SECTION 6
BUS OPERATION

The MCF5206 bus interface supports synchronous data transfers that can be terminated synchronously or asynchronously and burst or burst-inhibited between the MCF5206 and other devices in the system. This section describes the function of the bus, the signals that control the bus, and the bus cycles provided for data-transfer operations. Operation of the bus is defined for transfers initiated by the MCF5206 as a bus master and for transfers initiated by an alternate bus master (Note: “alternate bus master” and “external bus master” are used interchangeably). The section includes descriptions of the error conditions, bus arbitration, and the reset operation.

6.1 FEATURES

The following list summarizes the key bus operation features:

- As many as 28 bits of address and 32 bits of data
- Access 8-, 16-, and 32-bit port sizes
- Generates byte, word, longword, and line size transfers
- Bus arbitration for alternate masters
- Burst and burst-inhibited transfer support
- Internal termination generation
- Termination generation for alternate masters

6.2 BUS AND CONTROL SIGNALS

6.2.1 Address Bus (A[27:0])

These three-state bidirectional signals provide the location of a bus transfer (except for interrupt-acknowledge transfers) when the MCF5206 is the bus master. When an alternate bus master controls the bus, the address signals are examined when transfer start ($\overline{TS}$) is asserted to determine if the MCF5206 should assert chip-select, DRAM control, and/or transfer terminal signals . During an interrupt-acknowledge access, address lines A[27:5] are driven high, A[1:0] are driven low, and the address lines A[4:2] indicate the interrupt level being acknowledged.

NOTE

The ColdFire core outputs 32 bits of address to the internal bus controller. Of these 32 bits, only A[27:0] are output to pins on the MCF5206. The output of A[27:24] depends on the setting of PAR[3:0] in the Pin Assignment Register (PAR) in the SIM. Refer to **Section 7.3.2.10 Pin Assignment Register (PAR)** on how to program the Pin Assignment Register (PAR).

6.2.2 Data Bus (D[31:0])

These three-state bidirectional signals provide the general-purpose data path between the MCF5206 and all other devices. The data bus can transfer 8, 16, 32, or 128 bits of data per bus transfer. A write cycle drives all 32 bits of the data bus regardless of the port width and operand size.

6.2.3 Transfer Start ($\overline{TS}$)

The MCF5206 asserts this three-state bidirectional signal for one clock period to indicate the start of each bus cycle. During alternate master accesses, the MCF5206 monitors transfer start ($\overline{TS}$) to detect the start of each alternate master bus cycle to determine if chip-select, DRAM, and/or transfer termination signals should be asserted.

6.2.4 Read/Write (R/$\overline{W}$)

This three-state bidirectional signal defines the data transfer direction for the current bus cycle. A high (logic one) level indicates a read cycle; a low (logic zero) level indicates a write cycle. When an alternate bus master is controlling the bus, the MCF5206 monitors this signal to determine if chip-select or DRAM control signals should be asserted.

6.2.5 Size (SIZ[1:0])

These three-state bidirectional signals indicate the data size for the bus cycle. When an alternate bus master is controlling the bus, the MCF5206 monitors these signals to determine the data size for asserting the appropriate memory control signals. Table 6-1 shows the definitions of the SIZx encoding.

Table 6-1. SIZx Encoding

SIZ1	SIZ0	TRANSFER SIZE
0	0	Longword (4 Bytes)
0	1	Byte
1	0	Word (2 Bytes)
1	1	Line (16 Bytes)

6.2.6 Transfer Type (TT[1:0])

These three-state output signals indicate the type of access for the current bus cycle. Table 6-2 lists the definitions of the TTx encodings.

Table 6-2. Transfer Type Encoding

TT1	TT0	TRANSFER TYPE
0	0	Normal Access
0	1	Reserved
1	0	Debug Access
1	1	CPU Space/Acknowledge Access

The MCF5206 does not sample TT[1:0] during alternate master transfers.

6.2.7 Access Type and Mode (ATM)

This three-state output signal provides supplemental information for each transfer cycle type. Table 6-3 lists the encoding for normal, debug and CPU space/acknowledge transfer types.

Table 6-3. ATM Encoding

TRANSFER TYPE	INTERNAL TRANSFER MODIFIER	ATM (TS=0)	ATM (TS=1)
00 (Normal Access)	Supervisor Code	1	1
	Supervisor Data	0	1
	User Code	1	0
	User Data	0	0
10 (Debug Access)	Supervisor Code	1	1
	Supervisor Data	0	1
11 (CPU Space/Acknowledge)	CPU Space - MOVEC Instruction	0	0
	Interrupt Acknowledge - Level 7	1	0
	Interrupt Acknowledge - Level 6	1	0
	Interrupt Acknowledge - Level 5	1	0
	Interrupt Acknowledge - Level 4	1	0
	Interrupt Acknowledge - Level 3	1	0
	Interrupt Acknowledge - Level 2	1	0
	Interrupt Acknowledge - Level 1	1	0

The MCF5206 does not sample ATM during alternate master transfers.

6.2.8 Asynchronous Transfer Acknowledge ($\overline{\text{ATA}}$)

This active-low asynchronous input signal indicates the successful completion of a requested data transfer operation. Asynchronous transfer acknowledge ($\overline{\text{ATA}}$) is an input signal from the referenced slave device indicating completion of the transfer. ($\overline{\text{ATA}}$) is synchronized internal to the MCF5206.

NOTE

The internal synchronized version of ($\overline{\text{ATA}}$) will be referred to as “internal asynchronous transfer acknowledge.’’ Because of the time required to internally synchronize $\overline{\text{ATA}}$, during a read cycle, data is latched on the rising edge of CLK when the

internal asynchronous transfer acknowledge is asserted. Consequently, data must remain valid for at least one CLK cycle after the assertion of $\overline{ATA}$. Similarly, during a write cycle, data is driven until the rising edge of CLK when the internal asynchronous transfer acknowledge is asserted.

$\overline{ATA}$ must be driven for one full CLK to ensure that the MCF5206 properly synchronizes the signal. For the MCF5206 to accept the transfer as successful with an $\overline{ATA}$, transfer error acknowledge $\overline{TEA}$ must be negated until the internal asynchronous transfer acknowledge is asserted or the transfer will be completed with a bus error.

Asynchronous transfer acknowledge ($\overline{ATA}$) is not used for termination during DRAM accesses.

6.2.9 Transfer Acknowledge ($\overline{TA}$)

This three-state bidirectional active-low synchronous signal indicates the successful completion of a requested data transfer operation. During MCF5206-initiated transfers, transfer acknowledge ($\overline{TA}$) is an input signal from the referenced slave device indicating completion of the transfer. For the MCF5206 to accept the transfer as successful with a transfer acknowledge, $\overline{TEA}$ must be negated throughout the transfer.

$\overline{TA}$ is not used for termination during MCF5206-initiated DRAM accesses.

When an alternate master is controlling the bus, the MCF5206 can drive $\overline{TA}$ to indicate the completion of the requested data transfer. If the alternate master-requested transfer is to a chip-select or default memory, the assertion of $\overline{TA}$ is controlled by the number of wait states and the setting of the alternate master automatic acknowledge (EMAA) bit in the Chip-Select Control Registers (CSCRs) or the Default Memory Control Register (DMCR). If the alternate master-requested transfer is a DRAM access, the MCF5206 drives $\overline{TA}$ as an output and is asserted at the completion of the transfer.

6.2.10 Transfer Error Acknowledge ($\overline{TEA}$)

The external slave asserts this active-low input signal to indicate an error condition for the current transfer. The assertion of $\overline{TEA}$ immediately aborts the bus cycle. The assertion of $\overline{TEA}$ has precedence over the assertion of asynchronous transfer acknowledge ($\overline{ATA}$) and transfer acknowledge ($\overline{TA}$).

NOTE

$\overline{TEA}$ can be asserted up to one clock after the assertion of asynchronous transfer acknowledge ($\overline{ATA}$) and still be recognized.

$\overline{TEA}$ has no effect during DRAM accesses.

6.3 BUS EXCEPTIONS

6.3.1 Double Bus Fault

If the MCF5206 experiences a double bus fault, it will enter the halted state. To exit the halt state, reset the MCF5206.

6.4 BUS CHARACTERISTICS

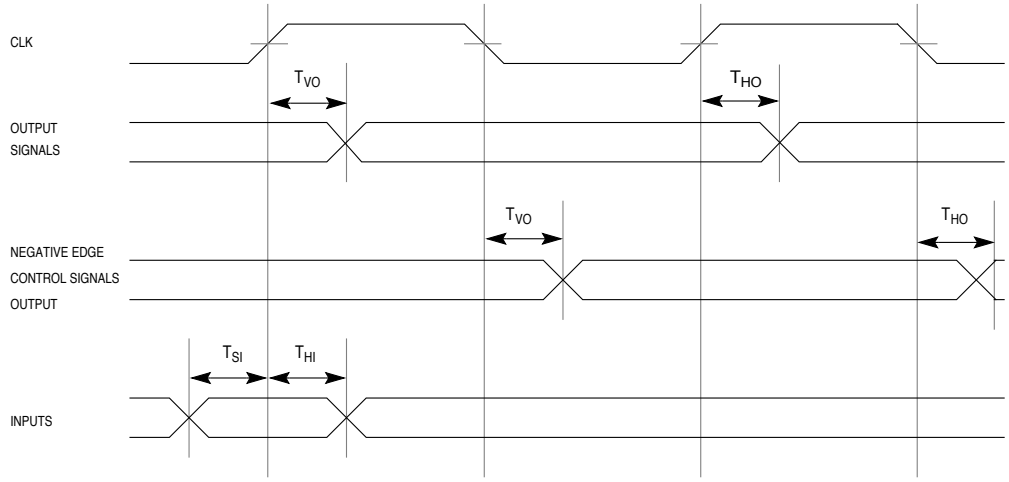
The MCF5206 uses the address bus (A[27:0]) to specify the location for a data transfer and the data bus (D[31:0]) to transfer the data. Control and attribute signals indicate the beginning and type of a bus cycle as well as the address space, direction, and size of the transfer. The selected device or the number of wait states programmed in the memory control register (the Chip-Select Control Register (CSCR), the DRAM Controller Control Registers (DCCR, including the DRAM Controller Timing Register (DCTR)), or the Default Memory Control Register (DMCR)) control the length of the cycle.

The MCF5206 CLK is distributed internally to provide logic timing. All bus signals are synchronous with the rising edge of CLK with the exception of row address strobes ($\overline{RAS}$[1:0]) and column address strobes ($\overline{CAS}$[3:0]), which can be asserted and negated synchronous with the falling edge of CLK.

Inputs to the MCF5206 (other than the interrupt priority level signals ($\overline{IPLx}$), reset in ($\overline{RSTI}$) and $\overline{ATA}$ signals) are synchronously sampled and must be stable during the sample window defined by t_{si} and t_{hi} (as shown in Figure 6-1) to guarantee proper operation. The asynchronous $\overline{IPLx}$, $\overline{RSTI}$ and $\overline{ATA}$ signals are internally synchronized to resolve the input to a valid level before being used.

Outputs to the MCF5206 begin to transition on the rising CLK edges, with the exception of $\overline{RAS}$[1:0] and $\overline{CAS}$[3:0], which begin to transition on the falling CLK edges. Specifically, $\overline{RAS}$[1:0] is asserted and negated synchronous with the falling edge of CLK, while $\overline{CAS}$[3:0] is asserted synchronous with the falling edge of CLK and can be negated synchronous with either the falling edge or the rising edge of CLK.

During alternate master accesses where the MCF5206 drives $\overline{TA}$ as an output, $\overline{TA}$ will always be driven negated for one CLK cycle before being placed in a high-impedence state.



T_{VO} = PROPAGATION DELAY OF SIGNAL RELATIVE TO CLK EDGE

T_{HO} = OUTPUT HOLD TIME RELATIVE TO CLK EDGE

T_{SI} = REQUIRED INPUT SETUP TIME RELATIVE TO CLK EDGE

T_{HI} = REQUIRED INPUT HOLD TIME RELATIVE TO CLK EDGE

Figure 6-1. Signal Relationships to CLK

6.5 DATA TRANSFER MECHANISM

The MCF5206 supports byte, word, and longword operands and allows accesses to 8-, 16-, and 32-bit data ports. With the MCF5206, you can select the port size of the specific memory, enable internal generation of transfer termination, and set the number of wait states for the external slave being accessed by programming the Chip-Select Control Registers (CSCRs), the DRAM Controller Control Registers (DCCRs), and the Default Memory Control Register (DMCR). For more information on programming these registers, refer to the SIM, Chip-Select, and DRAM Controller sections.

NOTE

The MCF5206 compares the address for the current bus transfer with the address and mask bits in the Chip-Select Address Registers (CSAR), DRAM Controller Address Registers (DCARs), the Chip-Select Mask Registers (CSMR), and DRAM Controller Mask Registers (DCMR), looking for a match. The priority is listed in Table 6-4 (from highest priority to lowest priority):

Table 6-4. Chip Select, DRAM and Default Memory Address Decoding Priority

HIGHEST PRIORITY	Chip Select 0
	Chip Select 1
	Chip Select 2
	Chip Select 3
	Chip Select 4
	Chip Select 5
	Chip Select 6
	Chip Select 7
	DRAM Bank 0
	DRAM Bank 1
LOWEST PRIORITY	Default Memory

The MCF5206 will compare the address and mask in chip-select 0 - 7 control registers (chip-select 0 is compared first), then the address and mask in DRAM bank 0 - 1 control registers. If the address does not match in either or these, the MCF5206 will use the control bits in the Default Memory Control Register (DMCR) to control the bus transfer. If the Default Memory Control Register (DMCR) control bits are used, no chip-select or DRAM control signals will be asserted during the transfer.

6.5.1 Bus Sizing

The MCF5206 reads the port size for each transfer from either the Chip-Select Control Registers (CSCRs), the DRAM Controller Control Registers (DCCRs), or the Default Memory Control Register (DMCR) at the start of each bus cycle. This allows the MCF5206 to transfer operands from 8-, 16-, or 32-bit ports. The size of the transfer is adjusted to accommodate the port size indicated. A 32-bit port must reside on data bus bits D[31:0], a 16-bit port must reside on data bus bits D[31:16], and an 8-bit port must reside on data bus bits D[31:24]. This requirement ensures that the MCF5206 correctly transfers valid data to 8-, 16-, and 32-bit ports.

The bytes of operands are designated as shown in Figure 6-2. The most significant byte of a longword operand is OP0; OP3 is the least significant byte. The two bytes of a word length operand are OP2 (most significant) and OP3. The single byte of a byte length operand is OP3. These designations are used in the figures and descriptions that follow.

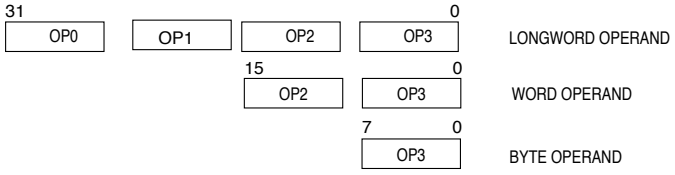


Figure 6-2. Internal Operand Representation

Figure 6-3 shows the required organization of data ports on the MCF5206 for 8-, 16-, and 32 bit devices. The four bytes shown are connected through the internal data bus and data multiplexer to the external data bus. This path is how the MCF5206 supports programmable port sizing and operand misalignment. The data multiplexer establishes the necessary connections for different combinations of address and data sizes.

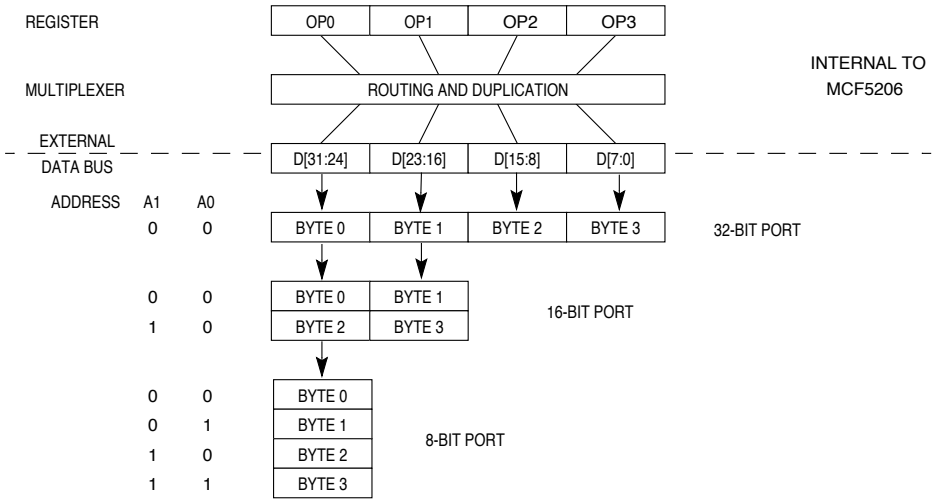


Figure 6-3. MCF5206 Interface to Various Port Sizes

The multiplexer takes the four bytes of the 32-bit bus and routes them to their required positions. For example, OP3 can be routed to D[7:0], as would be the normal case when interfacing to a 32-bit port. OP3 can be routed to D[23:16] for interfacing to a 16-bit port, or it can be routed to D[31:24] for interfacing to an 8-bit port. The operand size, address, and port size of the memory being accessed determines the positioning of bytes.

The MCF5206 can burst anytime the port size of the external slave being accessed is smaller than the operand size. If bursting is enabled, the MCF5206 will burst transfers depending on the port size and operand alignment. For any transfer, the number of bytes transferred during a bus cycle is equal to or less than the size indicated by the SIZx outputs. For example, during the first bus cycle of a longword transfer to a 16-bit port where bursting is enabled, the SIZx outputs will remain constant throughout the transfer and will indicate that four bytes are to be transferred, although only two bytes are moved at a time.Table 6-5 lists the encodings for the SIZx bits for each port size for transfers where bursting is both enabled and disabled.

A[0] and A[1] also affect operation of the data multiplexer. During an operand transfer, A[31:2] indicate the longword base address of that portion of the operand to be accessed; A[1] and A[0] indicate the byte offset from the base. Table 6-6 lists the encoding of A[1] and A[0] and the corresponding byte offset from the longword base.

Table 6-5. SIZx Encoding for Burst- and Bursting-Inhibited Ports

OPERAND SIZE	32-BIT PORT				16 -BIT PORT				8-BIT PORT			
	BURSTING ENABLED		BURSTING INHIBITED		BURSTING ENABLED		BURSTING INHIBITED		BURSTING ENABLED		BURSTING INHIBITED	
	SIZ1	SIZ0	SIZ1	SIZ0	SIZ1	SIZ0	SIZ1	SIZ0	SIZ1	SIZ0	SIZ1	SIZ0
BYTE	0	1	0	1	0	1	0	1	0	1	0	1
WORD	1	0	1	0	1	0	1	0	1	0	0	1
LONGWORD	0	0	0	0	0	0	1	0	0	0	0	1
LINE	1	1	0	0	1	1	1	0	1	1	0	1

Table 6-6. Address Offset Encoding

A1	A0	OFFSET
0	0	+0 Byte
0	1	+1 Byte
1	0	+2 Bytes
1	1	+3 Bytes

Table 6-7 lists the bytes that should be driven on the data bus during read cycles by the slave device being accessed. The entries shown as Byte X are portions of the requested operand that are read. The operand being read is defined by SIZ[1], SIZ[0], A[0], and A[1] for the bus cycle. Bytes labeled X are “don’t cares” and are not required during that read cycle. Bytes labeled “-” indicates that this transfer is not valid.

Table 6-7. Data Bus Requirement for Read Cycles

TRANSFER SIZE	SIZE		ADDRESS		32 BIT PORT EXTERNAL DATA BYTES REQUIRED				16 BIT PORT EXTERNAL DATA BYTES REQUIRED		8 BIT PORT EXTERNAL DATA BYTES REQUIRED
	SIZ1	SIZ0	A1	A0	D[31:24]	D[23:16]	D[15:8]	D[7:0]	D[31:24]	D[23:16]	D[31:24]
BYTE	0	1	0	0	Byte 0	X	X	X	Byte 0	X	Byte 0
			0	1	X	Byte 1	X	X	X	Byte 1	Byte 1
			1	0	X	X	Byte 2	X	Byte 2	X	Byte 2
			1	1	X	X	X	Byte 3	X	Byte 3	Byte 3
WORD	1	0	0	0	Byte 0	Byte 1	X	X	Byte 0	Byte 1	Byte 0
			0	1	-	-	-	-	-	-	Byte 1
			1	0	X	X	Byte2	Byte 3	Byte 2	Byte 3	Byte 2
			1	1	-	-	-	-	-	-	Byte 3
LONGWORD	0	0	0	0	Byte 0	Byte 1	Byte 2	Byte 3	Byte 0	Byte 1	Byte 0
			0	1	-	-	-	-	-	-	Byte 1
			1	0	-	-	-	-	Byte 2	Byte 3	Byte 2
			1	1	-	-	-	-	-	-	Byte 3

Table 6-7. Data Bus Requirement for Read Cycles (Continued)

TRANSFER SIZE	SIZE		ADDRESS		32 BIT PORT EXTERNAL DATA BYTES REQUIRED				16 BIT PORT EXTERNAL DATA BYTES REQUIRED		8 BIT PORT EXTERNAL DATA BYTES REQUIRED
	SIZ1	SIZ0	A1	A0	D[31:24]	D[23:16]	D[15:8]	D[7:0]	D[31:24]	D[23:16]	D[31:24]
LINE	1	1	0	0	Byte 0	Byte 1	Byte 2	Byte 3	Byte 0	Byte 1	Byte 0
			0	1	-	-	-	-	-	-	Byte 1
			1	0	-	-	-	-	Byte 2	Byte 3	Byte 2
			1	1	-	-	-	-	-	-	Byte 3

Figure 6-4 is a flowchart for read transfers to 8-, 16-, or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer.

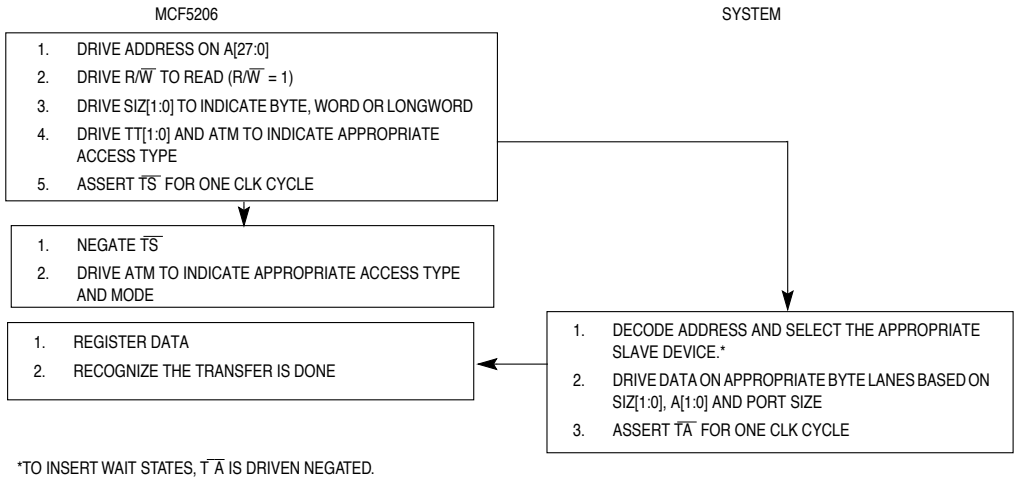


Figure 6-4. Byte-, Word-, and Longword-Read Transfer Flowchart

Figure 6-5 shows a longword supervisor code read from a 32-bit port.

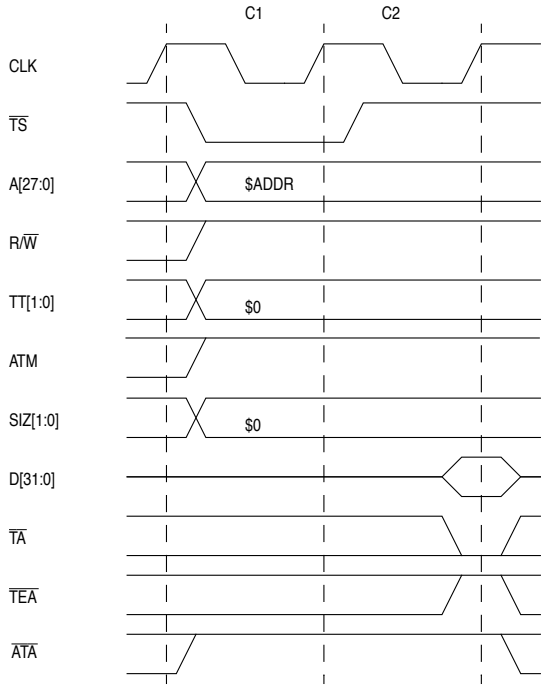


Figure 6-5. Longword-Read Transfer From a 32-Bit Port (No Wait States)

Clock 1 (C1)

The read cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type. Access type and mode (ATM) identifies the transfer as reading code. The read/write (R/$\overline{W}$) signal is driven high for a read cycle, and the size signals (SIZ[1:0]) are driven low to indicate a longword transfer. The MCF5206 asserts transfer start ($\overline{TS}$) to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates transfer start ($\overline{TS}$), drives access type and mode (ATM) high to identify the transfer as supervisor. The selected device(s) places the addressed data onto D[31:0] and asserts the transfer acknowledge ($\overline{TA}$). At the end of C2, the MCF5206 samples the level of $\overline{TA}$ and if $\overline{TA}$ is asserted, latches the current value of D[31:0]. If $\overline{TA}$ is asserted, the transfer of the longword is complete and the transfer terminates. If $\overline{TA}$ is negated, the MCF5206 continues to sample $\overline{TA}$ and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted. If the bus monitor timer is enabled and $\overline{TA}$ is not

asserted before the programmed bus monitor time is reached, the cycle will be terminated with an internal bus error.

Table 6-8 lists the combinations of SIZ[1:0], A[1:0] and the corresponding pattern of the data transfer for write cycles from the internal multiplexer of the MCF5206 to the external data bus. For example, if a longword transfer is generated to a 16-bit port, the MCF5206 will start the cycle with A[1:0] set to $0 and read the first word. The MCF5206 will then increment A[1:0] to $2 and will read the second word. The data for both word reads will be sampled from DATA[31:16]. Bytes labeled X are “don’t cares.”

Table 6-8. Internal to External Data Bus Multiplexer - Write Cycle

TRANSFER SIZE	SIZE		ADDRESS		EXTERNAL DATA BUS CONNECTION			
	SIZ1	SIZ0	A1	A0	D[31:24]	D[23:16]	D[15:8]	D[7:0]
BYTE	0	1	0	0	OP3	X	X	X
			0	1	OP3	OP3	X	X
			1	0	OP3	X	OP3	X
			1	1	OP3	OP3	X	OP3
WORD	1	0	0	0	OP2	OP3	X	X
			0	1	OP3	X	X	X
			1	0	OP2	OP3	OP2	OP3
			1	1	OP3	X	X	X
LONGWORD	0	0	0	0	OP0	OP1	OP2	OP3
			0	1	OP1	X	X	X
			1	0	OP2	OP3	X	X
			1	1	OP3	X	X	X
LINE	1	1	0	0	OP0	OP1	OP2	OP3
			0	1	OP1	X	X	X
			1	0	OP2	OP3	X	X
			1	1	OP3	X	X	X

Figure 6-6 is a flowchart for write transfers to 8-, 16-, or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer and the specific number of cycles needed for each transfer.

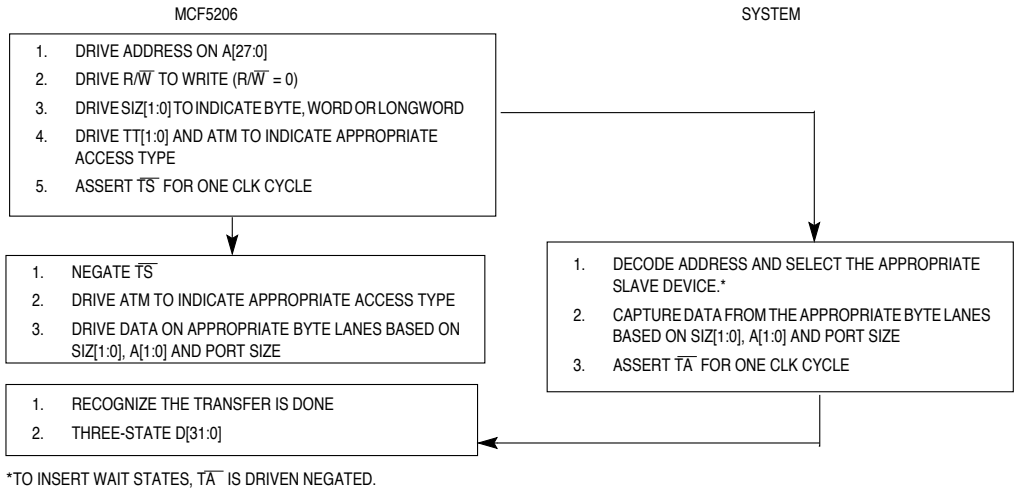


Figure 6-6. Byte-, Word-, and Longword-Write Transfer Flowchart

Figure 6-7 shows a supervisor data word-write transfer to a 16-bit port.

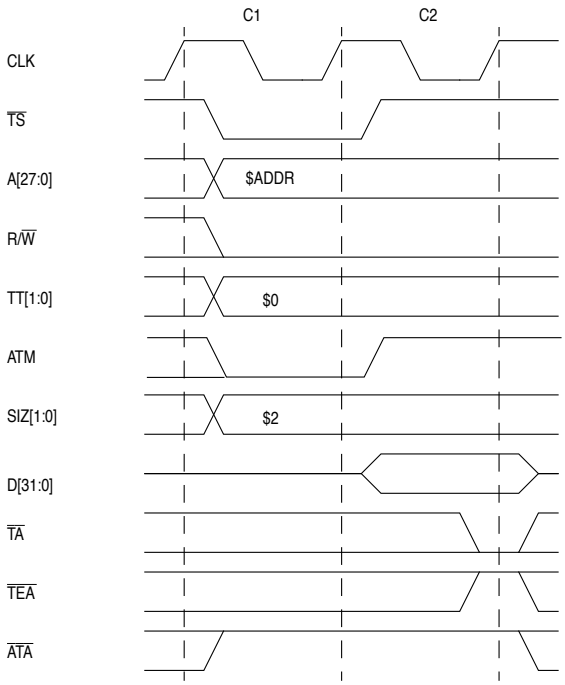


Figure 6-7. Word-Write Transfer to a 16-Bit Port (No Wait States)

Clock 1 (C1)

The write cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type. Access type and mode (ATM) identifies the transfer as writing data. The read/write (R/$\overline{W}$) signal is driven low for a write cycle, and the size signals (SIZ[1:0]) are driven to $2 to indicate a word transfer. The MCF5206 asserts transfer start ($\overline{TS}$) to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates transfer start ($\overline{TS}$), drives ATM high to identify the transfer as supervisor, and places the data on the data bus (D[31:0]). The selected device(s) asserts the transfer acknowledge ($\overline{TA}$) if it is ready to latch the data. At the end of C2, the selected device latches the current value of D[31:16], and the MCF5206 samples the level of $\overline{TA}$. If $\overline{TA}$ is asserted, the transfer of the word is complete and the transfer terminates. If $\overline{TA}$ is negated, the MCF5206 continues to output the data and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

6.5.2 Bursting Read Transfers: Word, Longword, and Line

If the burst-enable bit in the appropriate control register (Chip-Select Control Register or Default Memory Control Register) is set to 1or the transfer is to DRAM, and the operand size is larger than the port size of the memory being accessed, the MCF5206 performs word, longword, and line transfers in burst mode. When burst mode is selected, the size of the transfer (indicated by SIZ[1:0]) will reflect the size of the operand being read, not the size of the port being accessed (i.e., a line transfer will be indicated by SIZ[1:0] = $3 and a longword transfer will be indicated by SIZ[1:0] = $0, regardless of the size of the port or the number of transfers required to access the entire set of data).

The MCF5206 supports burst-inhibited transfers for memory devices that cannot support bursting. For this type of bus cycle, you should clear the burst-enable bit in the Chip-Select Control Registers (CSCRs) or Default Memory Control Register (DMCR).

NOTE

No burst-enable bit is provided for DRAM accesses. DRAM transfers are always bursted if the operand size is larger than the port size.

The MCF5206 uses line read transfers to access a 16-byte operand to support cache line filling and for a MOVEM instruction, when appropriate. A line read accesses a block of four longwords, aligned to a longword memory boundary, by supplying a starting address that points to one of the longwords and incrementing A3, A2, A1, and A0 of the supplied address for each transfer. A longword read accesses a single longword aligned to a longword boundary and increments A1 and A0 if the accessed port size is smaller than 32 bits. A word read accesses a single word of data, aligned to a word boundary and increments A0 if the accessed port size is smaller than 16 bits.

Figure 6-8 is a flowchart for bursting read transfers to 8-, 16-, or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each

transfer. A bursted read transfer can be from two to sixteen transfers long. The flowchart shown in Figure 6-8 is for a bursting transfer of four transfers long.

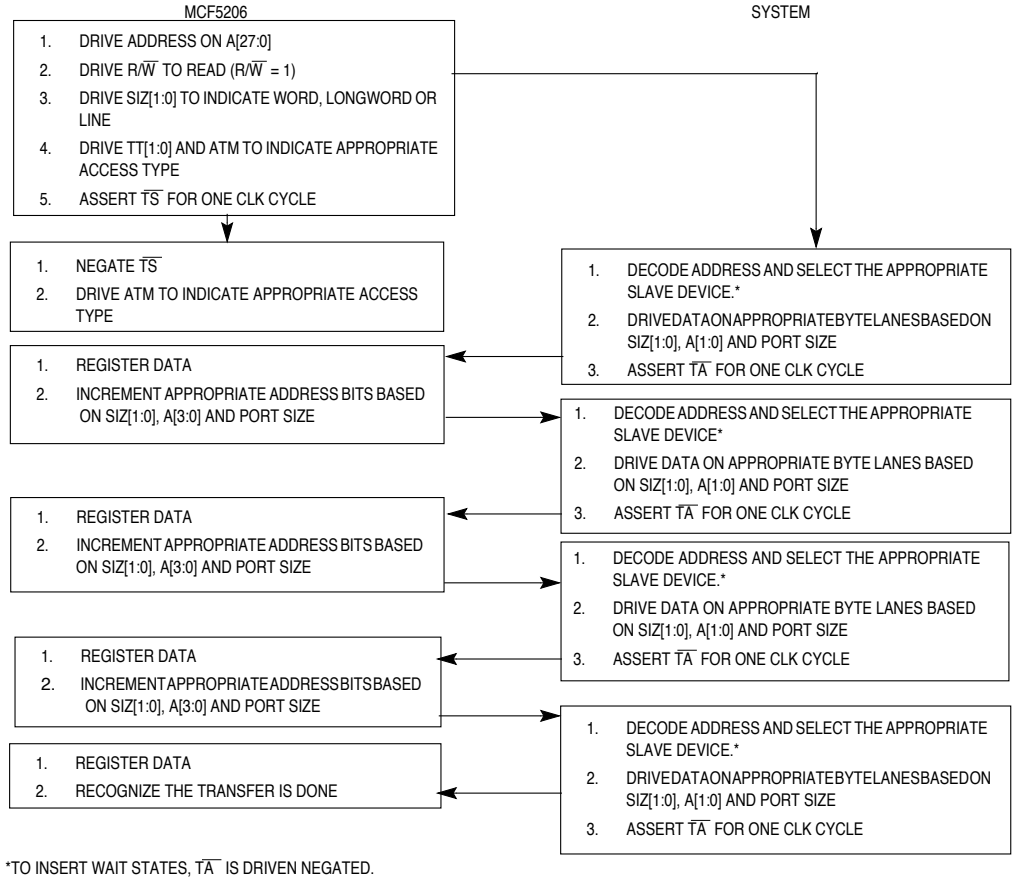


Figure 6-8. Bursting Word-, Longword-, and Line-Read Transfer Flowchart

Figure 6-9 shows a bursting user code word-read transfer from an 8-bit port.

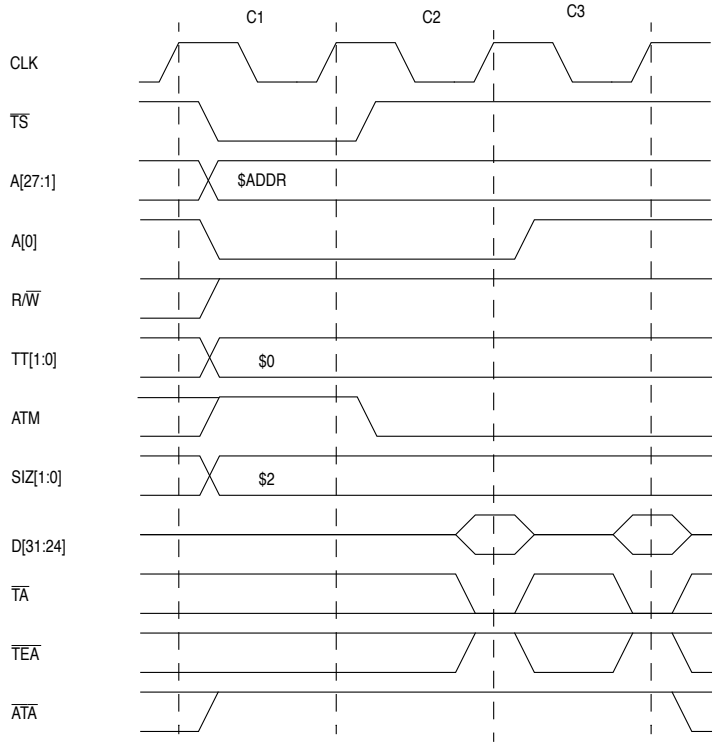


Figure 6-9. Bursting Word-Read From an 8-Bit Port (No Wait States)

Clock 1 (C1)

The read cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type. Access transfer and mode (ATM) identifies the transfer as reading code. The read/write (R/$\overline{W}$) signal is driven high for a read cycle, and the size signals (SIZ[1:0]) are driven to $2 to indicate a word transfer. The MCF5206 asserts transfer start ($\overline{TS}$) to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$, drives ATM low to identify the transfer as user. The selected device(s) places the first byte of the addressed data on to D[31:24] and asserts the transfer acknowledge ($\overline{TA}$). At the end of C2, the MCF5206 samples the level of $\overline{TA}$ and if $\overline{TA}$ is asserted, latches the current value of D[31:24]. If $\overline{TA}$ is asserted, the transfer of the first byte of the word read is complete. If $\overline{TA}$ is negated, the MCF5206 continues to sample $\overline{TA}$ and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

Clock 3 (C3)

The MCF5206 increments A0 to address the next byte of the word transfer. The selected device(s) places the second byte of the addressed data onto D[31:24] and asserts the transfer acknowledge ($\overline{TA}$). At the end of C3, the MCF5206 samples the level of $\overline{TA}$ and if $\overline{TA}$ is asserted, latches the current value of D[31:24]. If $\overline{TA}$ is asserted, the transfer of the word read is complete and the transfer is terminated. If $\overline{TA}$ is negated, the MCF5206 continues to sample $\overline{TA}$ and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

6.5.3 Bursting Write Transfers: Word, Longword, and Line

The MCF5206 uses line-write transfers to access a 16-byte operand for a MOVEM instruction, when appropriate. A line write accesses a block of four longwords, aligned to a longword memory boundary, by supplying a starting address that points to one of the longwords and increments A3, A2, A1, and A0 of the supplied address for each transfer. A longword write accesses a single longword aligned to a longword boundary and increments A1 and A0 if the accessed port size is smaller than 32 bits. A word write accesses a single word of data, aligned to a word boundary and increments A0 if the accessed port size is smaller than 16 bits. Table 6-9 lists the encodings for the SIZx bits for each port size for transfers where bursting is both enabled and disabled.

Table 6-9. SIZx Encoding for Burst- and Bursting-Inhibited Ports

OPERAND SIZE	32-BIT PORT				16 -BIT PORT				8-BIT PORT			
	BURSTING ENABLED		BURSTING INHIBITED		BURSTING ENABLED		BURSTING INHIBITED		BURSTING ENABLED		BURSTING INHIBITED	
	SIZ1	SIZ0	SIZ1	SIZ0	SIZ1	SIZ0	SIZ1	SIZ0	SIZ1	SIZ0	SIZ1	SIZ0
BYTE	0	1	0	1	0	1	0	1	0	1	0	1
WORD	1	0	1	0	1	0	1	0	1	0	0	1
LONGWORD	0	0	0	0	0	0	1	0	0	0	0	1
LINE	1	1	0	0	1	1	1	0	1	1	0	1

Figure 6-10 is a flowchart for bursting write transfers to 8-, 16-, or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer and the specific number of cycles needed for each transfer. A bursted write transfer can be from two to sixteen transfers long. The flowchart in Figure 6-10 is for a bursting transfer of four transfers long.

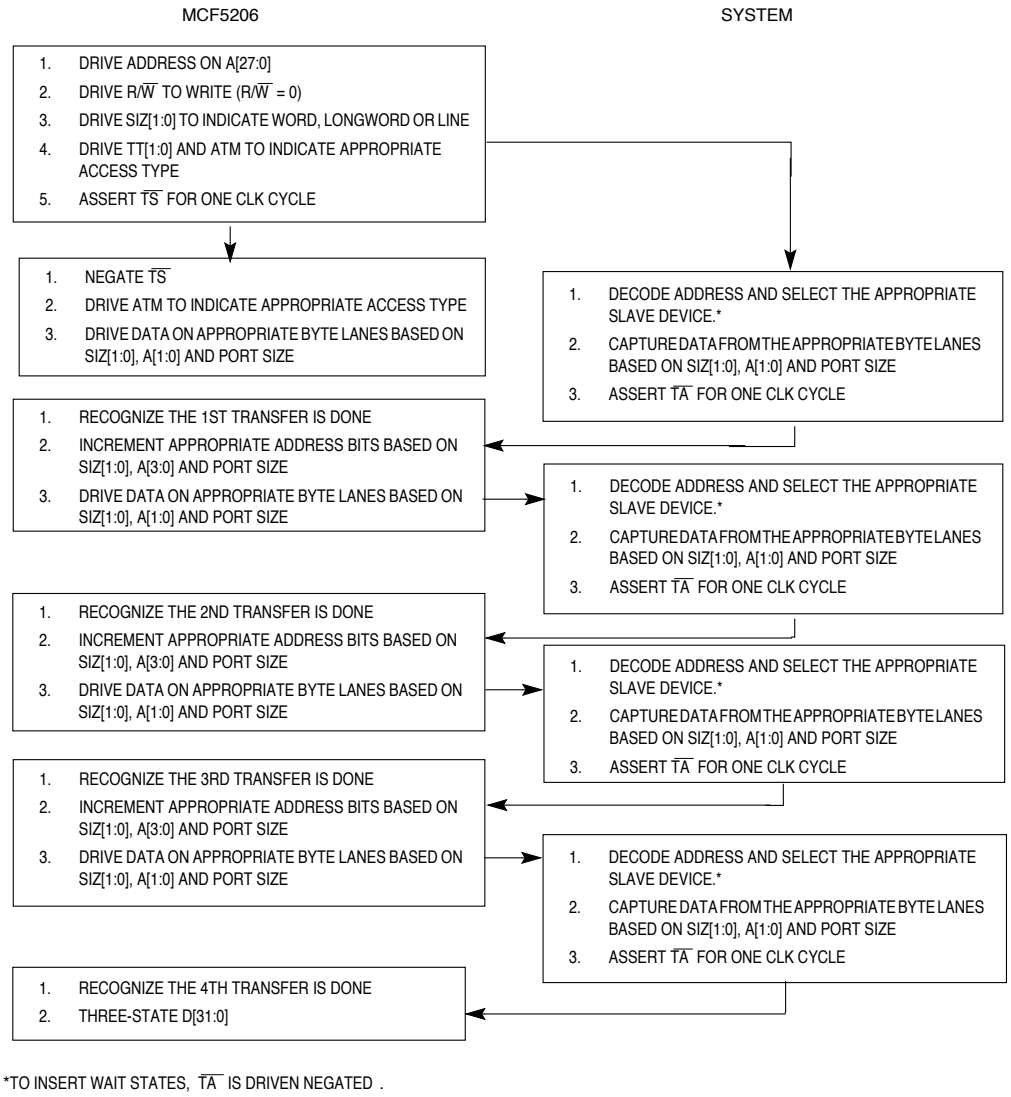


Figure 6-10. Word-, Longword-, and Line-Write Transfer Flowchart

Figure 6-11 shows a user data bursting line-write transfer to a 32-bit port.

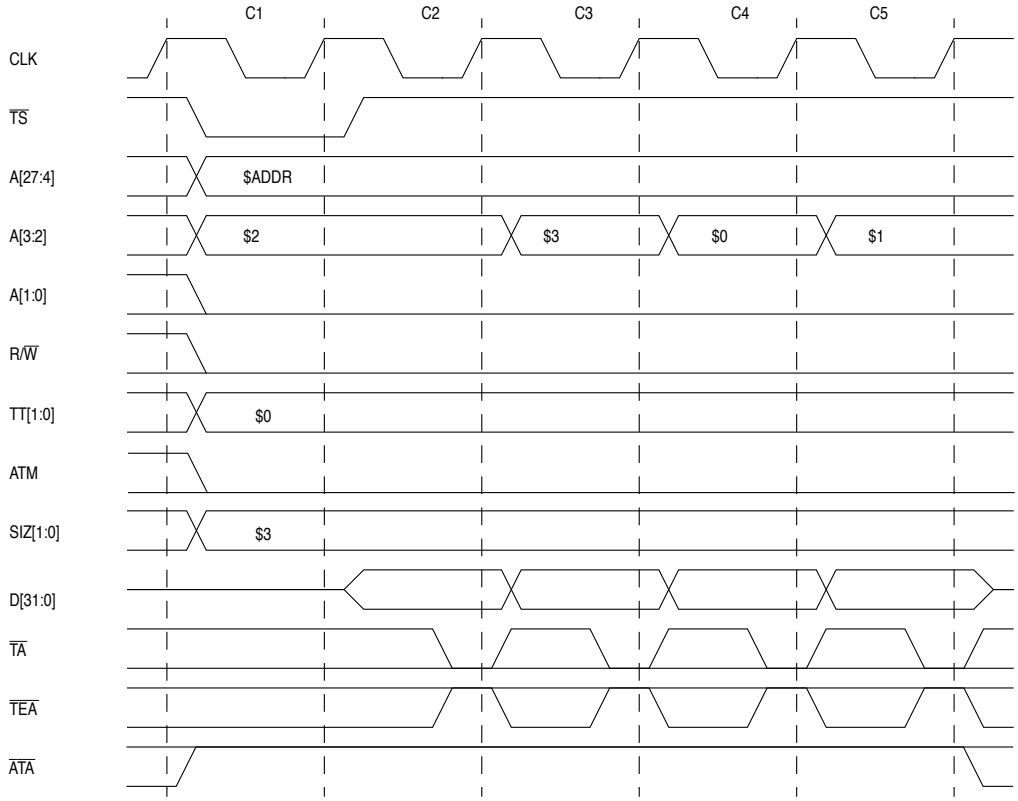


Figure 6-11. Line-Write Transfer to a 32-Bit Port (No Wait States)

Clock 1 (C1)

The write cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type. Access type and mode (ATM) identifies the transfer as data. The read/write (R/$\overline{W}$) signal is driven low for a write cycle, and the size signals (SIZ[1:0]) are driven to $3 to indicate a line transfer. The MCF5206 asserts transfer start ($\overline{TS}$) to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$, drives ATM low to identify the transfer as user, and places the data on the data bus (D[31:0]). The selected device(s) asserts the transfer acknowledge ($\overline{TA}$) if it is ready to latch the data. At the end of C2, the selected device latches the current value of D[31:0], and the MCF5206 samples the level of $\overline{TA}$. If $\overline{TA}$ is asserted, the transfer of the first longword is complete. If $\overline{TA}$ is negated, the MCF5206

continues to output the data and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

Clock 3 (C3)

The MCF5206 increments A[3:2] to address the next longword of the line transfer and drives D[31:0] with the second longword of data. The selected device(s) asserts the $\overline{TA}$ if it is ready to latch the data. At the end of C3, the MCF5206 samples the level of $\overline{TA}$ and if $\overline{TA}$ is asserted, the second longword transfer of the line write is complete. If $\overline{TA}$ is negated, the MCF5206 continues to sample $\overline{TA}$ and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

Clock 4 (C4)

This clock is identical to C3, except that once $\overline{TA}$ is asserted, the value corresponds to the third longword of data for the burst.

Clock 5 (C5)

This clock is identical to C3, except that once $\overline{TA}$ is asserted, the data value corresponds to the fourth longword of data for the burst. This is the last CLK cycle of the line-write transfer and the MCF5206 three-states D[31:0] at the start of the next CLK cycle.

6.5.4 Burst-Inhibited Read Transfer: Word, Longword, and Line

If the burst-enable bit in the appropriate control register (Chip-Select Control Register or Default Memory Control Register) is cleared and the operand size is larger than the port size of the memory being accessed, the MCF5206 performs word, longword, and line transfers in burst-inhibited mode. When burst-inhibit mode is selected, the size of the transfer (indicated by SIZ[1:0]) will reflect the port size if the operand being read is larger than the port size or the operand size if the port size is larger than the operand size. A transfer size of line (SIZ[1:0] = $3) will never be indicated in burst-inhibited mode. If the operand size is line, the size pins (SIZ[1:0]) will always indicate the port size. Refer to Table 6-9 for SIZx encodings for each port size for burst-inhibited transfers.

NOTE

All transfers to DRAM that have an operand size larger than the port size are bursted. Burst-inhibited transfers cannot be generated for DRAM accesses.

The basic transfer of a burst-inhibited read is the same as a “normal” read with the addition of more transfers until the entire operand has been accessed. Burst-inhibited read transfers can be from two to sixteen transfers long. Figure 6-12 is a flowchart for burst-inhibited read transfers (4 transfers long) to 8-, 16-, or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer.

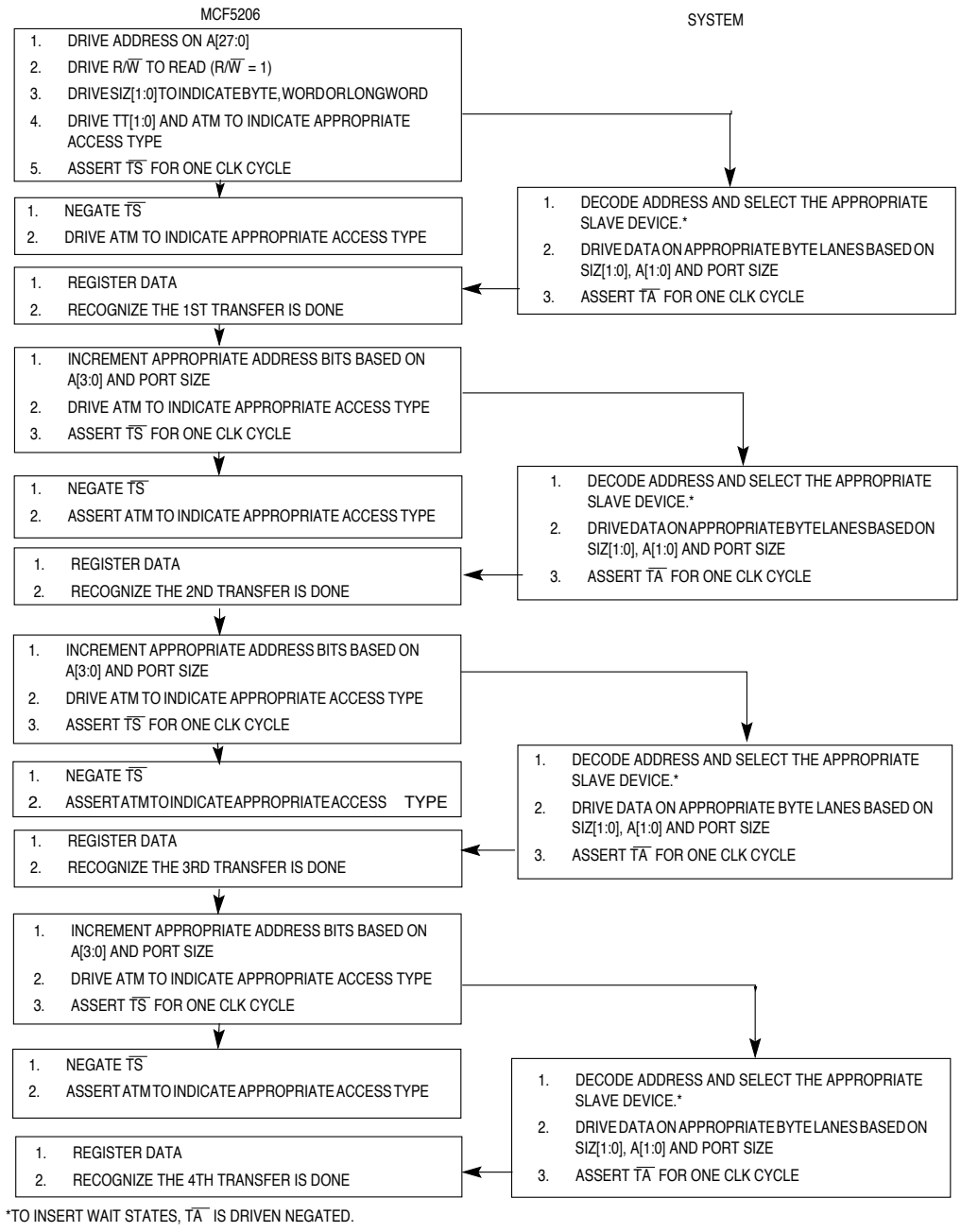


Figure 3-12. Burst-Inhibited Word-, Longword-, and Line-Read Transfer Flowchart

Figure 6-13 shows a burst-inhibited supervisor code longword-read transfer from an 8-bit port.

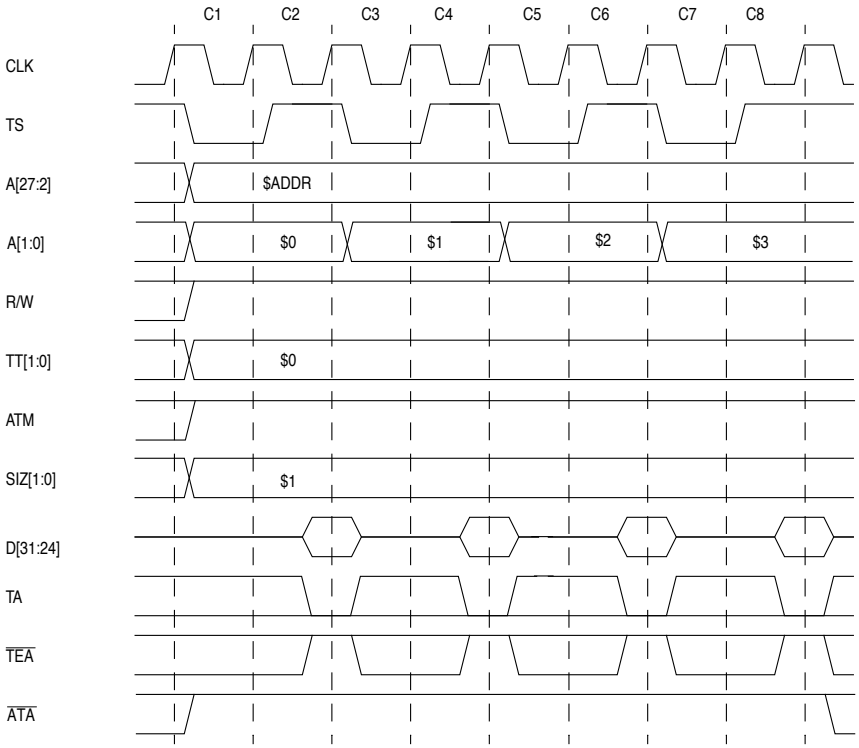


Figure 3-13. Burst-Inhibited Longword Read From an 8-Bit Port (No Wait States)

Clock 1 (C1)

The read cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type and drives ATM high to identify the transfer as code. The read/write (R/$\overline{W}$) signal is driven high for a read cycle, and the size signals (SIZ[1:0]) are driven to $1 to indicate a byte transfer. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$ and drives ATM high to identify the transfer as supervisor. The selected device(s) places the first byte of the addressed data onto D[31:24] and asserts $\overline{TA}$. At the end of C2, the MCF5206 samples the level of $\overline{TA}$ and if $\overline{TA}$ is asserted, latches the current value of D[31:24]. If $\overline{TA}$ is asserted, the transfer of the first byte of the longword read is complete. If $\overline{TA}$ is negated, the MCF5206 continues to

sample $\overline{TA}$ and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

Clock 3 (C3)

The MCF5206 increments A[1:0] to address the second byte of the longword transfer. The MCF5206 continues to drive transfer type (TT[1:0]), read/write (R/W) and size (SIZ[1:0]) signals to indicate a byte read. Access transfer mode (ATM) is driven high to indicate the transfer as code. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of the second transfer of the bus cycle.

Clock 4 (C4)

This clock is identical to C2, except that once $\overline{TA}$ is recognized asserted, the latched value corresponds to the second byte of data for the longword transfer.

Clock 5 (C5)

This clock is identical to C3, except the address is incremented to address the third byte of the longword transfer.

Clock 6 (C6)

This clock is identical to C2, except that once $\overline{TA}$ is recognized asserted, the latched value corresponds to the third byte of data for the longword transfer.

Clock 7 (C7)

This clock is identical to C3, except the address is incremented to address the fourth byte of the longword transfer.

Clock 8 (C8)

This clock is identical to C2, except that once $\overline{TA}$ is recognized asserted, the latched value corresponds to the fourth byte of data for the longword. This is the last CLK cycle of the longword-read transfer. The selected device negates $\overline{TA}$ signal and three-states D[31:24] after the next rising edge of CLK.

6.5.5 Burst-Inhibited Write Transfer: Word, Longword, and Line

The basic transfer of a burst-inhibited write is the same as “normal” write with the addition of more transfers until the entire operand has been accessed. Burst-inhibited write transfers can be from 2 to 16 transfers long. Figure 6-14 is a flowchart for burst-inhibited write transfers (4 transfers long) to 8-, 16-, or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer.

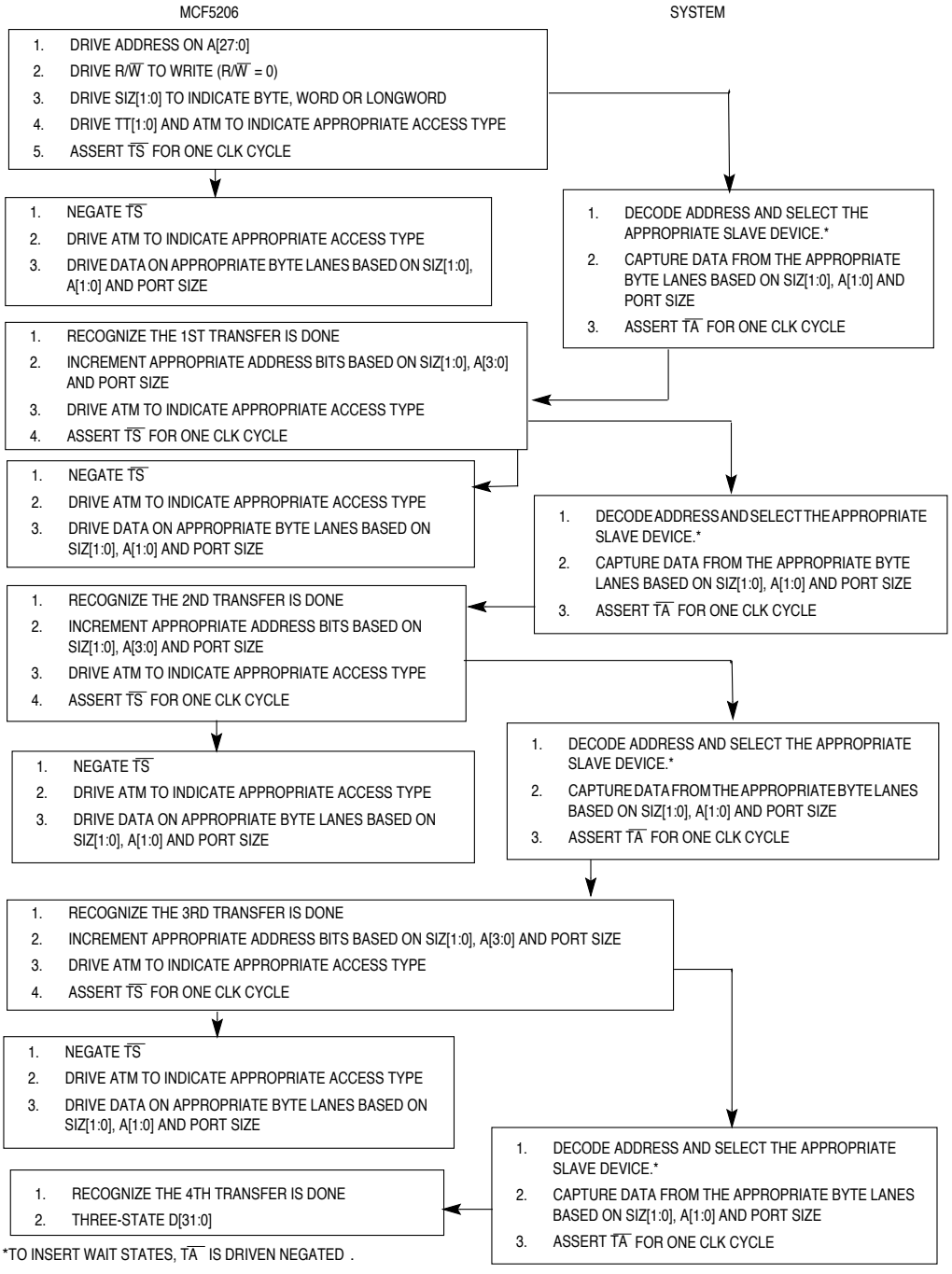

MCF5206
SYSTEM
1. DRIVE ADDRESS ON A[27:0]
2. DRIVE R/W TO WRITE (R/W = 0)
3. DRIVE SIZ[1:0] TO INDICATE BYTE, WORD OR LONGWORD
4. DRIVE TT[1:0] AND ATM TO INDICATE APPROPRIATE ACCESS TYPE
5. ASSERT TS FOR ONE CLK CYCLE
1. NEGATE TS
2. DRIVE ATM TO INDICATE APPROPRIATE ACCESS TYPE
3. DRIVE DATA ON APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
1. DECODE ADDRESS AND SELECT THE APPROPRIATE SLAVE DEVICE.*
2. CAPTURE DATA FROM THE APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
3. ASSERT TA FOR ONE CLK CYCLE
1. RECOGNIZE THE 1ST TRANSFER IS DONE
2. INCREMENT APPROPRIATE ADDRESS BITS BASED ON SIZ[1:0], A[3:0] AND PORT SIZE
3. DRIVE ATM TO INDICATE APPROPRIATE ACCESS TYPE
4. ASSERT TS FOR ONE CLK CYCLE
1. NEGATE TS
2. DRIVE ATM TO INDICATE APPROPRIATE ACCESS TYPE
3. DRIVE DATA ON APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
1. DECODE ADDRESS AND SELECT THE APPROPRIATE SLAVE DEVICE.*
2. CAPTURE DATA FROM THE APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
3. ASSERT TA FOR ONE CLK CYCLE
1. RECOGNIZE THE 2ND TRANSFER IS DONE
2. INCREMENT APPROPRIATE ADDRESS BITS BASED ON SIZ[1:0], A[3:0] AND PORT SIZE
3. DRIVE ATM TO INDICATE APPROPRIATE ACCESS TYPE
4. ASSERT TS FOR ONE CLK CYCLE
1. NEGATE TS
2. DRIVE ATM TO INDICATE APPROPRIATE ACCESS TYPE
3. DRIVE DATA ON APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
1. DECODE ADDRESS AND SELECT THE APPROPRIATE SLAVE DEVICE.*
2. CAPTURE DATA FROM THE APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
3. ASSERT TA FOR ONE CLK CYCLE
1. RECOGNIZE THE 3RD TRANSFER IS DONE
2. INCREMENT APPROPRIATE ADDRESS BITS BASED ON SIZ[1:0], A[3:0] AND PORT SIZE
3. DRIVE ATM TO INDICATE APPROPRIATE ACCESS TYPE
4. ASSERT TS FOR ONE CLK CYCLE
1. NEGATE TS
2. DRIVE ATM TO INDICATE APPROPRIATE ACCESS TYPE
3. DRIVE DATA ON APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
1. DECODE ADDRESS AND SELECT THE APPROPRIATE SLAVE DEVICE.*
2. CAPTURE DATA FROM THE APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
3. ASSERT TA FOR ONE CLK CYCLE
1. RECOGNIZE THE 4TH TRANSFER IS DONE
2. THREE-STATE D[31:0]
*TO INSERT WAIT STATES, TA IS DRIVEN NEGATED .

Figure 6-14. Burst-Inhibited Byte-, Word-, and Longword-Write Transfer Flowchart

Figure 6-15 shows a burst-inhibited supervisor data longword-write transfer to a 16-bit port.

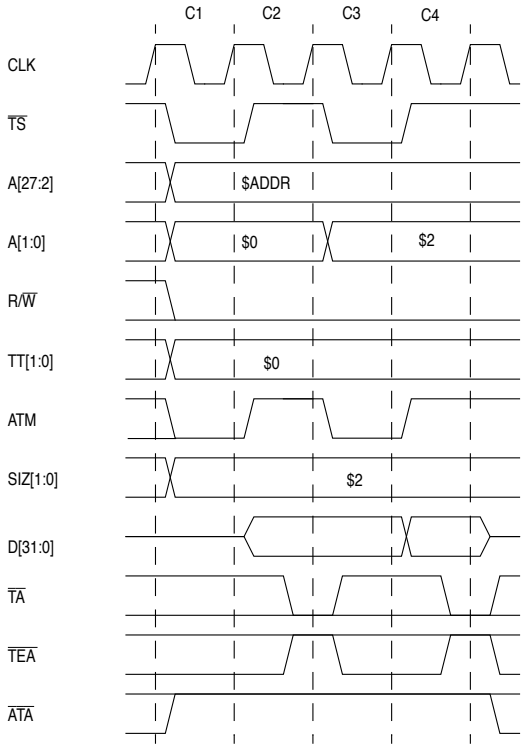


Figure 6-15. Burst-Inhibited Longword-Write Transfer to a 16-Bit Port (No Wait States)

Clock 1 (C1)

The write cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type and ATM identifies the transfer as data. The read/write (R/$\overline{W}$) signal is driven low for a write cycle, and the size signals (SIZ[1:0]) are driven to $2 to indicate a word transfer. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$, drives ATM high to identify the transfer as supervisor and places the data on the data bus (D[31:0]). The selected device(s) asserts $\overline{TA}$ if it is ready to latch the data. At the end of C2, the selected device latches the current

value of D[31:16], and the MCF5206 samples the level of $\overline{TA}$. If $\overline{TA}$ is asserted, the transfer of the first word is complete. If $\overline{TA}$ is negated, the MCF5206 continues to output the data and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

Clock 3 (C3)

The MCF5206 increments A[1:0] to address the next word, asserts $\overline{TS}$ and drives ATM low to identify the transfer as code or data.

Clock 4 (C4)

This clock is identical to C2, except that the data driven corresponds to the second word of data. This is the last CLK cycle of the longword-write transfer and the MCF5206 three-states D[31:0] at the start of the next CLK cycle.

6.5.6 Asynchronous-Acknowledge Read Transfer

The MCF5206 provides an asynchronous acknowledge that can be used for termination of all MCF5206 transfers except accesses to DRAM. $\overline{ATA}$ is synchronized internally before being used and must meet the specified setup and hold times to CLK only if recognition by a specific CLK rising edge is required. Because of the internal synchronization of $\overline{ATA}$, data must be driven on the bus until the asynchronous transfer acknowledge is recognized internally. If transfer error acknowledge ($\overline{TEA}$) is asserted while ATA is being synchronized internally, the transfer will be terminated in an error.

NOTE

The internal synchronized version of ($\overline{ATA}$) will be referred to as ‘‘internal asynchronous transfer acknowledge.’’ Because of the time required to internally synchronize $\overline{ATA}$ during a read cycle, data is latched on the rising edge of CLK when the internal asynchronous transfer acknowledge is asserted. Consequently, data must remain valid for at least one CLK cycle after the assertion of $\overline{ATA}$. Similarly, during a write cycle, data is driven until the rising edge of CLK when the internal asynchronous transfer acknowledge is asserted.

Figure 6-16 is a flowchart for read transfers to 8-, 16-, or 32-bit ports with asynchronous termination. Bus operations are similar for each case and vary only with the size indicated,

the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer.

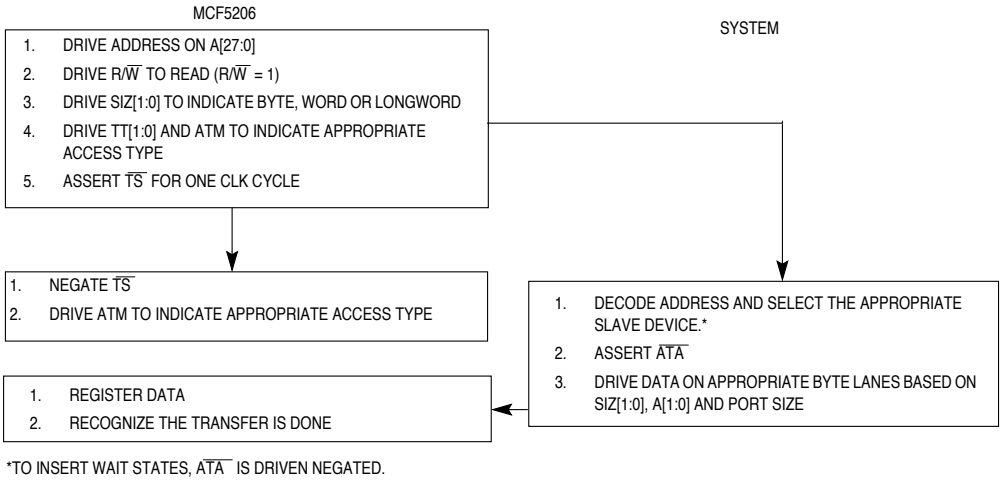


Figure 6-16. Byte-, Word-, and Longword-Read Transfer with Asynchronous Termination Flowchart (One Wait State)

NOTE

Zero-wait-state operation can be achieved with asynchronous termination by asserting asynchronous termination acknowledge ($\overline{ATA}$) during the CLK cycle transfer start ($\overline{TS}$) is asserted. This may only be practical if $\overline{ATA}$ is tied to GND. Refer to **3.5.12 Termination Tied to GND** for more information.

Figure 6-17 shows a user code byte read from an 8-bit port.

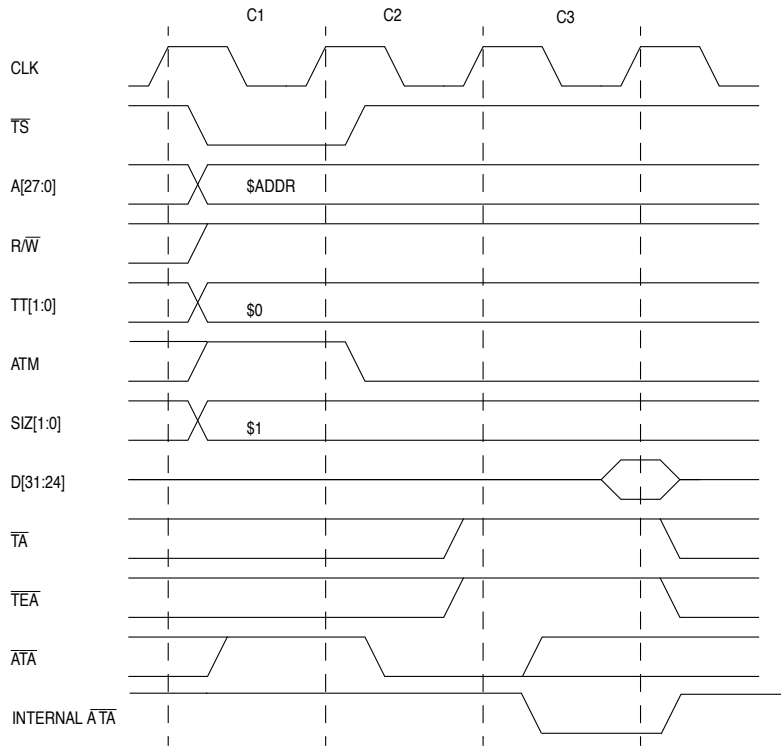


Figure 6-17. Byte-Read Transfer from an 8-Bit Port Using Asynchronous Termination (One Wait State)

Clock 1 (C1)

The read cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type and ATM identifies the transfer as code. The read/write (R/$\overline{W}$) signal is driven high for a read cycle, and the size signals (SIZ[1:0]) are driven to $1 to indicate a byte transfer. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$ and drives ATM low to identify the transfer as user. The selected device(s) asserts $\overline{ATA}$.

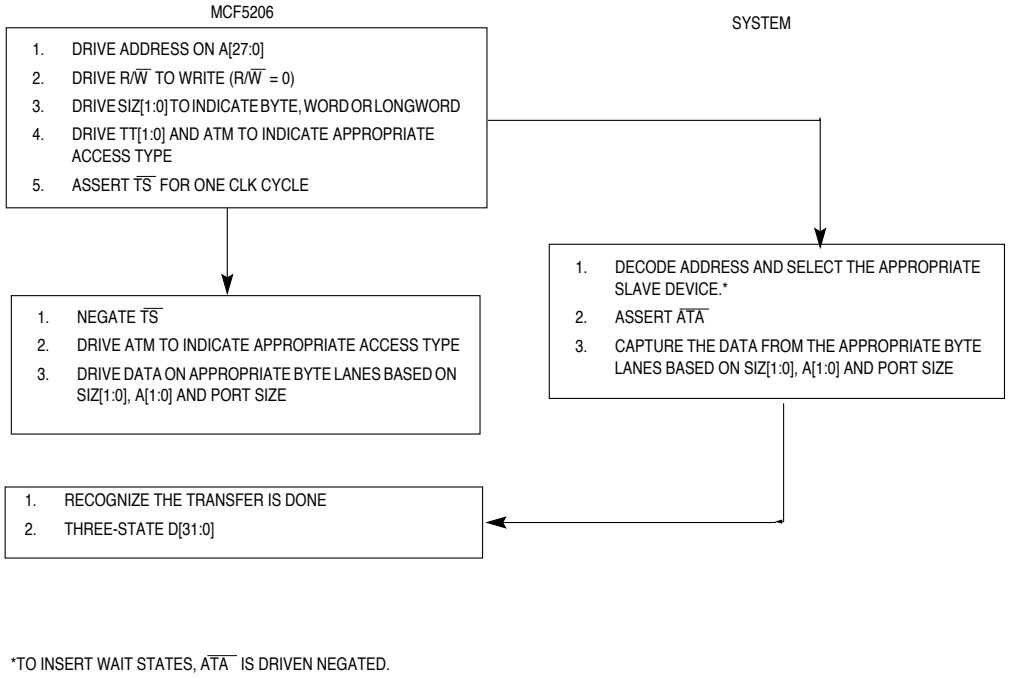
Clock 3 (C3)

At the end of C3, the MCF5206 samples the level of internal asynchronous transfer acknowledge and if it is asserted, latches the current value of D[31:24]. If internal

asynchronous transfer acknowledge is asserted, the byte transfer is complete and the transfer terminates. If internal asynchronous transfer acknowledge is negated, the MCF5206 continues to sample internal asynchronous transfer acknowledge and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample internal asynchronous transfer acknowledge until it is asserted. As long as $\overline{ATA}$ is asserted by the falling edge of C2, internal asynchronous transfer acknowledgethe will be asserted by the rising edge of C3.

6.5.7 Asynchronous Acknowledge Write Transfer

Figure 6-18 is a flowchart for write transfers to 8-, 16-, or 32-bit ports with asynchronous termination. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer.



*TO INSERT WAIT STATES, ATA IS DRIVEN NEGATED.

Figure 6-18. Byte-, Word-, and Longword-Write Transfer with Asynchronous Termination Flowchart

Figure 6-19 shows a user data byte transfer to a 32-bit port with asynchronous termination.

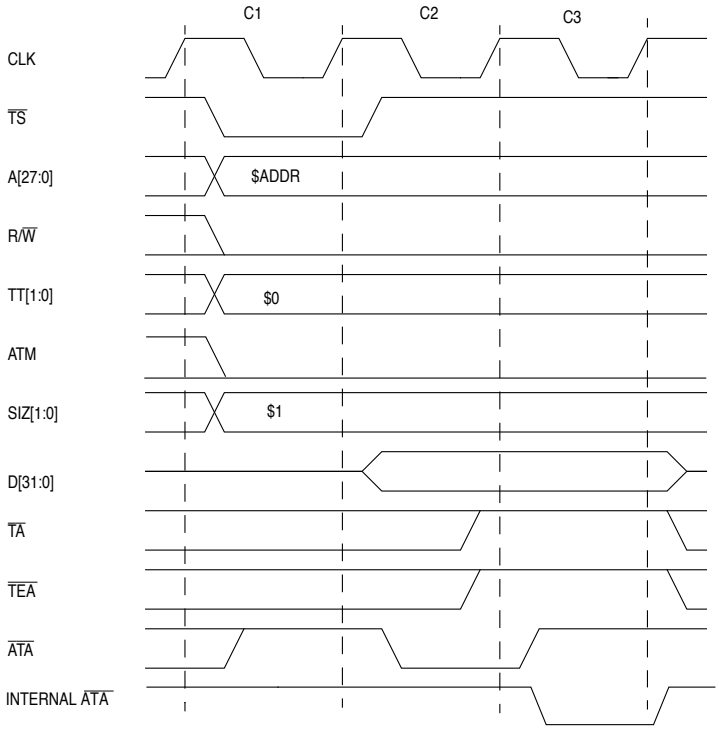


Figure 6-19. Byte-Write Transfer to a 32-Bit Port Using Asynchronous Termination (One Wait State)

Clock 1 (C1)

The write cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type and ATM identifies the transfer as data. The read/write (R/$\overline{W}$) signal is driven low for a write cycle, and the size signals (SIZ[1:0]) are driven to $1 to indicate a byte transfer. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$, drives ATM low to identify the transfer as user and places the data on the data bus (D[31:0]). The selected slave device asserts $\overline{ATA}$. The selected slave device may latch the data present on the data bus or may wait until the end of Clock 3 (after internal asynchronous transfer acknowledge has been asserted).

Clock 3 (C3)

At the end of C3, the MCF5206 samples the level of internal asynchronous transfer acknowledge and if it is asserted, the transfer of the byte is complete and the transfer terminates. If internal asynchronous transfer acknowledge is negated, the MCF5206 continues to sample internal asynchronous transfer acknowledge and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample internal asynchronous transfer acknowledge until it is asserted. As long as $\overline{ATA}$ is asserted by the falling edge of C2, internal asynchronous transfer acknowledge will be asserted by the rising edge of C3.

6.5.8 Bursting Read Transfers: Word, Longword, and Line with Asynchronous Acknowledge

If the burst-enable bit in the appropriate Chip-Select Control Register (CSCR) or Default Memory Control Register (DMCR) is set to 1 and the operand size is larger than the port size of the memory being accessed, the MCF5206 performs word, longword, and line transfers in burst mode. When burst mode is selected and the transfer is not to DRAM, the transfer can be terminated synchronously using $\overline{TA}$, or asynchronously using $\overline{ATA}$. The transfer attributes are the same for both the synchronous and asynchronously terminated burst transfers.

Figure 6-20 is a flowchart for bursting read transfers to 8-, 16-, or 32-bit ports using asynchronous termination. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus needed for the transfer, and the specific number

of cycles used for each transfer. A bursted transfer can be from two to 16 transfers long. The flow chart shown is for four bursting transfers.

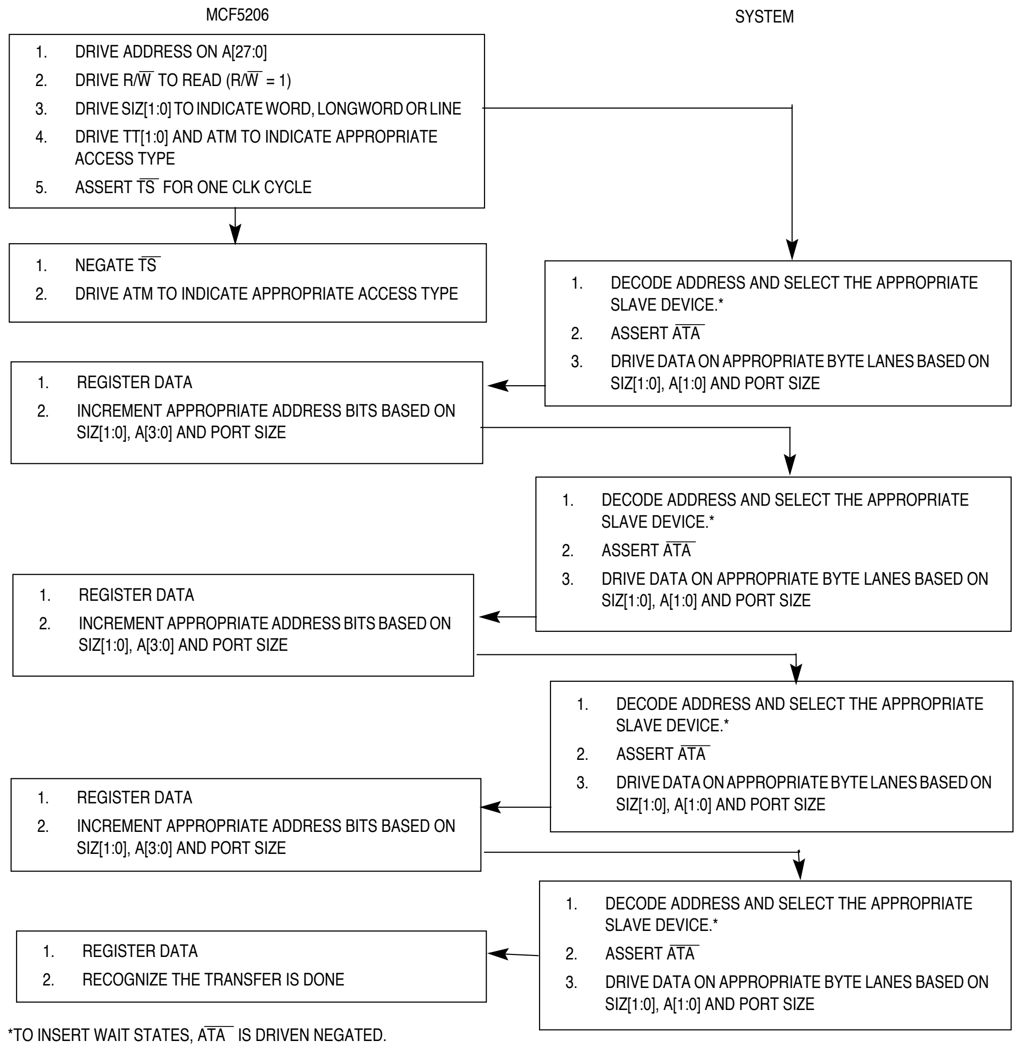


Figure 6-20. Bursting Word-, Longword-, and Line-Read Transfer with Asynchronous Termination Flowchart

Figure 6-21 shows a bursting supervisor data longword-read transfer from a 16-bit port.

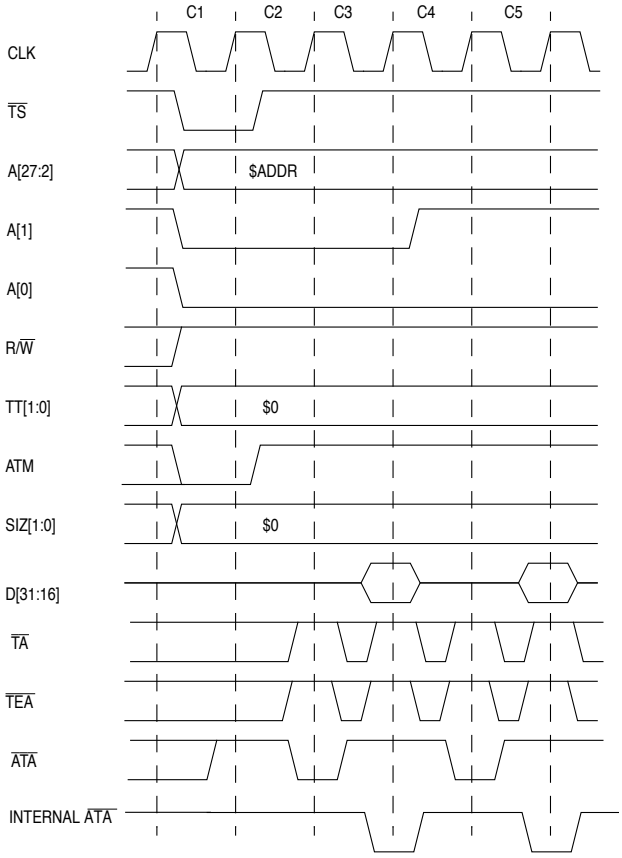


Figure 6-21. Bursting Longword-Read from 16-Bit Port Using Asynchronous Termination (One Wait State)

Clock 1 (C1)

The read cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type and ATM identifies the transfer as reading data. The read/write (R/$\overline{W}$) signal is driven high for a read cycle, and the size signals (SIZ[1:0]) are driven to $0 to indicate a longword transfer. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$, drives ATM high to identify the transfer as supervisor. The selected device(s) asserts $\overline{ATA}$.

Clock 3 (C3)

At the end of C3, the MCF5206 samples the level of internal asynchronous transfer acknowledge and if it is asserted, latches the current value of D[31:16]. If internal asynchronous transfer acknowledge is asserted, the transfer of the first word is complete. If internal asynchronous transfer acknowledge is negated, the MCF5206 continues to sample internal asynchronous transfer acknowledge and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample internal asynchronous transfer acknowledge until it is asserted. As long as $\overline{ATA}$ is asserted by the falling edge of C2, internal asynchronous transfer acknowledge will be asserted by the rising edge of C3.

Clock 4 (C4)

The MCF5206 increments A[1:0] to address the next word. The selected device(s) asserts $\overline{ATA}$.

Clock 5 (C5)

At the end of C5, the MCF5206 samples the level of internal asynchronous transfer acknowledge and if it is asserted, latches the current value of D[31:16]. If internal asynchronous transfer acknowledge is asserted, the transfer of the second word is complete and the transfer is terminated. If internal asynchronous transfer acknowledge is negated, the MCF5206 continues to sample internal asynchronous transfer acknowledge and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample internal asynchronous transfer acknowledge until it is asserted. As long as $\overline{ATA}$ is asserted by the falling edge of C4, internal asynchronous transfer acknowledge will be asserted by the rising edge of C5.

6.5.9 Bursting Write Transfers: Word, Longword, and Line with Asynchronous Acknowledge

Figure 6-22 is a flowchart for bursting write transfers (four transfers long) to 8-, 16-, or 32-bit ports using asynchronous termination. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer. A bursted transfer can be from two to 16 transfers long.

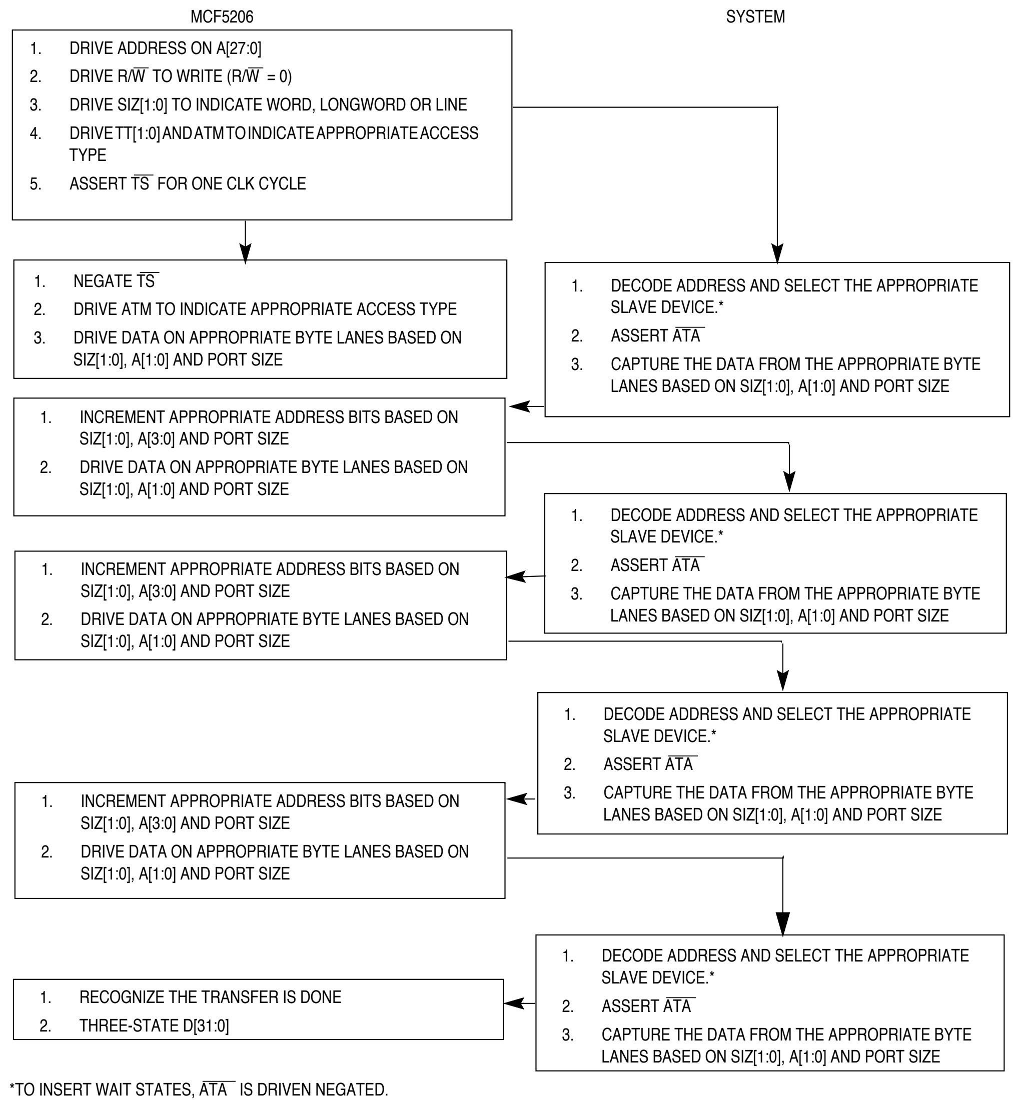


Figure 6-22. Word-, Longword-, and Line-Write Transfer Flowchart with Asynchronous Termination

Figure 6-23 shows a bursting user data line-write transfer to a 32-bit port using asynchronous termination.

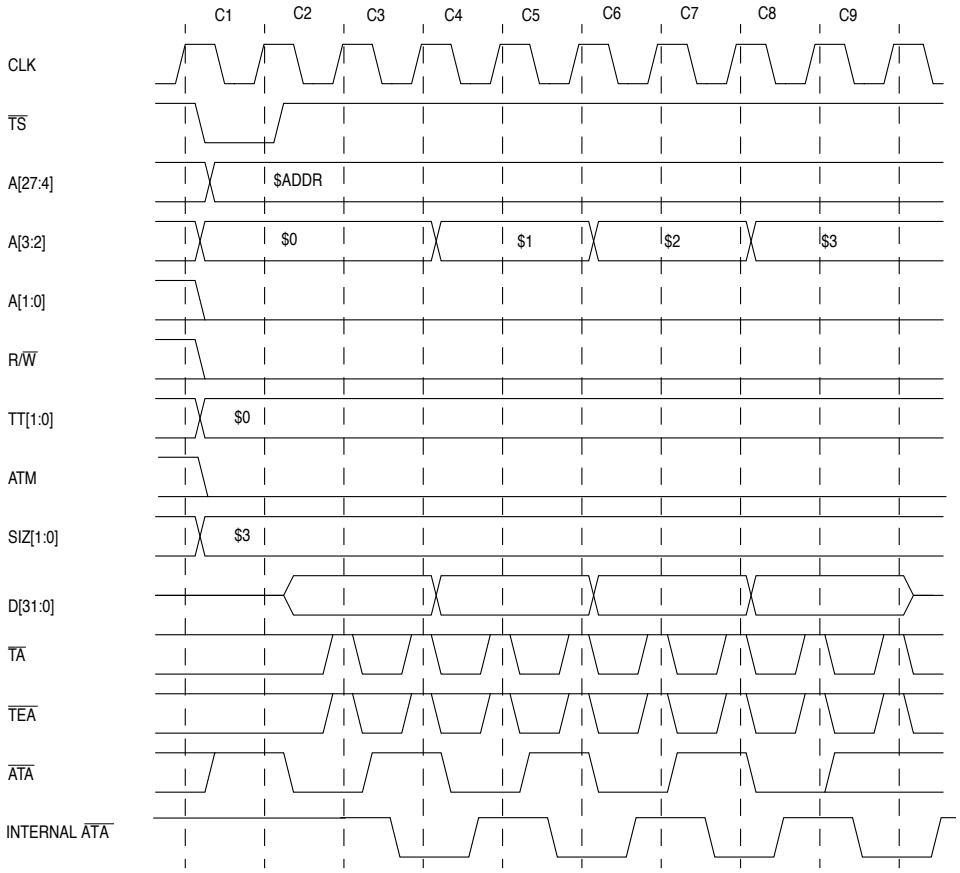


Figure 6-23. Bursting Line-Write from 32-Bit Port Using Asynchronous Termination (One Wait State)

Clock 1 (C1)

The write cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type and ATM identifies the transfer as data. The read/write (R/$\overline{W}$) signal is driven low for a write cycle, and the size signals (SIZ[1:0]) are driven to $3 to indicate a line transfer. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$, drives ATM low to identify the transfer as user and places the data on the data bus (D[31:0]). The selected device(s) asserts $\overline{ATA}$ if it is ready to latch the data.

Clock 3 (C3)

If the selected device asserted asynchronous transfer acknowledge during C2, the selected device must latch the data by the end of C3. At the end of C3, the MCF5206 samples the level of internal asynchronous transfer acknowledge. If internal asynchronous transfer acknowledge is asserted, the transfer of the first longword is complete. If internal asynchronous transfer acknowledge is negated, the MCF5206 continues to sample internal asynchronous transfer acknowledge and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample internal asynchronous transfer acknowledge until it is asserted. As long as $\overline{ATA}$ is asserted by the falling edge of C2, the internal asynchronous transfer acknowledgewill be asserted by the rising edge of C3.

Clock 4 (C4)

The MCF5206 increments A[3:2] to address the next longword of the line transfer and drives D[31:0] with the second longword of data. The selected device(s) asserts $\overline{ATA}$ if it is ready to latch the data. At the end of C4, the MCF5206 samples the level of internal $\overline{ATA}$ and if it is asserted, the second longword transfer of the line write is complete. If internal $\overline{ATA}$ is negated, the MCF5206 continues to sample internal $\overline{ATA}$ and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample internal $\overline{ATA}$ on successive rising edge of CLK until it is asserted.

Clock 5 (C5)

This clock is identical to C3, except that the data value corresponds to the second longword of data for the burst.

Clock 6 (C6)

This clock is identical to C4, except that once internal $\overline{ATA}$ is asserted, the address and the data values correspond to the third longword of data for the burst.

Clock 7 (C7)

This clock is identical to C3, except that the data value corresponds to the third longword of data for the burst.

Clock 8 (C8)

This clock is identical to C4, except that once internal $\overline{ATA}$ is asserted the address and data value correspond to the fourth longword of data for the burst.

Clock 9 (C9)

This clock is identical to C3, except that the data value corresponds to the fourth longword of data for the line. This is the last CLK cycle of the line write transfer and the MCF5206 three-states D[31:0] at the start of the next CLK cycle.

6.5.10 Burst-Inhibited Read Transfers: Word, Longword, and Line with Asynchronous Acknowledge

If the burst-enable bit is cleared in the appropriate Chip-Select Control Register (CSCR) or Default Memory Control Register (DMCR) and the operand size is larger than the port size of the memory being accessed, the MCF5206 performs word, longword, and line transfers in burst-inhibited mode. When burst-inhibit mode is selected, the size of the transfer (indicated by SIZ[1:0]) will reflect the port size if the operand being read is larger than the port size, or the operand size if the port size is larger than the operand size. A transfer size of line (SIZ[1:0] = $3) will never be indicated in burst-inhibited mode. If the operand size is line, the size pins (SIZ[1:0]) will always indicate the port size.

The basic transfer of a burst-inhibited read using asynchronous termination is the same as “normal” read using asynchronous termination with the addition of more transfers, until the entire operand has been accessed. Figure 6-24 is a flowchart for burst-inhibited read transfers to 8-, 16-, or 32-bit ports with asynchronous termination. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer. The flowchart is specifically for a burst-inhibited transfer of four transfers long.

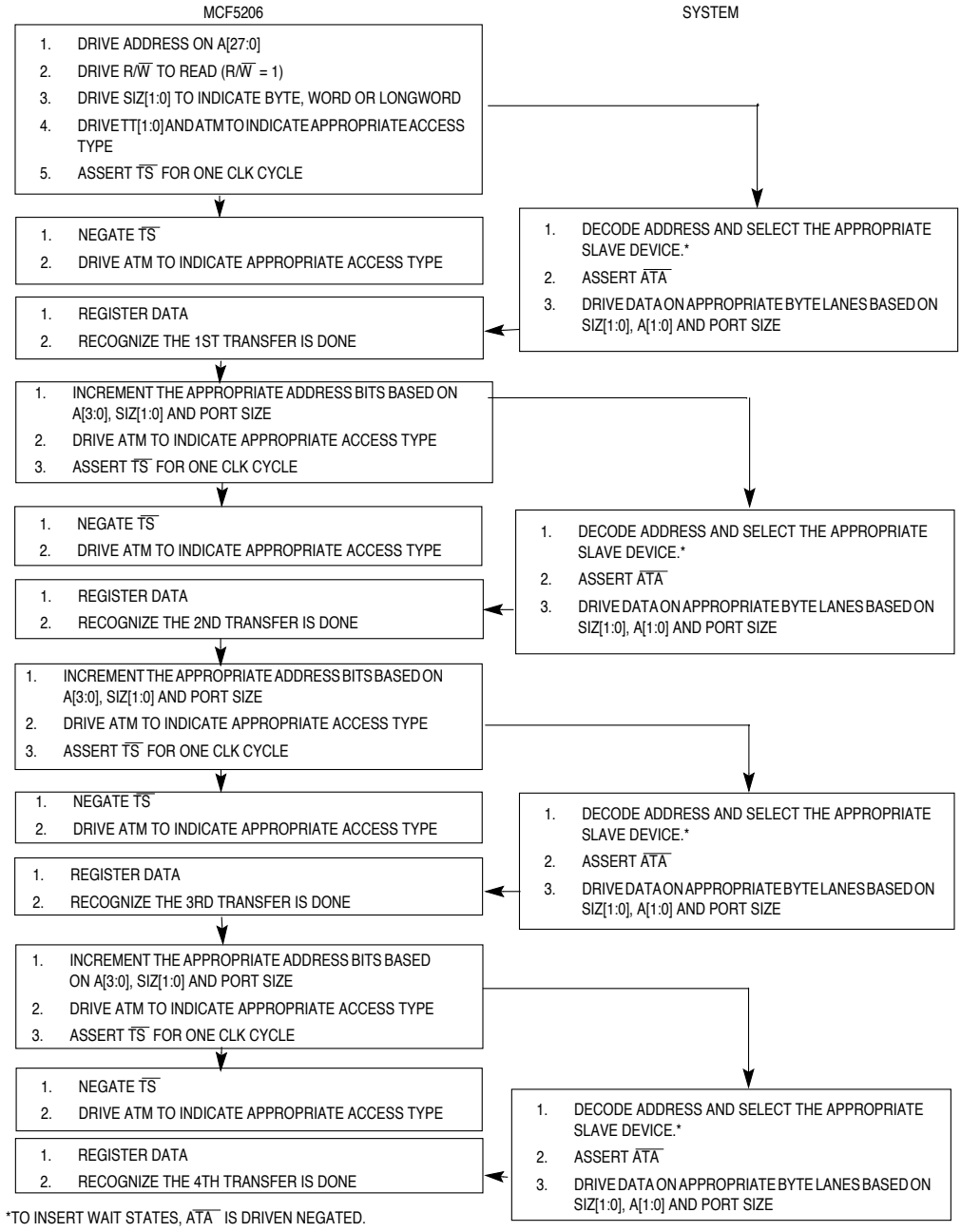


Figure 6-24. Burst-Inhibited Word-, Longword-, and Line-Read Transfer with Asynchronous Termination Flowchart

Figure 6-25 shows a burst-inhibited user code word-read transfer from an 8-bit port.

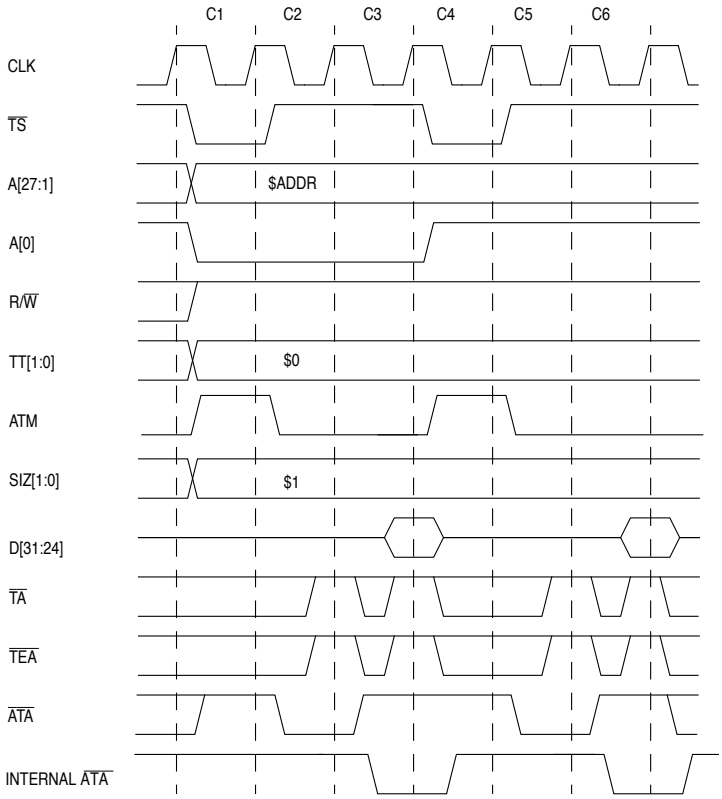


Figure 6-25. Burst-Inhibited Word Read from 8-Bit Port Using Asynchronous Termination

Clock 1 (C1)

The read cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type and ATM identifies the transfer as code. The read/write (R/$\overline{W}$) signal is driven high for a read cycle and the size signals (SIZ[1:0]) are driven to $1 to indicate a byte transfer. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$, drives ATM low to identify the transfer as user. The selected device(s) asserts $\overline{ATA}$.

Clock 3 (C3)

At the end of C3, the MCF5206 samples the level of internal asynchronous transfer acknowledge and if it is asserted, latches the current value of D[31:24]. If internal asynchronous transfer acknowledge is asserted, the transfer of the first byte is complete. If internal asynchronous transfer acknowledge is negated, the MCF5206 continues to sample internal asynchronous transfer acknowledge and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample internal asynchronous transfer acknowledge until it is asserted. As long as $\overline{ATA}$ is asserted by the falling edge of C2, internal asynchronous transfer acknowledge will be asserted by the rising edge of C3.

Clock 4 (C4)

This clock is identical to C1, except the address bus is incremented to point to the second byte of data.

Clock 5 (C5)

This clock is identical to C2.

Clock 6 (C6)

This clock is identical to C3, except once internal $\overline{ATA}$ is recognized, the data corresponds to the second byte of data.

6.5.11 Burst-Inhibited Write Transfers: Word, Longword, and Line with Asynchronous Acknowledge

The basic transfer of a burst-inhibited write using asynchronous termination is the same as “normal” write transfers with asynchronous termination but with the addition of more transfers until the entire operand has been accessed. Figure 6-26 is a flowchart for burst-inhibited write transfers to 8-, 16-, or 32-bit ports using asynchronous termination. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer. The flowchart specifically depicts a burst-inhibited transfer of four accesses long.

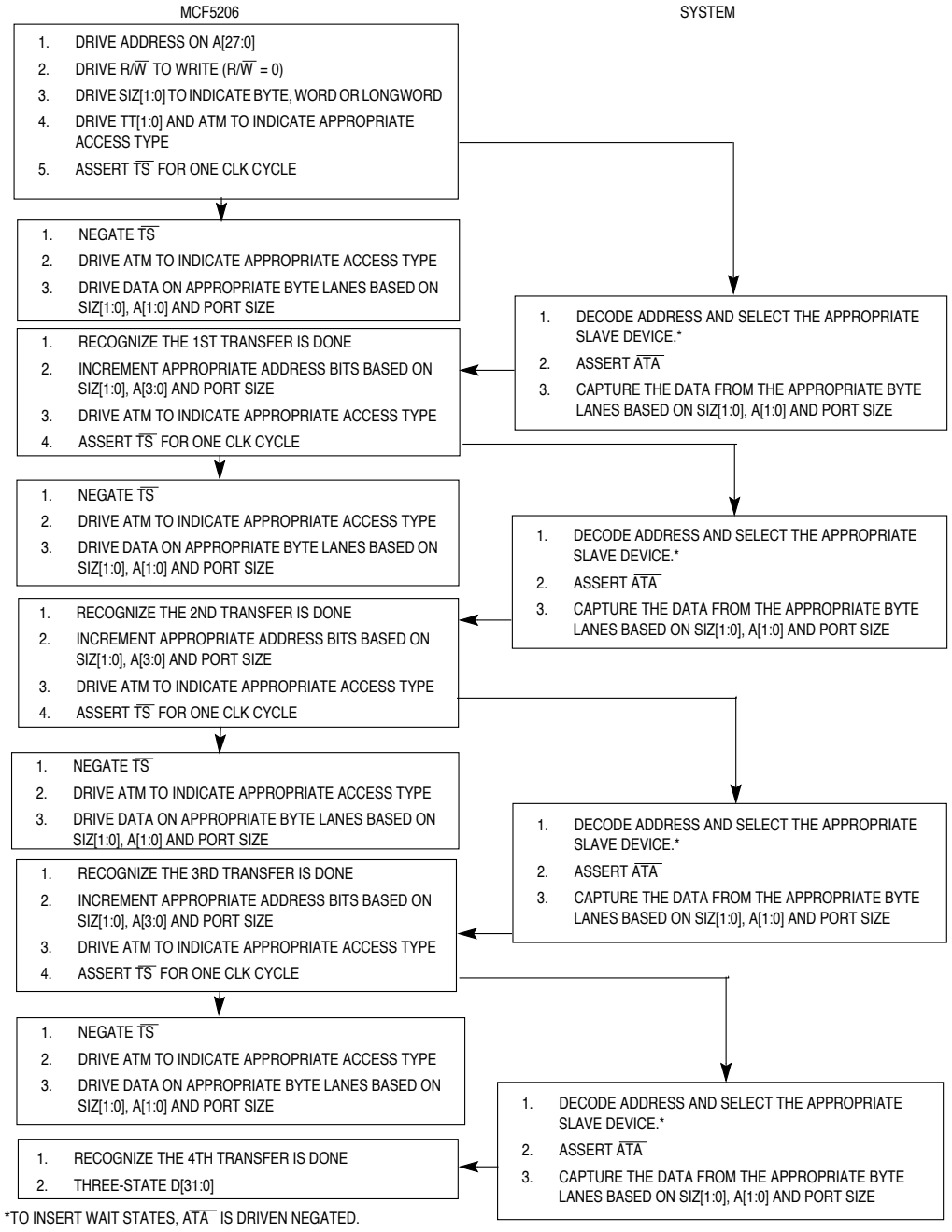


Figure 6-26. Burst-Inhibited Word-, Longword-, and Line-Write Transfer with

Asynchronous Termination Flowchart

Figure 6-27 shows a burst-inhibited supervisor data longword-write transfer to a 16-bit port.

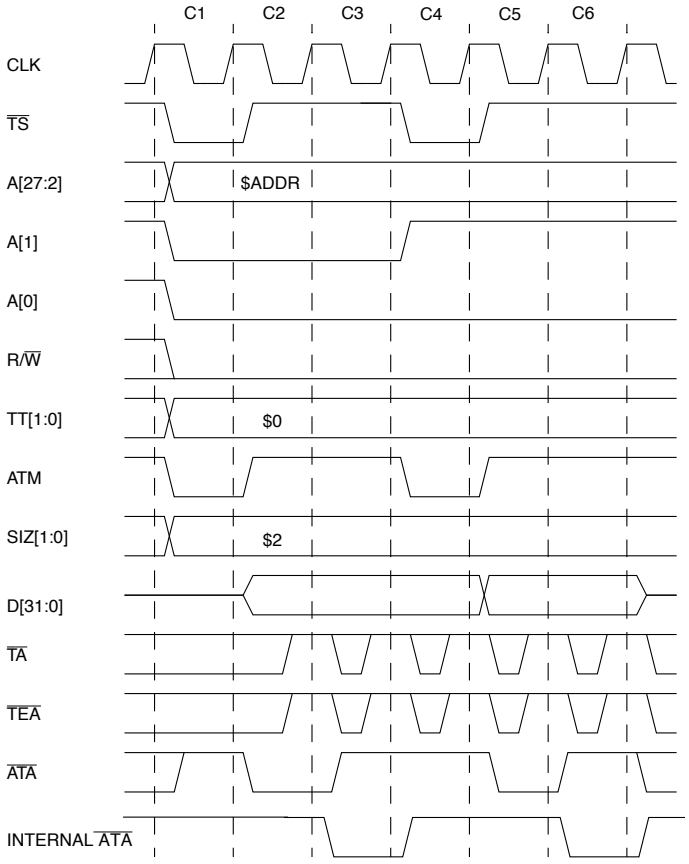


Figure 6-27. Burst-Inhibited Longword-Write Transfer to 16-Bit Port Using Asynchronous Termination (One Wait State)

Clock 1 (C1)

The write cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type and ATM identifies the transfer as data. The read/write (R/$\overline{W}$) signal is driven low for a write cycle, and the size signals (SIZ[1:0]) are driven to $2 to indicate a word transfer. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates $\overline{TS}$, drives ATM high to identify the transfer as supervisor and drives the data on the data bus (D[31:0]). The selected device(s) asserts $\overline{ATA}$ if it is ready to latch the data.

Clock 3 (C3)

At the end of C3, the MCF5206 samples the level of internal asynchronous transfer acknowledge and if it is asserted, terminates the first word transfer. If internal asynchronous transfer acknowledge is asserted, the transfer of the first word is complete. If internal asynchronous transfer acknowledge is negated, the MCF5206 continues to sample internal asynchronous transfer acknowledge and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample internal asynchronous transfer acknowledge until it is asserted. As long as $\overline{ATA}$ is asserted by the falling edge of C2, the rising edge of C3 will assert the internal asynchronous transfer acknowledge.

Clock 4 (C4)

This clock is identical to C1, except the MCF5206 increments the address to indicate the next word.

Clock 5 (C5)

This clock is identical to C2, except that the data driven corresponds to the second word of data.

Clock 6 (C6)

This clock is identical to C3, except after asynchronous transfer acknowledge is recognized, the MCF5206 three-states the data bus after the next rising edge of CLK.

6.5.12 Termination Tied to GND

If the MCF5206 is in a system with multiple masters and you require zero wait-state operation, you can tie $\overline{ATA}$ to GND to achieve zero wait-state operation for nonDRAM transfers. $\overline{ATA}$ must be used in this case as the MCF5206 can drive $\overline{TA}$ during alternate master accesses. When $\overline{ATA}$ is tied to GND, all nonDRAM transfers follow the timing shown with $\overline{TA}$ asserted with zero wait states.

If the MCF5206 is the only master in the system, $\overline{TA}$ and $\overline{BG}$ can be tied to GND to grant mastership of the external bus to the MCF5206 and achieve zero wait-state operation.

NOTE

$\overline{TA}$ cannot be tied to GND if the MCF5206 is not the only bus master in the system. Damage to the part could occur if $\overline{TA}$ is tied to GND and alternate master accesses using 5206 generated termination.

6.6 MISALIGNED OPERANDS

All MCF5206 data formats can be located in memory on any byte boundary. 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 that is not evenly divisible by four. However, because operands can reside at any byte boundary, they can be misaligned. Although the MCF5206 does not enforce any alignment restrictions for data operands (including program counter (PC) relative data addressing), some performance degradation occurs when additional bus cycles are required for longword or word operands that are misaligned. For maximum performance, data items should be aligned on their natural boundaries. All instruction words and extension words must reside on word boundaries. An address error exception will occur with any attempt to prefetch an instruction word at an odd address.

The MCF5206 converts misaligned operand accesses that are noncacheable to a sequence of aligned accesses. Figure 6-28 illustrates the transfer of a longword operand from a byte address to a 32-bit port, requiring more than one bus cycle. In this example, the SIZ[1:0] signals specify a byte transfer, and the byte offset of $1. The slave device supplies the byte and acknowledges the data transfer. When the MCF5206 starts the second cycle, the SIZ[1:0] signals specify a word transfer with a byte offset of $2. The next two bytes are transferred during this cycle. The MCF5206 then initiates the third cycle, with the SIZ[1:0] signals indicating a byte transfer. The byte offset is now $0; the port supplies the final byte and the operation is complete. Figure 6-29 is similar to the example illustrated in Figure 6-28 except that the operand is word-sized and the transfer requires only two bus cycles.

	31 – 24	23 – 16	15 – 8	7 – 0
TRANSFER 1	-	OP 3	-	-
TRANSFER 2	-	-	OP 2	OP 1
TRANSFER 3	OP 0	-	-	-

Figure 6-28. Example of a Misaligned Longword Transfer

	31 – 24	23 – 16	15 – 8	7 – 0
TRANSFER 1	-	-	-	OP 1
TRANSFER 2	OP 0	-	-	-

Figure 6-29. Example of a Misaligned Word Transfer

NOTE

Alternate masters that are using internal MCF5206 chip-select, DRAM, and default memory control signals must initiate aligned transfers only.

6.7 ACKNOWLEDGE CYCLES

When a peripheral device requires the services of the MCF5206 or is ready to send information that the ColdFire core requires, it can signal the ColdFire core to take an interrupt exception. The interrupt exception transfers control to a routine that responds appropriately. The peripheral device uses the interrupt priority-level/interrupt-request signals ($\overline{IPLx}/\overline{IRQx}$) to signal an interrupt condition to the MCF5206.

The MCF5206 has two levels of interrupt masking. The first level of interrupt masking is in the interrupt controller in the System Integration Module (SIM) which masks individual interrupt inputs and then outputs the interrupt priority level of the highest pending unmasked interrupt to the ColdFire core. The Status Register (SR) provides the second level of interrupt masking in the ColdFire core which contains an interrupt priority mask. The value of the SR interrupt mask is the highest priority level that the ColdFire core ignores. When an interrupt request has a priority higher than the value in the mask, the ColdFire core makes the request a pending interrupt. For more information about the Status Register refer to **Section 3.2.2.1 Status Register** in the ColdFire Core Section.

The MCF5206 continuously samples the external interrupt input signals and synchronizes and debounces these signals. An interrupt request must be held constant for at least two consecutive CLK periods to be considered a valid input. If the external interrupt inputs are programmed to individual interrupt requests (at levels 1, 4, and 7), the interrupt request must maintain the interrupt request level until the MCF5206 acknowledges the interrupt to guarantee that the interrupt is recognized. If the external interrupt inputs are programmed to be interrupt priority levels, the interrupt request must maintain the interrupt request level or a higher priority level until the MCF5206 acknowledges the interrupt to guarantee that the interrupt is recognized.

NOTE

All interrupts are level sensitive only. Interrupts must remain stable and held valid for the interrupt to be detected.

The MCF5206 takes an interrupt exception for a pending interrupt within one instruction boundary after processing any other pending exception with a higher priority. Thus, the MCF5206 executes at least one instruction in an interrupt exception handler before recognizing another interrupt request.

If the AVEC bit in the Interrupt Control Register (ICR) for the interrupt being acknowledged is set to 1 (enabling autovectoring), the interrupt acknowledge vector will be generated internally and no interrupt acknowledge cycle will be generated on the external bus. Refer

to the SIM section **Section 7.3.2.3 Interrupt Control Register (ICR)** for ICR programming.

NOTE

If autovector generation is used for external interrupts, no interrupt acknowledge cycle will be generated on the external bus. Consequently, you must clear the external interrupt in the interrupt service routine.

6.7.1 Interrupt Acknowledge Cycle

When the MCF5206 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. It differs from a normal read cycle in the following respects:

- TT[1:0] = $3 to indicate a CPU space/acknowledge bus cycle
- ATM = $1 when $\overline{TS}$ is asserted and ATM = $0 when $\overline{TS}$ is negated
- Address signals A[27:5] are set to all ones ($7FFFFF)
- Address signals A[4:2] are set to the interrupt request level being acknowledged
- Address signals A[1:0] are set to all zeros ($0)

The responding device places the vector number on D[31:24] of the data bus during the interrupt acknowledge bus cycle and the cycle is terminated normally with $\overline{TA}$ or $\overline{ATA}$.

Figure 6-30 and Figure 6-31 illustrate a flowchart and functional timing diagram for an interrupt-acknowledge cycle terminated with $\overline{TA}$.

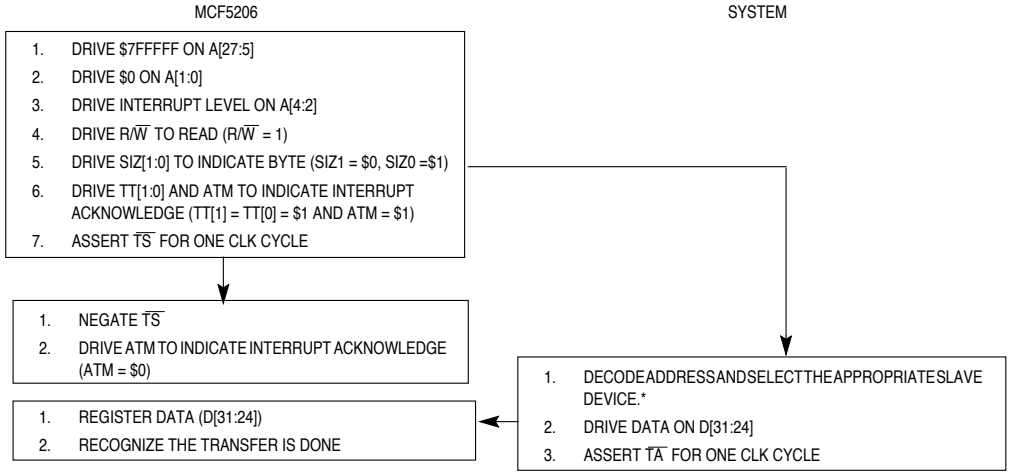


Figure 6-30. Interrupt-Acknowledge Cycle Flowchart

Figure 6-31 shows an interrupt acknowledge cycle.

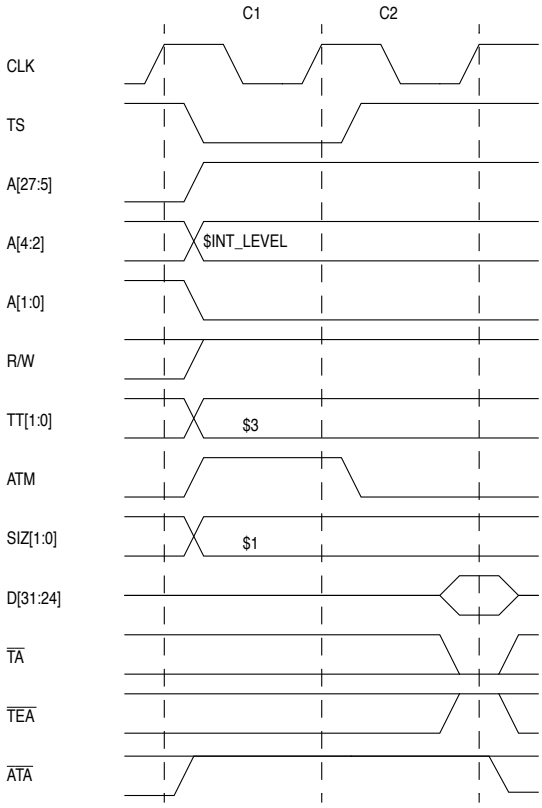


Figure 6-31. Interrupt Acknowledge Bus Cycle Timing (No Wait States)

Clock 1 (C1)

The interrupt acknowledge cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The address bus is driven with $7FFFFF on A[27:5], $0 on A[1:0] and the interrupt level being acknowledged on A[4:2]. The transfer type (TT[1:0]) signals are driven to $3 and the ATM is driven high to identify the access as an interrupt acknowledge cycle. The read/write (R/$\overline{W}$) signal is driven high for a read cycle, and the size signals (SIZ[1:0]) are driven to $1 to indicate a byte transfer. The MCF5206 asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates transfer start ($\overline{TS}$), drives access type and mode (ATM) low to identify the transfer as an interrupt acknowledge cycle. The selected device(s)

places the interrupt vector number onto D[31:24] and asserts $\overline{TA}$. At the end of C2, the MCF5206 samples the level of $\overline{TA}$ and if $\overline{TA}$ is asserted, latches the current value of D[31:24] which contains the interrupt vector number. If $\overline{TA}$ is asserted, the transfer of the interrupt vector is complete and the transfer terminates. If $\overline{TA}$ is negated, the MCF5206 continues to sample $\overline{TA}$ and inserts wait states instead of terminating the transfer. The MCF5206 continues to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

NOTE

Interrupt acknowledge cycles can be asynchronously acknowledged using $\overline{ATA}$. As long as $\overline{ATA}$ is asserted by the falling edge of C2, internal asynchronous transfer acknowledge will be asserted by the rising edge of C3. The interrupt vector must remain driven on D[31:24] until internal asynchronous transfer acknowledge is asserted.

6.8 BUS ERRORS

The system hardware can use the transfer error acknowledge ($\overline{TEA}$) signal to abort the current bus cycle when a fault is detected. A bus error is recognized during a bus cycle when $\overline{TEA}$ is asserted.

When the MCF5206 recognizes a bus error condition for an access, the access is terminated immediately. An access that requires more than one transfer, aborts without completing the remaining transfers if $\overline{TEA}$ is asserted, regardless of whether the access uses burst or burst-inhibited transfers.

Figure 6-32 shows a bursting supervisor code longword-read access from a 16-bit port with a transfer error.

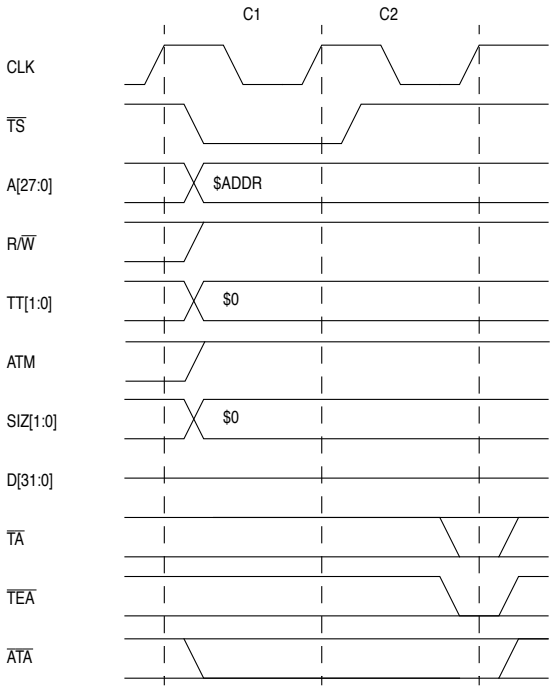


Figure 6-32. Bursting Longword-Read Access from 16-Bit Port Terminated with TEA Timing

Clock 1 (C1)

The read cycle starts in C1. During C1, the MCF5206 places valid values on the address bus (A[27:0]) and transfer control signals. The transfer type (TT[1:0]) signals identify the specific access type and ATM identifies the transfer as code. The read/write (R/W) signal is driven high for a read cycle, and the size signals (SIZ[1:0]) are driven to $0 to indicate a longword transfer. The MCF5206 asserts transfer start (TS) to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206 negates TS and drives ATM high to identify the transfer as supervisor. The selected device detects an error and asserts TEA. At the end of C2, the MCF5206 samples the level of TEA. If it is asserted, the transfer of the longword is aborted and the transfer terminates.

NOTE

If $\overline{TA}$ is asserted when transfer error-acknowledge ($\overline{TEA}$) is asserted, the transfer will be terminated with a bus error.

NOTE

For the MCF5206 to accept the transfer as successful with an $\overline{ATA}$, $\overline{TEA}$ must be negated until the internal asynchronous transfer acknowledge is asserted or the transfer will be completed with a bus error.

6.9 BUS ARBITRATION

The MCF5206 bus protocol provides for one bus master at a time: either the MCF5206 or an external device. If more than one external bus master is connected to the bus, an external arbiter can prioritize requests and determine which device is granted access to the bus. Bus arbitration is the protocol by which the MCF5206 or an external device becomes the bus master. When the MCF5206 is the bus master, it uses the bus to read instructions and transfer data not contained in its internal cache or memory to and from external memory. When an alternate bus master owns the bus, the MCF5206 can monitor the alternate bus master's transfers and assert chip-select and DRAM control, and transfer termination signals. This capability is discussed in more detail in **Section 6.10 Alternate Bus Master Operation**.

The MCF5206 bus arbitration can be used in two modes. A two-wire mode is provided for systems where the MCF5206 and a single external bus master are the only two masters arbitrating for use of the external bus. This arbitration mode uses the bus grant ($\overline{BG}$) and bus driven ($\overline{BD}$) signals. The bus request ($\overline{BR}$) signal can be ignored by the external bus master.

The second mode is provided for systems where multiple external bus masters are arbitrating for use of the external bus. This arbitration mode requires an external bus arbiter and uses the bus grant ($\overline{BG}$), bus driven ($\overline{BD}$) and bus request ($\overline{BR}$) signals to control usage of the external bus.

In either arbitration mode, the bus arbitration unit in the MCF5206 operates synchronously and transitions between states on the rising edge of CLK.

For systems where the MCF5206 is the only possible bus master, the bus can be continuously granted to the MCF5206 by tying bus grant ($\overline{BG}$) to $\overline{GND}$. An arbiter is not required.

6.9.1 Two Master Bus Arbitration Protocol (Two-Wire Mode)

The two-wire mode of bus arbitration allows the MCF5206 to share the external bus with a single external bus master without requiring an external bus arbiter. Figure 6-33 is a block diagram showing the MCF5206 connecting to an external bus master using the two-

wire mode. In this mode, the active-low bus grant ($\overline{BG}$) input of the MCF5206 is connected to the active-high HOLDREQ output of the external bus master and the active-low bus-driven ($\overline{BD}$) output of the MCF5206 is connected to the active-high HOLDACK input of the external bus master. Because the external bus master controls the assertion/negation of HOLDREQ, it controls when the MCF5206 is granted the bus, making the MCF5206 the lower priority master. You can program the bus lock (BL) bit in the SIM Configuration Register (SIMR) to a 1, instructing the MCF5206 to retain control of the external bus, even when bus grant ($\overline{BG}$) is negated. This lets you control the priority of the MCF5206 with respect to the alternate master when in two-wire mode.

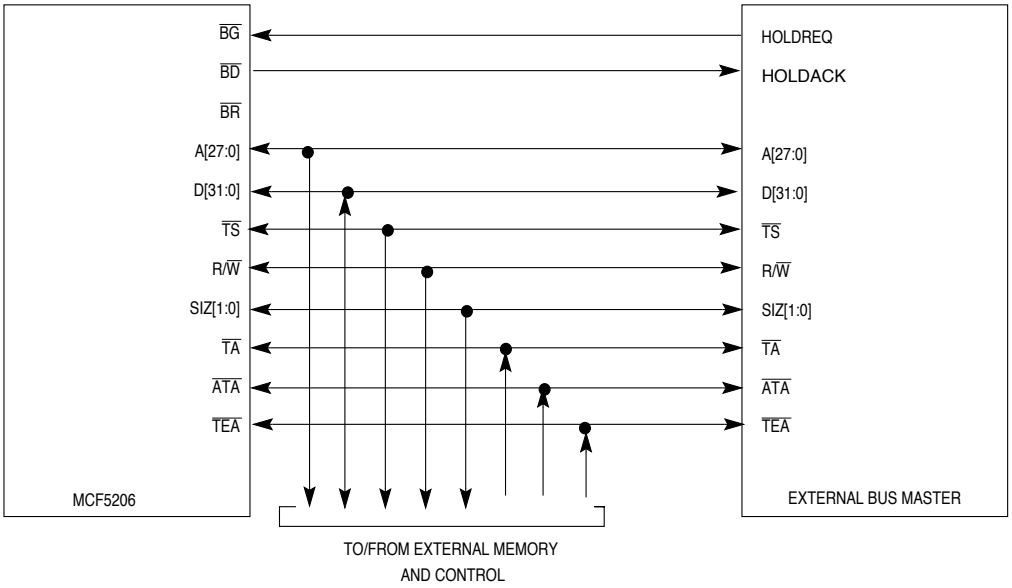


Figure 6-33. MCF5206 Two-Wire Mode Bus Arbitration Interface

When the alternate master is not using the bus, it negates HOLDREQ driving bus grant ($\overline{BG}$) low, granting the bus to the MCF5206. When the MCF5206 has an internal bus request pending and bus grant ($\overline{BG}$) is low, the MCF5206 will drive $\overline{BD}$ low, negating HOLDACK to the external bus master. When the external bus master requires use of the external bus, it asserts HOLDREQ, driving bus grant ($\overline{BG}$) high, requesting the MCF5206 to relinquish the bus. If $\overline{BG}$ is negated while a bus cycle is in progress and if the bus lock bit is cleared, the MCF5206 will relinquish the bus at the completion of the bus cycle. Note that the MCF5206 considers the individual transfers of a burst or burst-inhibited access to be a single bus cycle and does not relinquish the bus until the completion of the last transfer of the series.

When the bus has been granted to the MCF5206, one of two situations can occur. In the first case, the MCF5206 has an internal bus request pending, the MCF5206 asserts $\overline{BD}$ to indicate explicit bus ownership and begins the pending bus cycle by asserting $\overline{TS}$. The

MCF5206 continues to assert $\overline{BD}$ until the completion of the bus cycle. If $\overline{BG}$ is negated by the end of the bus cycle and the Bus Lock bit in the SIMR is 0, the MCF5206 will negate $\overline{BD}$. As long as $\overline{BG}$ is asserted, $\overline{BD}$ remains asserted to indicate the bus is owned by the MCF5206 and the MCF5206 continuously drives the address bus, attributes and control signals.

In the second situation, the bus is granted to the MCF5206, but the MCF5206 does not have an internal bus request pending and the Bus Lock bit in the SIMR is 0, so it takes implicit ownership of the bus. Implicit ownership of the bus occurs when the MCF5206 is granted the bus, but there are no pending bus cycles and the bus lock bit (BL) in the SIMR is set to 0. The MCF5206 does not drive the bus and does not assert bus driven $\overline{BD}$ if the bus is implicitly owned. If an internal bus request is generated or the bus lock bit in the SIM Configuration Register (SIMR) is set to 1, the MCF5206 assumes explicit ownership of the bus. If explicit ownership was assumed because of an internal request being generated, the MCF5206 immediately begins an access and simultaneously asserts bus driven $\overline{BD}$ and $\overline{TS}$. If explicit ownership was assumed because of the bus lock bit being set to 1, the MCF5206 asserts bus driven $\overline{BD}$ and drives the address, attributes and control signals but does no assert $\overline{TS}$ and does not begin a bus transfer.

In the case where the bus lock bit is set to 1, the MCF5206 will be the explicit master of the external bus, but will not begin an access until an internal request is generated.

Figure6-34 illustrates implicit and explicit bus ownership because of the bus lock bit being set then an internal bus request being generated.

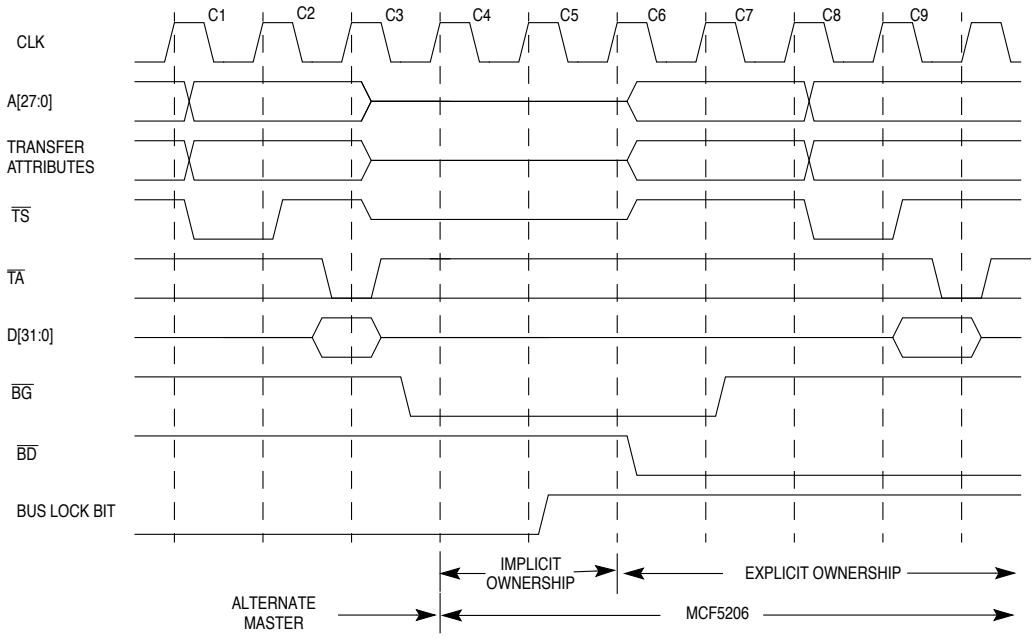


Figure 6-34. Two-Wire Implicit and Explicit Bus Ownership

In Figure 6-34, the alternate master has ownership of the external bus during Clock 1 (C1) and Clock 2 (C2). In Clock 3 (C3) the alternate master releases control of the bus by asserting bus grant ($\overline{BG}$) to the MCF5206. During Clock 4 (C4) and Clock 5 (C5) the MCF5206 is implicit owner because an internal access is not pending and the bus lock bit is cleared. In C5, the bus lock bit is set to 1, causing the MCF5206 to take explicit ownership of the bus in Clock 6 (C6) by asserting $\overline{BD}$. In Clock 7 (C7) the alternate master removes the bus grant to the MCF5206. Because the bus lock bit is set to 1, the MCF5206 does not relinquish the bus (the MCF5206 continues to assert $\overline{BD}$).

NOTE

The MCF5206 can start a transfer during the CLK cycle after $\overline{BG}$ is asserted. The alternate master should not assert $\overline{BG}$ to the MCF5206 until it has stopped driving the bus. $\overline{BG}$ cannot be asserted while the alternate master transfer is still in progress or damage to the part could occur.

When the bus has been removed from the MCF5206, one of two situations can occur. In the first case, the bus lock bit in the SIM Configuration Register (SIMR) is cleared and the

MCF5206 has implicit ownership of the bus. When the external bus master negates $\overline{BG}$, the MCF5206 will negate $\overline{BD}$ and three-state the address, data, $\overline{TS}$, R/$\overline{W}$, and SIZ signals after completing the current bus cycle. Figure 6-35 illustrates two-wire bus arbitration with the bus lock bit cleared.

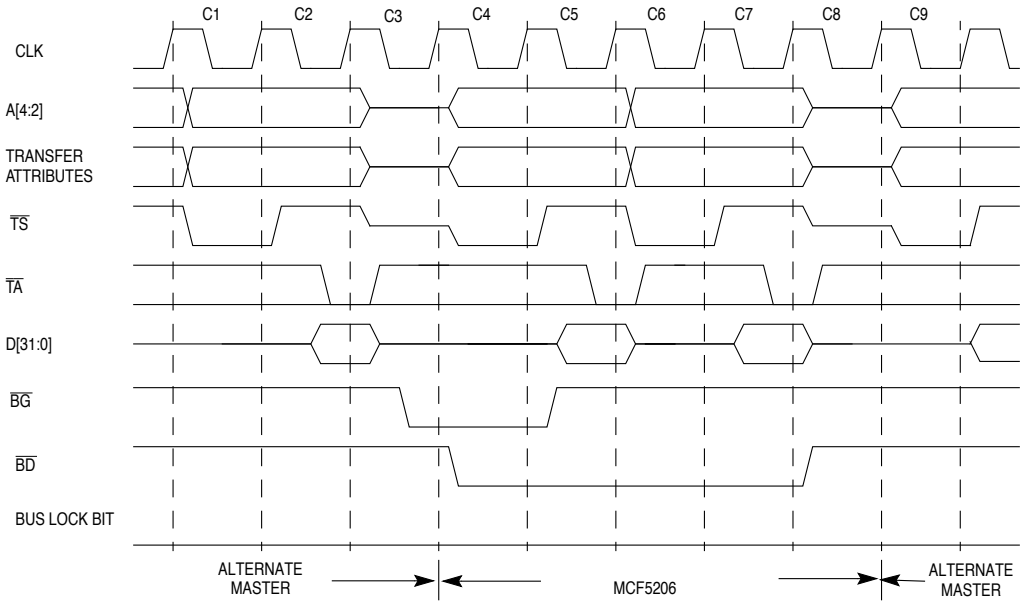


Figure 6-35. Two-Wire Bus Arbitration with Bus Lock Negated

In Figure 6-35 during clocks C1 and C2, the alternate master is the bus owner. During C3, the alternate master relinquishes control of the bus by asserting $\overline{BG}$ to the MCF5206. At this point, the bus lock bit is cleared, but because there is an internal access pending, the MCF5206 asserts $\overline{BD}$ during C4 and begins the access. Thus, the MCF5206 becomes the explicit master of the external bus. This access is a burst-inhibited access. During C5, the alternate master removes the grant from the MCF5206 by negating $\overline{BG}$. Because the MCF5206 is performing a burst-inhibited access, it continues to assert $\overline{BD}$ until the final transfer of the access has completed. The MCF5206 negates $\overline{BD}$ during C8, returning ownership of the external bus to the alternate master.

In the second case, the bus lock bit in the SIM Configuration Register (SIMR) is set to 1 and the MCF5206 has explicit ownership of the bus. In this case, when the external bus master negates $\overline{BG}$, the MCF5206 will continue to assert $\overline{BD}$ and will continue to drive address, attributes, and control signals. The MCF5206 will retain mastership of the bus until the bus lock bit in the SIM Configuration Register (SIMR) is cleared. By setting the bus lock bit to 1, you can select the MCF5206 to be the highest priority master, even when mastership of the bus is controlled by an alternate master. In this fashion, the MCF5206 can be guaranteed mastership of the bus when executing time critical, bus intensive

operations. Figure 6-36 illustrates bus arbitration using the bus lock bit to control the arbitration.

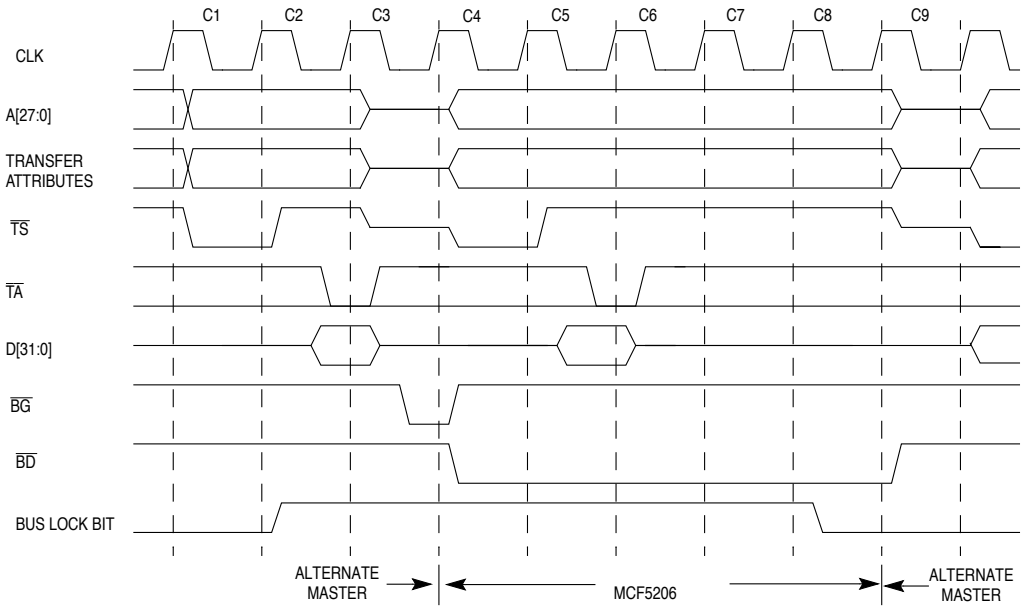


Figure 6-36. Two-Wire Bus Arbitration with Bus Lock Bit Asserted

In Figure 6-36 above, the alternate master is owner of the external bus during C1 and C2. During C3 the alternate master relinquishes control of the bus by asserting bus grant ($\overline{BG}$) to the MCF5206. At this point the bus lock bit is set to 1, and there is an internal access pending so the MCF5206 asserts bus driven ($\overline{BD}$) during C4 and begins the access. Thus, the MCF5206 becomes the explicit master of the external bus. Also during C4, the alternate master removes the grant from the MCF5206 by negating bus grant ($\overline{BG}$). Because the MCF5206 is the current bus master and the bus lock bit in the SIM Configuration Register (SIMR) is set to 1, it continues to assert $\overline{BD}$ even after the current transfer has completed. The MCF5206 negates the bus lock bit in SIMR during C8. Because bus grant ($\overline{BG}$) is negated, the MCF5206 negates bus driven ($\overline{BD}$) during C9 and three-states the external bus, thereby passing ownership of the external bus back to the alternate master.

Figure 6-37 is a bus arbitration state diagram for the MCF5206 bus arbitration protocol. Table 6-9 lists the conditions that cause bus arbitration state changes. Table 6-10

describes the MCF5206 bus ownership, bus driving and assertion of bus driven (BD) for each state of the bus arbitration state machine.

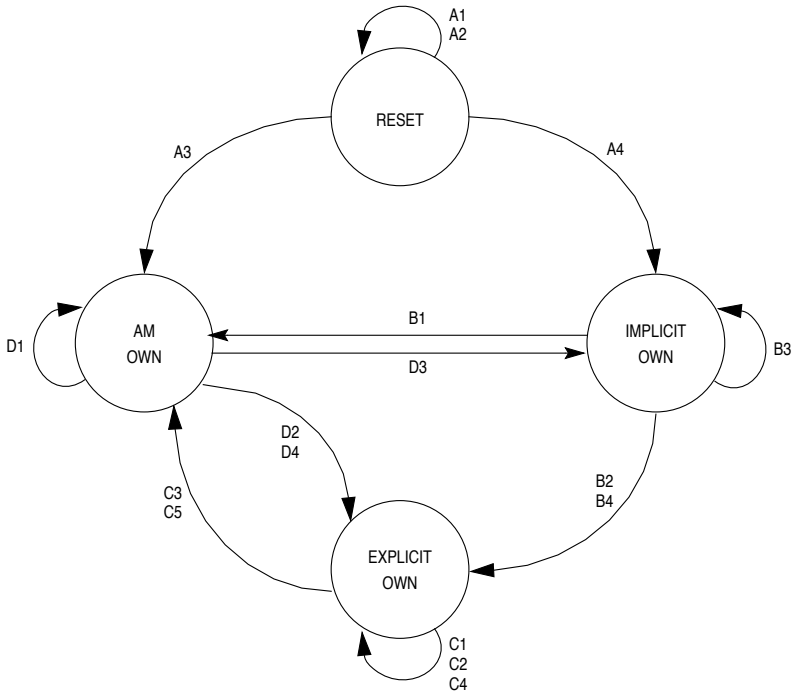


Figure 6-37. MCF5206 Two-Wire Bus Arbitration Protocol State Diagram

Table 6-9. MCF5206 Two-Wire Bus Arbitration Protocol Transition Conditions

PRESENT STATE	CONDITION LABEL	RSTI	SOFTWARE WATCHDOG RESET	BG	BUS LOCK BIT	INTERNAL BUS REQUEST	TRANSFER IN PROGRESS	END OF CYCLE	NEXT STATE
RESET	A!	A	-	-	-	-	-	-	Reset
	A2	N	A	-	-	-	-	-	Reset
	A3	N	N	N	-	-	-	-	AM Own
	A4	N	N	A	-	-	-	-	Implicit Own
IMPLICIT OWN	B1	N	N	N	-	-	-	-	AM Own
	B2	N	N	A	A	-	-	-	Explicit Own
	B3	N	N	A	N	N	-	-	Implicit Own
	B4	N	N	A	N	A	-	-	Explicit Own
EXPLICIT OWN	C1	N	N	A	-	-	-	-	Explicit Own
	C2	N	N	N	A	-	-	-	Explicit Own
	C3	N	N	N	N	-	N	-	AM Own
	C4	N	N	N	-	-	A	N	Explicit Own
	C5	N	N	N	N	-	A	A	AM Own

Table 6-9. MCF5206 Two-Wire Bus Arbitration Protocol Transition Conditions

AM OWN	D1	N	N	N	-	-	-	-	AM Own
	D2	N	N	A	A	-	-	-	Explicit Own
	D3	N	N	A	N	N	-	-	Implicit Own
	D4	N	N	A	N	A	-	-	Explictit Own

NOTES

1)"N" means negated; "A" means asserted; "AM" means alternate master.

2)End of Cycle: Whatever terminates a bus transaction whether it is normal or bus error. Note that bus cycles that result from a burst inhibited transfer are considered part of that original transfer.

Table 6-10. MCF5206 Two-Wire Arbitration Protocol State Diagram

STATE	OWN	BUS STATUS	BD
Reset	No	Not Driven	Negated
Implicit Own	Yes	Not Driven	Negated
Explicit Own	Yes	Driven	Asserted
Am Own	No	Not Driven	Negated

The MCF5206 can be in any one of four arbitration states during bus operation: reset, alternate master ownership, implicit ownership, or explicit ownership.

The MCF5206 enters the reset state whenever $\overline{\text{RSTI}}$ or software watchdog reset is asserted in any bus arbitration state. When $\overline{\text{RSTI}}$ and the software watchdog reset are negated, the MCF5206 proceeds to the implicit ownership state or alternate master ownership state, depending on $\overline{\text{BG}}$.

The alternate master ownership state denotes the MCF5206 does not have ownership ($\overline{\text{BG}}$ negated) of the bus and the MCF5206 does not drive the bus. The MCF5206 can assert memory control signals (i.e., $\overline{\text{CS}}$[7:0], $\overline{\text{WE}}$[3:0], $\overline{\text{RAS}}$[1:0] or $\overline{\text{CAS}}$[3:0]) and transfer acknowledge ($\overline{\text{TA}}$) during this state.

The implicit ownership state indicates that the MCF5206 owns the bus because $\overline{\text{BG}}$ is asserted to it. The MCF5206, however, is not ready to begin a bus cycle and the bus lock bit in the SIM Configuration Register (SIMR) is cleared. In this case, the MCF5206 keeps the bus three-stated until an internal bus request occurs or the bus lock bit in the SIMR is set to 1.

The MCF5206 explicitly owns the bus when the bus is granted to it ($\overline{\text{BG}}$ asserted) and at least one bus cycle has been initiated or the bus lock bit in the SIMR is set to 1. The MCF5206 asserts $\overline{\text{BD}}$ in this state to indicate the MCF5206 has explicit ownership of the bus. Until $\overline{\text{BG}}$ is negated, the MCF5206 retains explicit ownership of the bus whether or not active bus cycles are being executed. Once $\overline{\text{BG}}$ is negated and the bus lock bit in the SIMR is cleared, the MCF5206 will relinquish the bus at the end of the current bus cycle. When the MCF5206 is ready to relinquish the bus, it negates $\overline{\text{BD}}$ and three-states the bus signals.

6.9.2 Multiple External Bus Master Arbitration Protocol (Three-Wire Mode)

The three-wire mode of bus arbitration allows the MCF5206 to share the external bus with any number of external bus masters. In this mode, an external arbiter must be provided to assign priorities to each of the possible bus masters and determine which master should be allowed use of the external bus. The bus arbitration signals of the MCF5206, $\overline{BR}$, $\overline{BD}$, and $\overline{BG}$ connect to the bus arbiter, allowing the bus arbiter to control use of the external bus by the MCF5206.

The MCF5206 requests the bus from the external bus arbiter by asserting bus request ($\overline{BR}$) whenever an internal bus request is pending (the ColdFire core requests an access). The MCF5206 continues to assert $\overline{BR}$ until after the start of the external bus transfer. The MCF5206 can negate $\overline{BR}$ at any time regardless of the bus grant ($\overline{BG}$) status. If the bus is granted to the MCF5206 when an internal bus request is generated, the MCF5206 will assert bus driven ($\overline{BD}$) simultaneously with transfer start, allowing the access to begin immediately. The MCF5206 always drives $\overline{BR}$ and $\overline{BD}$. They cannot be directly wire-ORed with other devices.

The external arbiter asserts $\overline{BG}$ to indicate to the MCF5206 that it has been granted the bus and may begin a bus cycle after the rising edge of the next CLK. If $\overline{BG}$ is negated while a bus cycle is in progress, the MCF5206 relinquishes the bus at the completion of the bus cycle, except if the bus lock (BL) bit in the SIMR is set. To guarantee that the bus is relinquished, BL must be cleared and $\overline{BG}$ must be negated prior to the rising edge of the CLK in which the last $\overline{TA}$, $\overline{TEA}$ or internal asynchronous transfer acknowledge is asserted. Note that the MCF5206 considers any series of bus transfers of a burst or a burst-inhibited transfer to be a single bus cycle and does not relinquish the bus until completion of the last transfer of the series.

When the bus has been granted to the MCF5206 in response to the assertion of $\overline{BR}$, one of two situations can occur. In the first case, the MCF5206 has an internal bus request pending, the MCF5206 asserts $\overline{BD}$ to indicate explicit bus ownership and begins the pending bus cycle by asserting $\overline{TS}$. The MCF5206 continues to assert $\overline{BD}$ until the external bus master 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 the bus is owned by the MCF5206 and the MCF5206 continuously drives the address bus, attributes and control signals.

In the second situation, the bus is granted to the MCF5206, but the MCF5206 does not have an internal bus request pending and the bus lock bit in the SIMR is cleared. In this case, the MCF5206 takes implicit ownership of the bus. Implicit ownership of the bus occurs when the MCF5206 is granted the bus, but there are no pending bus cycles. The MCF5206 does not drive the bus and does not assert $\overline{BD}$ if the bus is implicitly owned. If an internal bus request is generated or the bus lock bit in the SIMR is set to 1, the MCF5206 assumes explicit ownership of the bus. If explicit ownership was assumed due to an internal request being generated, the MCF5206 immediately begins an access and simultaneously asserts $\overline{BD}$ and $\overline{TS}$. If explicit ownership was assumed due to the bus lock

bit being set to 1, the MCF5206 asserts $\overline{BD}$ and drives the address, attributes, and control signals. In this case, the MCF5206 will be the explicit master of the external bus, but will not begin an access until an internal request is generated. Figure 6-38 illustrates implicit and explicit bus ownership due to the bus lock bit being set then an internal bus request being generated.

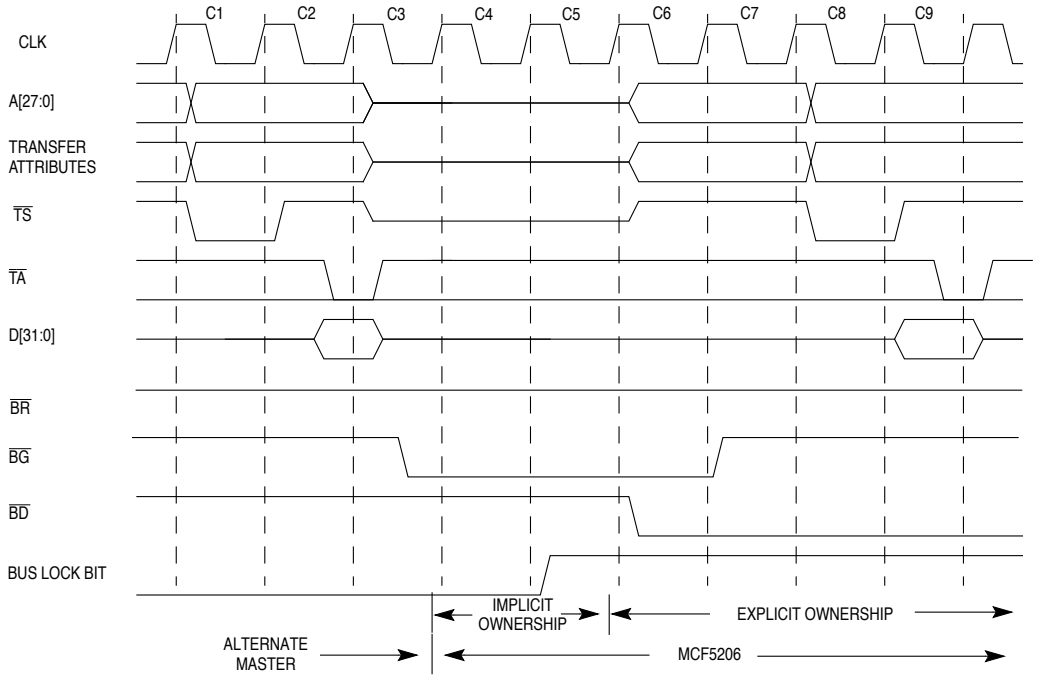


Figure 6-38. Three-Wire Implicit and Explicit Bus Ownership

In Figure 6-38, the alternate master has ownership of the external bus during C1 and C2. In C3, the alternate master releases control of the bus and the external arbiter asserts bus grant ($\overline{BG}$) to the MCF5206. During C4 and C5, the MCF5206 is implicit owner because an internal access is not pending and the bus lock bit in the SIMR is cleared. During C5, the bus lock bit is set to 1, causing the MCF5206 to take explicit ownership of the bus during C6 by asserting $\overline{BD}$. During C7, the external arbiter removes the bus grant from the MCF5206 by negating $\overline{BG}$. Because the bus lock bit is set to 1, the MCF5206 does not relinquish the bus (the MCF5206 continues to assert $\overline{BD}$).

NOTE

The MCF5206 can start a transfer during the CLK cycle after $\overline{BG}$ is asserted. The external arbiter should not assert $\overline{BG}$ to the MCF5206 until the previous alternate master has stopped driving the bus. $\overline{BG}$ cannot be asserted while another alternate master transfer is still in progress or damage to the part could occur.

When the bus has been removed from the MCF5206, one of two situations can occur. In the first case, the bus lock bit in the SIMR is cleared and the MCF5206 has explicit ownership of the bus. When the external bus master negates $\overline{BG}$, the MCF5206 will complete the current transfer, then negate $\overline{BD}$ and three-state the address, data, $\overline{TS}$, R/$\overline{W}$, and SIZ signals after completing the current bus cycle.

In the second case, the bus lock bit in the SIMR is set to 1 and the MCF5206 has explicit ownership of the bus. In this case, when the external bus master negates $\overline{BG}$, the MCF5206 will continue to assert $\overline{BD}$ and will continue to drive address, attributes, and control signals. The MCF5206 will retain mastership of the bus until the bus lock bit in the SIMR is cleared. By asserting the bus lock bit, you can select the MCF5206 to be the highest priority master, even when mastership of the bus is controlled by an external arbiter. In this fashion, the MCF5206 can be guaranteed mastership of the bus when

executing time-critical, bus-intensive operations. Figure 6-39 illustrates bus arbitration using the bus lock bit to control the arbitration.

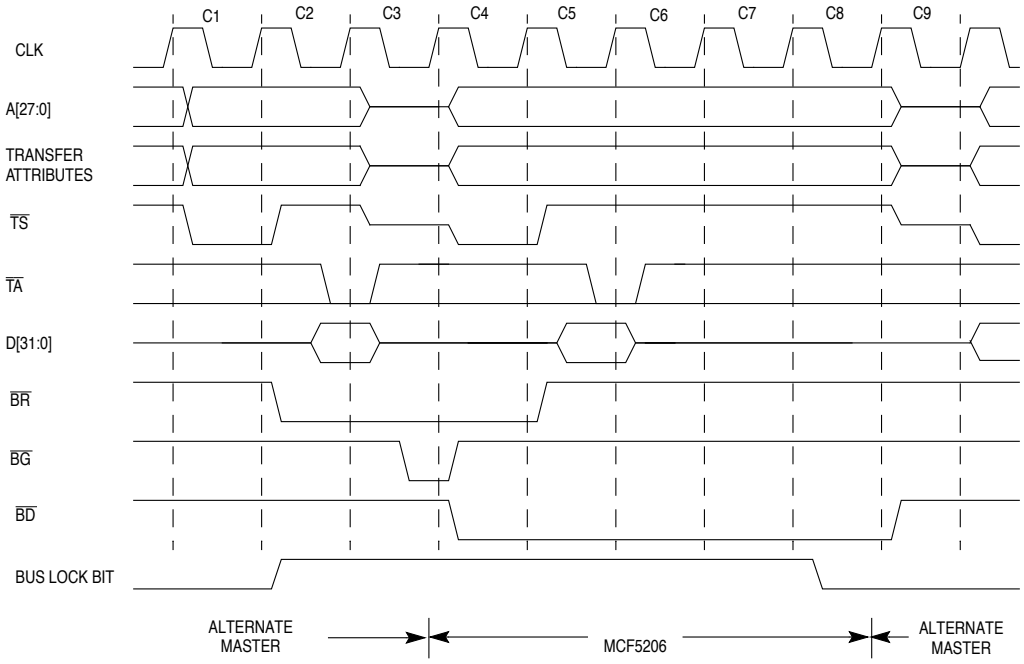


Figure 6-39. Three-Wire Bus Arbitration with Bus Lock Bit Asserted

In Figure 6-39, the alternate master is owner of the external bus during C1 and C2. During C2, the MCF5206 requests the external bus due to a pending internal transfer. On Clock C3, the alternate master relinquishes control of the bus and the external arbiter grants the bus to the MCF5206 by asserting bus grant ($\overline{BG}$). At this point the bus lock bit is set to 1, and there is an internal access pending so the MCF5206 asserts bus driven ($\overline{BD}$) during Clock C4, and begins the access. Thus, the MCF5206 becomes the explicit master of the external bus. Also during C4, the external arbiter removes the grant from the MCF5206 by negating bus grant ($\overline{BG}$). Because the MCF5206 is the current bus master and the bus lock bit in the SIMR is set to 1, it continues to assert $\overline{BD}$ even after the current transfer has completed. The MCF5206 negates the bus lock bit in the SIMR during C8. Because bus grant ($\overline{BG}$) is negated, the MCF5206 negates bus driven ($\overline{BD}$) during C9 and three-states the external bus, thereby passing ownership of the external bus to an alternate master.

$\overline{BR}$ can be used by the external arbiter as an indication that the MCF5206 needs the bus. However, there is no guarantee that when the bus is granted to the MCF5206, that a bus cycle will be performed. At best, $\overline{BR}$ must be used as a status output that indicates when the MCF5206 needs the bus, but not as an indication that the MCF5206 is in a certain bus arbitration state.

Figure 6-40, a high level bus arbitration state diagram for the MCF5206 bus arbitration protocol, can be used by external arbiters to predict how the MCF5206 operates as a function of external signals. Table 6-11 lists conditions that cause a change to and from the various states. Table 6-12 describes the MCF5206 bus ownership, bus driving, and assertion of bus driven ($\overline{BD}$) for each state of the bus arbitration state machine.

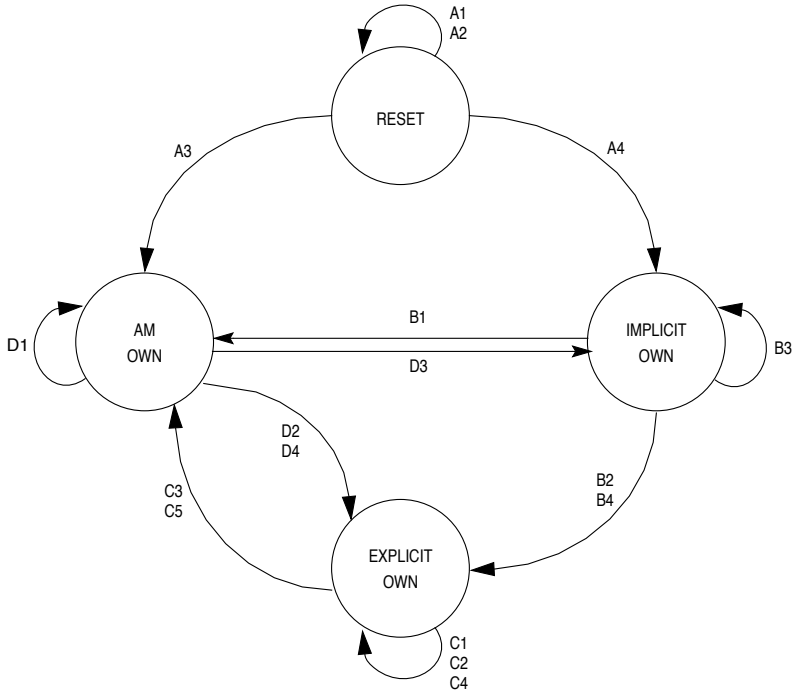


Figure 6-40. MCF5206 Bus Arbitration Protocol State Diagram

Table 6-11. MCF5206 Three-Wire Bus Arbitration Protocol Transition Conditions

PRESENT STATE	CONDITION LABEL	RSTI	SOFTWARE WATCHDOG RESET	BG	BUS LOCK BIT	INTERNAL BUS REQUEST ($\overline{IBR}$)	TRANSFER IN PROGRESS	END OF CYCLE	NEXT STATE
RESET	A1	A	-	-	-	-	-	-	Reset
	A2	N	A	-	-	-	-	-	Reset
	A3	N	N	N	-	-	-	-	AM Own
	A4	N	N	A	-	-	-	-	Implicit Own
IMPLICIT OWN	B1	N	N	N	-	-	-	-	AM Own
	B2	N	N	A	A	-	-	-	Explicit Own
	B3	N	N	A	N	N	-	-	Implicit Own
	B4	N	N	A	-	A	-	-	Explicit Own

Table 6-11. MCF5206 Three-Wire Bus Arbitration Protocol Transition Conditions

EXPLICIT OWN	C1	N	N	A	-	-	-	-	Explicit Own
	C2	N	N	N	Y	-	-	-	Explicit Own
	C3	N	N	N	N	-	N	-	AM Own
	C4	N	N	N	-	-	Y	N	Explicit Own
	C5	N	N	N	N	-	Y	Y	AM Own
AM OWN	D1	N	N	N	-	-	-	-	AM Own
	D2	N	N	A	A	-	-	-	Explicit Own
	D3	N	N	A	N	N	-	-	Implicit Own
	D4	N	N	A	N	A	-	-	Explicit Own

NOTES:

1)"N" means negated; "A" means asserted; "AM" means alternate master.

2)End of Cycle: Whatever terminates a bus transaction whether it is normal or bus error. Note that bus cycles that result from a burst inhibited transfer are considered part of that original transfer.

3)$\overline{IBR}$ refers to an internal bus request.The output signals $\overline{BR}$ is a registered version of $\overline{IBR}$ when $\overline{BG}$ is negated and $\overline{BD}$ is negated. There is an internal bus request when the Coldfire core requires the external bus for an operand transfer.

Table 6-12. MCF5206 Three-Wire Arbitration Protocol State Diagram

STATE	OWN	BUS STATUS	BD
Reset	No	Not Driven	Negated
Implicit Own	Yes	Not Driven	Negated
Explicit Own	Yes	Driven	Asserted
AM Own	No	Not Driven	Negated

The MCF5206 can be in any one of four arbitration states during bus operation: reset, alternate master own, implicit ownership, and explicit ownership.

The reset state is entered whenever $\overline{RSTI}$ or software watchdog reset is asserted in any bus arbitration state. When $\overline{RSTI}$ and the software watchdog reset are negated, the MCF5206 proceeds to the implicit ownership state or alternate master ownership state, depending on bus grant ($\overline{BG}$).

The alternate master ownership state denotes the MCF5206 does not have ownership (bus grant ($\overline{BG}$) negated) of the bus and the MCF5206 does not drive the bus. The MCF5206 can assert memory control signals (i.e., $\overline{CS}$[7:0], $\overline{WE}$[3:0], $\overline{RAS}$[1:0] or $\overline{CAS}$[3:0]) $\overline{TA}$ and $\overline{BR}$ during this state.

The implicit ownership state indicates that the MCF5206 owns the bus because bus grant ($\overline{BG}$) is asserted to it. The MCF5206, however, is not ready to begin a bus cycle and the bus lock bit in the SIMR is cleared, and it keeps the bus three-stated until an internal bus request occurs or the bus lock bit in the SIMR is set to 1.

The MCF5206 explicitly owns the bus when the bus is granted to it (bus grant ($\overline{BG}$) asserted) and at least one bus cycle has initiated or the bus lock bit in the SIMR is set to 1. The MCF5206 asserts $\overline{BD}$ in this state to indicate the MCF5206 has explicit ownership of the bus. Until bus grant ($\overline{BG}$) is negated, the MCF5206 regains explicit ownership of the

bus whether or not active bus cycles are being executed. Once bus grant ($\overline{BG}$) is negated and the bus lock bit in the SIMR is cleared, the MCF5206 will relinquish the bus at the end of the current bus cycle. When the MCF5206 is ready to relinquish the bus, it negates $\overline{BD}$ and three-states the bus signals.

The bus arbitration state diagram for the MCF5206 three-wire bus arbitration protocol can be used to approximate the high level behavior of the MCF5206. It is assumed that all $\overline{TS}$ signals in a system are tied together and each bus master's $\overline{BD}$ and $\overline{BR}$ signals are connected individually to the external bus arbiter. The external bus arbiter must be careful to make sure any alternate bus master has relinquished the bus or will be relinquishing the bus after the next rising edge of CLK before asserting bus grant ($\overline{BG}$) to the MCF5206. The MCF5206 does not monitor external bus master operation regarding bus arbitration.

NOTE

The MCF5206 can start a transfer on the rising edge of CLK the cycle after $\overline{BG}$ is asserted. The external arbiter should not assert $\overline{BG}$ to the MCF5206 until the previous alternate master has stopped driving the bus. $\overline{BG}$ cannot be asserted while another alternate master transfer is still in progress or damage to the part could occur.

6.10 ALTERNATE BUS MASTER OPERATION

The MCF5206 can monitor bus transfers by other bus masters and can assert chip-select, DRAM control, and transfer termination signals during these transfers. Assertion of chip-select and DRAM control signals can occur when the bus is granted to another bus master and $\overline{TS}$ is asserted by the alternate master as an input to the MCF5206.

NOTE

Alternate masters that are using internal MCF5206 chip-select, DRAM, and default memory control signals must initiate aligned transfers only.

The MCF5206 registers the value of A[27:0], R/$\overline{W}$, and SIZ[1:0] on the rising edge of CLK in which $\overline{TS}$ is asserted.

NOTE

If the pins A[27:24]/$\overline{CS}$[7:4]/$\overline{WE}$[0:3] are not assigned to output address signals, a value of $0 is assigned internally to A[27:24]. Also, $\overline{TT}$[1:0] and ATM are not examined during alternate master transfers. The mask bits SC, SD, UC, UD and C/I in the Chip-Select Mask Registers (CSMR) and in the

DRAM Controller Mask Registers (DCMR) are not used during alternate master transfers.

If the assertion of chip-select, DRAM control, and transfer termination signals during alternate master accesses is not required, the MCF5206 TS pin should not be asserted when bus driven ($\overline{BD}$) is negated.

This subsection will concentrate on alternate master accesses to default memory. For more information on external master accesses to chip-select and DRAM memory spaces, refer to **Section 8 Chip-Select** and **Section 10 DRAM Controller**.

During alternate master transfers, the MCF5206 examines the address, direction, and size of the transfer, and on the next rising edge of CLK, begins assertion of the proper sequence of memory control signals. If the transfer is decoded to be a chip select address and the chip-select is enabled for the direction of the transfer (read- and/or write-enabled), the appropriate chip-select and write-enable signals will be asserted. If the chip-select is enabled for external master automatic acknowledge, $\overline{TA}$ will be driven and asserted at the appropriate time.

The MCF5206 does not drive addresses during external bus master accesses that are decoded as chip-select or default memory transfers. The alternate master must provide the correct address to the external memory at the appropriate time. If the transfer is decoded to be a DRAM address and the DRAM bank is enabled for the direction of the transfer (read- and/or write-enabled), the appropriate DRAM control address and the transfer- acknowledge ($\overline{TA}$) signals will be asserted. If the address of the transfer is neither a chip- select or a DRAM address, the SIM will read the DMCR. If the external master automatic acknowledge (EMAA) bit in the DMCR is set, the MCF5206 will drive $\overline{TA}$ and will assert transfer acknowledge after the number of clocks programmed in the wait state bits (WS) in the DMCR. For more information about programming the Default Memory Control Register, refer to the SIM section. Table 6-13 lists the signals and conditions under which the MCF5206 drives these signals during alternate master accesses.

Table 6-13. Signal Source During Alternate Master Accesses

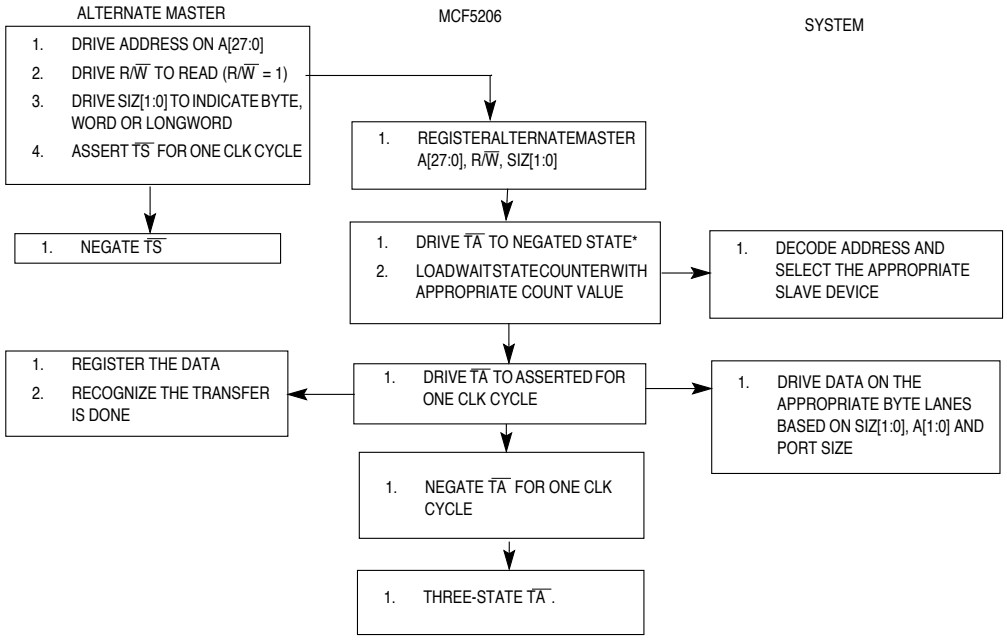
MEMORY SPACE	ADDRESS (DRIVEN BY)	CONTROL SIGNALS	TRANSFER ACKNOWLEDGE
Chip-Select	Alternate Master	$\overline{CS}$[7:0], $\overline{WE}$[3:0]	MCF5206: if EMAA in CSCR is set to 1
DRAM	MCF5206: if DCAR in DCCR is set to 1	$\overline{RAS}$[1:0], $\overline{CAS}$[3:0], $\overline{DRAMW}$	MCF5206
Default Memory	Alternate Master	-	MCF5206: if EMAA in DMCR is set to 1

6.10.1 Alternate Master Read Transfer Using MCF5206 Termination

The basic read cycle of an alternate master transfer using MCF5206-generated termination is the same as a ColdFire core initiated transfer with one additional CLK cycle

between the assertion of $\overline{TS}$ by the alternate master and the starting of the internal wait-state counter by the MCF5206. During this CLK cycle, the MCF5206 decodes the alternate master address to determine the appropriate memory control and termination signals that must be asserted. For more information on chip-select transfers and DRAM transfers, refer to **Section 8 Chip-Selects** and **Section 10 DRAM Controller**.

Figure 6-41 is a flow chart for alternate master read transfers using MCF5206-generated automatic acknowledge to access 8-, 16-, or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer.



*TA IS DRIVEN NEGATED IF THE APPROPRIATE WAIT STATE COUNT IS GREATER THAN ZERO. IF THE WAIT STATE COUNT IS ZERO, TA IS DRIVEN AND ASSERTED DURING THE SAME CLK .

Figure 6-41. Alternate Master Read Transfer using MCF5206-Generated Transfer Acknowledge Flowchart

Figure 6-42 illustrates transfer acknowledge ($\overline{TA}$) assertion by the MCF5206 during alternate master read transfers.

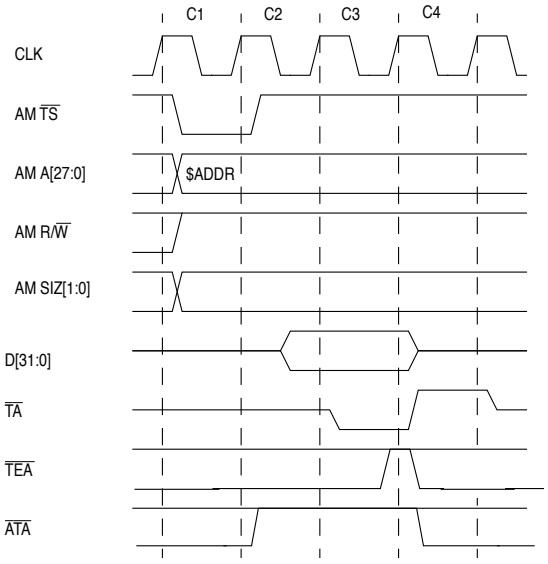


Figure 6-42. Alternate Master Read Transfer Using MCF5206 Transfer Acknowledge Timing (No Wait States)

Clock 1 (C1)

The read cycle starts in C1. During C1, the alternate master drives valid values on the address bus (A[27:0]) and transfer control signals. The read/write (R/$\overline{W}$) signal is driven high for a read cycle, and the size signals (SIZ[1:0]) are driven to indicate the transfer size. The alternate master asserts transfer start ($\overline{TS}$) to indicate the beginning of a bus cycle.

Clock 2 (C2)

At the start of C2, the MCF5206 registers the alternate master address bus, read/write and size signals. During C2, the MCF5206 decodes the registered address and read/write signals and if the external master automatic acknowledge (EMAA) bit in the Default Memory Control Register (DMCR) is set to 1, the MCF5206 selects the indicated number of wait states for loading into the internal wait state counter. During C2, the alternate master negates $\overline{TS}$ and samples the level of $\overline{TA}$. The selected device(s) decodes the address and drives the appropriate data onto the data bus.

Clock 3 (C3)

At the start of C3, if the EMAA bit in the Default Memory Control Register (DMCR) is set to 1 and the number of wait states is zero, the MCF5206 drives $\overline{TA}$ signal to the asserted

state. During C3, the alternate master samples the level of $\overline{TA}$ and if $\overline{TA}$ is asserted, latches the data and terminates the transfer. If $\overline{TA}$ is negated, the alternate master continues to insert wait states instead of terminating the transfer. The alternate master must continue to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

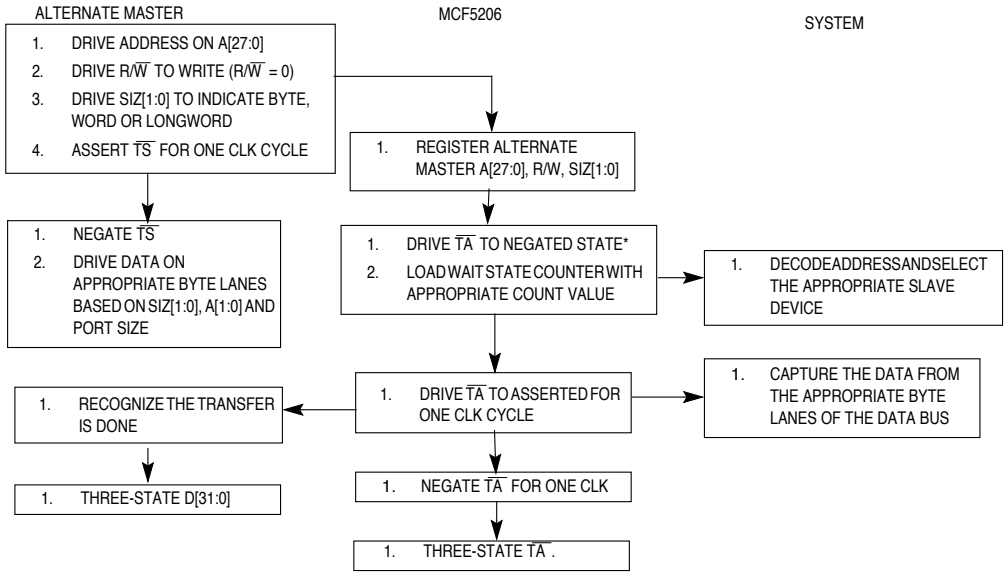
Clock 4 (C4)

During C4, the selected slave device drives the data bus to a high-impedence state. The MCF5206 negates $\overline{TA}$ and drives $\overline{TA}$ to a high-impedence state after the next rising edge of CLK.

6.10.2 Alternate Master Write Transfer Using MCF5206 Termination

The basic write cycle of an alternate master transfer using MCF5206-generated termination is the same as a ColdFire core-initiated transfer with one additional CLK cycle between the assertion of $\overline{TS}$ by the alternate master and the start of the internal wait state counter by the MCF5206. During this CLK cycle, the MCF5206 decodes the alternate master address to determine the appropriate memory control and termination signals that must be asserted. For more information on chip-select transfers, refer to the Chip-Selects section. For more information on DRAM transfers, refer to the DRAM Controller section.

Figure 6-43 is a flow chart for alternate master write transfers using MCF5206-generated automatic acknowledge to access 8-, 16-, or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer.



*TA IS DRIVEN AND NEGATED IF THE APPROPRIATE WAIT STATE COUNT IS GREATER THAN ZERO. IF THE WAIT STATE COUNT IS ZERO, TA IS DRIVEN AND ASSERTED DURING THE SAME CLK .

Figure 6-43. Alternate Master Write Transfer Using MCF5206-Generated

Transfer Acknowledge Flowchart

Figure 6-44 illustrates $\overline{TA}$ assertion by the MCF5206 during alternate master write transfers.

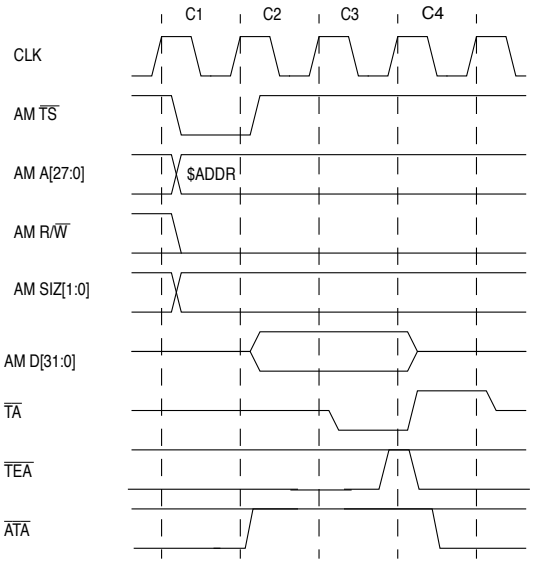


Figure 6-44. Alternate Master Write Transfer Using MCF5206 Transfer-Acknowledge Timing (No Wait States)

Clock 1 (C1)

The write cycle starts in C1. During C1, the alternate master drives valid values on the address bus (A[27:0]) and transfer control signals. The read/write (R/$\overline{W}$) signal is driven low for a write cycle, and the size signals (SIZ[1:0]) are driven to indicate the transfer size. The alternate master asserts transfer start ($\overline{TS}$) to indicate the beginning of a bus cycle.

Clock 2 (C2)

At the start of C2, the MCF5206 registers and decodes the alternate master address bus, read/write and size signals. If the external master automatic acknowledge (EMAA) bit in the Default Memory Control Register (DMCR) is set to 1, the MCF5206 selects the indicated number of wait states for loading into the internal wait state counter. During C2, the alternate master negates $\overline{TS}$, places the data on the data bus (D[31:0]), and samples the level of $\overline{TA}$. The selected device(s) decode the address and latch the data when it is ready.

Clock 3 (C3)

At the start of C3, if the EMAA bit in the Default Memory Control Register (DMCR) is set to 1 and the number of wait states is zero, the MCF5206 asserts $\overline{TA}$. During C3, the alternate master samples the level of $\overline{TA}$. If $\overline{TA}$ is asserted, the alternate master terminates the transfer. If $\overline{TA}$ is negated, the alternate master continues to output the data and inserts wait states instead of terminating the transfer. The alternate master must continue to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

Clock 4 (C4)

During C4, the alternate master places the data bus in a high-impedence state. The MCF5206 negates $\overline{TA}$ and drives $\overline{TA}$ to a high impedence state after the next rising edge of CLK.

6.10.3 Alternate Master Bursting Read Using MCF5206-Generated Transfer Termination

The bursting read transfer of an alternate master transfer using MCF5206-generated termination is similar to a ColdFire core initiated bursting transfer with the exception that one additional CLK cycle is inserted between the assertion of $\overline{TS}$ by the alternate master and the starting of the internal wait state counter by the MCF5206. If the transfer is to default memory, the alternate master must increment the address to the appropriate value after each assertion of transfer acknowledge. For more information on chip-select transfers, refer to **Section 8 Chip-Selects.** For more information on DRAM transfers, refer to **Section 10 DRAM Controller.**

NOTE

An alternate master cannot initiate a bursting read transfer for a chip-select or default memory space where the burst-enable bit (BRST) in the Chip-Select Control Register (CSCR) or the Default Memory Control Register (DMCR) is cleared. Undefined behavior will occur if you attempt such a transfer.

Figure 6-45 is a flowchart for an alternate master bursting read transfer using MCF5206-generated automatic acknowledge to access 8-, 16-, or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer, and the specific number of cycles needed for each transfer. A bursting read transfer can be from 2 to 16 transfers long. The flowchart in Figure 6-45 is for a bursting transfer 4 transfers long.

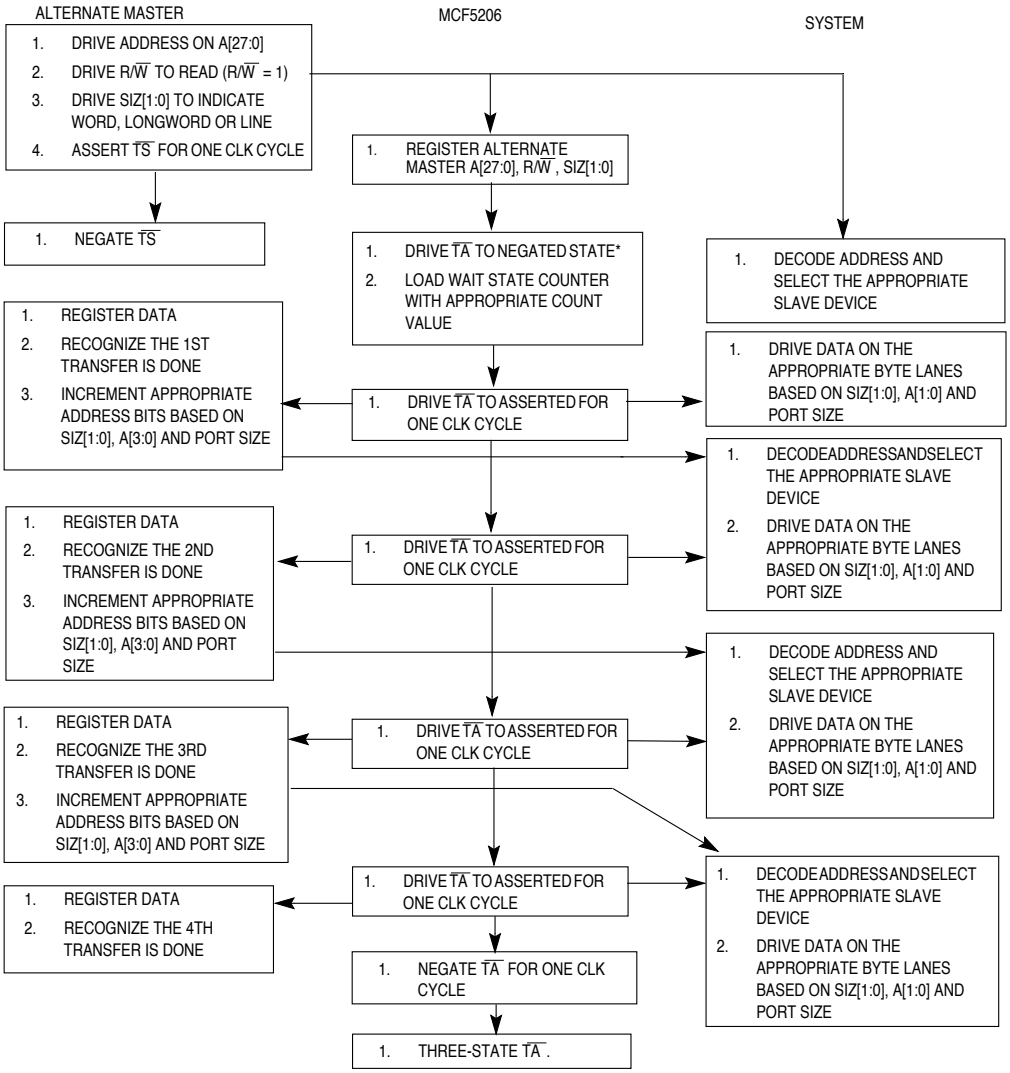


Figure 6-45. Alternate Master Bursting Read Transfer Using MCF5206-Generated Transfer-Acknowledge Flowchart

Figure 6-46 illustrates $\overline{TA}$ assertion by the MCF5206 during alternate master bursting read transfers.

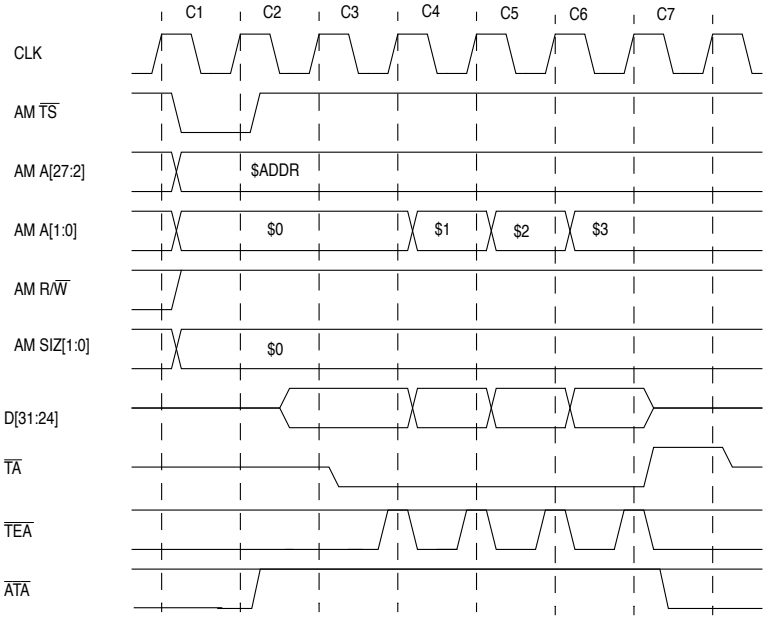


Figure 6-46. Alternate Master Bursting Longword Read Transfer to an 8-Bit Port Using MCF5206 Transfer-Acknowledge Timing (No Wait States)

Clock 1 (C1)

The read cycle starts in C1. During C1, the alternate master places valid values on the address bus (A[27:0]) and transfer control signals. The read/write (R/$\overline{W}$) signal is driven high for a read cycle, and the size signals (SIZ[1:0]) are driven to $0 to indicate a longword transfer. The alternate master asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

At the start of C2, the MCF5206 registers the alternate master address bus, read/write and size signals. During C2, the MCF5206 decodes the registered address and read/write signals and if the external master automatic acknowledge (EMAA) bit in the Default Memory Control Register (DMCR) is set to 1, the MCF5206 selects the indicated number of wait states for loading into the internal wait state counter. During C2, the alternate master negates $\overline{TS}$ and samples the level of $\overline{TA}$. The selected device(s) decodes the address and drives the appropriate data onto the data bus.

Clock 3 (C3)

At the start of C3, if the EMAA bit in the Default Memory Control Register (DMCR) is set to 1 and the number of wait states is zero, the MCF5206 drives $\overline{TA}$ signal to the asserted state. During C3, the alternate master samples the level of $\overline{TA}$. If $\overline{TA}$ is asserted, the alternate master latches the first byte of data from D[31:24]. If $\overline{TA}$ is negated, the alternate master continues to insert wait states instead of terminating the transfer. The alternate master must continue to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

Clock 4 (C4)

During C4, the alternate master increments the address by one to access the second byte of data in the longword transfer. The alternate master also samples the level of $\overline{TA}$. If $\overline{TA}$ is asserted, the alternate master latches the second byte of data from D[31:24]. If $\overline{TA}$ is negated, the alternate master continues to insert wait states instead of terminating the transfer. The alternate master must continue to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

The selected slave decodes the address and outputs the next byte of data on D[31:24]. The MCF5206 continues to assert $\overline{TA}$.

Clock 5 (C5)

This clock is identical to C4, except the alternate master increments the address to point to the third byte of data, and the selected slave decodes the address and outputs the third byte of data of the longword transfer.

Clock 6 (C6)

This clock is identical to C4, except the alternate master increments the address to point to the fourth byte of data, and the selected slave decodes the address and outputs the fourth byte of data of the longword transfer.

Clock 7 (C7)

During C7, the selected slave device drives the data bus to a high impedence state. The MCF5206 drives $\overline{TA}$ to the inactive state and then drives $\overline{TA}$ to a high-impedence state after the next rising edge of CLK.

6.10.4 Alternate Master Bursting Write Using MCF5206-Generated Transfer Termination

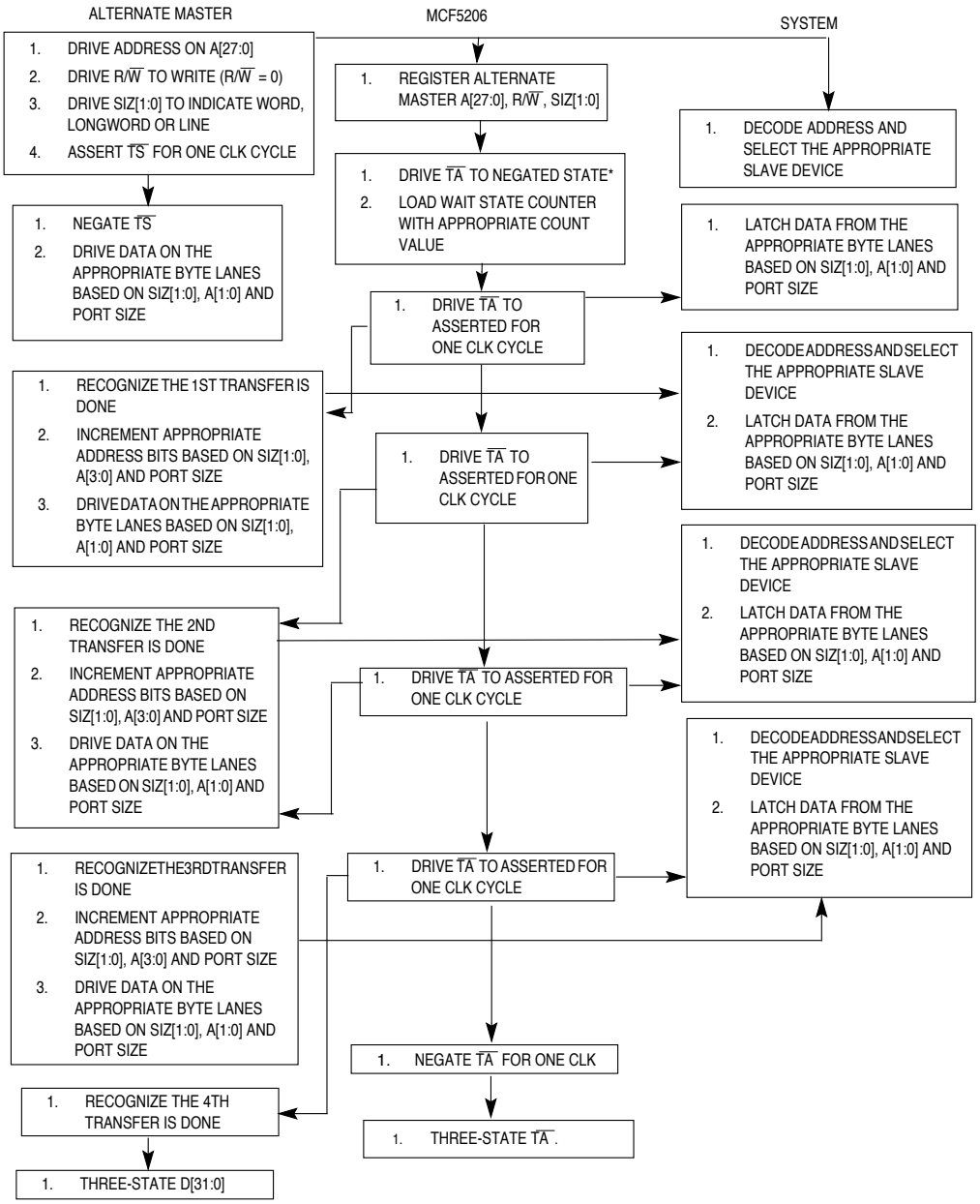
The bursting write transfer of an alternate master using MCF5206-generated termination is similar to a ColdFire core initiated bursting write transfer except that one additional CLK cycle is inserted between the assertion of $\overline{TS}$ by the alternate master and the start of the internal wait state counter by the MCF5206. If the transfer is to default memory, the alternate master must increment the address to the appropriate value after each assertion

of transfer acknowledge. For more information on chip-select transfers or DRAM transfers, refer to **Section 8 Chip-Selects** or to **Section 10 DRAM Controller**.

NOTE

An alternate master cannot initiate a bursting write transfer for a chip-select or default memory space where the burst-enable bit (BRST) in the Chip-Select Control Register (CSCR) or the Default Memory Control Register (DMCR) is cleared. Undefined behavior will occur if you try this.

Figure 6-47 is a flowchart for alternate master bursting write transfer using MCF5206 generated automatic acknowledge to access 8-, 16- or 32-bit ports. Bus operations are similar for each case and vary only with the size indicated, the portion of the data bus used for the transfer and the specific number of cycles needed for each transfer. A bursting write transfer can be from two to sixteen transfers long. The flowchart shown in Figure 6-47 is for a bursting write transfer of four transfers long.



*TA IS DRIVEN AND NEGATED IF THE APPROPRIATE WAIT STATE COUNT IS GREATER THAN ZERO. IF THE WAIT STATE COUNT IS ZERO, TA IS DRIVEN AND ASSERTED DURING THE SAME CLK.

Figure 6-47. Alternate Master Bursting Write Transfer using MCF5206-Generated

Transfer-Acknowledge Flowchart

Figure 6-48 illustrates $\overline{TA}$ assertion by the MCF5206 during alternate master bursting write transfers.

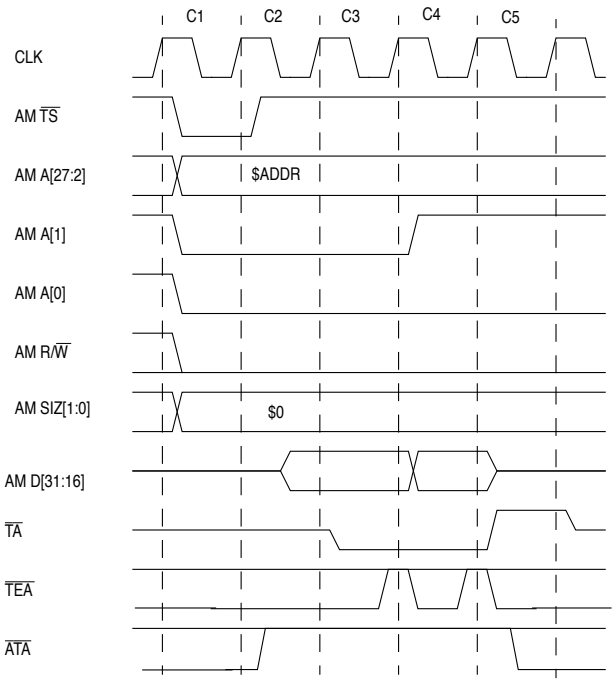


Figure 6-48. Alternate Master Bursting Longword Write Transfer to a 16-Bit Port Using MCF5206 Transfer Acknowledge Timing (No Wait States)

Clock 1 (C1)

The write cycle starts in C1. During C1, the alternate master places valid values on the address bus (A[27:0]) and transfer control signals. The read/write (R/$\overline{W}$) signal is driven low for a write cycle, and the size signals (SIZ[1:0]) are driven to $0 to indicate a longword transfer. The alternate master asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

At the start of C2, the MCF5206 registers and decodes the alternate master address bus, read/write and size signals. If the external master automatic acknowledge (EMAA) bit in the Default Memory Control Register (DMCR) is set to 1, the MCF5206 selects the indicated number of wait states for loading into the internal wait state counter. During C2, the alternate master negates $\overline{TS}$, drives the appropriate data onto the data bus, and

samples the level of $\overline{TA}$. The selected device(s) decodes the address and if ready, latches the appropriate data from the data bus.

Clock 3 (C3)

At the start of C3, if the EMAA bit in the Default Memory Control Register (DMCR) is set to 1 and the number of wait states is zero, the MCF5206 asserts $\overline{TA}$. During C3, the alternate master samples the level of $\overline{TA}$. If $\overline{TA}$ is asserted, the transfer of the first word is complete. If $\overline{TA}$ is negated, the alternate master continues to insert wait states instead of terminating the transfer. The alternate master must continue to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

Clock 4 (C4)

During C4, the alternate master increments the address by two to point to the second word of data in the longword transfer and outputs the second word of data onto the data bus. The alternate master also samples the level of $\overline{TA}$. If $\overline{TA}$ is asserted, the transfer of the second word of the longword transfer is complete. If $\overline{TA}$ is negated, the alternate master continues to insert wait states instead of terminating the transfer. The alternate master must continue to sample $\overline{TA}$ on successive rising edges of CLK until it is asserted.

The selected slave decodes the address and latches the next word of data on D[31:16]. The MCF5206 continues to assert $\overline{TA}$.

Clock 5 (C5)

During C5, the alternate master drives the data bus to a high-impedence state. The MCF5206 drives $\overline{TA}$ to the inactive state and places $\overline{TA}$ in a high-impedence state after the next rising edge of CLK.

6.11 RESET OPERATION

The MCF5206 supports three types of reset, two of which are external hardware resets (master reset and normal reset) and one internal reset—software watchdog reset. Master reset resets the entire MCF5206 including the DRAM controller. Normal reset resets all of the MCF5206 with the exception of the DRAM controller. Normal reset allows DRAM refresh cycles to continue at the programmed rate and with the programmed waveform timing while the remainder of the system is being reset, maintaining the data stored in DRAM. The software watchdog resets act as internally generated normal resets.

NOTE

Master reset must be asserted for all power-on resets. Failure to assert master reset during power-on sequences will result in unpredictable DRAM controller behavior.

6.11.1 MASTER RESET

To perform a master reset, an external device asserts the reset input pin ($\overline{RSTI}$) and the $\overline{HIZ}$ input pin ($\overline{HIZ}$) simultaneously. When power is applied to the system, external circuitry should assert $\overline{RSTI}$ for a minimum of six CLK cycles after Vcc is within tolerance. Figure 6-49 is a functional timing diagram of the master reset operation, illustrating relationships among Vcc, $\overline{RSTI}$, $\overline{HIZ}$, $\overline{RSTO}$, mode selects, and bus signals. CLK must be stable by the time Vcc reaches the minimum operating specification. CLK should start oscillating as Vcc is ramped up to clear out contention internal to the MCF5206 caused by the random manner in which internal flip-flops power up. $\overline{RSTI}$ and $\overline{HIZ}$ are internally synchronized for two CLKs before being used and must meet the specified setup and hold times to CLK only if recognition by a specific CLK rising edge is required.

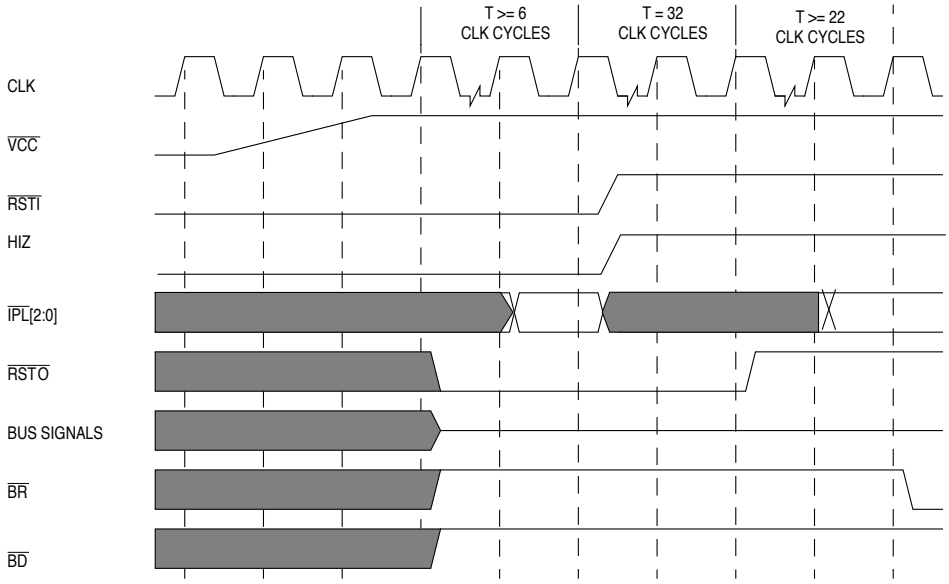


Figure 6-49. Master Reset Timing

$\overline{TS}$ must be pulled up or negated during master reset. When the assertion of $\overline{RSTI}$ is recognized internally, the MCF5206 will assert the reset out pin ($\overline{RSTO}$). $\overline{RSTO}$ will be asserted as long as $\overline{RSTI}$ is asserted and will remain asserted for 32 CLK cycles after $\overline{RSTI}$ is negated. For proper master reset operation, $\overline{RSTI}$ and HIZ must be asserted and negated simultaneously.

During the master reset period, all signals that can be are driven to a high-impedence state and all those that cannot be driven to a high-impedence state are driven to their negated states. Once $\overline{RSTI}$ negates, all bus signals continue to remain in a high-impedance state until the MCF5206 is granted the bus and the ColdFire core begins the first bus cycle for reset exception processing. A master reset causes any bus cycle (including DRAM refresh cycles) to terminate. In addition, master reset initializes registers

appropriately for a reset exception. During a master reset, the hard reset bit (HRST) bit in the Reset Status Register (RSR) is set and the software reset bit (SRST) in the RSR is cleared to indicate that an external hardware reset caused the previous reset.

The levels of the $\overline{IPLx}$ pins select the port size and acknowledge features of the global chip-select after a master reset occurs. The $\overline{IPLx}$ signals are synchronized and are registered on the last rising edge of CLK where $\overline{RSTI}$ and $\overline{HIZ}$ are asserted.

6.11.2 NORMAL RESET

External normal resets should be performed anytime it is important to maintain the data stored in DRAM during a reset. An external normal reset is performed when an external device asserts the reset input pin ($\overline{RSTI}$) while negating the $\overline{HIZ}$ input pin ($\overline{HIZ}$). During an external normal reset, $\overline{RSTI}$ must be asserted for a minimum of six CLKs. Figure 6-50 is a functional timing diagram of external normal reset operation, illustrating relationships among $\overline{RSTI}$, $\overline{HIZ}$, $\overline{RSTO}$, mode selects, and bus signals. $\overline{RSTI}$ and $\overline{HIZ}$ are internally synchronized for two CLKs before being used and must meet the specified setup and hold times to CLK only if recognition by a specific CLK rising edge is required.

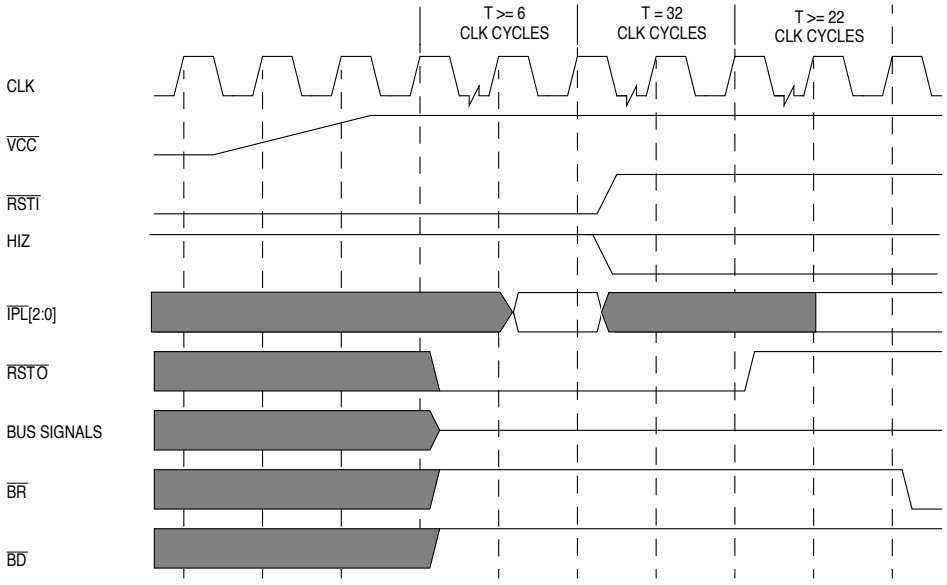


Figure 6-50. Normal Reset Timing

$\overline{TS}$ must be pulled up or negated during normal reset. When the assertion of $\overline{RSTI}$ is recognized internally, the MCF5206 will assert the reset out pin ($\overline{RSTO}$). $\overline{RSTO}$ will be asserted as long as $\overline{RSTI}$ is asserted and will remain asserted for 32 CLK cycles after $\overline{RSTI}$ is negated. For proper normal reset operation, $\overline{HIZ}$ must be negated as long as $\overline{RSTI}$ is asserted.

During the normal reset period, all signals that can be are driven to a high-impedence state and all those that cannot are driven to their negated states. Once $\overline{RSTI}$ negates, all bus signals continue to remain in a high-impedance state until the MCF5206 is granted the bus and the ColdFire core begins the first bus cycle for reset exception processing.

A normal reset causes all bus activity except DRAM refresh cycles to terminate. During a normal reset, DRAM refresh cycles will continue to occur at the programmed rate and with the programmed waveform timing. In addition, normal reset initializes registers appropriately for a reset exception. During an external normal reset, the hard reset (HRST) bit in the Reset Status Register (RSR) is set and the software reset (SRST) bit in the Reset Status Register (RSR) is cleared to indicate an external hardware reset caused the previous reset.

The levels of the $\overline{IPLx}$ pins select the port size and acknowledge features of the global chip-select after an external normal reset occurs. The $\overline{IPLx}$ signals are synchronized and are registered on the last rising edge of CLK where $\overline{RSTI}$ is asserted.

6.11.3 SOFTWARE WATCHDOG TIMER RESET OPERATION

If the software watchdog timer is programmed to generate a reset, when a timeout occurs an internal reset will be asserted for at least 31 CLKs, resetting internal registers as with a normal reset. The $\overline{RSTO}$ pin will assert for at least 31 CLKs after the software watchdog timeout. Figure 6-51 illustrates the timing of $\overline{RSTO}$ when asserted by a software watchdog timeout.

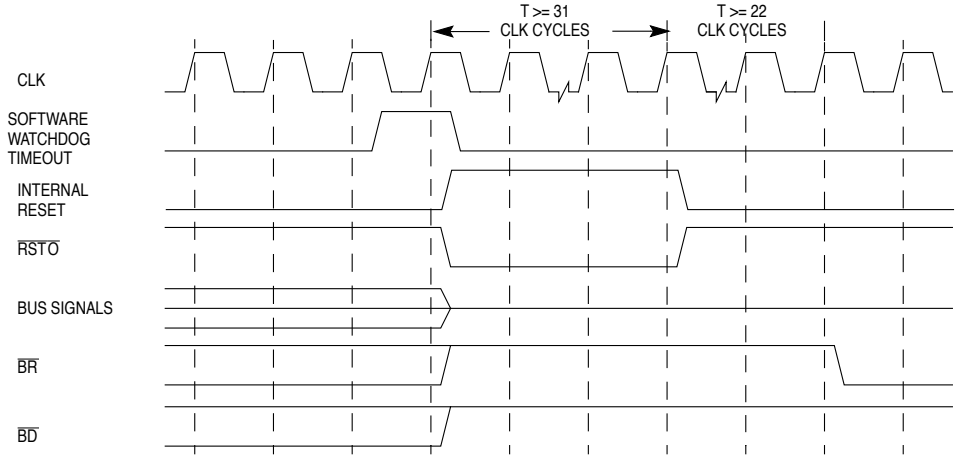


Figure 6-51. Software Watchdog Timer Reset Timing

NOTE

Like the normal reset, the internal reset generated by a software watchdog timeout does not reset the DRAM

controller. DRAM refreshes will continue to be generated during and after the software watchdog timout reset at the programmed rate and with the programmed waveform timing.

$\overline{TS}$ must be pulled up or negated during software watchdog reset. When the software watchdog timeout recognized internally, the reset out pin ($\overline{RTS2}$/$\overline{RSTO}$) will be asserted by the MCF5206. $\overline{RSTO}$ will be asserted for at least 31 CLK cycles after the internal software watchdog timer reset negated.

During the software watchdog timer reset period, all signals that can be are driven to a high-impedence state and all those that cannot are driven to their negated states. Once $\overline{RSTO}$ negates, all bus signals continue to remain in a high-impedance state until the MCF5206 is granted the bus and the ColdFire core begins the first bus cycle for reset exception processing.

A software watchdog timer reset causes all bus activity except DRAM refresh cycles to terminate. During a software watchdog timer reset, DRAM refresh cycles will continue to occur at the programmed rate and with the programmed waveform timing. In addition, software watchdog timer reset initializes registers appropriately for a reset exception. During a software watchdog timer reset, the hard reset (HRST) bit in the RSR is cleared and the software reset (SRST) bit in the RSR is set to 1 to indicate that a software watchdog timeout caused the previous reset.

NOTE

The levels of the $\overline{IPLx}$ pins are not sampled during a software watchdog reset. If the port size and acknowledge features of the global chip-select are different from the values programmed in the Chip-Select Control Register 0 (CSCR0) at the time of the software watchdog reset, you must assert $\overline{RSTI}$ during software watchdog reset to cause the IPLx/IRQx pins to be resampled.