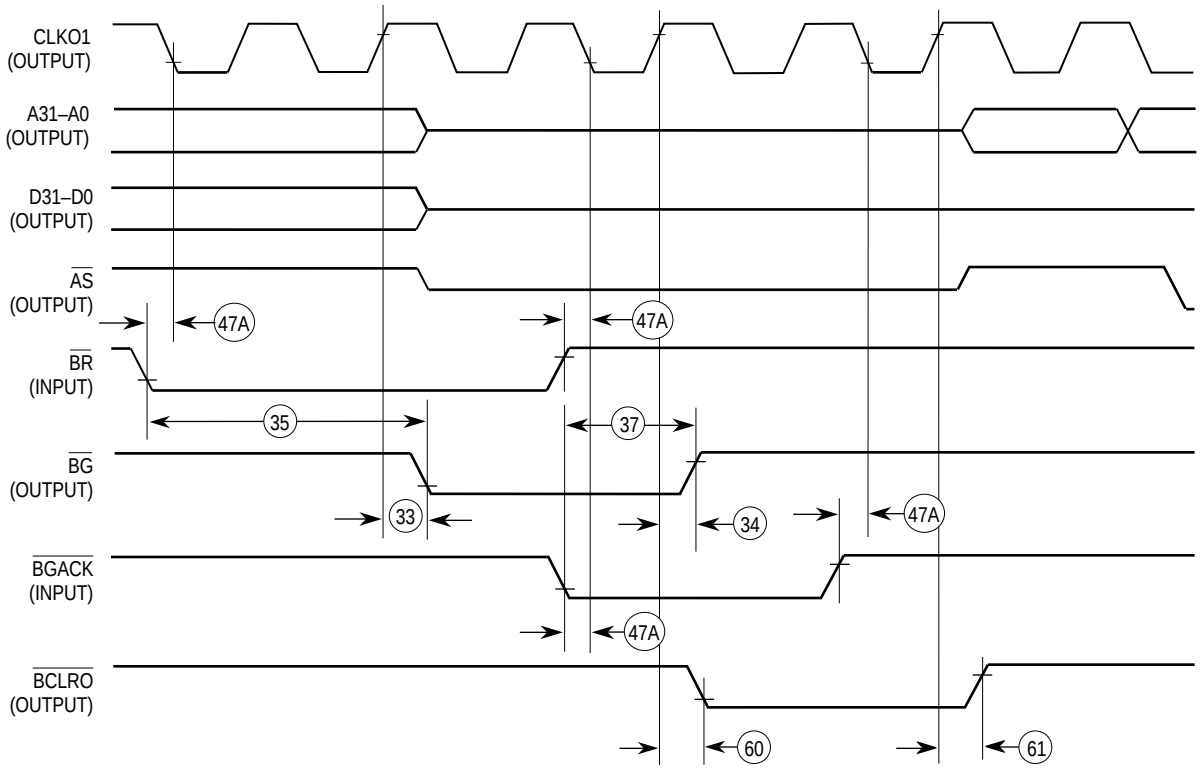


Figure 10-12. ASYNC Bus Arbitration – IDLE Bus Case

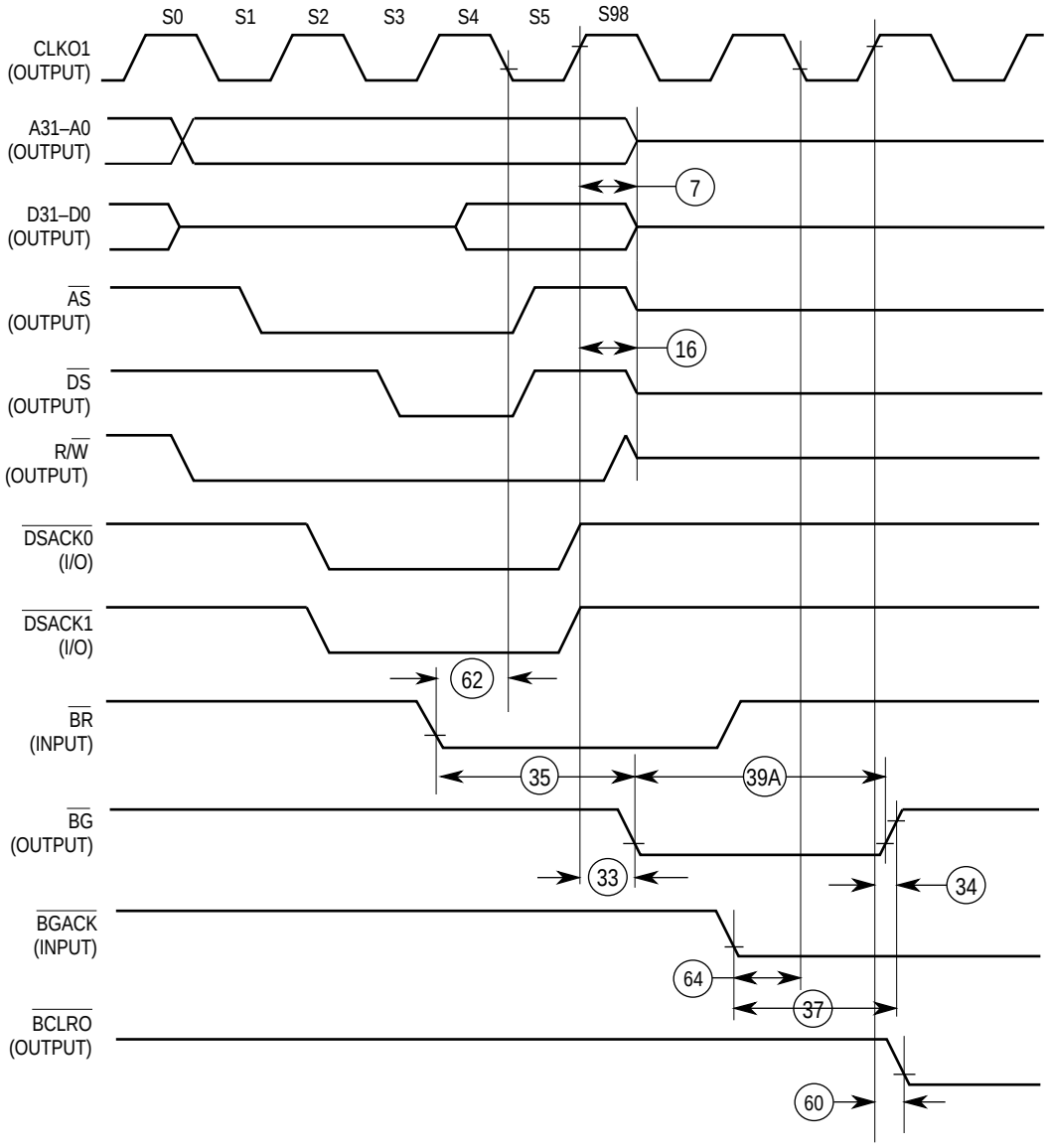


Figure 10-15. SYNC Bus Arbitration – Active Bus Case

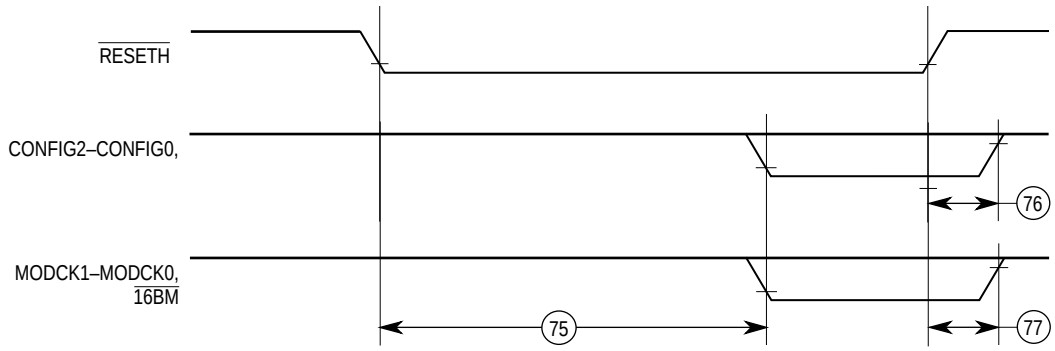


Figure 10-16. Configuration And Clock Mode Select Timing

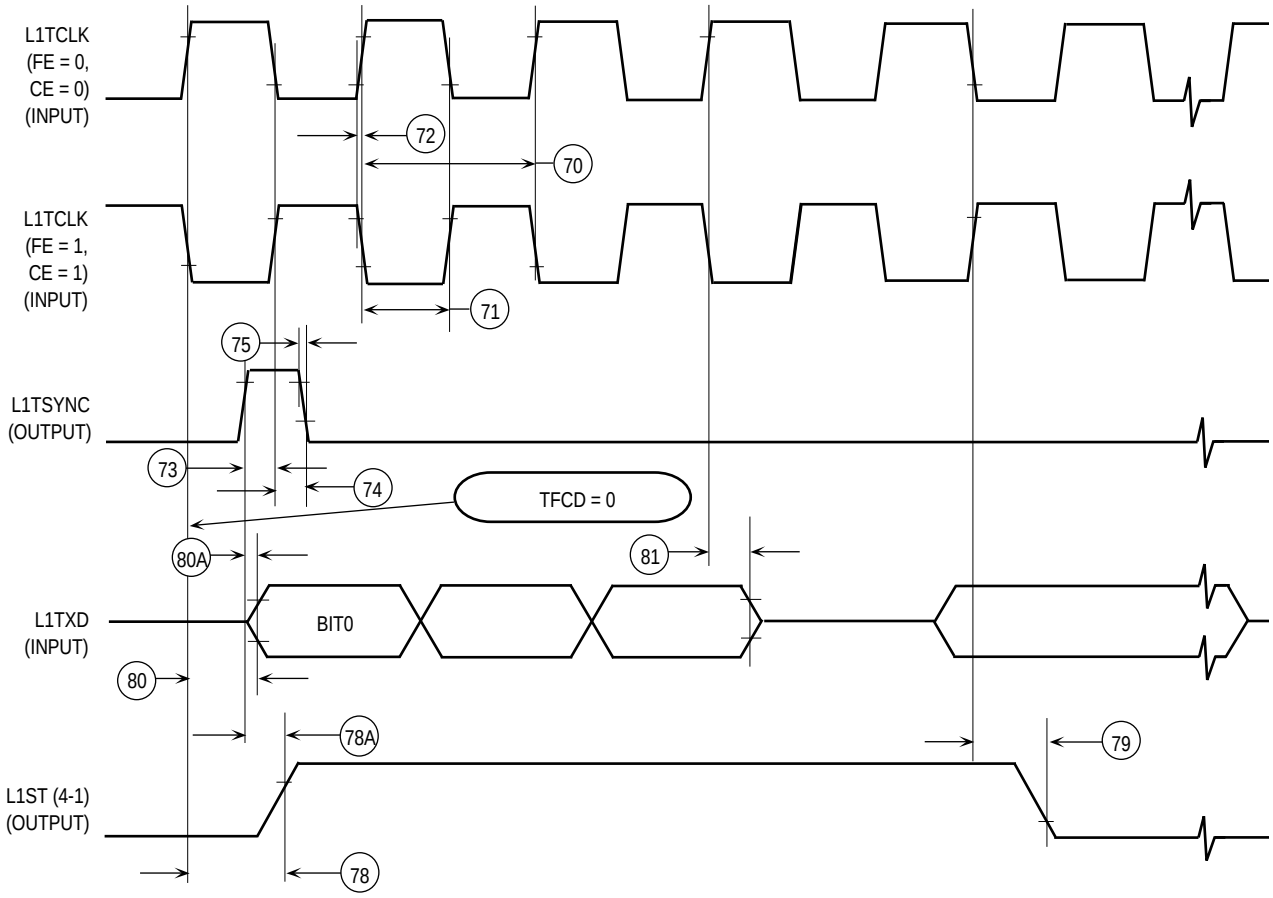


Figure 10-62. SI Transmit Timing with Normal Clocking (DSC = 0)

10.24 SCC IN NMSI MODE—EXTERNAL CLOCK ELECTRICAL SPECIFICATIONS

(The electrical specifications in this document are preliminary. See Figure 10-65–Figure 10-67)

Num.	Characteristic	3.3 V or 5.0 V		5.0 V		Unit
		25.0 MHz		33.34 MHz		
		Min	Max	Min	Max	
100[1]	RCLK1 and TCLK1 Width High	CLKO1	—	CLKO1	—	
101	RCLK1 and TCLK1 Width Low	CLKO1 + 5nS	—	CLKO1 + 5nS	—	
102	RCLK1 and TCLK1 Rise/Fall Time	—	15	—	15	ns
103	TXD1 Active Delay (From TCLK1 Falling Edge)	0	50	0	50	ns
104	RTS1 Active/Inactive Delay (From TCLK1 Falling Edge)	0	50	0	50	ns
105	CTS1 Setup Time to TCLK1 Rising Edge	5	—	5	—	ns
106	RXD1 Setup Time to RCLK1 Rising Edge	5	—	5	—	ns
107[2]	RXD1 Hold Time from RCLK1 Rising Edge	5	—	5	—	ns
108	CD1 Setup Time to RCLK1 Rising Edge	5	—	5	—	ns

NOTES:
1.The ratio SyncCLK/RCLK1 and SyncCLK/TCLK1 must be greater or equal to 2.25/1
2.Also applies to $\overline{CD}$ and $\overline{CTS}$ hold time when they are used as an external sync signals.

10.25 SCC IN NMSI MODE—INTERNAL CLOCK ELECTRICAL SPECIFICATIONS

(The electrical specifications in this document are preliminary. See Figure 10-65–Figure 10-67)

Num.	Characteristic	3.3 V or 5.0 V		5.0 V		Unit
		25.0 MHz		33.34 MHz		
		Min	Max	Min	Max	
100[1]	RCLK1 and TCLK1 Frequency	0	8.3	0	11	MHz
102	RCLK1 and TCLK1 Rise/Fall Time	—	—	—	—	ns
103	TXD1 Active Delay (From TCLK1 Falling Edge)	0	30	0	30	ns
104	RTS1 Active/Inactive Delay (From TCLK1 Falling Edge)	0	30	40	—	ns
105	CTS1 Setup Time to TCLK1 Rising Edge	40	—	40	—	ns
106	RXD1 Setup Time to RCLK1 Rising Edge	40	—	0	—	ns
107[2]	RXD1 Hold Time from RCLK1 Rising Edge	0	—	40	—	ns
108	CD1 Setup Time to RCLK1 Rising Edge	40	—	0	30	ns

NOTES:
1.The ratio SyncCLK/RCLK1 and SyncCLK/TCLK1 must be greater or equal to 3/1
2.Also applies to $\overline{CD}$ and $\overline{CTS}$ hold time when they are used as an external sync signals.

N

O

P