Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

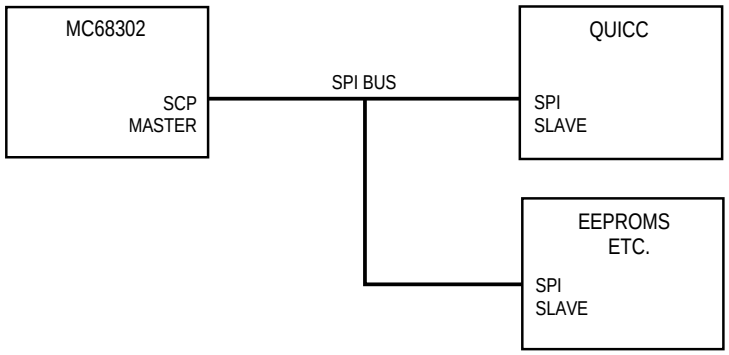
Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, 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	33MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	357-BGA
Supplier Device Package	357-PBGA (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68360zp33l

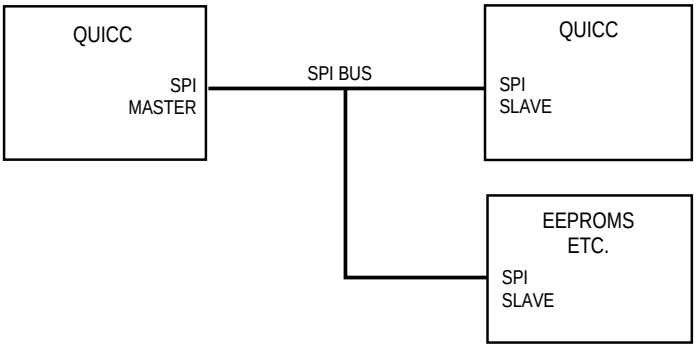
Figure 1-10 shows how the SCP on the MC68302 can be used to interface to the QUICC SPI.



NOTE: The MC68302 SCP can communicate with the QUICC SPI.

Figure 1-10. SPI Implementation Using SCP

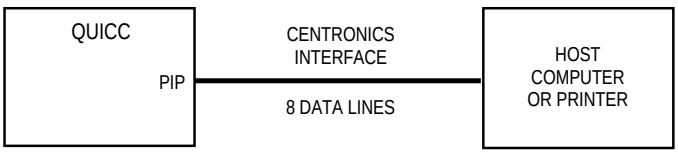
Figure 1-11 shows how the SPI on the QUICC can interface to another QUICC or SPI-based peripherals.



NOTE: Two QUICCs configured for a master-slave SPI connection.

Figure 1-11. SPI Master-Slave Implementation

Figure 1-12 shows how the parallel interface port (PIP) can be used to implement the Centronics interface connection. The QUICC may be the peripheral or the host.



NOTE: The QUICC can communicate over a Centronics Interface.

Figure 1-12. Centronics Interface Implementation

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

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

2.1.4 Parity

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

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

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

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

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

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

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

2.1.5 Memory Controller

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

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

NOTE

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

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

Table 3-4. QUICC CPM Registers Memory Map

REGB + 515		8	Reserved			
REGB + 516	CMAR1	8	Channel Mask Register	00		
REGB + 517		8	Reserved			
REGB + 518	CSR1	8	IDMA1 Channel Status Register	00		
REGB + 519		24	Reserved			
REGB + 51C	SDSR	8	SDMA Status Register	00		SDMA
REGB + 51D		8	Reserved			
REGB + 51E	SDCR	16	SDMA Configuration Register	0000	H	
REGB + 520	SDAR	32	SDMA Address Register	XXXX XXXX		
REGB + 524		16	Reserved			IDMA2
REGB + 526	CMR2	16	IDMA2 Mode Register	0000		
REGB + 528	SAPR2	32	IDMA2 Source Address Pointer	0000 0000		
REGB + 52C	DAPR2	32	IDMA2 Destination Address Pointer	0000 0000		
REGB + 530	BCR2	32	IDMA2 Byte Count Register	0000 0000		
REGB + 534	FCR2	8	IDMA2 Function Code Register	00		
REGB + 535		8	Reserved			
REGB + 536	CMAR2	8	Channel Mask Register	00		
REGB + 537		8	Reserved			
REGB + 538	CSR2	8	IDMA2 Channel Status Register	00		
REGB + 539 to REGB + 53F			Reserved			
REGB + 540	CICR	24	CP Interrupt Configuration Register	xx00 0000	H	CPIC
REGB + 544	CIPR	32	CP Interrupt Pending Register	0000 0000		
REGB + 548	CIMR	32	CP Interrupt Mask Register	0000 0000		
REGB + 54C	CISR	32	CP In-Service Register	0000 0000		
REGB + 550	PADIR	16	Port A Data Direction Register	0000	H	Parallel I/O
REGB + 552	PAPAR	16	Port A Pin Assignment Register	0000	H	
REGB + 554	PAODR	16	Port A Open Drain Register	0000	H	
REGB + 556	PADAT	16	Port A Data Register	XXXX		
REGB + 558 to REGB + 55f			Reserved			
REGB + 560	PCDIR	16	Port C Data Direction Register	0000	H	
REGB + 562	PCPAR	16	Port C Pin Assignment Register	0000	H	
REGB + 564	PCSO	16	Port C Special Options	0000	H	
REGB + 566	PCDAT	16	Port C Data Register	XXXX		
REGB + 568	PCINT	16	Port C Interrupt Control Register	0000	H	
REGB + 56a to REGB + 57f			Reserved			
REGB + 580	TGCR	16	Timer Global Configuration Register	0000	H	TIMER

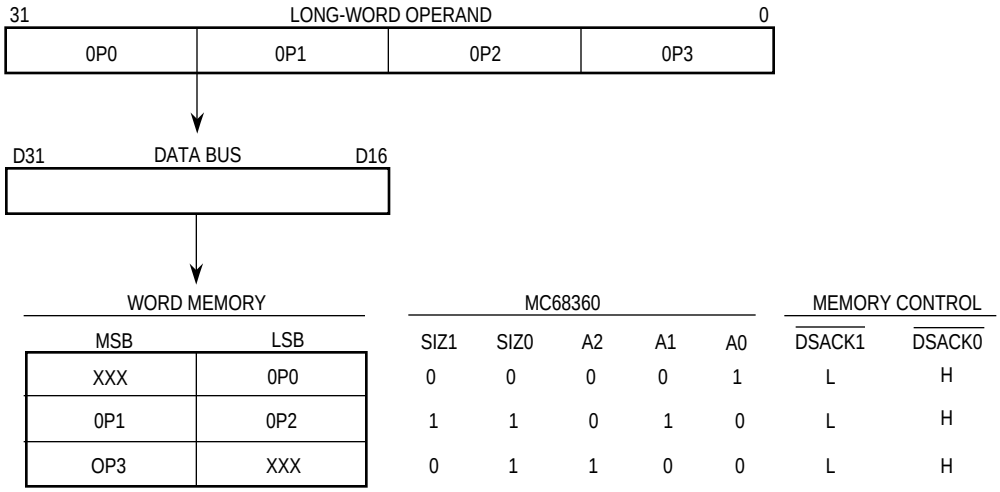
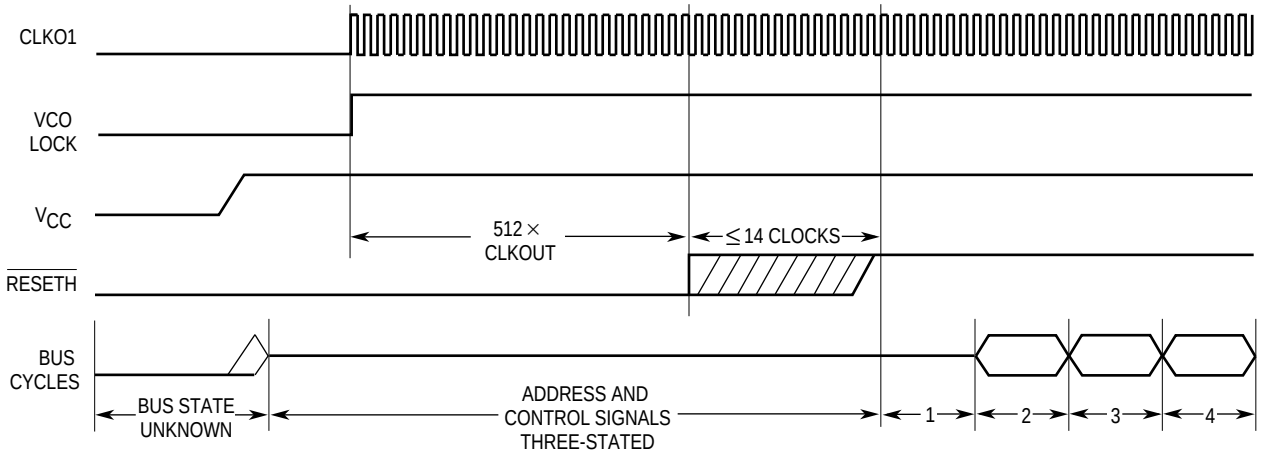


Figure 4-8. Misaligned Long-Word Transfer to Word Port Example

Once $\overline{\text{RESETH}}$ and $\overline{\text{RESETS}}$ negate, all control signals are driven to their inactive state, the data bus is in read mode, and the address bus is driven. After this, the first bus cycle of the reset exception processing begins.



NOTES:
1. Internal start-up time.
2. SSP read here.
3. PC read here.
4. First instruction fetched here.
5. This figure is true when MODCK is 11 or 10.
   When MODCK is 01 CLKO1 will be driven high at power up.

Figure 4-47. Initial Reset Operation Timing

NOTE

The PLL samples the MODCLK pins while in the first 512 clocks of RESET. The process starts with RESET being asserted, then MODCLK pins are sampled and the PLL is initialized according to the MODCLK pins. For the next 500-2000 EXTAL cycles the PLL is synchronizing. 512 clocks after the PLL synchronizes, the QUICC no longer drives RESET and does not sample the MOD-CLK pins.

User should make suer the ramp up time of Vcc will never be faster than 4mSec to ensure proper power on reset sequence.

When a RESET instruction is executed, the QUICC drives the $\overline{\text{RESETS}}$ signal for 512 clock cycles. In this case, the QUICC resets the external devices of the system, and many of the internal registers of the QUICC (see Section 3 QUICC Memory Map for a list of registers affected by each type of reset).

The bus arbitration circuitry is only reset during a power-on reset. It may be used during all other resets.

In QUICC slave mode (disable CPU32+) the reset operates the same as in the normal (mas-ter) mode except that the RESET instruction does not exist.

Interrupt recognition and subsequent processing are based on internal interrupt request signals ($\overline{IRQ7}$–$\overline{IRQ1}$) and the current priority set in SR priority mask I2–I0. Interrupt request level 0 ($\overline{IRQ7}$–$\overline{IRQ1}$ negated) indicates that no service is requested. When an interrupt of level 1 through 6 is requested via $\overline{IRQ6}$–$\overline{IRQ1}$, the processor compares the request level with the interrupt mask to determine whether the interrupt should be processed. Interrupt requests are inhibited for all priority levels less than or equal to the current priority. Level 7 interrupts are nonmaskable.

$\overline{IRQ7}$–$\overline{IRQ1}$ are synchronized and debounced by input circuitry on two consecutive rising edges of the processor clock.

Interrupt requests do not force immediate exception processing, but are left pending. A pending interrupt is detected between instructions or at the end of exception processing—all interrupt requests must be held asserted until they are acknowledged by the CPU. If the priority of the interrupt is greater than the current priority level, exception processing begins.

Exception processing occurs as follows. First, the processor makes an internal copy of the SR. After the copy is made, the processor state bits in the SR are changed—the S-bit is set, establishing supervisor access level, and bits T1 and T0 are cleared, disabling

tracing. Priority level is then set to the level of the interrupt, and the processor fetches a vector number from the interrupting device (CPU space $F). The fetch bus cycle is classified as an interrupt acknowledge, and the encoded level number of the interrupt is placed on the address bus.

If an interrupting device requests automatic vectoring, the processor generates a vector number (25 to 31) determined by the interrupt level number.

If the response to the interrupt acknowledge bus cycle is a bus error, the interrupt is taken to be spurious, and the spurious interrupt vector number (24) is generated.

The exception vector number, PC, and SR are saved on the supervisor stack. The saved value of the PC is the address of the instruction that would have executed if the interrupt had not occurred.

Priority level 7 interrupt is a special case. Level 7 interrupts are nonmaskable interrupts (NMI). $\overline{IRQ7}$ is a level sensitive input and must remain low until CPU32+ returns a n interrupt acknowledge cycle for level 7 interrupt.

Many M68000 peripherals provide for programmable interrupt vector numbers to be used in the system interrupt request/acknowledge mechanism. If the vector number is not initialized after reset and if the peripheral must acknowledge an interrupt request, the peripheral should return the uninitialized interrupt vector number (15).

See Section 4 Bus Operation for detailed information on interrupt acknowledge cycles.

5.5.2.12 RETURN FROM EXCEPTION. When exception stacking operations for all pending exceptions are complete, the processor begins execution of the handler for the last exception processed. After the exception handler has executed, the processor must restore

the system context in existence prior to the exception. The RTE instruction is designed to accomplish this task.

When RTE is executed, the processor examines the stack frame on top of the supervisor stack to determine if it is valid and determines what type of context restoration must be performed. See 5.5.4 CPU32+ Stack Frames for a description of stack frames.

For a normal four-word frame, the processor updates the SR and PC with data pulled from the stack, increments the SSP by 8, and resumes normal instruction execution. For a six-word frame, the SR and PC are updated from the stack, the active SSP is incremented by 12, and normal instruction execution resumes.

For a bus fault frame, the format value on the stack is first checked for validity. In addition, the version number on the stack must match the version number of the processor that is attempting to read the stack frame. The version number is located in the most significant byte (bits 15–8) of the internal register word at location SP + $14 in the stack frame. The validity check ensures that stack frame data will be properly interpreted in multiprocessor systems.

If a frame is invalid, a format error exception is taken. If it is inaccessible, a bus error exception is taken. Otherwise, the processor reads the entire frame into the proper internal registers, de-allocates the stack (12 words), and resumes normal processing. Bus error frames for faults during exception processing require the RTE instruction to rewrite the faulted stack frame. If an error occurs during any of the bus cycles required by rewrite, the processor halts.

If a format error occurs during RTE execution, the processor creates a normal four-word fault stack frame below the frame that it was attempting to use. If a bus error occurs, a bus-error stack frame will be created. The faulty stack frame remains intact, so that it may be examined and repaired by an exception handler or used by a different type of processor (e.g., MC68010, MC68020, or future M68000 processor) in a multiprocessor system.

5.5.3 Fault Recovery

There are four phases of recovery from a fault: recognizing the fault, saving the processor state, repairing the fault (if possible), and restoring the processor state. Saving and restoring the processor state are described in the following paragraphs.

The stack contents are identified by the special status word (SSW). In addition to identifying the fault type represented by the stack frame, the SSW contains the internal processor state corresponding to the fault.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP	MV	SZC1	TR	B1	B0	RR	RM	IN	RW	SZC0	SIZ		FUNC		

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)

NOTES

When used in slave mode, the QUICC must be configured with a 32-bit data bus.

Even without the use of the slave mode, another processor can be granted access to the QUICC's on-chip peripherals by requesting the bus with the $\overline{BR}$ pin.

6.8.1 MBAR in a Multiple QUICC System

The module base address register (MBAR) is used to configure the location of the QUICC's block of on-chip RAM and registers. In a multiple QUICC system, a technique must be provided to allow multiple MBARs on multiple QUICCs to be programmed with unique values. The QUICC has several provisions to support this.

First, any QUICC that is configured into slave mode with its global chip select disabled (CONFIG pins = 110) automatically has its MBAR location changed from $0003FF00 to $0003FF04. Second, the MBAR, newly located at address $0003FF04, can only be enabled for access after a keyed write operation is performed (see Figure 6-9). The keyed write allows the user to program the MBARs of multiple QUICC slaves without adding any external glue logic.

NOTES

If the QUICC is configured into slave mode with its global chip select enabled, the MBAR location does not change, and the keyed write is not required. Thus, a single QUICC configured as a slave to an MC68EC040 or MC68EC030 does not require a keyed write for its MBAR.

If there are N QUICCs sharing a bus, N–1 QUICCs would normally have their CONFIG pins configured as 110.

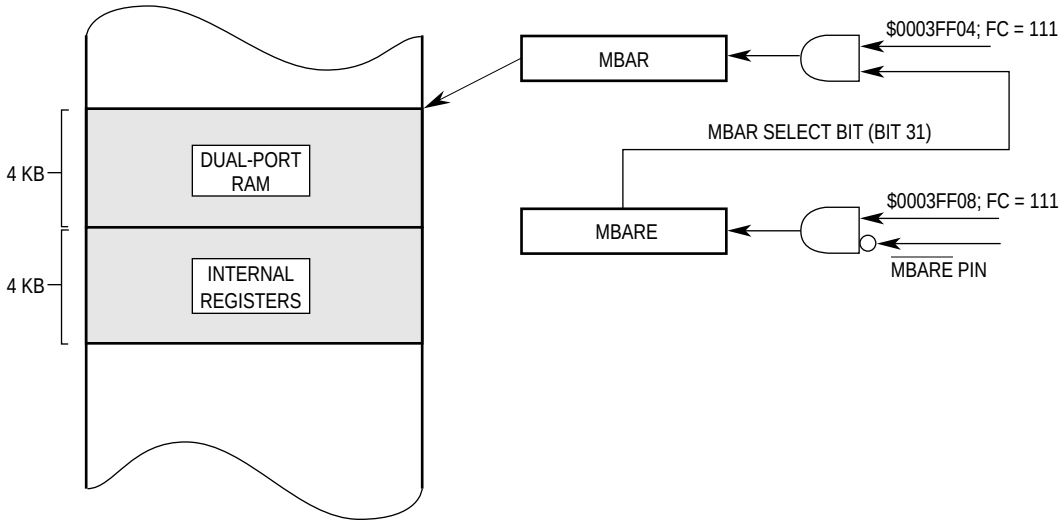


Figure 6-9. MBAR Access to a Multiple QUICC Slave System

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR040ID2–BR040ID0			—	—	—	—	—	—	—	—	—	—	—	—	BSTM
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASTM	FRZ1–FRZ0		BCLROID2–BCLROID0			SHEN1–SHEN0		SUPV	BCLRISM2–BCLRISM0 or BCLRIID2–BCLRIID0			IARB3–IARB0			
0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1

BR040ID2–BR040ID0—Bus Request MC68040 Arbitration ID

These bits contain the arbitration priority level for the MC68040 $\overline{BR}$ signal when the QUICC is in MC68040 companion mode; otherwise, this value is ignored. The MC68040 $\overline{BR}$ signal in companion mode) is reflected on the IMB with the bus arbitration level corresponding to these bits. This method gives the user a choice of where to place the arbitration level of the MC68040 (and other external masters in this system) relative to the IDMA, SDMA, or DRAM refresh cycles generated by the QIUCC.

NOTE

In a typical configuration, the user would program this value to a 3 to give the MC68040 priority over the IDMAs, but not over the SDMAs and the DRAM refresh cycle. If the SDMAs, however, are not of extremely high priority, the user may choose this value to be 5. User should never program this field to be 7.

Bits 28–17—Reserved

BSTM—Bus Synchronous Timing Mode

This bit determines whether the EBI will synchronize the $\overline{AS}$ and $\overline{DS}$ bus signals used for an external master's access into the QUICC peripherals and for $\overline{CS}$ and $\overline{RAS}$ generation by the QUICC. The synchronization will add a one-clock delay to the $\overline{RAS}$/$\overline{CS}$ assertion for an external master. The MC68EC040 signals must always be synchronized to the QUICC clock, regardless of the setting of this bit. See 6.10 Memory Controller for recommendations on the setting of BSTM in certain situations.

0 = Asynchronous timing on the bus signals may be used. The bus signals are synchronized internally by the QUICC and do not have to meet any timings relative to the system clock.

1 = Synchronous timing on the bus signals must be used. The bus control signals will not be synchronized internally and therefore must meet the system clock setup and hold timings.

NOTE

$\overline{BCLRI}$, Address, Data, $\overline{DSACK}$, $\overline{BERR}$, $\overline{HALT}$, $\overline{RESETH}$, and $\overline{RESETS}$ are always asynchronous.

IDMA channel will terminate the transfer of a block of memory if this register reaches zero during operation.

7.6.2.7 CHANNEL STATUS REGISTER (CSR). The CSR is an 8-bit register used to report events recognized by the IDMA controller. On recognition of an event, the IDMA sets its corresponding bit in the CSR, regardless of the corresponding bits in the CMAR. The CSR is a memory-mapped register that may be read at any time. A bit is reset by writing a one and is left unchanged by writing a zero. More than one bit may be reset at a time, and the register is cleared by reset.

7	6	5	4	3	2	1	0
—		AD	BRKP	OB	BES	BED	DONE

Bits 7–6—Reserved

AD—Auxiliary Done

This bit is valid in auto buffer and buffer chaining modes. It is set when the IDMA channel has completed a buffer transfer for a buffer descriptor (BD) that has its I-bit set. For AD to be set, the BCR must have been decremented to zero with no errors occurring during any IDMA transfer bus cycle. The IDMA will then move to the next BD and continue to transfer data.

BRKP—Breakpoint

This bit indicates that the breakpoint signal was asserted during an IDMA transfer. This bit is cleared by writing a one or by reset. Writing a zero has no effect on BRKP.

OB—Out of Buffers

This bit is valid only when the RISC controls the IDMA (RCI bit in the CMR is set). It is set when working with the RISC controller and there are no more valid buffers out of which to transfer data.

BES—Bus Error Source

This bit indicates that the IDMA channel terminated with an error during the read cycle. The channel terminates the IDMA operation without setting DONE. BES is cleared by writing a one or by setting RST in the CMR. Writing a zero has no effect on BES.

BED—Bus Error Destination

This bit indicates that the IDMA channel terminated with an error during the write cycle. The channel terminates the IDMA operation without setting DONE. BED is cleared by writing a one or by setting RST in the CMR. Writing a zero has no effect on BED.

DONE—Normal Channel Transfer Done

This bit indicates that the IDMA channel has terminated normally. Normal channel termination is defined as follows:

1. In single buffer mode, the BCR has decremented to zero, and no errors have occurred during any IDMA transfer bus cycle.

SMC2CS—SMC2 Clock Source (NMSI mode)

SMC2 can take its clocks from one of the baud rate generators or one of four pins from the bank of clocks. The SMC2 transmit and receive clocks must be the same when it is connected to the NMSI.

000 = SMC2 transmit and receive clocks are BRG1.
001 = SMC2 transmit and receive clocks are BRG2.
010 = SMC2 transmit and receive clocks are BRG3.
011 = SMC2 transmit and receive clocks are BRG4.
100 = SMC2 transmit and receive clocks are CLK5.
101 = SMC2 transmit and receive clocks are CLK6.
110 = SMC2 transmit and receive clocks are CLK7.
111 = SMC2 transmit and receive clocks are CLK8.

SMC1CS—SMC1 Clock Source (NMSI mode)

SMC1 can take its clocks from one of the baud rate generators or one of four pins from the bank of clocks. The SMC1 transmit and receive clocks must be the same when it is connected to the NMSI.

000 = SMC1 transmit and receive clocks are BRG1.
001 = SMC1 transmit and receive clocks are BRG2.
010 = SMC1 transmit and receive clocks are BRG3.
011 = SMC1 transmit and receive clocks are BRG4.
100 = SMC1 transmit and receive clocks are CLK1.
101 = SMC1 transmit and receive clocks are CLK2.
110 = SMC1 transmit and receive clocks are CLK3.
111 = SMC1 transmit and receive clocks are CLK4.

SDMx—SI Diagnostic Mode for TDM A or B

00 = Normal operation.
01 = Automatic Echo. In this mode, the channel_x transmitter automatically retransmits the TDM received data on a bit-by-bit basis. The receive section operates normally, but the transmit section can only retransmit received data. In this mode, the L1GRx line is ignored.
10 = Internal Loopback. In this mode, the TDM transmitter output is internally connected to the TDM receiver input (L1TXDx is connected to L1RXDx). The receiver and transmitter operate normally. The data appears on the L1TXDx pin. In this mode, the L1RQx line is asserted normally. The L1GRx line is ignored.
11 = Loopback Control. In this mode, the TDM transmitter output is internally connected to the TDM receiver input (L1TXDx is connected to L1RXDx). The transmitter output (L1TXDx) and the L1RQx pin will be inactive. This mode is used to accomplish loopback testing of the entire TDM without affecting the external serial lines.

NOTE

In modes 01,10, and 11, the receive and the transmit clocks should be identical.

Transmission of out-of-sequence characters is also supported by the UART and is normally used for the transmission of flow control characters such as XON or XOFF. This procedure is performed using the TOSEQ entry in the UART parameter RAM.

The UART will poll TOSEQ whenever the transmitter is enabled for UART operation. This includes during UART freeze operation, during UART buffer transmission, and when no buffer is ready for transmission. The TOSEQ character is transmitted at a higher priority than the other characters in the transmit buffer (if any), but does not preempt characters already in the transmit FIFO. This means that the XON or XOFF character may not be transmitted for eight character times (SCC1) or four character times (SCC2, SCC3, and SCC4). To reduce this latency, the TFL bit in the GSMR should be set to decrease the FIFO size to one character prior to enabling the SCC transmitter.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	REA	I	CT	0	0	A	CHARSEND							

Bits 15–14—Don't Care. May be written with ones or zeros.

The fact that these bits are don’t cares allows full compatibility between TOSEQ on the QUICC and CHARACTER8 on the MC68302.

REA—Ready

This bit is set by the CPU32+ core when the character is ready for transmission and will remain one while the character is being transmitted. The CP clears this bit after transmission.

I—Interrupt

If set, the CPU32+ core will be interrupted when this character has been transmitted. (The TX bit will be set in the UART event register.)

CT—Clear-to-Send Lost

This status bit indicates that the $\overline{\text{CTS}}$ signal was negated during transmission of this character. If this occurs, the CTS bit in the UART event register will also be set. This bit operates only if the $\overline{\text{CTS}}$ line is monitored by the SCC as determined by the DIAG bits.

NOTE

If the $\overline{\text{CTS}}$ signal was negated during transmission and the CP transmits this character in the middle of buffer transmission, the $\overline{\text{CTS}}$ signal could actually have been negated either during this character’s transmission or during a buffer character’s transmission. In this case, the CP sets the CT bit both here and in the Tx BD status word.

FLC—Flow Control

0 = Normal operation. The GSMR and port C registers determine the mode of the $\overline{\text{CTS}}$ pin.

1 = Asynchronous flow control. When the $\overline{\text{CTS}}$ pin is negated, the transmitter will stop transmitting at the end of the current character. (If $\overline{\text{CTS}}$ is negated past the middle of the current character, the next full character may be sent, and then transmission will be stopped.) When $\overline{\text{CTS}}$ is asserted once more, transmission will continue where it left off. No $\overline{\text{CTS}}$ lost error will be reported. No characters except idles will be transmitted while $\overline{\text{CTS}}$ is negated.

SL—Stop Length

The SL bit selects the number of the stop bits transmitted by the UART. This bit may be modified on the fly. The receiver is always enabled for one stop bit unless the UART is in synchronous mode and the RZS bit is set. Fractional stop bits are configured in the DSR.

0 = One Stop Bit
1 = Two Stop Bits

CL—Character Length

The CL bits determine the number of data bits in the character, not including the optional parity or multidrop address bits. When less than an 8-bit character is used, the MSBs in memory are written as zeros, and on transmission the MSBs in memory are a don’t care. These bits may be modified on the fly.

00 = 5 Data Bits
01 = 6 Data Bits
10 = 7 Data Bits
11 = 8 Data Bits

UM—UART Mode

The UART mode bits select the protocol that is implemented over the ASYNC channel. These bits may be modified on the fly.

00 = Normal UART operation. Multidrop mode is disabled, and an idle-line wake-up is selected. In the idle-line wake-up mode, the UART receiver is reenabled by receiving one character of all ones.

01 = Multidrop non-automatic mode. In the multidrop mode, an additional address/data bit is transmitted with each character. The multidrop asynchronous modes are compatible with the MC68681 DUART, the MC68HC11 SCI, the DSP56000 SCI, and the Intel 8051 serial interface. The UART receiver is reenabled when the last data bit received in the character (i.e., the address bit) is a one. This means that the received character is an address that has to be processed by all inactive processors. The UART receives the address character and writes it to a new buffer. The CPU32+ core then compares the written address with its own address to decide whether to ignore or process the following characters.

10 = Reserved

11 = Multidrop automatic mode. In this mode, the CP automatically checks the address of the incoming address character using the UADDR1 and UADDR2 parameter

than it is to the system integration block on the MC68302. The QUICC communications features, however, are very similar to those of the MC68302.

9.3.3 Notes About Porting

Although the following paragraphs show how to port MC68302 functions to the QUICC, it should not be assumed that the port operation will provide a complete initialization of the QUICC. The QUICC contains features not available on the MC68302 that may require additional initialization. For instance, the QUICC contains an on-chip PLL to generate a high-speed system clock frequency from a low-speed crystal, such as a 32-kHz crystal. Since this feature is not available on the MC68302, it is not mentioned in this discussion. Refer to 9.2 How to take A QUICC Software Test-Drive as a guide to complete QUICC initialization.

Although the QUICC was designed to allow a convenient upgrade path from the MC68302, this discussion does not guarantee that every MC68302 operation can be exactly and precisely duplicated on the QUICC. For instance, when using the serial channels, the time between setting the ready bit of a buffer descriptor and the assertion of the $\overline{\text{RTS}}$ pin may not be the same on the QUICC as the MC68302. Also, although the MC68302 hardware watchdog can be implemented in the QUICC bus monitor, the MC68302 maximum timeout is 16K clocks; whereas, the QUICC maximum timeout period is 1K clocks (before $\overline{\text{BERR}}$ is asserted on the system bus).

The QUICC offers features that simplify what would otherwise be a more code-intensive process. For specific examples, see the INIT TX AND RX PARAMETERS, the GRACEFUL STOP TRANSMIT, and the CLOSE BD commands. The user can do a direct port using the old commands and techniques; however, the new commands may be used in many instances to simplify the application process.

Additionally, the most direct port to the QUICC will not necessarily take advantage of a number of new QUICC features. For specific examples, see the dynamic allocation of SCC interrupt levels and the buffer descriptor capability of the independent DMA (IDMA) channels. Some comments will be made about these features where appropriate.

9.3.4 How To Port MC68302 Functions

The following paragraphs detail the different MC68302 functions and how/where to implement them on the QUICC. The MC68302 functions are listed in ascending order in the MC68302 memory map.

9.3.4.1 SYSTEM CONFIGURATION REGISTERS. The following paragraphs describe the MC68302 configuration registers.

9.3.4.1.1 Base Address Register (BAR). This register is most closely related to the MBAR on the QUICC:

The base address field is configured in the BA bits of the MBAR. Be sure to allow 8 Kbytes in the memory map of the QUICC system at this address. Also, the V-bit of MBAR must be set before the contents are valid.

The CFC bit and FC2–FC0 bits may be duplicated by setting all but one of the AS7–AS0 bits in the MBAR. Note that the MBAR offers many more options since multiple AS bits

BCYC1–BCYC0 may be set to zeros (no wait states) if the QUICC is controlling the bursting for the MC68EC040 and the timing supports one-clock MC68EC040 bursting. However, with 60- or 70-ns DRAMs at 25 MHz, BCYC1–BCYC0 should be set to 01 for two-clock MC68EC040 bursting.

PGME should be cleared.

SPS1–SPS0 should be cleared.

DSSEL should be set only if this is a DRAM bank.

9.4.4 Interfacing Multiple QUICCs to an MC68EC040

It is possible to interface multiple QUICCs to an MC68EC040. The first QUICC can be configured as previously shown in this subsection. Additional QUICCs should be configured as noted in the following list:

- The additional QUICCs should have their CONFIG2–CONFIG0 pins configured for slave mode, global chip select *disabled*, and MBAR at $003FF04. These QUICCs still recognize and respond to MC68040 cycles via the $\overline{TS}$ pin, even though their CONFIG2–CONFIG0 pins are not configured for MC68040 companion mode.
- The MBAR of the additional QUICCs should be programmed using the $\overline{MBARE}$ pin and MBARE register as described in Section 6 System Integration Module (SIM60).
- An external bus arbiter is required to take the bus request of the additional QUICC (which is an output because of the CONFIG2–CONFIG0 pins) and prioritize it with the MC68EC040, present it to the original QUICC, and issue a bus grant to the appropriate device.
- An external interrupt prioritizer is required to determine which QUICC $\overline{IOUT2}$–$\overline{IOUT0}$ pins are currently routed to the MC68EC040. Alternatively, the additional QUICC should have its interrupts brought out on a single $\overline{RQOUT}$ pin, which is routed to one of the original QUICC interrupt inputs. This would eliminate the external logic.
- The additional QUICCs should not be configured to perform any memory controller support functions for the MC68EC040. Only the original QUICC should be used for this purpose.

9.5 SELECTING CACHE MODES ON THE MC68EC040

When the QUICC is used in its MC68040 companion mode with the MC68040, it is recommended that the QUICC serial data buffers be cache inhibited by the MC68040. This avoids the overhead that would result from the cache coherency algorithms of the MC68040 following the frequent write accesses by the QUICC to one of its serial data buffers located in external memory. When using the MC68040, the MC68040 memory management unit (MMU) may be used to cache inhibit the data buffers. However, the lower cost MC68EC040 does not have an MMU. Therefore, what technique can be used by the MC68EC040 to cache inhibit serial data buffers? The following paragraphs discuss a method for selecting caching modes on 16-byte boundaries for an MC68EC040.

The MC68EC040 delivers high performance at a low system cost for embedded control by providing the same high integer performance and large 4K instruction/data caches as the MC68040 without the floating-point unit and MMU. The MC68040 MMU makes the caching

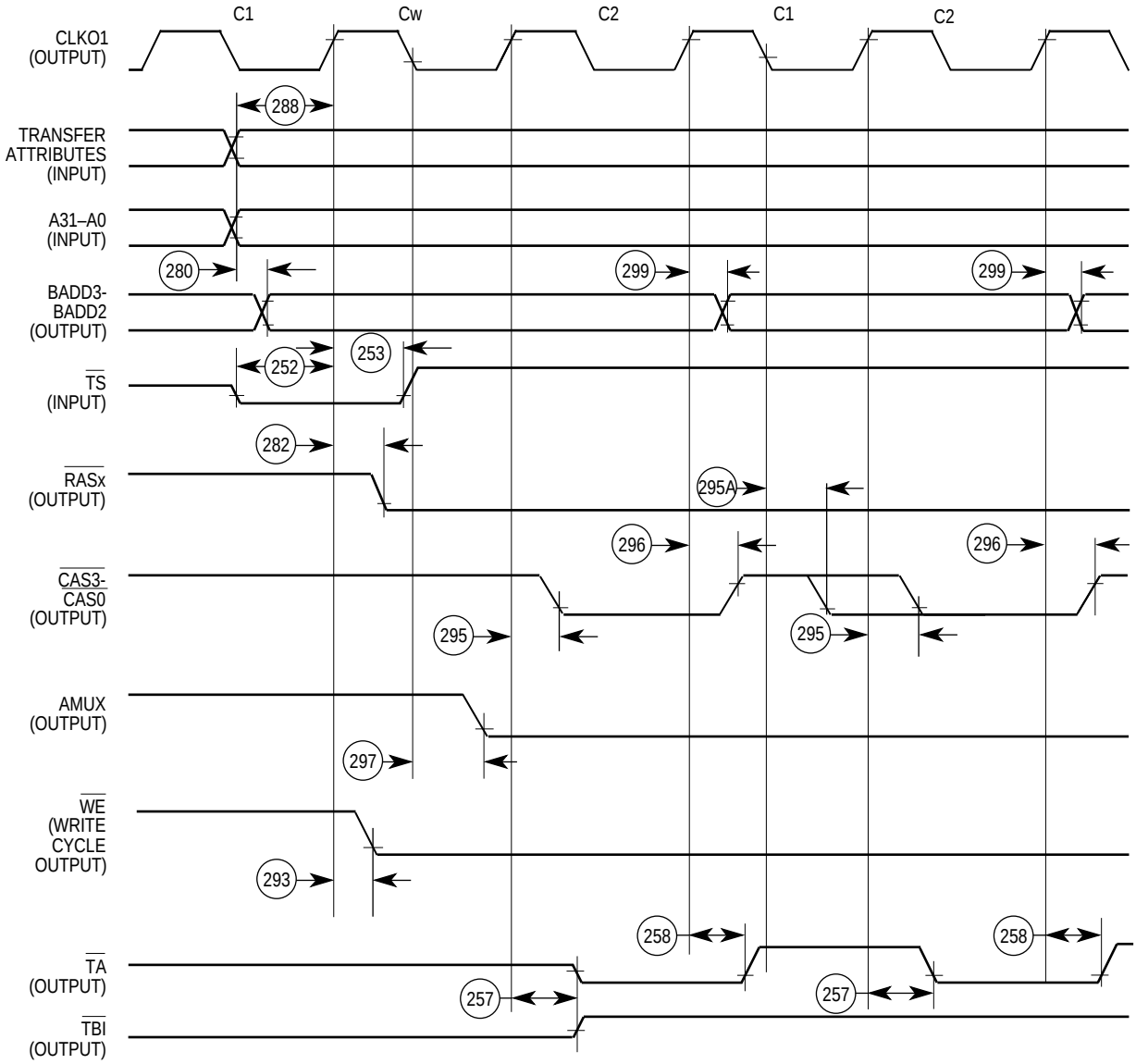


Figure 10-47. External MC68040 DRAM Burst Cycles Timing Diagram

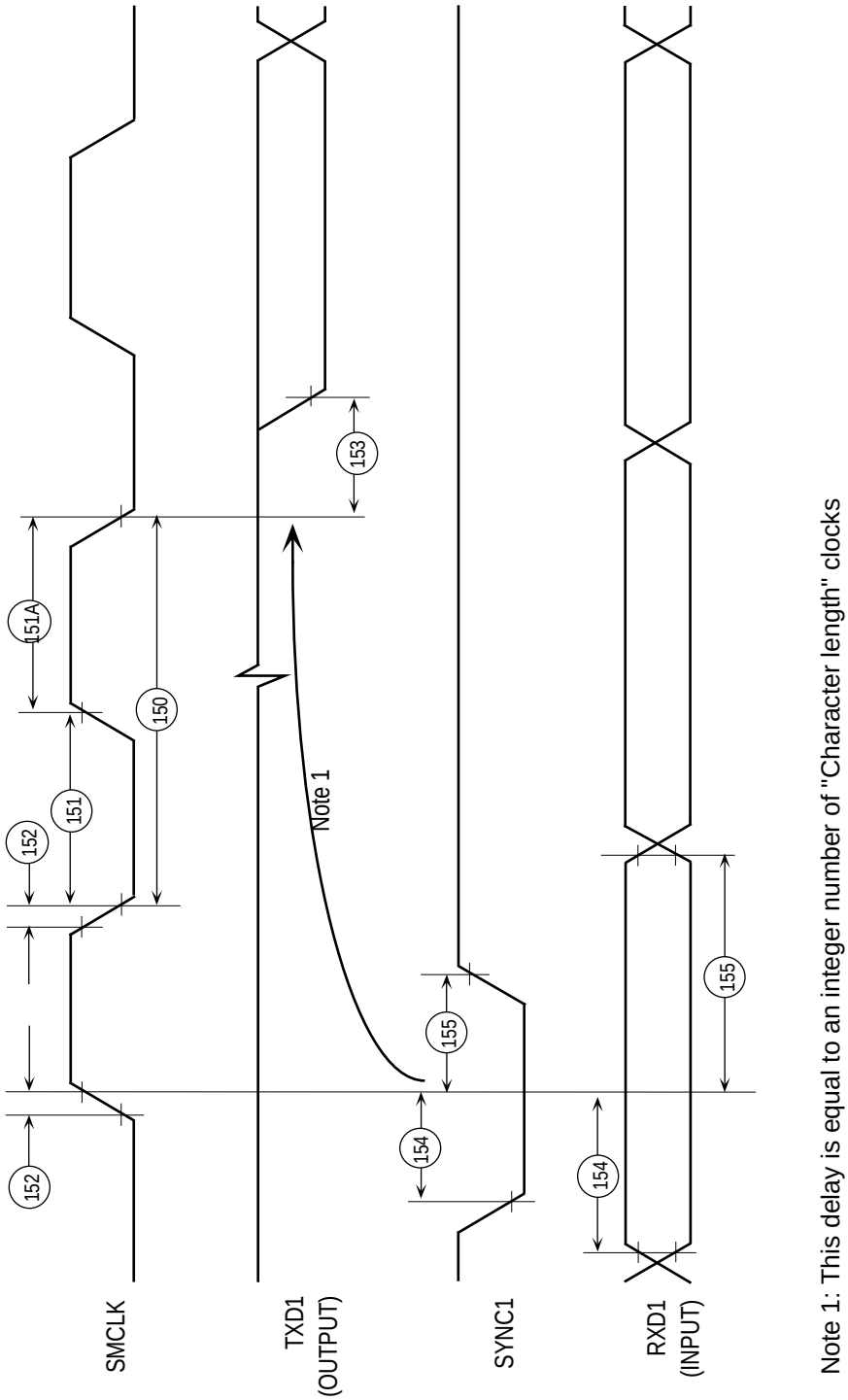


Figure 10-74. SMC Transparent

INDEX