Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	CPU32+
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	25MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	240-BFQFP
Supplier Device Package	240-FQFP (32x32)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68360em25k

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

4.1.1 Bus Control Signals

The QUICC initiates a bus cycle by driving the address, size, function code, and read/write outputs. At the beginning of a bus cycle, SIZ1 and SIZ0 are driven with the FC signals. SIZ1 and SIZ0 indicate the number of bytes remaining to be transferred during an operand cycle (consisting of one or more bus cycles). Table 4-3 lists the encoding of SIZ1 and SIZ0. These signals are valid while $\overline{AS}$ is asserted.

The R/$\overline{W}$ signal determines the direction of the transfer during a bus cycle. Driven at the beginning of a bus cycle, R/$\overline{W}$ is valid while $\overline{AS}$ is asserted. R/$\overline{W}$ only transitions when a write cycle is preceded by a read cycle or vice versa. The signal may remain low for consecutive write cycles.

The $\overline{RMC}$ signal is asserted at the beginning of the first bus cycle of a read-modify-write operation and remains asserted until completion of the final bus cycle of the operation.

4.1.2 Function Codes (FC3–FC0)

The FCx signals are outputs that indicate one of 16 address spaces to which the address applies. Fifteen of these spaces are designated as either a normal or DMA cycle, user or supervisor, and program or data spaces. One other address space is designated as CPU space to allow the CPU32+ to acquire specific control information not normally associated with read or write bus cycles. The FCx signals are valid while $\overline{AS}$ is asserted.

Function codes (see Table 4-1) can be considered as extensions of the 32-bit address that can provide up to eight different 4-Gbyte address spaces. Function codes are automatically generated by the CPU32+ to select address spaces for data and program at both user and supervisor privilege levels, and a CPU address space for processor functions. User programs access only their own program and data areas to increase protection of system integrity and can be restricted from accessing other information. The S-bit in the CPU32+ status register is set for supervisor accesses and cleared for user accesses to provide differentiation. Refer to 4.4 CPU Space Cycles for more information.

Table 4-1. Address Space Encoding

Function Code Bits				Address Spaces
3	2	1	0	
0	0	0	0	Reserved (Motorola)
0	0	0	1	User Data Space
0	0	1	0	User Program Space
0	0	1	1	Reserved (User)
0	1	0	0	Reserved (Motorola)
0	1	0	1	Supervisor Data Space
0	1	1	0	Supervisor Program Space
0	1	1	1	Supervisor CPU Space
1	x	x	x	DMA space

16 bits. SIZ0 and SIZ1 indicate that a word remains to be transferred; A0 and A1 indicate that the word corresponds to an offset of two from the base address. The multiplexer follows the pattern corresponding to this configuration of the size and address signals and places the two least significant bytes of the long word on the word portion of the bus (D16–D31). The bus cycle transfers the remaining bytes to the word-size port. Figure 4-5 shows the timing of the bus transfer signals for this operation.

Table 4-6. QUICC Internal to External Data Bus Multiplexer—Write Cycle

Transfer Size	Size		Address		External Data Bus Connection			
	SIZ1	SIZ0	A1	A0	D31:D24	D23:D16	D15:D8	D7:D0
Byte	0	1	0	0	OP3	x	x	x
	0	1	0	1	OP3	OP3	x	x
	0	1	1	0	OP3	x	OP3	x
	0	1	1	1	OP3	OP3	x	OP3
Word	1	0	0	0	OP2	OP3	x	x
	1	0	0	1	OP2	OP2	OP3	x
	1	0	1	0	OP2	OP3	OP2	OP3
	1	0	1	1	OP2	OP2	x	OP2
3 Bytes	1	1	0	0	OP1	OP2	OP3	x
	1	1	0	1	OP1	OP1	OP2	OP3
	1	1	1	0	OP1	OP2	OP1	OP2
	1	1	1	1	OP1	x	OP2	OP1
Long Word	0	0	0	0	OP0	OP1	OP2	OP3
	0	0	0	1	OP0	OP0	OP1	OP2
	0	0	1	0	OP0	OP1	OP0	OP1
	0	0	1	1	OP0	OP0	x	OP0

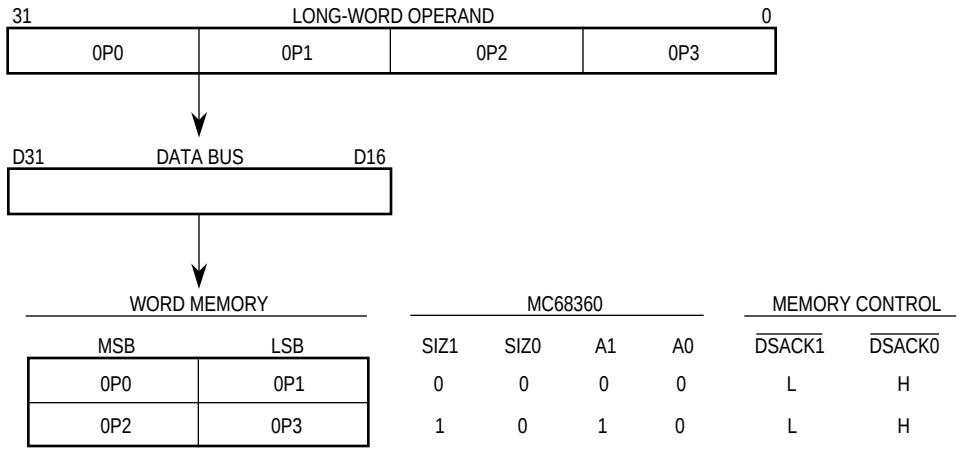


Figure 4-4. Example of Long-Word Transfer to Word Port

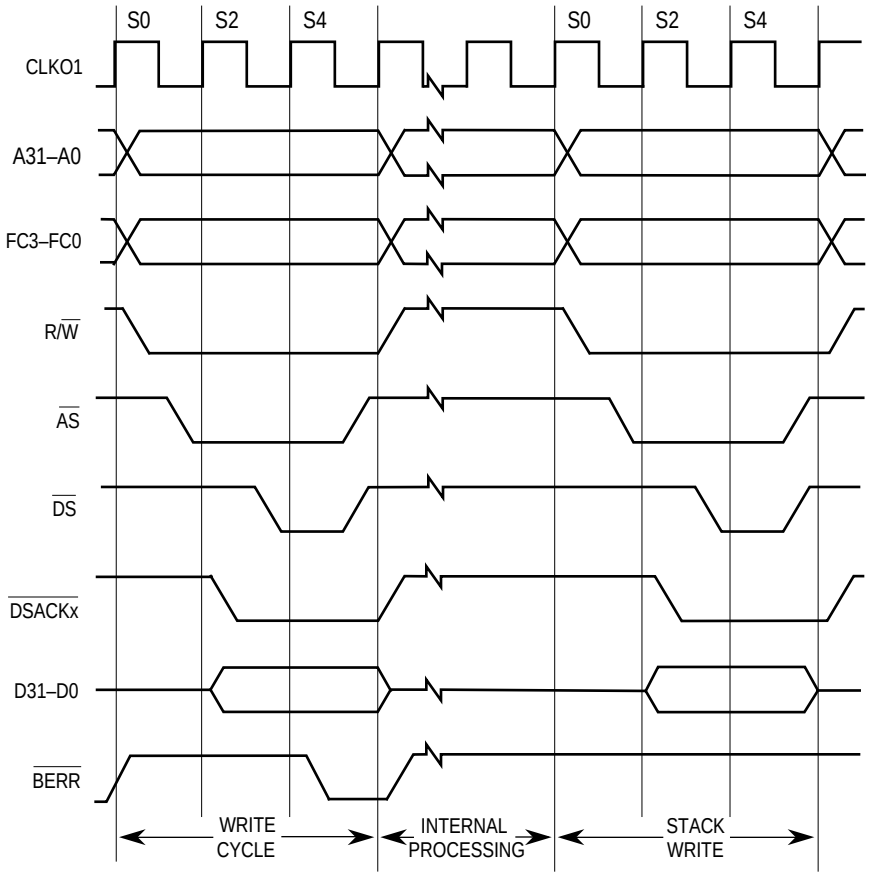


Figure 4-30. Late Bus Error with DSACKx

In the second case, in which BERR is asserted after DSACKx is asserted, BERR must be asserted within the time specified for purely asynchronous operation, or it must be asserted and remain stable during the sample window around the next falling edge of the clock after DSACKx is recognized. If BERR is not stable at this time, the QUICC may exhibit erratic behavior. BERR has priority over DSACKx. In this case, data may be present on the bus but may not be valid. This sequence can be used by systems that have memory error detection and correction logic and by external cache memories.

4.5.2 Retry Operation

When both BERR and HALT are asserted by an external device during a bus cycle, the QUICC enters the retry sequence shown in Figure 4-31. A delayed retry, which is similar to the delayed bus error signal described previously, can also occur (see Figure 4-32). The QUICC terminates the bus cycle, places the control signals in their inactive state, and does not begin another bus cycle until the BERR and HALT signals are negated by external logic. After a synchronization delay, the QUICC retries the previous cycle using the same access information (address, function code, size, etc.). BERR should be negated before S2 of the retried cycle to ensure correct operation of the retried cycle.

4.6.2 Bus Grant

The QUICC supports operand coherency; thus, if an operand transfer requires multiple bus cycles, the QUICC does not release the bus until the entire transfer is complete. The assertion of $\overline{BG}$ is therefore subject to the following constraints:

- The minimum time for $\overline{BG}$ assertion after $\overline{BR}$ is asserted depends on internal synchronization.
- When working in synchronous mode (ASTM bit in the MCR is set), the minimum time can be one clock.
- During an external operand transfer, the QUICC does not assert $\overline{BG}$ until after the last cycle of the transfer (determined by SIZx and $\overline{DSACKx}$).
- During an external operand transfer, the QUICC does not assert $\overline{BG}$ as long as $\overline{RMC}$ is asserted.
- If the show cycle bits SHEN1–SHEN0 = 1x and if one of the QUICC internal masters is making internal accesses, the QUICC does not assert $\overline{BG}$ until the transfer is terminated.
- If SHEN1–SHEN0 = 00 and if one of the QUICC internal masters is making internal accesses, the external bus is granted away, and the QUICC continues to execute internal bus cycles. In this case, the arbitration overhead (external bus idle time) is minimal.
- If SHEN1–SHEN0 = 01, the QUICC does not assert $\overline{BG}$ to an external master.

Externally, the $\overline{BG}$ signal can be routed through a daisy-chained network or a priority-encoded network. The QUICC is not affected by the method of arbitration as long as the protocol is obeyed.

4.6.3 Bus Grant Acknowledge

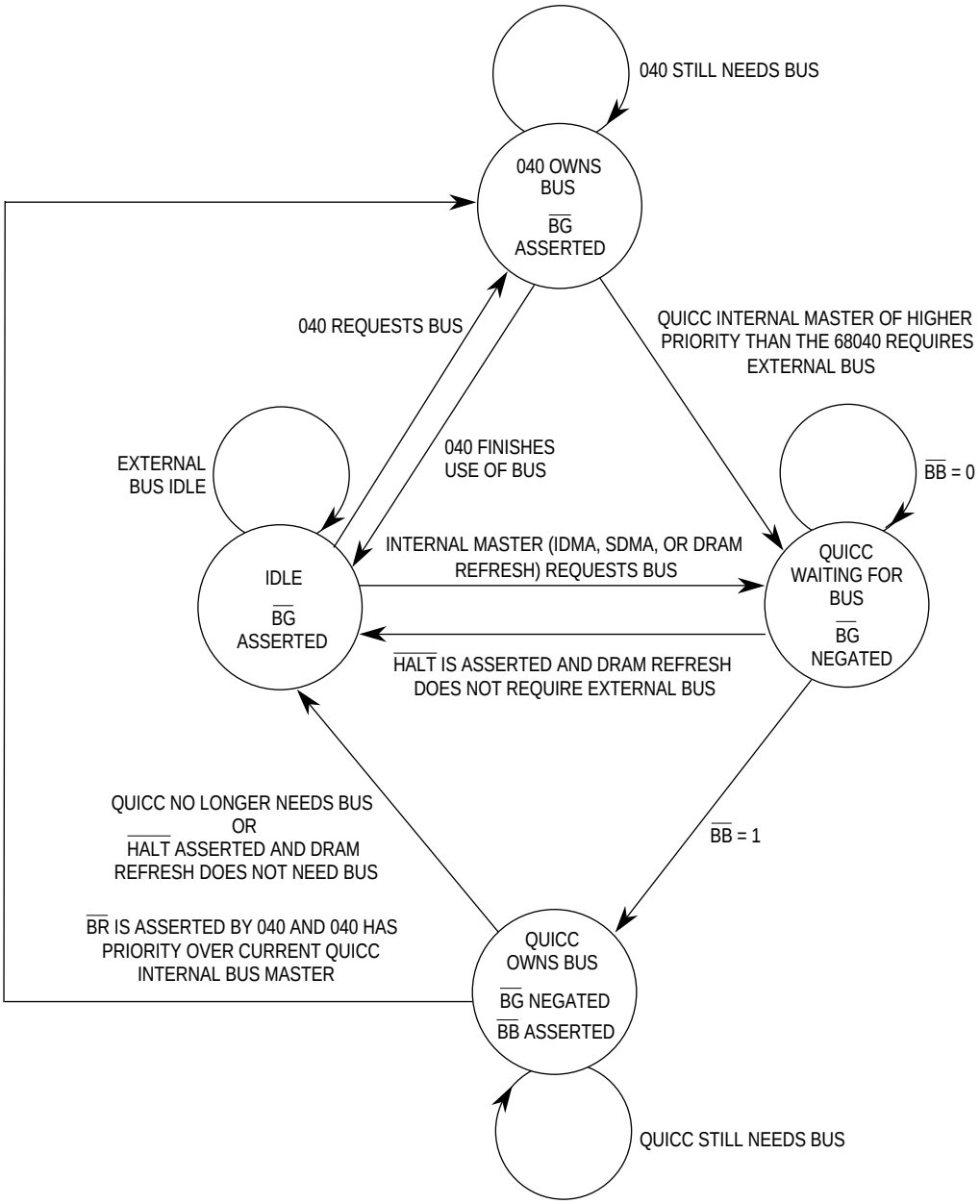
An external device cannot request and be granted the external bus while another device is the active bus master. A device that asserts $\overline{BGACK}$ remains the bus master until it negates $\overline{BGACK}$. $\overline{BGACK}$ should not be negated until all required bus cycles are completed. Bus mastership is terminated at the negation of $\overline{BGACK}$. When no other device requests the bus after $\overline{BGACK}$ is negated, the QUICC will regain bus mastership.

The minimum time for the first bus cycle after $\overline{BGACK}$ negation depends on internal synchronization and internal bus arbitration. This timing is therefore subject to the following constraints:

- When working in synchronous mode (ASTM bit in the MCR is set) and SHEN0–SHEN1 = 00 and one of the QUICC internal masters requests an external accesses, the minimum time can be one clock.
- When working in asynchronous mode (ASTM bit in the MCR is cleared) and SHEN0–1 = 00 and one of the QUICC internal masters requests an external accesses, the minimum time depends on internal synchronization plus one clock.
- If SHEN1–SHEN0 = 1×, another clock is added for internal bus arbitration.

3. If the 68040 requests the bus at the same time that a QUICC internal master is requesting the bus, the BR040ID bits are used to determine who will acquire the bus first.
4. When the QUICC no longer needs the bus, it deasserts $\overline{BB}$ and asserts $\overline{BG}$.

The state machine for the MC68040 companion mode arbitration is shown in Figure 4-39.



NOTES:
1. If the 68040 and the QUICC Internal Master requests the bus at the same time, the highest priority requester wins.
2. The transition from "040 Owns Bus" to "QUICC Waiting for Bus" may be delayed, until the write portion of an 040 locked cycle if an 040 locked cycle is in progress when the higher priority QUICC internal master requests the bus.
3. $\overline{BB}$ is only asserted by QUICC during the state "QUICC Owns Bus", otherwise $\overline{BB}$ is three-stated by the QUICC.

Figure 4-39. MC68040 Companion Mode Bus Arbitration State Machine

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

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

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

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

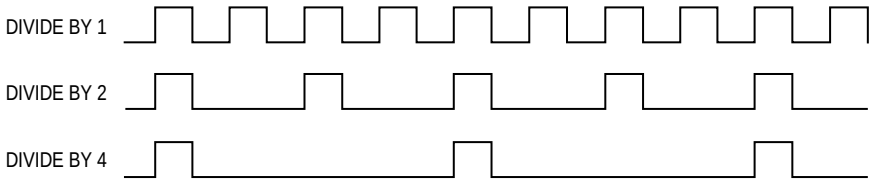


Figure 6-8. Divided Clocks

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

NOTES

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

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

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

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

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.

ASTM—Arbitration Synchronous Timing Mode

This bit determines whether the EBI will synchronize the arbitration signals: $\overline{BR}$, $\overline{BG}$, and $\overline{BGACK}$. The synchronization will add a one-clock delay to the external bus arbitration.

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

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

FRZ1—Freeze SWT and PIT Enable

0 = When FREEZE is asserted, the SWT and the PIT counters continue to run. See 6.3.3 Freeze Support for more information.

1 = When FREEZE is asserted, the SWT and the PIT counters are disabled, preventing interrupts from occurring during software debugging.

FRZ0—Freeze Bus Monitor Enable

0 = When FREEZE is asserted, the bus monitor continues to operate as programmed.

1 = When FREEZE is asserted, both the internal and external bus monitors are disabled.

BCLROID2–BCLROID0—Bus Clear Out Arbitration ID

These bits contain the arbitration priority level for the assertion of the $\overline{BCLRO}$ signal. When internal masters (IDMA, SDMA, or DRAM refresh cycle) request the bus and the arbitration level on the IMB is greater than the bus clear out arbitration ID, the $\overline{BCLRO}$ signal will be asserted until the arbitration level is less than or equal to the bus clear out arbitration ID. $\overline{BCLRO}$ can be used to clear an external master from the external bus when a refresh cycle is pending. It may also be used to clear an external master from the bus when an SDMA or IDMA channel requests the external bus.

NOTE

Program this value to 3 in a normal system to allow the SDMA and DRAM refresh controller to clear other bus masters off the external bus.

SHEN1–SHEN0—Show Cycle Enable

These two control bits determine what the EBI does with the external bus during internal transfer operations (see Table 6-3). A show cycle allows internal transfers to be externally monitored. The address, data, and control signals (except for $\overline{AS}$) are driven externally. $\overline{DS}$ is used to signal address strobe timing for show cycles. Data is valid on the next falling clock edge after $\overline{DS}$ is negated. However, data is not driven externally and $\overline{AS}$ and $\overline{DS}$ are not asserted externally for internal accesses unless show cycles are enabled.

If external bus arbitration is disabled, the EBI will not recognize an external bus request until arbitration is enabled again. When SHEN1 is set, an external bus request causes an internal master to stop its current cycle and relinquish the internal bus. The internal master resumes running cycles on the bus after $\overline{BR}$ and $\overline{BGACK}$ are negated. To prevent bus

NOTES

This mode is used in QUICC slave operation to assert either the $\overline{\text{BKPTO}}$ line for the external CPU or the internal IMB $\overline{\text{BKPT}}$ line for an internal-to-internal IDMA/SDMA access. When the external bus is used, the breakpoint line will be asserted as if the SIZM bit is set.

In the case of an external MC68040 burst, only the first address of the burst is checked.

When the QUICC is in master mode this bit should be zero to prevent external breakpoint from being ignored.

RW1–RW0—Read/Write Selection

Assert a breakpoint match on read cycles only, write cycles only, or on both.

00 = Assert breakpoint on read cycles.
01 = Assert breakpoint on write cycles.
10 = Assert breakpoint on read or write cycles.
11 = Reserved.

SIZM—Size Mask

This bit determines whether the breakpoint logic will use the SIZ bits to determine whether a breakpoint match has occurred.

0 = Compare the size lines as programmed in the SIZ bits to determine whether a breakpoint match has occurred.

NOTE

This mode would normally be used to break on an access to a location that contains data.

1 = Mask the size lines. The size of the access is not used in determining whether a breakpoint match has occurred. The breakpoint logic will assert the break signal when the address and size overlaps the programmable value. For example if the programmable address is xxx2, the breakpoint line for the low word will be asserted when the access address is xxx2 with a word size or when the address is xxx0 with a long-word size.

NOTE

This mode would normally be used to break on an instruction fetch.

SIZ1–SIZ10—Size Bits

The breakpoint logic can cause a breakpoint match for accesses that correspond to the size of the access. Set the SIZM bit to disable this feature.

—CSR1 = $FF. Clear any CSR bits that are currently set.
—CMAR1 = $00. Disable interrupts for this example.
—CMR1 = $47A1. Internal maximum transfer rate; starts IDMA.

Bus Access #	Address (Hex)	Operation	No. Bytes	No. Bytes in DHR
1	$00000001	Read	1	1
2	$00000002	Read	2	3
3	$20000000	Write	2	1
4	$00000002	Write	1	0

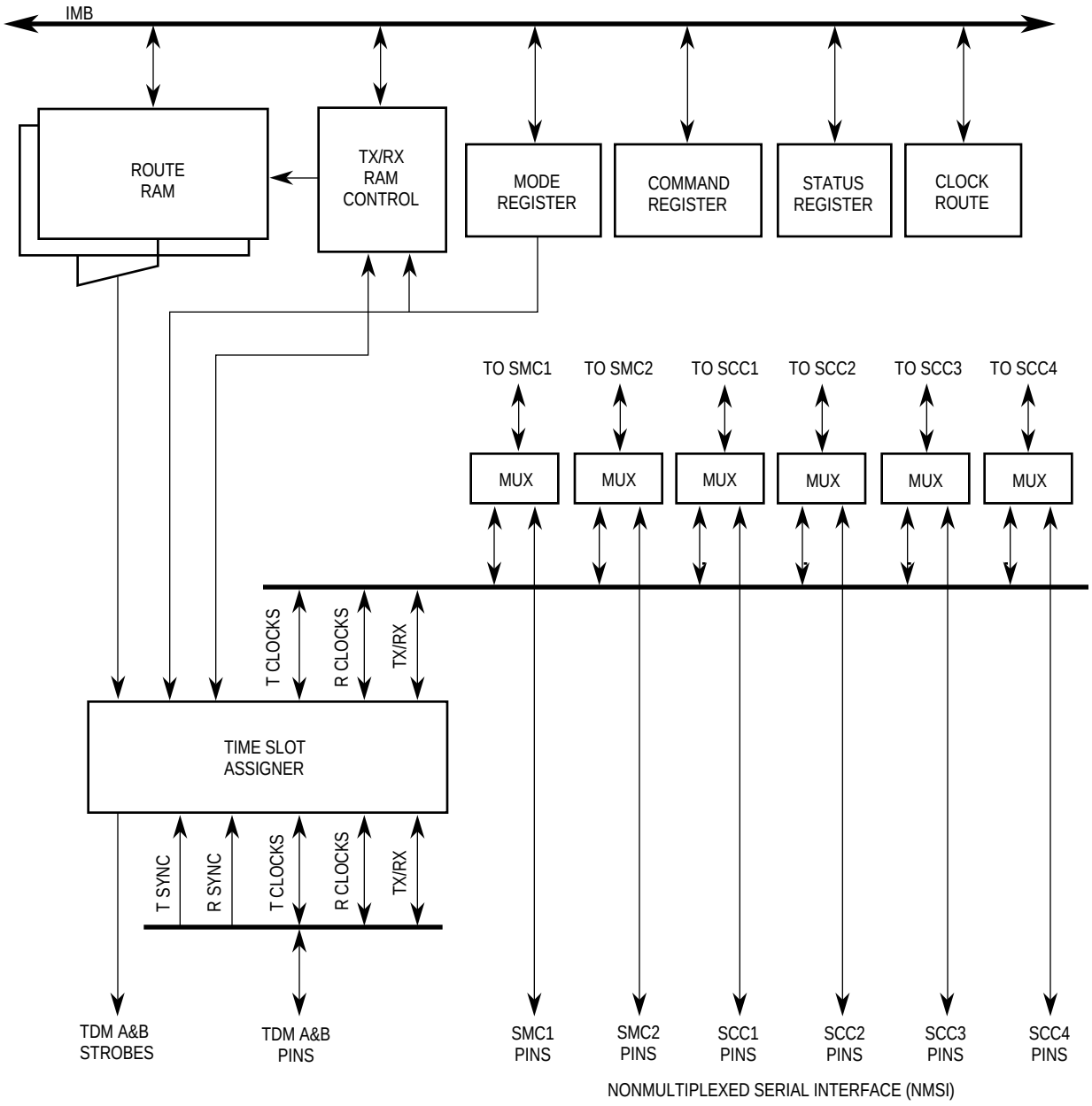
Example 2. This more complicated example shows how packing is performed when the source and destination sizes are the same—long word. This example also shows the entire 7-byte DHR in use. The source address is $00000000, and the destination address is $20000003. The number of bytes to be transferred is 16.

IDMA channel 1 initialization required for this example:

- ICCR = $0720. Recommended normal configuration.
- FCR1 = $89. Source function code is 1000; destination function code is 1001.
- SAPR1 = $00000000. Source address.
- DAPR1 = $20000003. Destination address.
- BCR1 = $00000010. Byte transfer count.
- CSR1 = $FF. Clear any CSR bits that are currently set.
- CMAR1 = $00. Disable interrupts for this example.
- CMR1 = $4701. Internal maximum transfer rate; starts IDMA.

Bus Access #	Address (Hex)	Operation	No. Bytes	No. Bytes in DHR
1	$00000000	Read	4	4
2	$20000003	Write	1	3
3	$00000004	Read	4	7
4	$20000004	Write	4	3
5	$00000008	Read	4	7
6	$20000008	Write	4	3
7	$0000000C	Read	4	7
8	$2000000C	Write	4	3
9	$20000010	Write	2	1
10	$20000012	Write	1	0

Example 3. This example shows how packing operates when the source and destination sizes are different. The source address is $00000002, and the destination address is $20000002. The source size is long word, and the destination size is byte. The number of bytes to be transferred is 8.



NOTE: NMSI clocking paths are not shown.

Figure 7-19. SI Block Diagram

The NMSI contains the following features:

- Each SCC and SMC Can Be Independently Programmed To Work with Its Own Set of Pins in a Nonmultiplexed Manner.
- Each SCC Can Have Its Own Set of Modem Control Pins (TXD, RXD, TCLK, RCLK, $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{CD}}$).
- Each SMC Can Have Its Own Set of Four Pins (SMTXD, SMRXD, CLK, and SMSYN).
- Each SCC and SMC Can Derive Clocks Externally from a Bank of Eight Clock Pins (CLK1–CLK8) or a Bank of Four Baud Rate Generators (BRG1–BRG4).

any dynamic change in its parallel I/O ports or serial channel physical interface configuration. A full reset using the RST bit in the CR is a comprehensive reset that may also be used.

7.10.10 SCC Interrupt Handling

The following describes what would normally take place within an interrupt handler for the SCC.

1. Once an interrupt occurs, the SCCE should be read by the user to see which sources have caused interrupts. The SCCE bits that are going to be "handled" in this interrupt handler would normally be cleared at this time.
2. Process the Tx BDs to reuse them if the TX bit or TXE bit was set in SCCE. If the transmit speed is fast or the interrupt delay is long, more than one transmit buffer may have been sent by the SCC. Thus, it is important to check more than just one Tx BD during the interrupt handler. One common practice is to process all Tx BDs in the interrupt handler until one is found with its R-bit set.
3. Extract data from the Rx BD if the RX, RXB, or RXF bit was set in SCCE. If the receive speed is fast or the interrupt delay is long, more than one receive buffer may have been received by the SCC. Thus, it is important to check more than just one Rx BD during the interrupt handler. One common practice is to process all Rx BDs in the interrupt handler until one is found with its E-bit set.
4. Clear the SCCx bit in the CISR.
5. Execute the RTE instruction.

7.10.11 SCC Timing Control

When the DIAG bits of the GSMR are programmed to normal operation, the $\overline{CD}$ and $\overline{CTS}$ lines are controlled automatically by the SCC. The following paragraphs describe the behavior in this mode. In the following description, the TCI bit in the GSMR is assumed to be cleared, implying normal transmit clock operation.

7.10.11.1 SYNCHRONOUS PROTOCOLS. The $\overline{RTS}$ pin is asserted when the SCC data is loaded into the transmit FIFO and a falling transmit clock occurs. At this point, the SCC begins transmitting the data, once the appropriate conditions occur on the $\overline{CTS}$ pin. In all cases, the first bit of data is the first bit of the opening flag, sync pattern, or the preamble (if a preamble was programmed to be sent prior to the frame).

Figure 7-39 shows that the delay between $\overline{RTS}$ and data is 0 bit times, regardless of the CTSS bit in the GSMR. This operation assumes that the $\overline{CTS}$ pin is already asserted to the SCC or that the $\overline{CTS}$ pin is reprogrammed to be a parallel I/O line, in which case the $\overline{CTS}$ signal to the SCC is always asserted. $\overline{RTS}$ is negated one clock after the last bit in the frame.

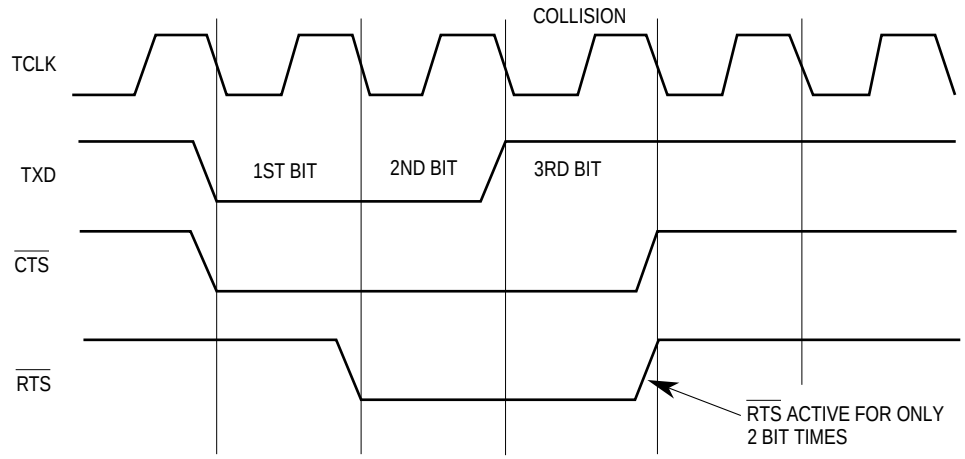
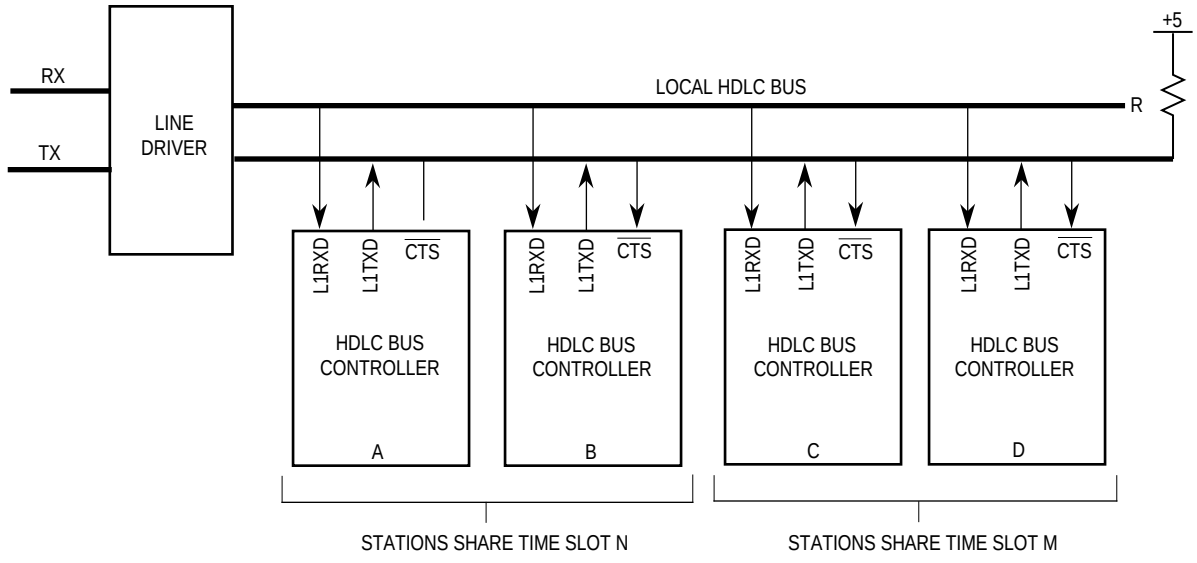


Figure 7-59. Delayed RTS Mode

7.10.18.2.4 Using the TSA. Sometimes HDLC bus may be used in a configuration that has a local HDLC bus, and a TDM transmission line that is not an HDLC bus. Figure 7-60 shows such a case. The local HDLC bus controllers all communicate over time slots; however, more than one HDLC bus controller is assigned to a given time slot, and the HDLC bus protocol is used to control access during that time slot.



NOTES:
1. All Tx pins of slave devices should be configured to open-drain in the port C parallel I/O port.
2. The TSA in the SI of each station is used to configure the desired time slot.
3. The choice of the number of stations to share a time slot is user-defined. It is two in this example.

Figure 7-60. HDLC Bus TSA Transmission Line Configuration

Once again, the local HDLC controllers do not communicate with each other, only with the transmission line. If the SCC is configured to operate using the TSA of the SI, then the data will be received and transmitted using the L1TXDx and L1RXDx pins. The collision sensing

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. This value should normally be greater than zero. The data length may be equal to zero with the P-bit set, and only a preamble will be sent.

If the number of data bits in the UART character is greater than 8, then the data length should be even. Example: to transmit three UART characters of 8-bit data, 1 start, and 1 stop, the data length field should be initialized to 3. However, to transmit three UART characters of 9-bit data, 1 start, and 1 stop, the data length field should be initialized to 6, since the three 9-bit data fields occupy three words in memory (the 9 LSBs of each word).

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 (unless the number of actual data bits in the UART character is greater than 8 bits, in which case the transmit buffer pointer must be even.) For instance, the pointer to 8-bit data, 1 start, and 1 stop characters may be even or odd, but the pointer to 9-bit data, 1 start, and 1 stop characters must be even. The buffer may reside in either internal or external memory.

7.11.7.14 SMC UART EVENT REGISTER (SMCE). When the UART protocol is selected, the SMCE register is called the SMC UART event register. It is an 8-bit register used to report events recognized by the SMC UART channel and to generate interrupts. On recognition of an event, the UART will set the corresponding bit in the SMC UART event register.

The SMC UART event register 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.

An example of the timing of various events in the SMC UART event register is shown in Figure 7-77.

7	6	5	4	3	2	1	0
—	BRKe[1]	—	BRK	—	BSY	TX	RX

NOTES:
1: Only available on REV C mask or later. NOT Available on REV A or B.
Rev A mask is C63T
Rev B mask are C69T, and F35G
Current Rev C mask are E63C, E68C and F15W

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 transmission order of bytes within a buffer word is reversed as compared to the Motorola mode.

1 = Motorola byte ordering—normal operation. It is also called big-endian byte ordering. As data is transmitted onto the serial line from the data buffer, the most significant byte of the buffer word contains data to be transmitted 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.

7.12.5.3.3 Maximum Receive Buffer Length Register (MRBLR). The SPI has one MRBLR to define the receive buffer length for that SPI. MRBLR defines the maximum number of bytes that the QUICC will write to a receive buffer on that SPI before moving to the next buffer. The QUICC may write fewer bytes to the buffer than the MRBLR value if a condition such as an error or end-of-frame occurs, but it will never write more bytes than the MRBLR value. It follows, then, that buffers supplied by the user for use by the QUICC should always be of size MRBLR (or greater) in length.

The transmit buffers for an SPI are not affected in any way by the value programmed into MRBLR. Transmit buffers may be individually chosen to have varying lengths, as needed. The number of bytes to be transmitted is chosen by programming the data length field in the Tx BD.

NOTES

MRBLR is not intended to be changed dynamically while an SPI is operating. However, if it is modified in a single bus cycle with one 16-bit move (NOT two 8-bit bus cycles back-to-back), then a dynamic change in receive buffer length can be successfully achieved. This takes place when the CP moves control to the next Rx BD in the table. Thus, a change to MRBLR will not have an immediate effect. To guarantee the exact Rx BD on which the change will occur, the user should change MRBLR only while the SPI receiver is disabled.

The MRBLR value should be greater than zero and should be even if the character length of the data is greater than eight bits.

12. Write $00000020 to the CIMR to allow the SPI to generate a system interrupt. (The CICR should also be initialized.)
13. Write $0370 to SPMODE to enable normal operation (not loopback), master mode, SPI enabled, 8-bit characters, and the fastest speed possible.
14. Write PBDAT bit 0 with zero to assert the SPI select pin.
15. Set the STR bit in the SPCOM to start the transfer.

NOTE

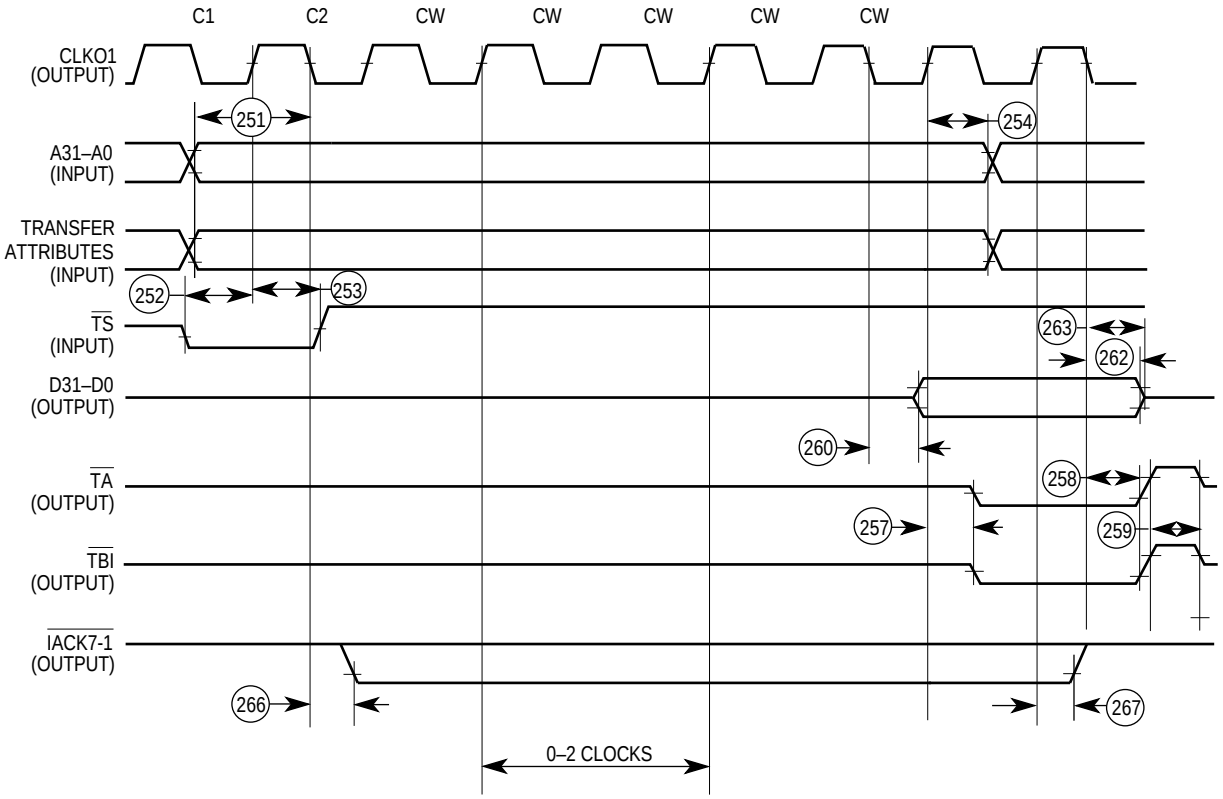
After 5 bytes have been transmitted, the Tx BD is closed. Additionally, the receive buffer is closed after 5 bytes have been received because the L-bit of the Tx BD was set.

7.12.7 SPI Slave Example

The following list is an initialization sequence for use of the SPI as a slave. It is very similar to the SPI master example except that the $\overline{\text{SPISEL}}$ pin is used, rather than a general-purpose I/O pin.

1. The SDCR (SDMA Configuration Register) should be initialized to $0740, rather than being left at its default value of $0000.
2. Configure the port B pins to enable the SPIMOSI, SPIMISO, $\overline{\text{SPISEL}}$, and SPICLK pins. Write PBPAR bits 0, 1, 2, and 3 with ones. Write PBDIR bits 0, 1, 2, and 3 with ones. Write PBODR bits 0, 1, 2, and 3 with zeros.
3. Write RBASE and TBASE in the SPI parameter RAM to point to the Rx BD and Tx BD in the dual-port RAM. Assuming one Rx BD at the beginning of dual-port RAM and one Tx BD following that Rx BD, write RBASE with $0000 and TBASE with $0008.
4. Program the CR to execute the INIT RX & TX PARAMS command for this channel. For instance, to execute this command for SCC1, write $0001 to the CR. This command causes the RBPTR and TBPTR parameters of the serial channel to be updated with the new values just programmed into RBASE and TBASE.
5. Write RFCR with $18 and TFCR with $18 for normal operation.
6. Write MRBLR with the maximum number of bytes per receive buffer. For this case, assume 16 bytes, so MRBLR = $0010.
7. Initialize the Rx BD. Assume the Rx data buffer is at $00001000 in main memory. Write $B000 to Rx_BD_Status. Write $0000 to Rx_BD_Length (not required—done for instructional purposes only). Write $00001000 to Rx_BD_Pointer.
8. Initialize the Tx BD. Assume the Tx data buffer is at $00002000 in main memory and contains five 8-bit characters. Write $B800 to Tx_BD_Status. Write $0005 to Tx_BD_Length. Write $00002000 to Tx_BD_Pointer.
9. Write $FF to the SPIE to clear any previous events.
10. Write $37 to the SPIM to enable all possible SPI interrupts.
11. Write $00000020 to the CIMR to allow the SPI to generate a system interrupt. (The

3. Decide which events in the SCCE1 will be handled in this handler and clear those bits as soon as possible. (SCCE bits are cleared by writing ones.)
4. Handle events in the SCC1 Rx or Tx BD tables.
5. Clear the SCC1 bit in the CISR.
6. Execute the RTE instruction. If any unmasked bits in SCCE1 remain at this time (either not cleared by the software or set by the QUICC during the execution of this handler), this interrupt source will be made pending again immediately following the RTE instruction.



NOTES:
1. MC68040 Transfer Attribute Signals = SIZx, TTx, TMx, R/W, LOCK.
2. Up to two wait states may be inserted for internal arbitration.

Figure 10-42. MC68040 IACK Cycles (Vector Driven)