Chapter 18
Bus Operation

This chapter describes data-transfer operations, error conditions, bus arbitration, and reset operations. It describes transfers initiated by the MCF5307 and by an external bus master, and includes detailed timing diagrams showing the interaction of signals in supported bus operations. Chapter 11, “Synchronous/Asynchronous DRAM Controller Module,” describes DRAM cycles.

18.1 Features

The following list summarizes bus operation features:

- Up to 32 bits of address and data
- 8-, 16-, and 32-bit port sizes
- Byte, word, longword, and line size transfers
- Bus arbitration for external devices
- Burst and burst-inhibited transfer support
- Internal termination for core and DMA bus cycles
- External termination of bus cycles controlled by an external bus master

Note that, throughout this manual, an overbar indicates an active-low signal.

18.2 Bus and Control Signals

Table 18-1 summarizes MCF5307 bus signals described in Chapter 17, “Signal Descriptions.”

Table 18-1. ColdFire Bus Signal Summary

Signal Name	Description	MCF5307 Master	External Master	Edge
$\overline{AS}$	Address strobe	O	I	Falling
A[31:0]	Address bus	O	I	Rising
$\overline{BE}$/$\overline{BWE}$ [1]	Byte enable/Byte write enable	O	O	Falling
$\overline{CS}$[7:0] [1]	Chip selects	O	O	Falling
D[31:0]	Data bus	I/O	I/O	Rising

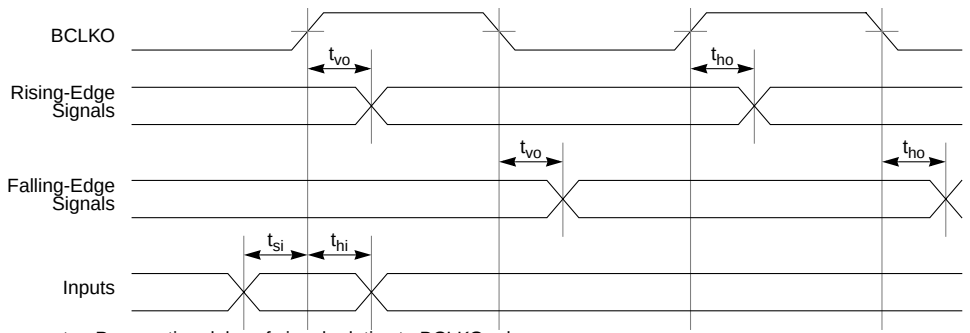
Table 18-1. ColdFire Bus Signal Summary (Continued)

Signal Name	Description	MCF5307 Master	External Master	Edge
$\overline{IRQ}$[7,5,3,1]	Interrupt request	I	I	Rising
$\overline{OE}$ [1]	Output enable	O	I	Falling
R/$\overline{W}$	Read/write	O	I	Rising
SIZ[1:0]	Transfer size	O	I	Rising
$\overline{TA}$	Transfer acknowledge	I	O	Rising
$\overline{TIP}$	Transfer in progress	O	Three-state	Rising
TM[2:0]	Transfer modifier	O	Three-state	Rising
$\overline{TS}$	Transfer start	O	I	Rising
TT[1:0]	Transfer type	O	Three-state	Rising

1 These signals change after the falling edge. In Chapter 20, “Electrical Specifications,” these signals are specified off the rising edge because CLKIN is squared up internally.

18.3 Bus Characteristics

The MCF5307 uses an input clock signal (CLKIN) to generate its internal clock. BCLKO is the bus clock rate, where all bus operations are synchronous to the rising edge of BCLKO. Some of the bus control signals ($\overline{BE}$/$\overline{BWE}$, $\overline{OE}$, $\overline{CSx}$, and $\overline{AS}$) are synchronous to the falling edge, shown in Figure 18-1. Bus characteristics may differ somewhat for interfacing with external DRAM.



t_{vo}=Propagation delay of signal relative to BCLKO edge
t_{ho}=Output hold time relative to BCLKO edge
t_{si}=Required input setup time relative to BCLKO edge
t_{hi}=Required input hold time relative to BCLKO edge

Figure 18-1. Signal Relationship to BCLKO for Non-DRAM Access

18.4 Data Transfer Operation

Data transfers between the MCF5307 and other devices involve the following signals:

- Address bus (A[31:0])
- Data bus (D[31:0])
- Control signals ($\overline{TS}$ and $\overline{TA}$)
- $\overline{AS}$, $\overline{CSx}$, $\overline{OE}$, $\overline{BE}$/$\overline{BWE}$
- Attribute signals (R/$\overline{W}$, SIZ, TT, TM, and $\overline{TIP}$)

The address bus, write data, $\overline{TS}$, and all attribute signals change on the rising edge of BCLKO. Read data is latched into the MCF5307 on the rising edge of BCLKO. $\overline{AS}$, $\overline{CSx}$, $\overline{OE}$, and $\overline{BE}$/$\overline{BWE}$ change on the falling edge.

The MCF5307 bus supports byte, word, and longword operand transfers and allows accesses to 8-, 16-, and 32-bit data ports. Transfer parameters such as port size, the number of wait states for the external slave being accessed, and whether internal transfer termination is enabled, can be programmed in the chip-select control registers (CSCRs) and DRAM control registers (DACRs).

For aligned transfers larger than the port size, SIZ[1:0] behaves as follows:

- If bursting is used, SIZ[1:0] stays at the size of transfer.
- If bursting is inhibited, SIZ[1:0] first shows the size of the transfer and then shows the port size.

Table 18-2 shows encoding for SIZ[1:0].

Table 18-2. Bus Cycle Size Encoding

SIZ[1:0]	Port Size
00	Longword
01	Byte
10	Word
11	Line

Figure 18-2 shows the byte lanes that external memory should be connected to and the sequential transfers if a longword is transferred for three port sizes. For example, an 8-bit memory should be connected to D[31:24] ($\overline{BE0}$). A longword transfer takes four transfers on D[31:24], starting with the MSB and going to the LSB.

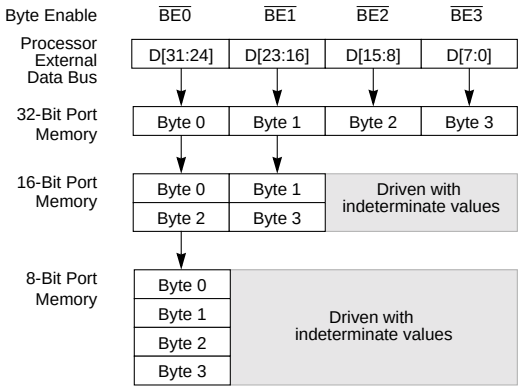


Figure 18-2. Connections for External Memory Port Sizes

The timing relationships between BCLKOchip select ($\overline{CS}$[7:0]), byte enable/byte write enables ($\overline{BE}$/$\overline{BWE}$[3:0]), and output enable ($\overline{OE}$) are similar to their relationships with address strobe ($\overline{AS}$) in that all transitions occur during the low phase of BCLKO. However, as shown in Figure 18-3, differences in on-chip signal routing and external loading may prevent signals from asserting simultaneously.

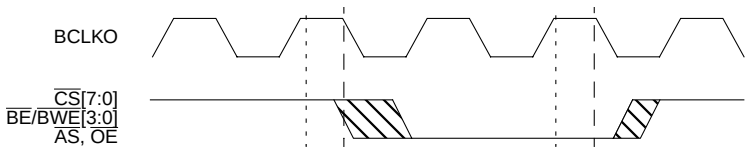


Figure 18-3. Chip-Select Module Output Timing Diagram

18.4.1 Bus Cycle Execution

When a bus cycle is initiated, the MCF5307 first compares its address with the base address and mask configurations programmed for chip selects 0–7 (CSCR0–CSCR7) and for DRAM blocks 0 and 1 address and control registers (DACR0 and DACR1). If the driven address matches a programmed chip select or DRAM block, the appropriate chip select is asserted or the DRAM block is selected using the specifications programmed in the respective configuration register. Otherwise, the following occurs:

- If the address and attributes do not match in CSCR or DACR, the MCF5307 runs an external burst-inhibited bus cycle with a default of external termination on a 32-bit port.
- If an address and attribute match in multiple CSCRs, the matching chip-select signals are driven; however, the MCF5307 runs an external burst-inhibited bus cycle with external termination on a 32-bit port.
- If an address and attribute match both DACRs or a DACR and a CSCR, the operation is undefined.

Table 18-3 shows the type of access as a function of match in the CSCRs and DACRs.

Table 18-3. Accesses by Matches in CSCRs and DACRs

Number of CSCR Matches	Number of DACR Matches	Type of Access
0	0	External
1	0	Defined by CSCRs
Multiple	0	External, burst-inhibited, 32-bit
0	1	Defined by DACRs
1	1	Undefined
Multiple	1	Undefined
0	Multiple	Undefined
1	Multiple	Undefined
Multiple	Multiple	Undefined

Basic bus operations occur in three clocks, as follows:

1. During the first clock, the address, attributes, and $\overline{TS}$ are driven. $\overline{AS}$ is asserted at the falling edge of the clock to indicate that address and attributes are valid and stable.
2. Data and $\overline{TA}$ are sampled during the second clock of a bus-read cycle. During a read, the external device provides data and is sampled at the rising edge at the end of the second bus clock. This data is concurrent with $\overline{TA}$, which is also sampled at the rising clock edge.

   During a write, the MCF5307 drives data from the rising clock edge at the end of the first clock to the rising clock edge at the end of the bus cycle. Wait states can be added between the first and second clocks by delaying the assertion of $\overline{TA}$. $\overline{TA}$ can be configured to be generated internally through the DACRs and CSCRs. If $\overline{TA}$ is not generated internally, the system must provide it externally.
3. The last clock of the bus cycle uses what would be an idle clock between cycles to provide hold time for address, attributes, and write data. Figure 18-6 and Figure 18-8 show the basic read and write operations.

18.4.2 Data Transfer Cycle States

The data transfer operation in the MCF5307 is controlled by an on-chip state machine. Each bus clock cycle is divided into two states. Even states occur when BCLKO is high and odd states occur when BCLKO is low. The state transition diagram for basic and fast-termination read and write cycles is shown in Figure 18-4.

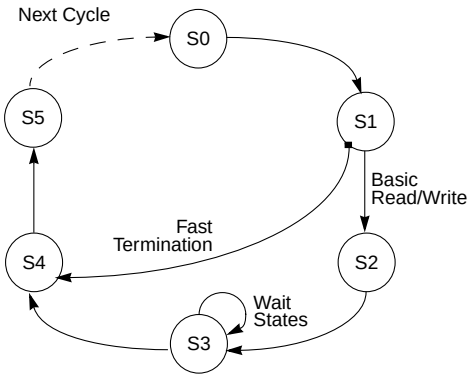


Figure 18-4. Data Transfer State Transition Diagram

Table 18-4 describes the states as they appear in subsequent timing diagrams. Note that the TT[1:0], TM[2:0], and $\overline{\text{TIP}}$ functions are chosen in the PAR, as described in Section 15.1.1, "Pin Assignment Register (PAR)."

Table 18-4. Bus Cycle States

State	Cycle	BCLKO	Description
S0	All	High	The read or write cycle is initiated. On the rising edge of BCLKO, the MCF5307 places a valid address on the address bus, asserts $\overline{\text{TIP}}$, and drives R/$\overline{\text{W}}$ high for a read and low for a write, if these signals are not already in the appropriate state. The MCF5307 asserts TT[1:0], TM[2:0], SIZ[1:0], and $\overline{\text{TS}}$ on the rising edge of BCLKO.
S1	All	Low	$\overline{\text{AS}}$ asserts on the falling edge of BCLKO, indicating that the address and attributes are stable. The appropriate $\overline{\text{CS}}$x, $\overline{\text{BE}}$/$\overline{\text{BWE}}$, and $\overline{\text{OE}}$ signals assert on the BCLKO falling edge.
	Fast termination		$\overline{\text{TA}}$ must be asserted during S1. Data is made available by the external device and is sampled on the rising edge of BCLKO with $\overline{\text{TA}}$ asserted.
S2	Read/write (skipped for fast termination)	High	$\overline{\text{TS}}$ is negated on the rising edge of BCLKO.
	Write		The data bus is driven out of high impedance as data is placed on the bus on the rising edge of BCLKO.
S3	Read/write (skipped for fast termination)	Low	The MCF5307 waits for $\overline{\text{TA}}$ assertion. If $\overline{\text{TA}}$ is not sampled as asserted before the rising edge of BCLKO at the end of the first clock cycle, the MCF5307 inserts wait states (full clock cycles) until $\overline{\text{TA}}$ is sampled as asserted.
	Read		Data is made available by the external device on the falling edge of BCLKO and is sampled on the rising edge of BCLKO with $\overline{\text{TA}}$ asserted.
S4	All	High	The external device should negate $\overline{\text{TA}}$.
	Read (including fast termination)		The external device can stop driving data after the rising edge of BCLKO. However, data could be driven up to S5.

Table 18-4. Bus Cycle States (Continued)

State	Cycle	BCLKO	Description
S5	S5	Low	$\overline{AS}$, $\overline{CS}$, $\overline{BE}$/$\overline{BWE}$, and $\overline{OE}$ are negated on the BCLKO falling edge. The MCF5307 stops driving address lines and R/$\overline{W}$ on the rising edge of BCLKO, terminating the read or write cycle. At the same time, the MCF5307 negates TT[1:0], TM[2:0], $\overline{TIP}$, and SIZ[1:0] on the rising edge of BCLKO. Note that the rising edge of BCLKO may be the start of S0 for the next access cycle; in this case, $\overline{TIP}$ remains asserted and R/$\overline{W}$ may not transition, depending on the nature of the back-to-back cycles.
	Read		The external device stops driving data between S4 and S5.
	Write		The data bus returns to high impedance on the rising edge of BCLKO. The rising edge of BCLKO may be the start of S0 for the next access.

NOTE:

An external device has at most two BCLKO cycles after the start of S4 to three-state the data bus after data is sampled in S3. This applies to basic read cycles, fast-termination cycles, and the last transfer of a burst.

18.4.3 Read Cycle

During a read cycle, the MCF5307 receives data from memory or from a peripheral device. Figure 18-5 is a read cycle flowchart.

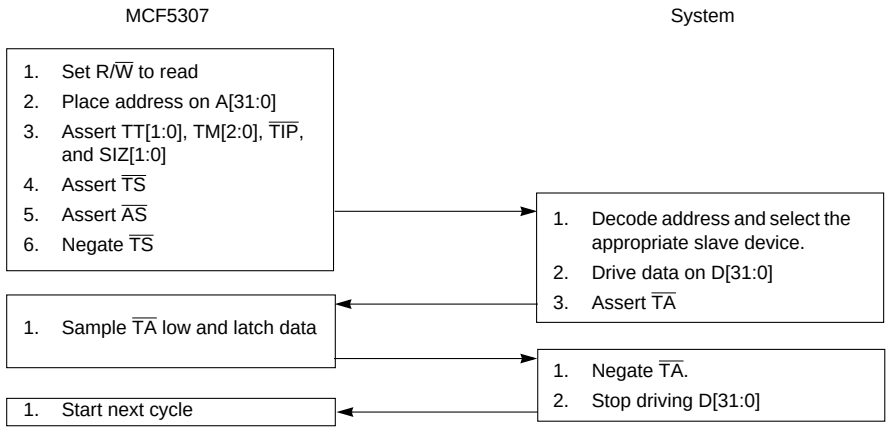


Figure 18-5. Read Cycle Flowchart

The read cycle timing diagram is shown in Figure 18-6.

NOTE:

In the following timing diagrams, $\overline{TA}$ waveforms apply for chip selects programmed to enable either internal or external termination. $\overline{TA}$ assertion should look the same in either case.

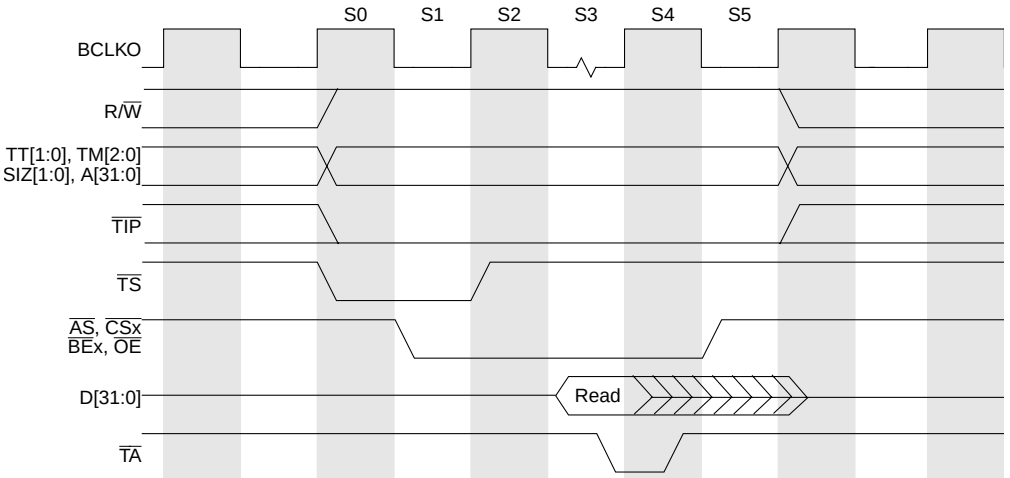


Figure 18-6. Basic Read Bus Cycle

Note the following characteristics of a basic read:

- In S3, data is made available by the external device on the falling edge of BCLKO and is sampled on the rising edge of BCLKO with $\overline{TA}$ asserted.
- In S4, the external device can stop driving data after the rising edge of BCLKO. However, data could be driven up to S5.
- For a read cycle, the external device stops driving data between S4 and S5.

States are described in Table 18-4.

18.4.4 Write Cycle

During a write cycle, the MCF5307 sends data to memory or to a peripheral device. The write cycle flowchart is shown in Figure 18-7.

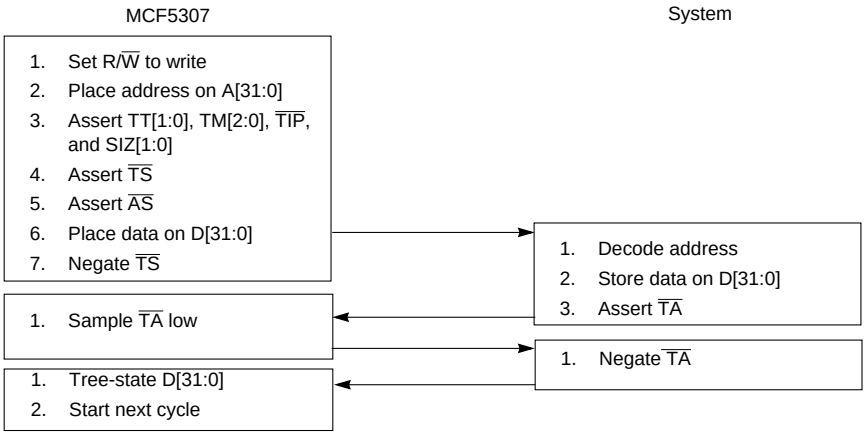


Figure 18-7. Write Cycle Flowchart

The write cycle timing diagram is shown in Figure 18-8.

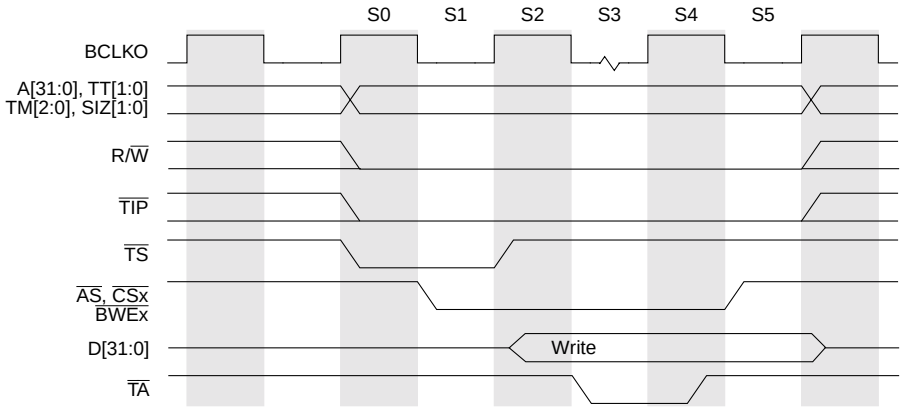


Figure 18-8. Basic Write Bus Cycle

Table 18-4 describes the six states of a basic write cycle.

18.4.5 Fast-Termination Cycles

Two clock-cycle transfers are supported on the MCF5307 bus. In most cases, this is impractical to use in a system because the termination must take place in the same half clock during which AS is asserted. Because this is atypical, it is not referred to as the zero-wait-state case but is called the fast-termination case. A fast-termination cycle is one in which an external device or memory asserts TA as soon as TS is detected. This means that the MCF5307 samples TA on the rising edge of the second cycle of the bus transfer. Figure 18-9 shows a read cycle with fast termination. Note that fast termination cannot be used with internal termination.

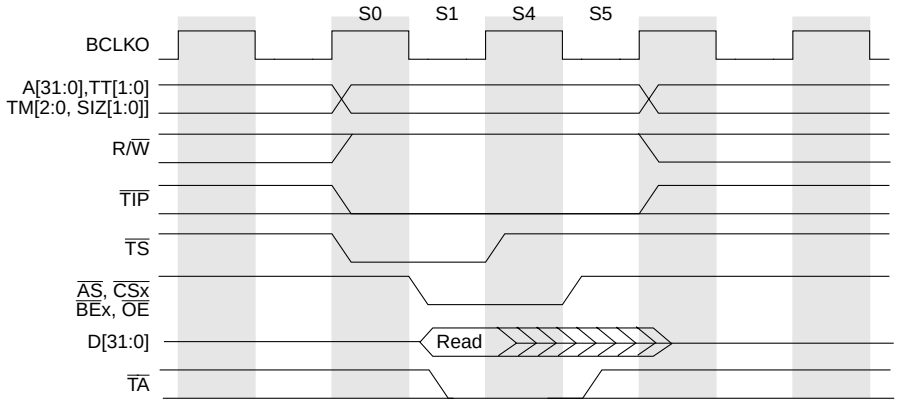


Figure 18-9. Read Cycle with Fast Termination

Figure 18-10 shows a write cycle with fast termination.

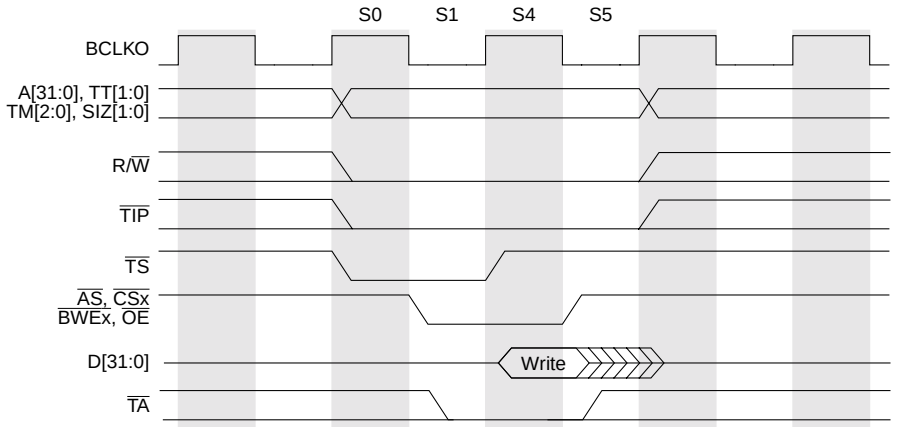


Figure 18-10. Write Cycle with Fast Termination

18.4.6 Back-to-Back Bus Cycles

The MCF5307 runs back-to-back bus cycles whenever possible. For example, when a longword read is started on a word-size bus, the processor performs two back-to-back word read accesses. Back-to-back accesses are distinguished by the continuous assertion of TIP throughout the cycle. Figure 18-11 shows a read back-to-back with a write.

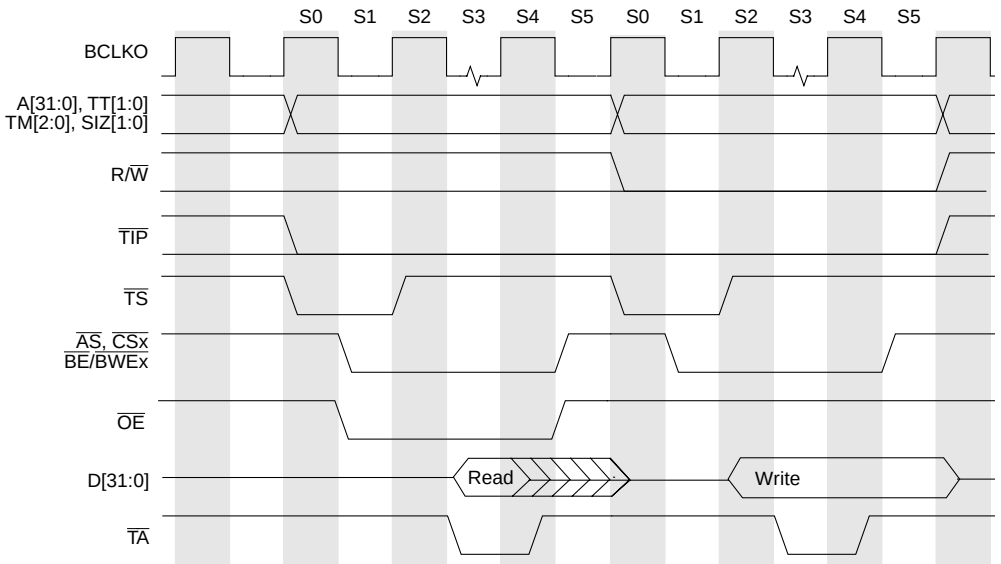


Figure 18-11. Back-to-Back Bus Cycles

Basic read and write cycles are used to show a back-to-back cycle, but there is no restriction as to the type of operations to be placed back to back. The initiation of a back-to-back cycle is not user definable.

18.4.7 Burst Cycles

The MCF5307 can be programmed to initiate burst cycles if its transfer size exceeds the size of the port it is transferring to. For example, with bursting enabled, a word transfer to an 8-bit port would take a 2-byte burst cycle for which SIZ[1:0] = 10 throughout. A line transfer to a 32-bit port would take a 4-longword burst cycle, for which SIZ[1:0] = 11 throughout.

The MCF5307 bus can support 2-1-1-1 burst cycles to maximize cache performance and optimize DMA transfers. A user can add wait states by delaying termination of the cycle. The initiation of a burst cycle is encoded on the size pins. For burst transfers to smaller port sizes, SIZ[1:0] indicates the size of the entire transfer. For example, if the MCF5307 writes a longword to an 8-bit port, SIZ[1:0] = 00 for the first byte transfer and does not change.

CSCRs are used to enable bursting for reads, writes, or both. MCF5307 memory space can be declared burst-inhibited for reads and writes by clearing the appropriate CSCR*x*[BSTR,BSTW]. A line access to a burst-inhibited region is broken into separate port-width accesses. Unlike a burst access, SIZ[1:0] = 11 only for the first port-width access; for the remaining accesses, SIZ[1:0] reflects the port width, with individual accesses separated by $\overline{AS}$ negations. The address changes if internal termination is used but does not change if external termination is used, as shown in Figure 18-12 and Figure 18-14.

18.4.7.1 Line Transfers

A line is a 16-byte-aligned, 16-byte value. Despite the alignment, a line access may not begin on the aligned address; therefore, the bus interface supports line transfers on multiple address boundaries. Table 18-5 shows allowable patterns for line accesses.

Table 18-5. Allowable Line Access Patterns

A[3:2]	Longword Accesses
00	0–4–8–C
01	4–8–C–0
10	8–C–0–4
11	C–0–4–8

18.4.7.2 Line Read Bus Cycles

Figure 18-12 shows line read with zero wait states. The access starts like a basic read bus cycle with the first data transfer sampled on the rising edge of S4, but the next pipelined burst data is sampled a cycle later on the rising edge of S6. Each subsequent pipelined data burst is single cycle until the last one, which can be held for up to 2 BCLKO cycles after $\overline{TA}$ is asserted. Note that $\overline{AS}$ and $\overline{CSx}$ are asserted throughout the burst transfer. This example shows the timing for external termination, which differs only from the internal termination example in Figure 18-13 in that the address lines change only at the beginning (assertion of $\overline{TS}$ and $\overline{TIP}$) and end (negation of $\overline{TIP}$) of the transfer.

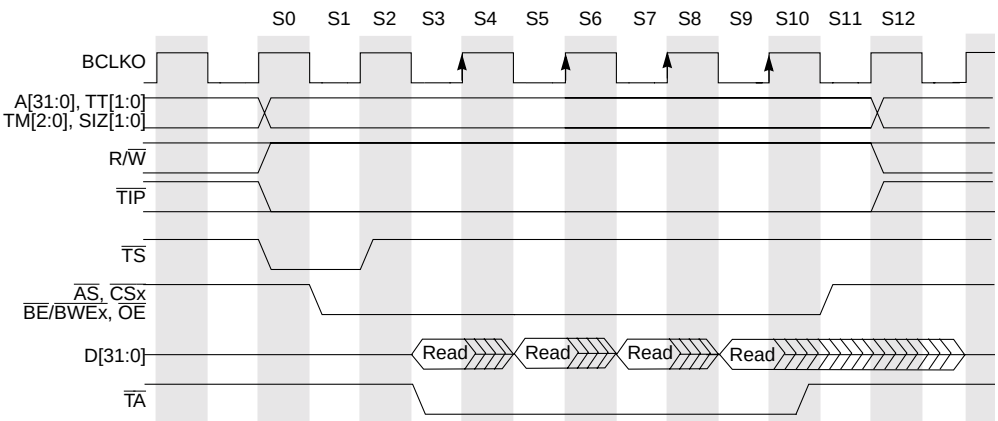


Figure 18-12. Line Read Burst (2-1-1-1), External Termination

Figure 18-13 shows timing when internal termination is used.

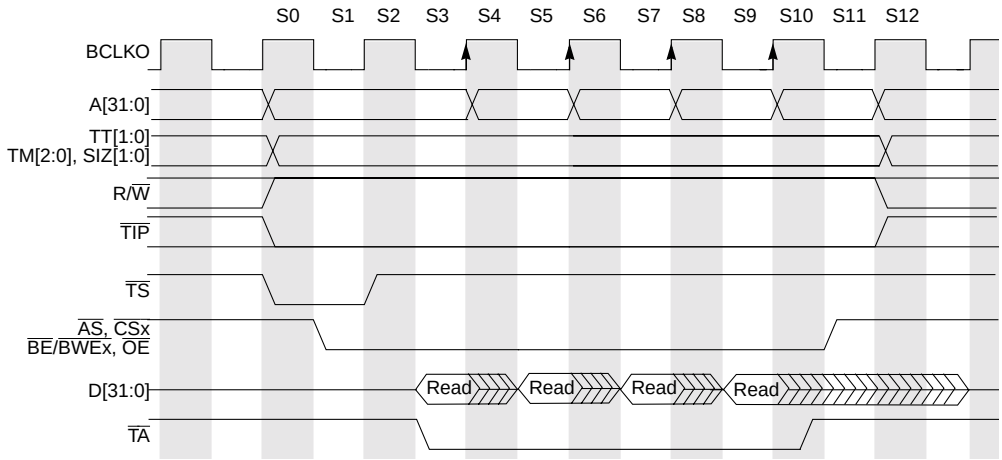


Figure 18-13. Line Read Burst (2-1-1-1), Internal Termination

Figure 18-14 shows a line access read with one wait state programmed in CSCR*x* to give the peripheral or memory more time to return read data. This figure follows the same execution as a zero-wait state read burst with the exception of an added wait state.

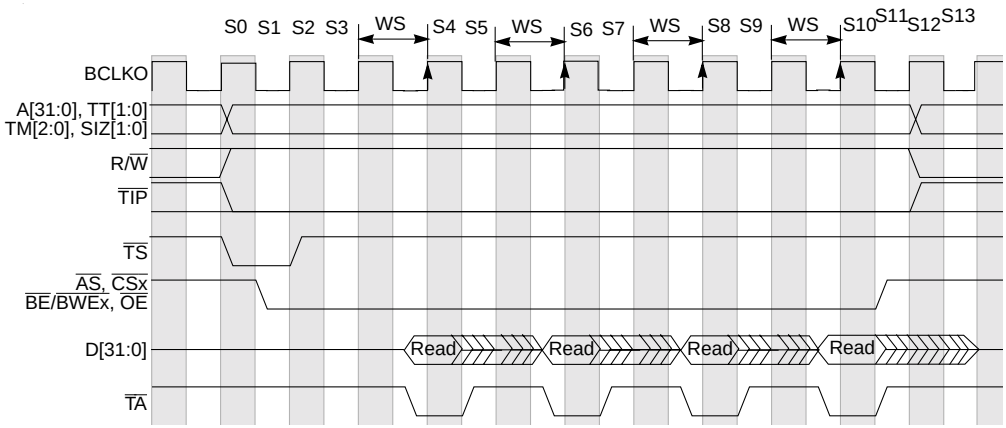


Figure 18-14. Line Read Burst (3-2-2-2), External Termination

Figure 18-15 shows a burst-inhibited line read access with fast termination. The external device executes a basic read cycle while determining that a line is being transferred. The external device uses fast termination for subsequent transfers.

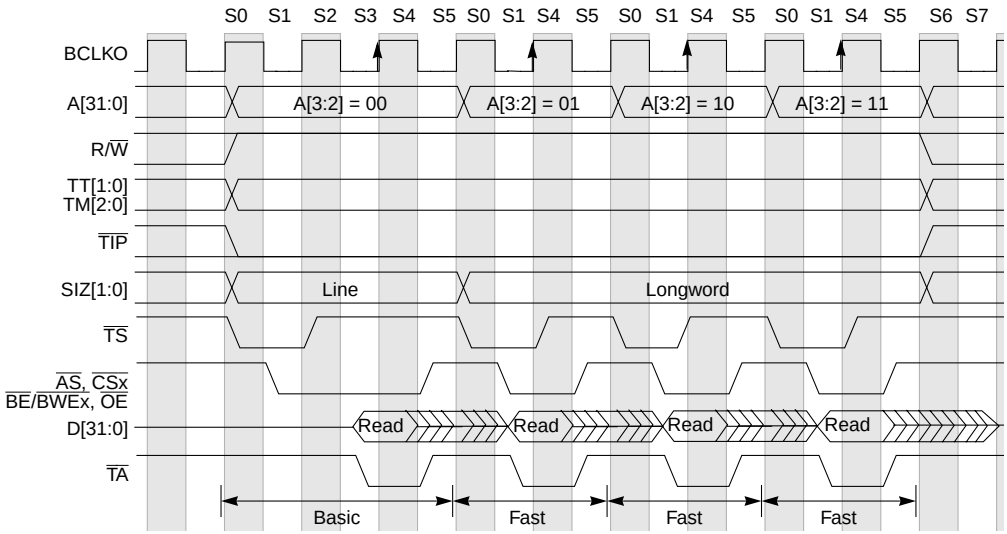


Figure 18-15. Line Read Burst-Inhibited, Fast, External Termination

18.4.7.3 Line Write Bus Cycles

Figure 18-16 shows a line access write with zero wait states. It begins like a basic write bus cycle with data driven one clock after $\overline{TS}$. The next pipelined burst data is driven a cycle after the write data is registered (on the rising edge of S6). Each subsequent burst takes a single cycle. Note that as with the line read example in Figure 18-12, $\overline{AS}$ and $\overline{CS}x$ remain asserted throughout the burst transfer. This example shows the behavior of the address lines for both internal and external termination. Note that with external termination, address lines, like SIZ, TT, and TM, hold the same value for the entire transfer.

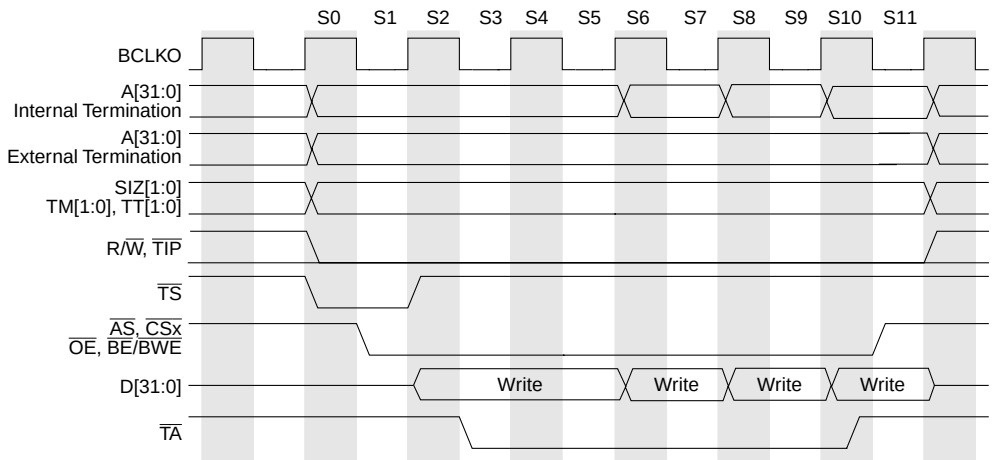


Figure 18-16. Line Write Burst (2-1-1-1), Internal/External Termination

Figure 18-17 shows a line burst write with one wait-state insertion.

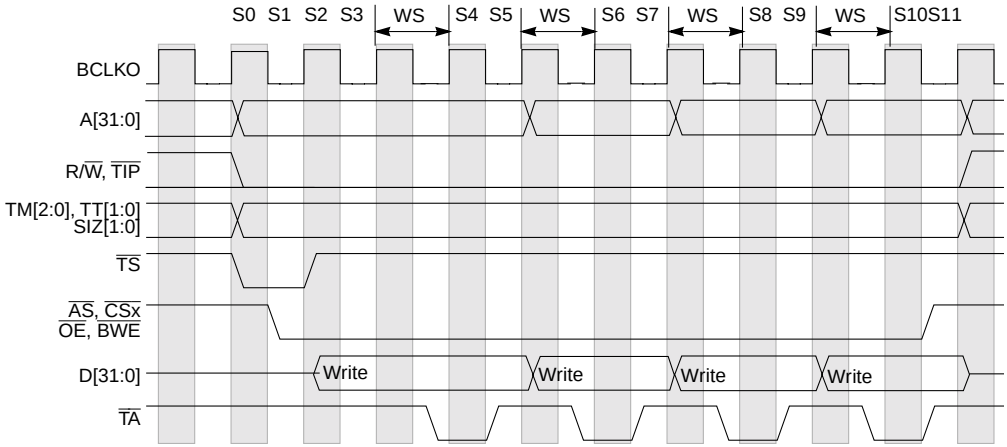


Figure 18-17. Line Write Burst (3-2-2-2) with One Wait State, Internal Termination

Figure 18-18 shows a burst-inhibited line write. The external device executes a basic write cycle while determining that a line is being transferred. The external device uses fast termination to end each subsequent transfer.

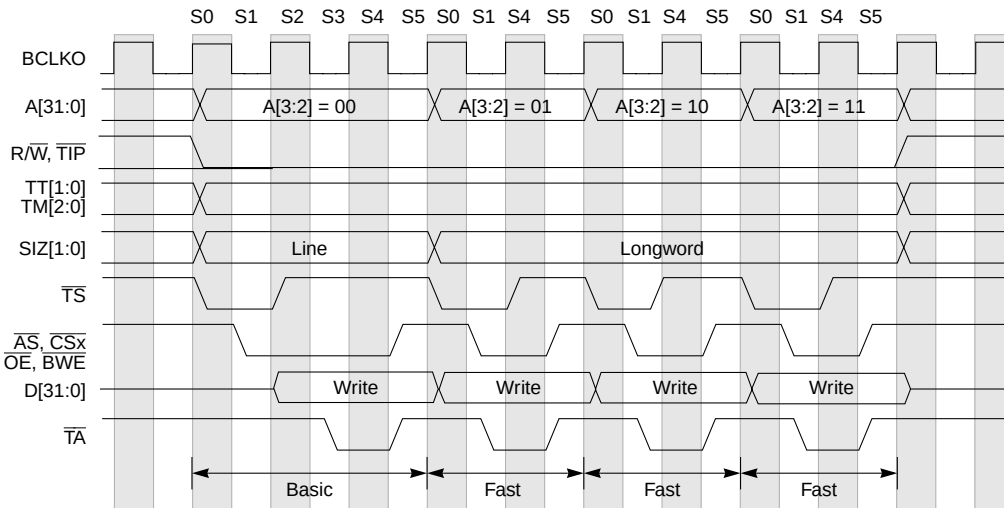


Figure 18-18. Line Write Burst-Inhibited, Internal Termination

18.4.7.4 Transfers Using Mixed Port Sizes

Figure 18-19 shows timing for a longword read from an 8-bit port using external termination. Figure 18-20 shows the same transfer with internal termination. For both, SIZ[1:0] change only at the start of a new transfer because this burst is implemented as one

transfer.

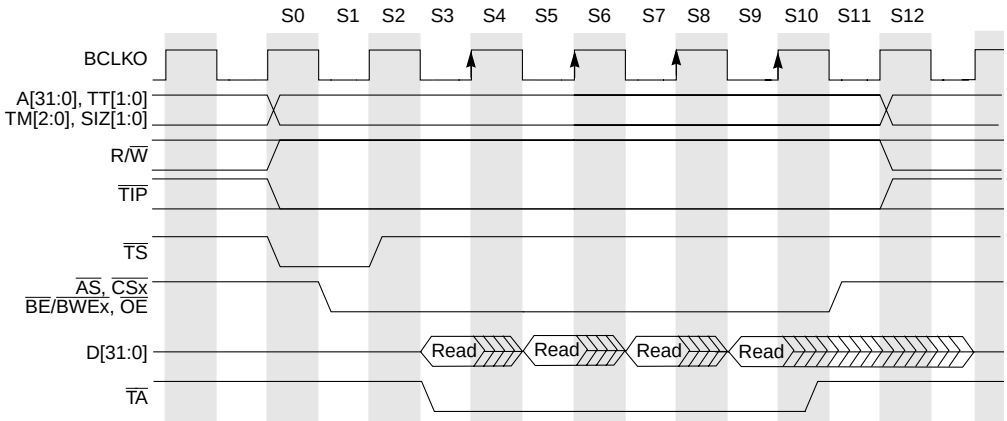


Figure 18-19. Longword Read from an 8-Bit Port, External Termination

Note that with external termination, address signals do not change. With internal termination, Figure 18-20, A[1:0] increment for the same longword transfer.

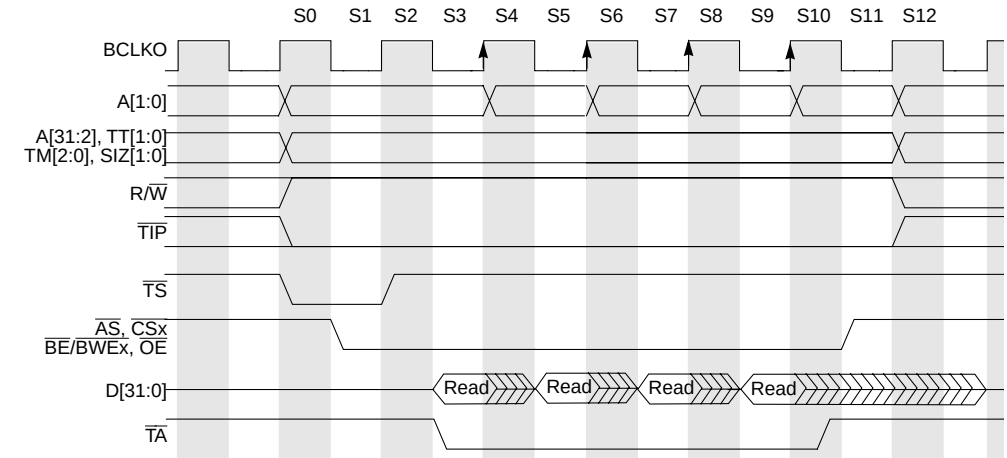


Figure 18-20. Longword Read from an 8-Bit Port, Internal Termination

18.5 Misaligned Operands

Because operands, unlike opcodes, can reside at any byte boundary, they are allowed to be misaligned. A byte operand is properly aligned at any address, a word operand is misaligned at an odd address, and a longword is misaligned at an address not a multiple of four. Although the MCF5307 enforces no alignment restrictions for data operands (including program counter (PC) relative data addressing), additional bus cycles are required for misaligned operands.

Instruction words and extension words (opcodes) must reside on word boundaries. Attempting to prefetch a misaligned instruction word causes an address error exception.

The MCF5307 converts misaligned, cache-inhibited operand accesses to multiple aligned accesses. Figure 18-21 shows the transfer of a longword operand from a byte address to a 32-bit port. In this example, SIZ[1:0] specify a byte transfer and a byte offset of 0x1. The slave device supplies the byte and acknowledges the data transfer. When the MCF5307 starts the second cycle, SIZ[1:0] specify a word transfer with a byte offset of 0x2. The next two bytes are transferred in this cycle. In the third cycle, byte 3 is transferred. The byte offset is now 0x0, the port supplies the final byte, and the operation is complete.

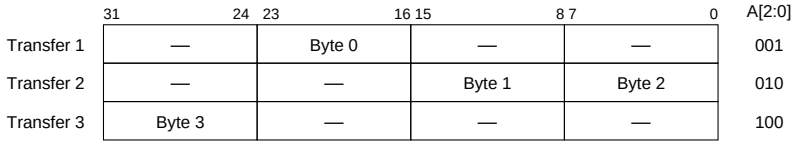


Figure 18-21. Example of a Misaligned Longword Transfer (32-Bit Port)

If an operand is cacheable and is misaligned across a cache-line boundary, both lines are loaded into the cache. The example in Figure 18-22 differs from the one in Figure 18-21 in that the operand is word-sized and the transfer takes only two bus cycles.

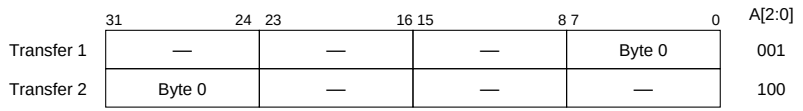


Figure 18-22. Example of a Misaligned Word Transfer (32-Bit Port)

NOTE:

External masters using internal MCF5307 chip selects and default memory control signals must initiate aligned transfers.

18.6 Bus Errors

The MCF5307 has no bus monitor. If the auto-acknowledge feature is not enabled for the address that generates the error, the bus cycle can be terminated by asserting $\overline{TA}$ or by using the software watchdog timer. If it is required that the MCF5307 handle a bus error differently, an interrupt handler can be invoked by asserting an interrupt to the core along with $\overline{TA}$ when the bus error occurs.

18.7 Interrupt Exceptions

A peripheral device uses the interrupt-request signals ($\overline{IRQx}$) to signal the core to take an interrupt exception when it needs the MCF5307 or is ready to send information to it. The interrupt transfers control to an appropriate routine.

The MCF5307 has the following two levels of interrupt masking:

- Interrupt mask registers in the SIM compare interrupt inputs with programmable interrupt mask levels. The SIM outputs only unmasked interrupts.
- The status register uses a 3-bit interrupt priority mask. The core recognizes only interrupt requests of higher priority than the value in the mask. See Section 2.2.2.1, “Status Register (SR).”

NOTE:

To mask a level 1–6 interrupt source, write a higher-level SR interrupt mask before setting IMR. Then restore the mask to its previous value. Do not mask a level 7 interrupt source.

The MCF5307 continuously samples and synchronizes external interrupt inputs. An interrupt request must be held for at least two consecutive BCLKO periods to be considered valid. To guarantee that the interrupt is recognized, the request level must be maintained until the MCF5307 acknowledges the interrupt with an interrupt-acknowledge cycle.

NOTE:

Interrupt levels 1–7 are level-sensitive. Level 7 is also edge-triggered. See Section 18.7.1, “Level 7 Interrupts.”

The MCF5307 takes an interrupt exception for a pending interrupt within one instruction boundary after processing any higher-priority pending exception. Thus, the MCF5307 executes at least one instruction in an interrupt exception handler before recognizing another interrupt request.

If autovector generation is used for internal interrupts (ICR*n*[AVEC] = 1), the interrupt acknowledge vector is generated internally and no interrupt acknowledge cycle is generated on the external bus.

If autovector generation is used for external interrupts, no interrupt acknowledge cycle is shown on the external bus ($\overline{AS}$ is not asserted) unless AVR[BLK] is 0. Consequently, the external interrupt must be cleared in the interrupt service routine. See Section 9.2.2, “Autovector Register (AVR).”

18.7.1 Level 7 Interrupts

Level 7 interrupts are nonmaskable and are handled differently than other interrupts. Level 7 interrupts are edge triggered by a transition from a lower priority request to the level 7 request. Interrupts at all other levels are level sensitive. Therefore, if $\overline{IRQ7}$ remains asserted, the MCF5307 recognizes only one level 7 interrupt because only one transition from a lower level request to a level 7evel 7 request occurred. For the processor to recognize two consecutive level 7 interrupts, one of the following must occur:

- The interrupt request on the interrupt control pins is raised to level 7 and stays there until an interrupt-acknowledge cycle begins. The level later drops but then returns to level 7, causing a second transition on the interrupt control lines.
- The interrupt request on the interrupt control pins is raised to level 7 and stays there. If the level 7 interrupt routine lowers the mask level, a second level 7 interrupt is recognized without a transition of the interrupt control pins. After the level 7 routine completes, the MCF5307 compares the mask level to the request level on the $\overline{\text{IRQ}}x$ signals. Because the mask level is lower than the requested level, the interrupt mask is set back to level 7. To ensure it is recognized, the level 7 request on $\overline{\text{IRQ7}}$ must be held until the second interrupt-acknowledge bus cycle begins.

18.7.2 Interrupt-Acknowledge Cycle

When the MCF5307 processes an interrupt exception, it performs an interrupt-acknowledge bus cycle to obtain the vector number that contains the starting location of the interrupt exception handler. The interrupt-acknowledge bus cycle is a read transfer that differs from normal read cycles in the following respects:

- TT[1:0] = 0x3 to indicate a CPU space or acknowledge bus cycle.
- TM[2:0] = the level of interrupt being acknowledged.
- A[31:5] = 0x7F_FFFF.
- A[4:2] = the interrupt request level being acknowledged (same as TM[2:0]).
- A[1:0] = 00.

During the interrupt-acknowledge bus cycle (a read cycle), the responding device places the vector number on D[31:24] and the cycle is terminated normally with $\overline{\text{TA}}$. Figure 18-23 is a flow diagram for an interrupt-acknowledge cycle terminated with $\overline{\text{TA}}$.

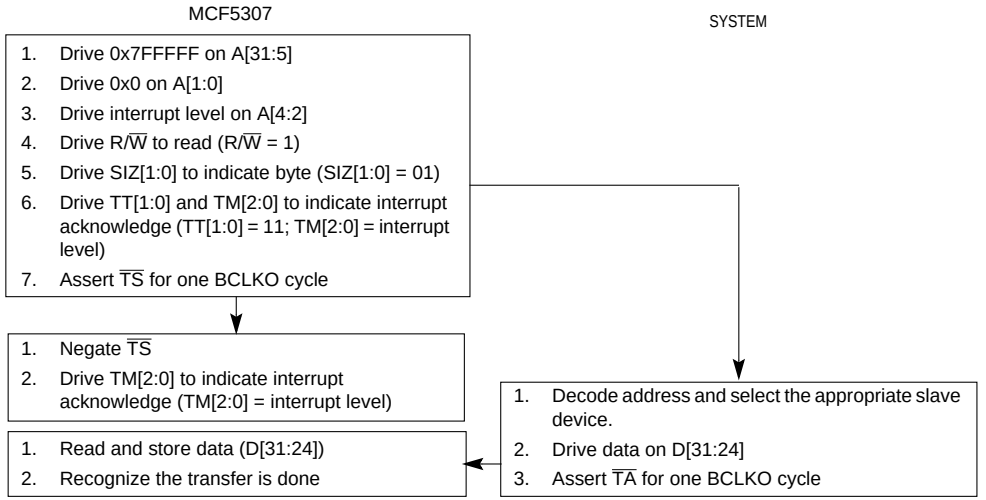


Figure 18-23. Interrupt-Acknowledge Cycle Flowchart

18.8 Bus Arbitration

The MCF5307 bus protocol gives either the MCF5307 or an external device access to the external bus. If more than one external device uses the bus, an external arbiter can prioritize requests and determine which device is bus master. When the MCF5307 is bus master, it uses the bus to fetch instructions and transfer data to and from external memory. When an external device is bus master, the MCF5307 can monitor the external master's transfers and interact through its chip-select, DRAM control, and transfer termination signals. See Section 10.4.1.3, "Chip-Select Control Registers (CSCR0–CSCR7)," and Chapter 11, "Synchronous/Asynchronous DRAM Controller Module."

Two-wire bus arbitration is used where the MCF5307 shares the bus with a single external device. This mode uses $\overline{BG}$ and $\overline{BD}$. The external device can ignore $\overline{BR}$. Three-wire mode is used where the MCF5307 shares the bus with multiple external devices. This requires an external bus arbiter and uses $\overline{BG}$, $\overline{BD}$, and $\overline{BR}$. In either mode, the MCF5307 bus arbiter operates synchronously and transitions between states on the rising edge of BCLKO.

Table 18-6 shows the four arbitration states the MCF5307 can be in during bus operation.

Table 18-6. MCF5307 Arbitration Protocol States

State	Master	Bus	$\overline{BD}$	Description
Reset	None	Not driven	Negated	The MCF5307 enters reset state from any other state when $\overline{RSTI}$ or software watchdog reset is asserted. If both are negated, the MCF5307 enters implicit or external device mastership state, depending on $\overline{BG}$.
Implicit master	MCF5307	Not driven	Negated	The MCF5307 is bus master ($\overline{BG}$ input is asserted) but is not ready to begin a bus cycle. It continues to three-state the bus until an internal bus request.

Table 18-6. MCF5307 Arbitration Protocol States (Continued)

State	Master	Bus	$\overline{BD}$	Description
Explicit master	MCF5307	Driven	Asserted	The MCF5307 is explicit bus master when $\overline{BG}$ is asserted and at least one bus cycle has been initiated. It asserts $\overline{BD}$ and retains explicit mastership until $\overline{BG}$ is negated even if no active bus cycles are executed. It releases the bus at the end of the current bus cycle, then negates $\overline{BD}$ and three-states the bus signals.
External master	External	Not driven	Negated	An external device is bus master ($\overline{BG}$ negated to MCF5307). The MCF5307 can assert $\overline{OE}$, $\overline{CS}$[7:0], $\overline{BE}$/$\overline{BWE}$[3:0], $\overline{TA}$, and all DRAM controller signals ($\overline{RAS}$[1:0], $\overline{CAS}$[3:0], $\overline{SRAS}$, $\overline{SCAS}$, $\overline{DRAMW}$, SCKE).

If the MCF5307 is the only possible master, $\overline{BG}$ can be tied to GND—no arbiter is needed.

18.8.1 Bus Arbitration Signals

Bus arbitration signal timings in Table 18-7 are referenced to the system clock, which is not considered a bus signal. Clock routing is expected to meet application requirements.

Table 18-7. ColdFire Bus Arbitration Signal Summary

Signal	I/O	Description
$\overline{BR}$	O	Bus request. Indicates to an external arbiter that the processor needs to become bus master. $\overline{BR}$ is negated when the MCF5307 begins an access to the external bus with no other internal accesses pending. $\overline{BR}$ remains negated until another internal request occurs.
$\overline{BG}$	I	Bus grant. An external arbiter asserts $\overline{BG}$ to indicate that the MCF5307 can control the bus at the next rising edge of BCLKO. When the arbiter negates $\overline{BG}$, the MCF5307 must release the bus as soon as the current transfer completes. The external arbiter must not grant the bus to any other device until both $\overline{BD}$ and $\overline{BG}$ are negated.
$\overline{BD}$	O	Bus driven. The MCF5307 asserts $\overline{BD}$ to indicate it is current master and is driving the bus. If it loses bus mastership during a transfer, it completes the last transfer of the current access, negates $\overline{BD}$, and three-states all bus signals on the rising edge of BCLKO. If it loses mastership during an idle clock cycle, it three-states all bus signals on the rising edge of BCLKO.

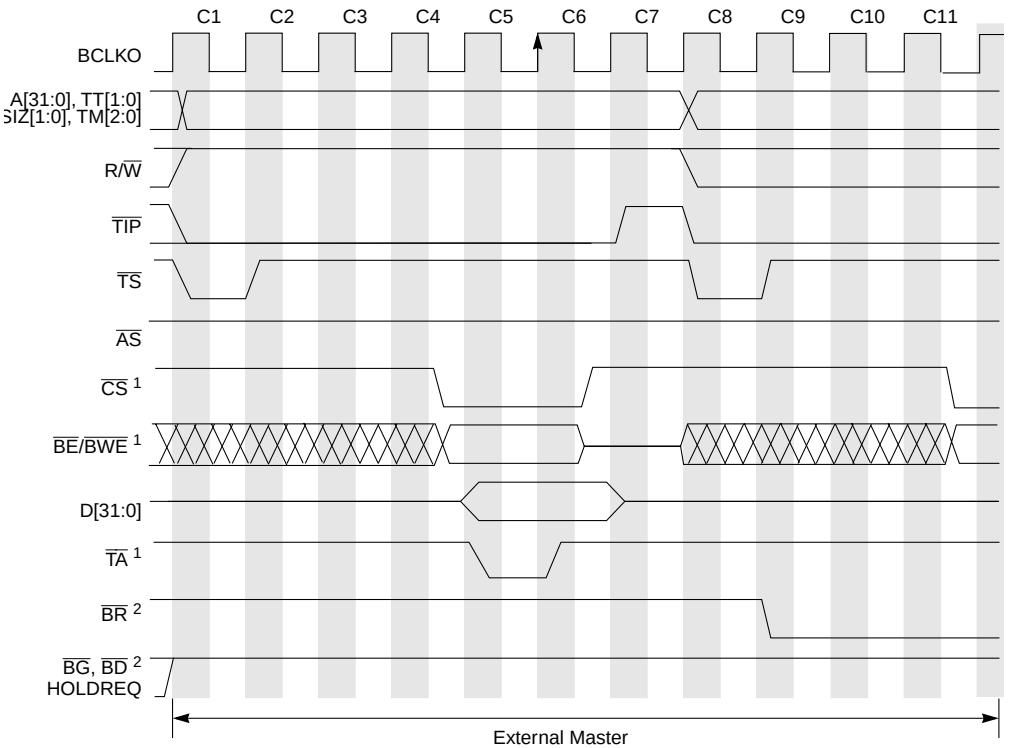
18.9 General Operation of External Master Transfers

An external master asserts its hold signal (such as HOLDREQ) when it executes a bus cycle, driving $\overline{BG}$ high and forcing the MCF5307 to hold all bus requests. During an external master cycle, the MCF5307 can provide memory control signals ($\overline{OE}$, $\overline{CS}$[7:0], $\overline{BE}$/$\overline{BWE}$[3:0], $\overline{RAS}$[1:0], $\overline{CAS}$[3:0]) and $\overline{TA}$ while the external master drives the address and data bus and other required bus control signals. When the external master asserts $\overline{TS}$ or $\overline{AS}$ to the MCF5307, the beginning of a bus cycle is identified and the MCF5307 starts decoding the address driven.

Note the following regarding external master accesses:

- For the MCF5307 to assert a $\overline{CSx}$ during external master accesses, CSMR*n*[AM] must be set. External master hits use the corresponding CSCR*n* settings for auto-acknowledge, byte enables, and wait states. See Section 10.4.1.3, “Chip-Select Control Registers (CSCR0–CSCR7).”
- To enable DRAM control signals during external master accesses, DCMR*n*[AM] must be set.
- During external master bus cycles, either $\overline{TS}$ or $\overline{AS}$ (but not both) should be driven to the MCF5307. Driving both during a bus cycle causes indeterminate results.

External master transfers that use the MCF5307 to drive memory control signals and $\overline{TA}$ are like normal MCF5307 transfers. Figure 18-24 shows timing for basic back-to-back bus cycles during an external master transfer.



1 Depending on programming, these signals may or may not be driven by the processor.
2 This signal is driven by the processor for an external master transfer.

Figure 18-24. Basic No-Wait-State External Master Access

R/$\overline{W}$ is asserted high for reads and low for writes; otherwise, the transfers are the same. In Figure 18-24, the MCF5307 chip select’s internal transfer acknowledge is enabled and the MCF5307 drives $\overline{TA}$ as an output after a programmed number of wait states.

NOTE:

Bus timing diagrams for external master transfers are not valid for on-chip internal four-channel DMA accesses on the MCF5307.

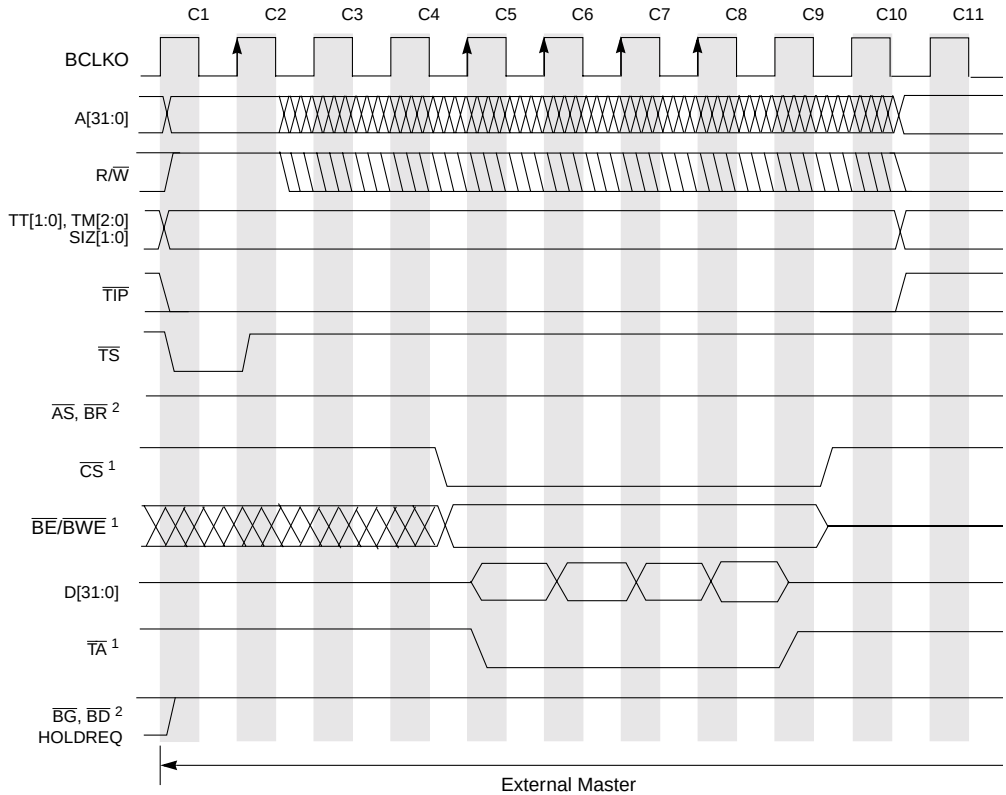
Timing diagrams describe transactions in general terms of bus cycles (C*n*) rather than the states (S*n*) used in the bus diagrams.

Table 18-8 defines the cycles for Figure 18-24.

Table 18-8. Cycles for Basic No-Wait-State External Master Access

Cycle	Definition
C1	The external master asserts HOLDREQ, signaling the MCF5307 to hold bus requests. $\overline{BD}$ should not be asserted. The external master drives address, $\overline{TS}$, R/$\overline{W}$, TT[1:0], TM[2:0], $\overline{TIP}$, and SIZ[1:0] as MCF5307 inputs.
C2–C3	The MCF5307 decodes the external master's address and control signals to identify the proper chip select and byte enable assertion. The external master negates $\overline{TS}$ in C2.
C4	On the falling edge of BCLKO, the MCF5307 asserts the appropriate chip select for the external master access along with the appropriate byte enables.
C5	On the rising edge of BCLKO, data is driven onto the bus by the device selected by $\overline{CS}$. On the rising edge, the MCF5307 asserts $\overline{TA}$ to indicate the cycle is complete.
C6	$\overline{TA}$ negates on the rising edge of BCLKO. On the falling edge, the MCF5307 negates the chip select and byte enables and the next cycle can begin.
C7	The external master negates $\overline{TIP}$ on the rising edge of BCLKO.
C8	The external device retains bus mastership and drives the address bus, $\overline{TS}$, R/$\overline{W}$, TT[1:0], TM[2:0], $\overline{TIP}$, and SIZ[1:0] as inputs to the MCF5307.
C9	The MCF5307 decodes the external master's address and control signals to identify the proper chip select and byte enable assertion. The external master negates $\overline{TS}$. The MCF5307 asserts $\overline{BR}$ on the rising edge of BCLKO, signalling that it wants to arbitrate for the bus when the current cycle completes.
C10	The MCF5307 continues to decode the external device's address and control signals to identify the proper chip select and byte enable assertion.
C11	On the falling edge of BCLKO, the MCF5307 asserts the appropriate chip select for the external master access along with the appropriate byte enables.

Figure 18-25 shows a burst line access for an external master transfer with the chip select set to no-wait states and with internal transfer-acknowledge assertion enabled.



1 Depending on programming, these signals may or may not be driven by the processor.
2 These signals are driven by the processor for an external master transfer.

Figure 18-25. External Master Burst Line Access to 32-Bit Port

Table 18-9 defines the cycles for Figure 18-25.

Table 18-9. Cycles for External Master Burst Line Access to 32-Bit Port

Cycle	Definition
C1	The external device is bus master and asserts HOLDREQ, indicating to the MCF5307 to hold all bus requests. In other words, BD should not be asserted. The external master drives address, TS, R/W, TT[1:0], TM[2:0], TIP, and SIZ[1:0] as inputs to the MCF5307. SIZ[1:0] inputs indicate a line transfer. The MCF5307 is not asserting BR.
C2–C3	The MCF5307 decodes the external device's address and control signals to identify the proper chip-select and byte-enable assertion. The external device negates TS in C2. Address and R/W are latched in the MCF5307 on the rising edge of BCLKO in C2. After C2, the address and R/W are ignored for the rest of the burst transfer.
C4	On the falling edge of BCLKO, the MCF5307 asserts the appropriate chip select for the external device access along with the appropriate byte enables.
C5	On the rising edge of BCLKO, data is driven onto the bus by the device selected by CS. The MCF5307 asserts TA on the rising edge of BCLKO, indicating the first data transfer is complete.

Table 18-9. Cycles for External Master Burst Line Access to 32-Bit Port (Continued)

Cycle	Definition
C6–C8	No-wait state data transfers 2–4 occur on the rising edges of BCLKO. $\overline{TA}$ continues to be asserted indicating completion of each transfer. $\overline{TIP}$, $\overline{CSx}$, and $\overline{BE}/\overline{BWE}$[3:0] are driven.
C9	$\overline{TA}$ negates on the rising edge of BCLKO along with external device's negation of $\overline{TIP}$. On the falling edge, the MCF5307 negates chip select and byte enables, creating an opportunity for another cycle to begin.

18.9.1 Two-Device Bus Arbitration Protocol (Two-Wire Mode)

Two-wire mode bus arbitration lets the MCF5307 share the external bus with a single external bus device without requiring an external bus arbiter. Figure 18-26 shows the MCF5307 connecting to an external device using the two-wire mode. The MCF5307 $\overline{BG}$ input is connected to the HOLDREQ output of the external device; the MCF5307 $\overline{BD}$ output is connected to the HOLDACK input of the external device. Because the external device controls the state of HOLDREQ, it controls when the MCF5307 is granted the bus, giving the MCF5307 lower priority.

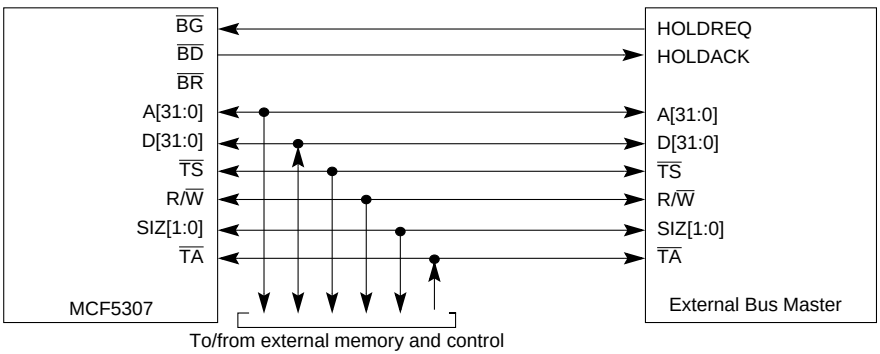


Figure 18-26. MCF5307 Two-Wire Mode Bus Arbitration Interface

When the external device is not using the bus, it negates HOLDREQ, driving $\overline{BG}$ low and granting the bus to the MCF5307. When the MCF5307 has an internal bus request pending and $\overline{BG}$ is low, the MCF5307 drives $\overline{BD}$ low, negating HOLDACK to the external device. When the external bus device needs the external bus, it asserts HOLDREQ, driving $\overline{BG}$ high, requesting the MCF5307 to release the bus. If $\overline{BG}$ is negated while a bus cycle is in progress, the MCF5307 releases the bus at the completion of the bus cycle. Note that the MCF5307 considers the individual transfers of a burst or burst-inhibited access to be a single bus cycle and does not release the bus until the last transfer of the series completes.

When the bus has been granted to the MCF5307, one of two situations can occur. In the first case, if the MCF5307 has an internal bus request pending, the MCF5307 asserts $\overline{BD}$ to indicate explicit bus mastership and begins the pending bus cycle by asserting $\overline{TS}$. As

shown in Figure 18-25, the MCF5307 continues to assert $\overline{BD}$ until the completion of the bus cycle. If $\overline{BG}$ is negated by the end of the bus cycle, the MCF5307 negates $\overline{BD}$. While $\overline{BG}$ is asserted, $\overline{BD}$ remains asserted to indicate the MCF5307 is master, and it continuously drives the address bus, attributes, and control signals.

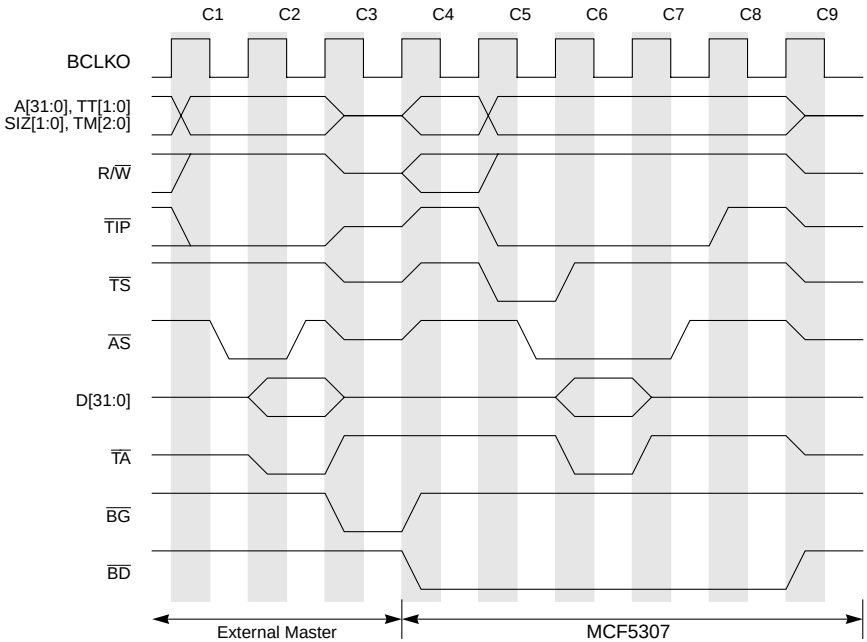


Figure 18-27. Two-Wire Bus Arbitration with Bus Request Asserted

In the second situation, the bus is granted to the MCF5307, but it does not have an internal bus request pending, so it takes implicit bus mastership. The MCF5307 does not drive the bus and does not assert $\overline{BD}$ if the bus has an implicit master. If an internal bus request is generated, the MCF5307 assumes explicit bus mastership. If explicit mastership was assumed because an internal request was generated, the MCF5307 immediately begins an access and asserts $\overline{BD}$.

In Figure 18-28, the external device is bus master during C1 and C2. During C3 the external device releases control of the bus by asserting $\overline{BG}$ to the MCF5307. At this point, there is an internal access pending so the MCF5307 asserts $\overline{BD}$ during C4 and begins the access. Thus, the MCF5307 becomes the explicit external bus master. Also during C4, the external device removes the grant from the MCF5307 by negating $\overline{BG}$. As the current bus master, the MCF5307 continues to assert $\overline{BD}$ until the current transfer completes. Because $\overline{BG}$ is negated, the MCF5307 negates $\overline{BD}$ during C9 and three-states the external bus, thereby returning external bus mastership to the external device.

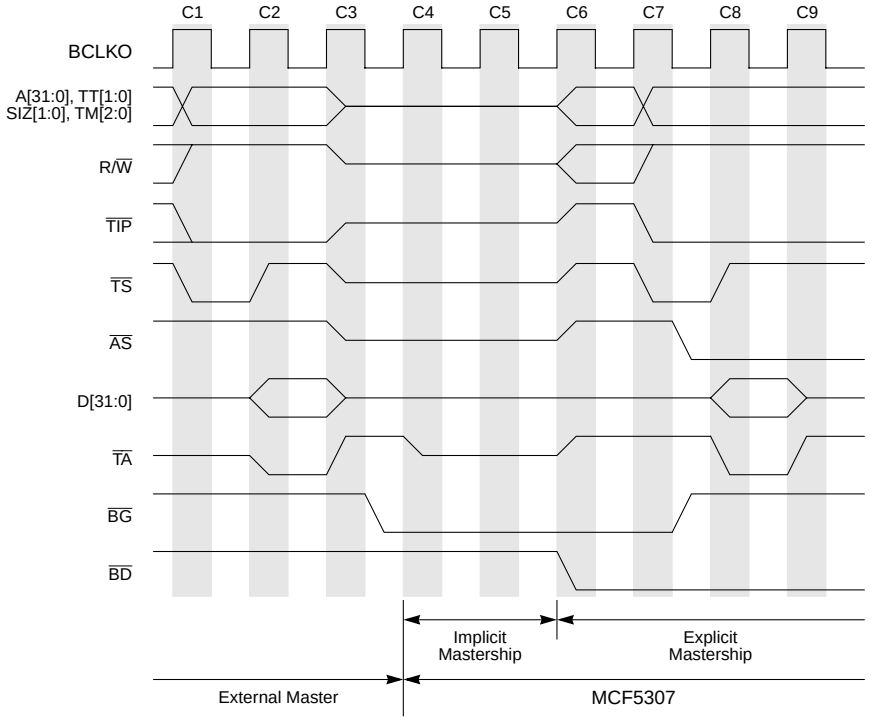


Figure 18-28. Two-Wire Implicit and Explicit Bus Mastership

In Figure 18-28, the external device is master during C1 and C2. It releases bus control in C3 by asserting $\overline{BG}$ to the MCF5307. During C4 and C5, the MCF5307 is implicit master because no internal access is pending. In C5, an internal bus request becomes pending, causing the MCF5307 to become explicit bus master in C6 by asserting $\overline{BD}$. In C7, the external device removes the bus grant to the MCF5307. The MCF5307 does not release the bus (the MCF5307 continues to assert $\overline{BD}$) until the transfer ends.

NOTE:

The MCF5307 can start a transfer in the clock cycle after $\overline{BG}$ is asserted. The external master must not assert $\overline{BG}$ to the MCF5307 while driving the bus or the part may be damaged.

Chapter 5, “Debug Support is a MCF5307 bus arbitration state diagram. States are described in Table 18-6.

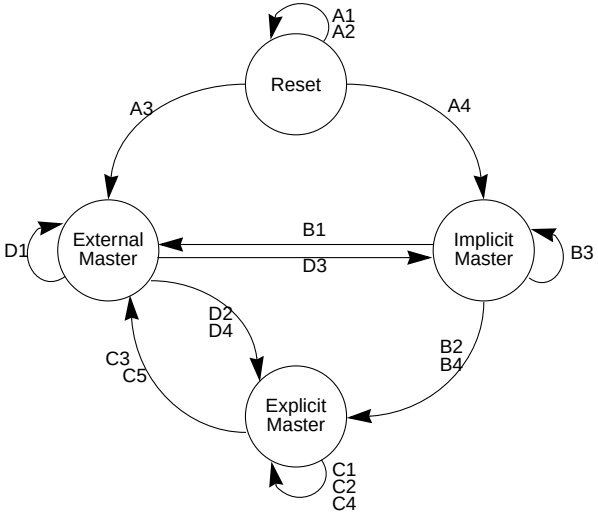


Figure 18-29. MCF5307 Two-Wire Bus Arbitration Protocol State Diagram

Table 18-10 describes the two-wire bus arbitration protocol transition conditions.

Table 18-10. MCF5307 Two-Wire Bus Arbitration Protocol Transition Conditions

Present State	Condition Label	RSTI	Software Watchdog Reset	BG	Bus Request	Transfer in Progress	End of Cycle[1]	Next State
Reset	A1	A[2]	—	—	—	—	—	Reset
	A2	N[3]	A	—	—	—	—	Reset
	A3	N	N	N	—	—	—	EM [4]
	A4	N	N	A	—	—	—	Implicit mas
Implicit Master	B1	N	N	N	—	—	—	EM
	B2	N	N	A	—	—	—	Explicit mas
	B3	N	N	A	N	—	—	Implicit mas
	B4	N	N	A	A	—	—	Explicit mas
Explicit Master	C1	N	N	A	—	—	—	Explicit mas
	C2	N	N	N	—	—	—	Explicit mas
	C3	N	N	N	—	N	—	EM
	C4	N	N	N	—	A	N	Explicit mas
	C5	N	N	N	—	A	A	EM
External Master	D1	N	N	N	—	—	—	EM mas
	D2	N	N	A	—	—	_	Explicit mas
	D3	N	N	A	N	—	—	Implicit mas
	D4	N	N	A	A	—	—	Explicit mas

1 Both normal terminations and terminations due to bus errors generate an end of cycle. Bus cycles resulting from a burst-inhibited transfer are considered part of that original transfer.

2 A means asserted.

3 N means negated.

4 EM means external master.

18.9.2 Multiple External Bus Device Arbitration Protocol (Three-Wire Mode)

Three-wire mode lets the MCF5307 share the external bus with multiple external devices. This mode requires an external arbiter to assign priorities to each potential master and to determine which device accesses the external bus. The arbiter uses the MCF5307 bus arbitration signals, $\overline{BR}$, $\overline{BD}$, and $\overline{BG}$, to control use of the external bus by the MCF5307.

The MCF5307 requests the bus from the external bus arbiter by asserting $\overline{BR}$ when the core requests an access. It continues to assert $\overline{BR}$ until after the transfer starts. It can negate $\overline{BR}$ at any time regardless of the $\overline{BG}$ status. If the MCF5307 is granted the bus when an internal bus request is generated, it asserts $\overline{BD}$ and the access begins immediately. The MCF5307 always drives $\overline{BR}$ and $\overline{BD}$, which cannot be directly wire-ORed with other devices.

The external arbiter asserts $\overline{BG}$ to grant the bus to MCF5307, which can begin a bus cycle after the next rising edge of BCLKO. If $\overline{BG}$ is negated during a bus cycle, the MCF5307 releases the bus when the cycle completes. To guarantee that the bus is released, $\overline{BG}$ must be negated before the rising edge of the BCLKO in which the last $\overline{TA}$ is asserted. Note that the MCF5307 treats any series of burst or a burst-inhibited transfers as a single bus cycle and does not release the bus until the last transfer of the series completes.

When the MCF5307 is granted the bus after it asserts $\overline{BR}$, one of two things can occur. If the MCF5307 has an internal bus request pending, it asserts $\overline{BD}$, indicating explicit bus mastership, and begins the pending bus cycle by asserting $\overline{TS}$. The MCF5307 continues to assert $\overline{BD}$ until the external bus arbiter negates $\overline{BG}$, after which $\overline{BD}$ is negated at the completion of the bus cycle. As long as $\overline{BG}$ is asserted, $\overline{BD}$ remains asserted to indicate that the MCF5307 is bus master, and the MCF5307 continuously drives the address bus, attributes, and control signals.

If no internal request is pending, the MCF5307 takes implicit bus mastership. It does not drive the bus and does not assert $\overline{BD}$ if the bus has an implicit master. If an internal bus request is generated, the MCF5307 assumes explicit bus mastership and immediately begins an access and asserts $\overline{BD}$. Figure 18-30 shows implicit and explicit bus mastership due to generation of an internal bus request.

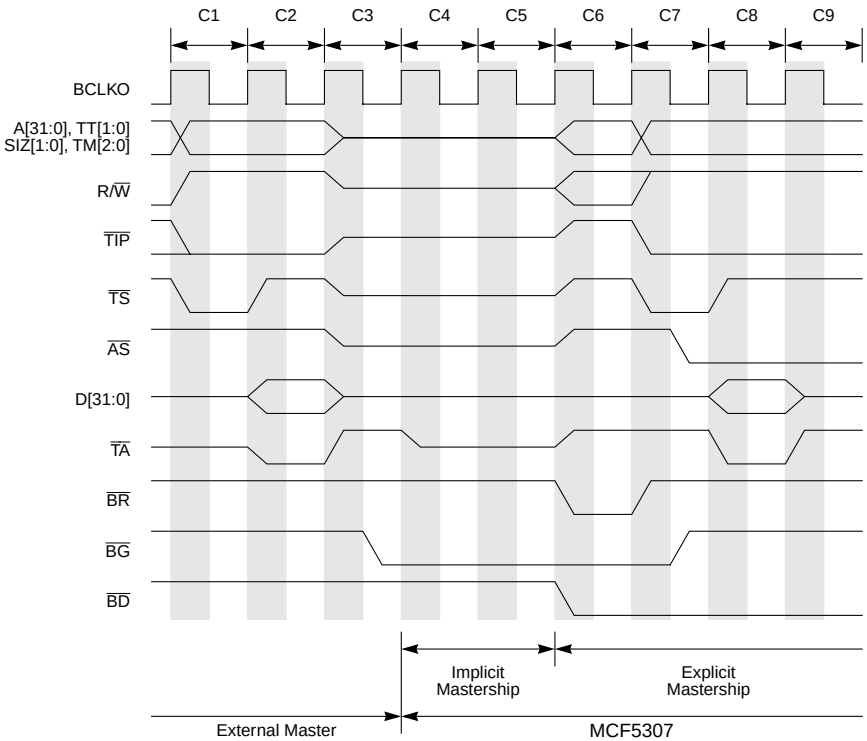


Figure 18-30. Three-Wire Implicit and Explicit Bus Mastership

In Figure 18-30, the external device is bus master during C1 and C2, releasing control in C3, at which time the external arbiter asserts $\overline{BG}$ to the MCF5307. During C4 and C5, the MCF5307 is implicit master because no internal access is pending. In C5, an internal bus request becomes pending, causing the MCF5307 to take explicit bus mastership in C6 by asserting $\overline{BR}$ and $\overline{BD}$. In C7, the external device removes the bus grant to the MCF5307. The MCF5307 does not release the bus (the MCF5307 asserts $\overline{BD}$) until the transfer ends.

NOTE:

The MCF5307 can start a transfer in the cycle after $\overline{BG}$ is asserted. The external arbiter should not assert $\overline{BG}$ to the MCF5307 until the previous external master stops driving the bus. Asserting $\overline{BG}$ during another external master's transfer may damage the part.

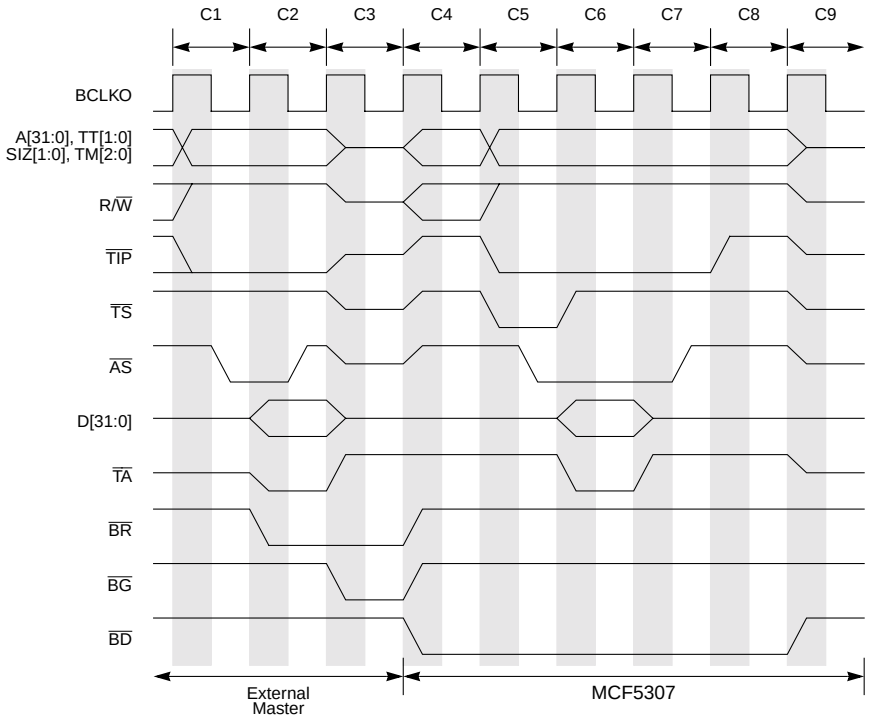


Figure 18-31. Three-Wire Bus Arbitration

In Figure 18-31, the external device is bus master during C1 and C2. During C2, the MCF5307 requests the external bus because of a pending internal transfer. On C3, the external releases mastership and the external arbiter grants the bus to the MCF5307 by asserting $\overline{BG}$. At this point, an internal is access pending so the MCF5307 asserts $\overline{BD}$ during C4 and begins the access. Thus, the MCF5307 becomes the explicit bus master. Also during C4, the external arbiter removes the grant from the MCF5307 by negating $\overline{BG}$. Because the MCF5307 is bus master, it continues to assert $\overline{BD}$ until the current transfer completes. Because $\overline{BG}$ is negated, the MCF5307 negates $\overline{BD}$ during C9 and three-states the external bus, thereby passing mastership to an external device.

The MCF5307 can assert $\overline{BR}$ to signal the external arbiter that it needs the bus. However, there is no guarantee that when the bus is granted to the MCF5307 that a bus cycle will be performed. At best, $\overline{BR}$ must be used as a status output that indicates when the MCF5307 needs the bus, but not as an indication that the MCF5307 is in a certain bus arbitration state.

Figure 18-32 is a high-level state diagram for MCF5307 bus arbitration protocol. Table 18-6 describes the four states shown in Figure 18-32.

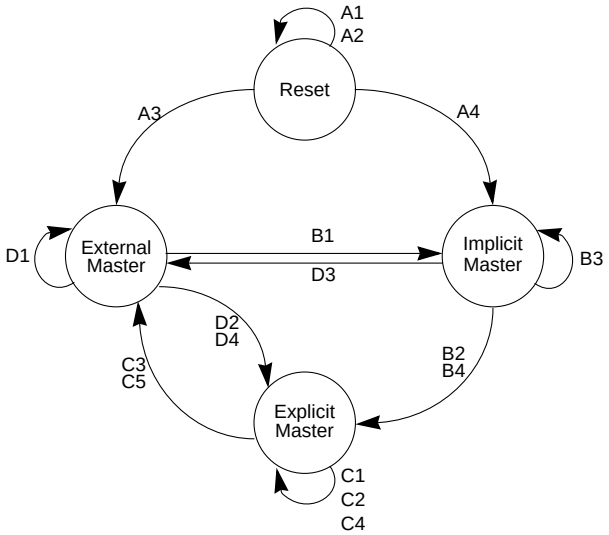


Figure 18-32. Three-Wire Bus Arbitration Protocol State Diagram

Table 18-11 lists conditions that cause state transitions.

Table 18-11. Three-Wire Bus Arbitration Protocol Transition Conditions

Current State	Condition Label	$\overline{\text{RSTI}}$	Software Watchdog Reset	$\overline{\text{BG}}$	Bus Request	Transfer in Progress	End of Cycle [1]	Next State
Reset	A1	Asserted	—	—	—	—	—	Reset
	A2	Negated	Asserted	—	—	—	—	Reset
	A3	Negated	Negated	Negated	—	—	—	EM
	A4	Negated	Negated	Asserted	—	—	—	Implicit master
Implicit master	B1	Negated	Negated	Negated	—	—	—	External device master
	B2	Negated	Negated	Asserted	—	—	—	Explicit master
	B3	Negated	Negated	Asserted	Negated	—	—	Implicit master
	B4	Negated	Negated	Asserted	Asserted	—	—	Explicit master

Table 18-11. Three-Wire Bus Arbitration Protocol Transition Conditions (Continued)

Current State	Condition Label	$\overline{\text{RSTI}}$	Software Watchdog Reset	$\overline{\text{BG}}$	Bus Request	Transfer in Progress	End of Cycle [1]	Next State
Explicit master	C1	Negated	Negated	Asserted	—	—	—	Explicit master
	C2	Negated	Negated	Negated	—	—	—	Explicit master
	C3	Negated	Negated	Negated	—	Negated	—	External device master
	C4	Negated	Negated	Negated	—	Yes	Negated	Explicit master
	C5	Negated	Negated	Negated	—	Yes	Yes	External device master
External master	D1	Negated	Negated	Negated[1]	—	—	—	External device master
	D2	Negated	Negated	Asserted	—	—	—	Explicit master
	D3	Negated	Negated	Asserted	Negated	—	—	Implicit master
	D4	Negated	Negated	Asserted	Asserted	—	—	Explicit master

[1] Both normal terminations and terminations due to bus errors generate an end of cycle. Bus cycles resulting from a burst-inhibited transfer are considered part of that original transfer.

The bus arbitration state diagram can be used for the MCF5307 three-wire bus arbitration protocol to approximate the high-level behavior of the MCF5307. It is assumed that all $\overline{\text{TS}}$ or $\overline{\text{AS}}$ signals in a system are tied together and each bus device's $\overline{\text{BD}}$ and $\overline{\text{BR}}$ signals are connected individually to the external arbiter. The external arbiter must ensure that any external masters will have released the bus after the next rising edge of before asserting $\overline{\text{BG}}$ to the MCF5307. The MCF5307 does not monitor external bus master operation regarding bus arbitration.

NOTE:

The MCF5307 can start a transfer on the rising edge of the cycle after $\overline{\text{BG}}$ is asserted. The external arbiter should not assert $\overline{\text{BG}}$ to the MCF5307 until the previous external master stops driving the bus or the part may be damaged.

18.10 Reset Operation

The MCF5307 supports two types of reset. Asserting $\overline{\text{RSTI}}$ resets the entire MCF5307. A software watchdog reset resets everything but the internal PLL module.

18.10.1 Master Reset

To perform a master reset, an external device asserts $\overline{\text{RSTI}}$. When power is applied to the system, external circuitry should assert $\overline{\text{RSTI}}$ for a minimum of 80 CLKIN cycles after Vcc is within tolerance. Figure 18-33 is a functional timing diagram of the master reset operation, showing relationships among Vcc, $\overline{\text{RSTI}}$, mode selects, and bus signals. CLKIN must be stable by the time Vcc reaches the minimum operating specification. CLKIN should start oscillating as Vcc is ramped up to clear out contention internal to the MCF5307 caused by the random states of internal flip-flops on power up. $\overline{\text{RSTI}}$ is internally synchronized for two CLKIN cycles before being used and must meet the specified setup and hold times in relationship to CLKIN to be recognized.

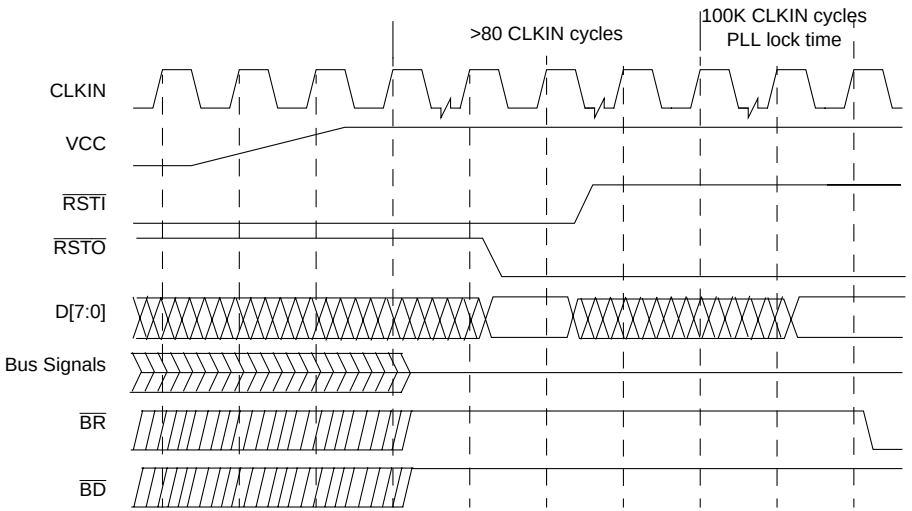


Figure 18-33. Master Reset Timing

During the master reset period, all signals capable of being three-stated are driven to a high-impedance; all others are negated. When $\overline{\text{RSTI}}$ negates, all bus signals remain in a high-impedance state until the MCF5307 is granted the bus and the core begins the first bus cycle for reset exception processing. A master reset causes any bus cycle (including DRAM refresh cycle) to terminate and initializes registers appropriately for a reset exception.

Note that during reset D[7:0] are sampled on the negating edge of $\overline{\text{RSTI}}$ for initial MCF5307 configurations listed in Table 18-12.

Table 18-12. Data Pin Configuration

Pin	Function
D7	Auto-Acknowledge Configuration (AA_CONFIG)
D[6:5]	Port Size Configuration (PS_CONFIG[1:0])
D4	Address Configuration (ADDR_CONFIG/D4)
D[3:2]	Frequency of CLKIN (FREQ[1:0])
D[1:0]	Ratio of BCLKO/Processor Clock {DIVIDE[1:0])

See Section 17.5.5, “Data/Configuration Pins (D[7:0]).” Motorola recommends that the data pins be driven rather than using a weak pull-up or pull-down resistor. Table 17-1 lists the encoding of these pins sampled at reset.

18.10.2 Software Watchdog Reset

A software watchdog reset is performed if the executing software does not provide the correct write data sequence with the enable-control bit set. This reset helps prevent runaway software or unterminated bus cycles. Figure 18-34 is a functional timing diagram of the software watchdog reset operation, showing $\overline{\text{RSTO}}$ and bus signal relationships.

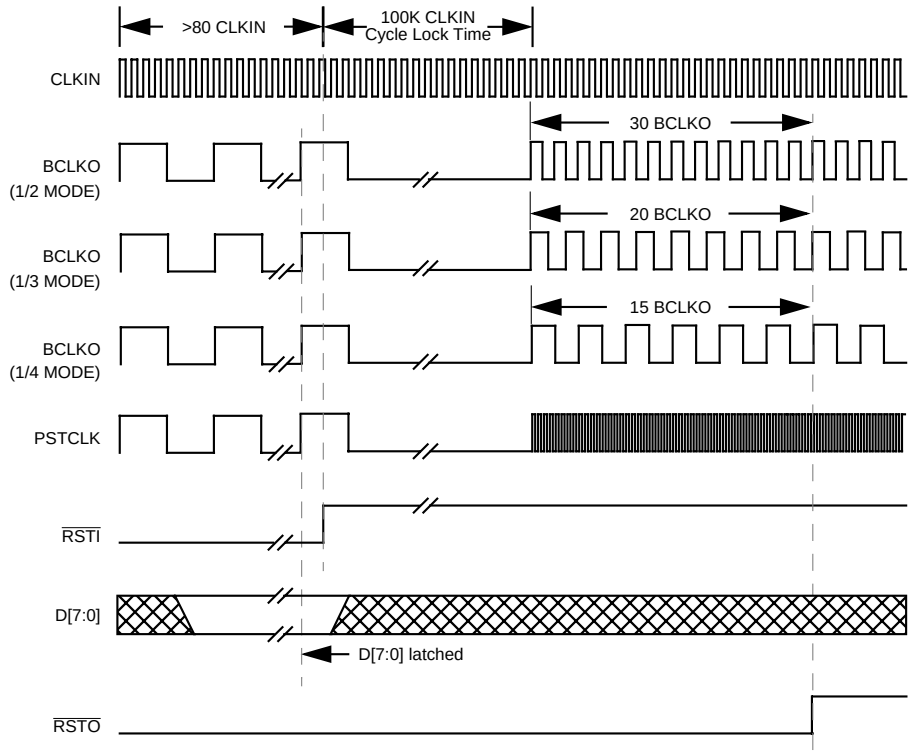


Figure 18-34. Software Watchdog Reset Timing

During the software watchdog reset period, all signals that can be are driven to a high-impedance state; all those that cannot be are negated. When $\overline{\text{RSTO}}$ negates, bus signals remain in a high-impedance state until the MCF5307 is granted the bus and the ColdFire core begins the first bus cycle for reset exception processing.