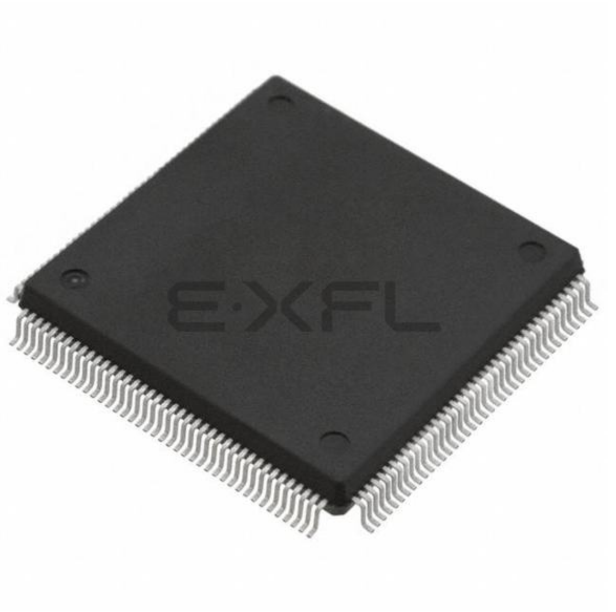


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	Coldfire V2
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	54MHz
Co-Processors/DSP	-
RAM Controllers	DRAM
Graphics Acceleration	-
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	3.3V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	160-BQFP
Supplier Device Package	160-QFP (28x28)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mcf5206eft54

LIST OF TABLES (Continued)

Figure Number	Title	Page Number
14-1.	Programming Model for Timers	14-3
15-1.	Processor PST Definition	15-2
15-2.	CPU-Generated Message Encoding	15-7
15-3.	BDM Command Summary	15-7
15-4.	BDM Size Field Encoding	15-8
15-5.	Control Register Map	15-23
15-6.	Definition of DRc Encoding - Read	15-25
15-7.	Definition of DRc Encoding - Write	15-26
15-8.	SIZE Encodings	15-29
15-9.	Transfer Type Encodings	15-29
15-10.	Transfer Modifier Encodings for Normal Transfers	15-30
15-11.	Transfer Modifier Encodings for Alternate Access Transfers	15-30
15-12.	Core Address, Access Size, and Operand Location	15-30
15-13.	DDATA, CSR[31:28] Breakpoint Response	15-36
15-14.	Shared BDM/Breakpoint Hardware	15-37
16-1.	JTAG Pin Descriptions	16-3
16-2.	JTAG Instructions	16-3
17-1.	Supply, Input Voltage and Storage Temperature	17-1
17-2.	Operating Temperature	17-1
17-3.	Thermal Resistance	17-2
17-4.	Output Loading	17-2
17-5.	DC Electrical Specifications	17-3
17-6.	I/O Driver Capability	17-4
17-7.	Clock Input Timing Specifications	17-5
17-8.	Processor Bus Input Timing Specifications	17-6
17-9.	Processor Bus Output Timing Specifications	17-8
17-10.	Timer Module AC Timing Specifications	17-16
17-11.	UART Module AC Timing Specifications	17-17
17-12.	INPUT Timing Specifications Between SCL and SDA	17-18
17-13.	Output Timing Specifications Between SCL and SDA	17-19
17-14.	Timing Specifications Between CLK and SCL, SDA	17-19
17-15.	General-Purpose I/O Port AC Timing Specifications	17-21
17-16.	IEEE 1149.1 (JTAG) AC Timing Specifications	17-22
18-1.	MCF5206e Package/Frequency Availability	18-3
18-2.	MCF5206e Documentation	18-3

INSTRUCTION	OPERAND SYNTAX	OPERAND SIZE	OPERATION
SUBX	Dy,Dx	32	Destination – Source – X → Destination
SWAP	Dx	16	MSW of Dx ←→ LSW of Dx

TRAP	none	none	SP – 4 → SP;PC → (SP);
			SP – 2 → SP;SR → (SP);
			SP – 2 → SP; Format → (SP);
			Vector Address → PC

TRAPF	none #<data>	none 16 32	PC + 2 → PC PC + 4 → PC PC + 6 → PC
TST	<ea>y	8,16,32	Set Condition Codes
UNLK	Ax	32	Ax →SP; (SP) → Ax; SP + 4 → SP
WDDATA	<ea>y	8,16,32	<ea>y →DDATA port
WDEBUG	<ea>y	2 x 32	<ea>y → Debug Module

1.3.2 MAC Module

The MAC unit provides a common set of simple DSP operations and speeds the execution of the integer multiply instructions in the ColdFire core. It provides functionality in three related areas: faster multiplications of signed and unsigned operands; and new miscellaneous register operations. Multiplies of 16x16 and 32x32 with 32-bit accumulates are supported. The MAC has a single clock issue for 16x16 multiplies and implements a 3-stage execution pipeline.

1.3.3 Hardware Divide Module

The MCF5206e processor includes a hardware divider which performs a number of interger divide operations. The supported divide functions include: 32/16 producing a 16-bit quotient and 16-bit remainder, 32/32 producing a 32-bit quotient, and 32/32 producing a 32-bit remainder.

The hardware divide function provides enhanced functionality, particularly in printing applications. With graphics-based printing this has resulted in as much as 15 percent performance improvement.

1.3.4 Instruction Cache

The instruction cache improves system performance by providing cached instructions to the execution unit in a single clock. The MCF5206e processor uses a 4K-byte, direct-mapped instruction cache to achieve 50 MIPS at 54MHz. The cache is accessed by physical addresses, where each 16-byte line consists of an address tag and a valid bit.

The instruction cache also includes a bursting interface for 32 bit, 16 bit, and 8 bit port sizes to quickly fill cache lines.

1.3.5 Internal SRAM

The 8 KByte on-chip SRAM provides one clock-cycle access for the ColdFire core. This SRAM can store processor stack and critical code or data segments to maximize performance.

Table 3-2. Format Field Encodings

ORIGINAL A7 @ TIME OF EXCEPTION, BITS 1:0	A7 @ 1ST INSTRUCTION OF HANDLER	FORMAT FIELD
00	Original A7 - 8	4
01	Original A7 - 9	5
10	Original A7 - 10	6
11	Original A7 - 11	7

- There is a 4-bit fault status field, FS[3:0], at the top of the system stack. This field is defined for access and address errors only and written as zeros for all other types of exceptions. See Table 3-3.

Table 3-3. Fault Status Encodings

FS[3:0]	DEFINITION
00xx	Reserved
0100	Error on instruction fetch
0101	Reserved
011x	Reserved
1000	Error on operand write
1001	Attempted write to write-protected space
101x	Reserved
1100	Error on operand read
1101	Reserved
111x	Reserved

- The 8-bit vector number, vector[7:0], defines the exception type and is calculated by the processor for all internal faults and represents the value supplied by the peripheral in the case of an interrupt. Refer to Table 3-1.

3.5 PROCESSOR EXCEPTIONS

3.5.1 Access Error Exception

The exact processor response to an access error depends on the type of memory reference being performed. For an instruction fetch, the processor postpones the error reporting until the faulted reference is needed by an instruction for execution. Therefore, faults that occur during instruction prefetches that are then followed by a change of instruction flow do not generate an exception. When the processor attempts to execute an instruction with a faulted opword and/or extension words, the access error is signaled and the instruction aborted. For this type of exception, the programming model has not been altered by the instruction generating the access error.

If the access error occurs on an operand read, the processor immediately aborts the current instruction's execution and initiates exception processing. In this situation, any address

3.9 MISCELLANEOUS INSTRUCTION EXECUTION TIMES

Table 3-9. Miscellaneous Instruction Execution Times

OPCODE	<EA>	EFFECTIVE ADDRESS							
		RN	(AN)	(AN)+	-(AN)	(D16,AN)	(D8,AN,XN*SF)	XXX.WL	#XXX
LINK.W	Ay,#imm	2(0/1)	—	—	—	—	—	—	—
MOVE.W	CCR,Dx	1(0/0)	—	—	—	—	—	—	—
MOVE.W	<ea>,CCR	1(0/0)	—	—	—	—	—	—	1(0/0)
MOVE.W	SR,Dx	1(0/0)	—	—	—	—	—	—	—
MOVE.W	<ea>,SR	7(0/0)	—	—	—	—	—	—	7(0/0) [2]
MOVEC	Ry,Rc	9(0/1)	—	—	—	—	—	—	—
MOVEM.L	<ea>,&list	—	1+n(n/0)	—	—	1+n(n/0)	—	—	—
MOVEM.L	&list,<ea>	—	1+n(0/n)	—	—	1+n(0/n)	—	—	—
NOP		3(0/0)	—	—	—	—	—	—	—
PEA	<ea>	—	2(0/1)	—	—	2(0/1) [4]	3(0/1) [5]	2(0/1)	—
PULSE		1(0/0)	—	—	—	—	—	—	—
STOP	#imm	—	—	—	—	—	—	—	3(0/0) [3]
TRAP	#imm	—	—	—	—	—	—	—	15(1/2)
TRAPF		1(0/0)	—	—	—	—	—	—	—
TRAPF.W		1(0/0)	—	—	—	—	—	—	—
TRAPF.L		1(0/0)	—	—	—	—	—	—	—
UNLK	Ax	2(1/0)	—	—	—	—	—	—	—
WDDATA	<ea>	—	3(1/0)	3(1/0)	3(1/0)	3(1/0)	4(1/0)	3(1/0)	3(1/0)
WDEBUG	<ea>	—	5(2/0)	—	—	5(2/0)	—	—	—

n is the number of registers moved by the MOVEM opcode.

[1]£ indicates that long multiplies have early termination after 9 cycles; thus, actual cycle count is operand independent

[2]If a MOVE.W #imm,SR instruction is executed and imm[13] = 1, the execution time is 1(0/0).

[3]The execution time for STOP is the time required until the processor begins sampling continuously for interrupts.

[4]PEA execution times are the same for (d16,PC)

[5] PEA execution times are the same for (d8,PC,Xn*SF)

- Addresses not assigned to the registers and undefined register bits are reserved for future expansion. Write accesses to these reserved address spaces and reserved register bits have no effect; read accesses will return zeros.
- The reset value column indicates the initial value of the register at reset. Certain registers may be uninitialized upon reset, i.e., they may contain random values after reset.
- The access column indicates if the corresponding register allows both read/write functionality (R/W), read-only functionality (R), or write-only functionality (W). If a read access to a write-only register is attempted, zeros will be returned. If a write access to a read-only register is attempted the access will be ignored and no write will occur.

Table 4-3. Memory Map of I-Cache Registers

ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE	ACCESS
MOVEC with $002	CACR	32	Cache Control Register	$0000	W
MOVEC with $004	ACR0	32	Access Control Register 0	$0000	W
MOVEC with $005	ACR1	32	Access Control Register 1	$0000	W

4.4.2 Instruction Cache Register

4.4.2.1 CACHE CONTROL REGISTER (CACR). The CACR controls the operation of the instruction cache. The CACR provides a set of default memory access attributes used when a reference address does not map into the spaces defined by the ACRs.

The CACR is a 32-bit write-only supervisor control register. It is accessed in the CPU address space via the MOVEC instruction with an Rc encoding of $002. The CACR can be read when in Background Debug mode (BDM). At system reset, the entire register is cleared.

Cache Control Register (CACR)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CENB	-	-	CPDI	CFRZ	-	-	CINV	-	-	-	-	-	-	-	-
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CEIB	DCM	DBWE	-	-	DWP	-	-	-	CLNF1	
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6-7. Data Bus Requirement for Read Cycles

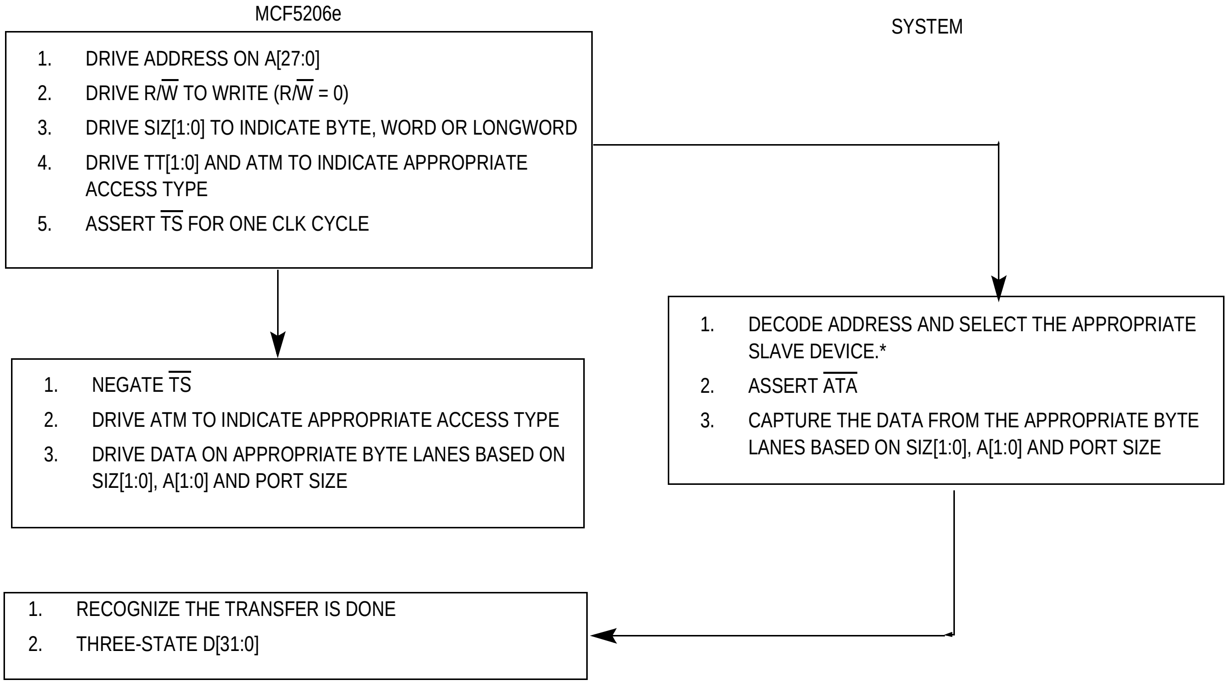
TRANSFER SIZE	SIZE		ADDRESS		32 BIT PORT EXTERNAL DATA BYTES REQUIRED				16 BIT PORT EXTERNAL DATA BYTES REQUIRED		8 BIT PORT EXTERNAL DATA BYTES REQUIRED
	SIZ[1]	SIZ[0]	A[1]	A[0]	D[31:24]	D[23:16]	D[15:8]	D[7:0]	D[31:24]	D[23:16]	D[31:24]
BYTE	0	1	0	0	Byte 0	X	X	X	Byte 0	X	Byte 0
			0	1	X	Byte 1	X	X	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
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

Clock 3 (C3)

At the end of C3, the MCF5206e 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 MCF5206e continues to sample internal asynchronous transfer acknowledge and inserts wait states instead of terminating the transfer. The MCF5206e continues to sample internal asynchronous transfer acknowledge until it is asserted. As long as $\overline{\text{ATA}}$ is asserted prior to the falling edge of C2, internal asynchronous transfer acknowledge is 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.


MCF5206e
SYSTEM
1. DRIVE ADDRESS ON A[27:0]
2. DRIVE R/$\overline{\text{W}}$ TO WRITE (R/$\overline{\text{W}}$ = 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 $\overline{\text{TS}}$ FOR ONE CLK CYCLE
1. DECODE ADDRESS AND SELECT THE APPROPRIATE SLAVE DEVICE.*
2. ASSERT $\overline{\text{ATA}}$
3. CAPTURE THE DATA FROM THE APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
1. NEGATE $\overline{\text{TS}}$
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. RECOGNIZE THE TRANSFER IS DONE
2. THREE-STATE D[31:0]

*TO INSERT WAIT STATES, $\overline{\text{ATA}}$ IS DRIVEN NEGATED.

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

Clock 3 (C3)

At the end of C3, the MCF5206e 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 MCF5206e continues to sample internal asynchronous transfer acknowledge and inserts wait states instead of terminating the transfer. The MCF5206e continues to sample internal asynchronous transfer acknowledge until it is asserted. As long as $\overline{\text{ATA}}$ is asserted prior to the falling edge of C2, internal asynchronous transfer acknowledge is 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)

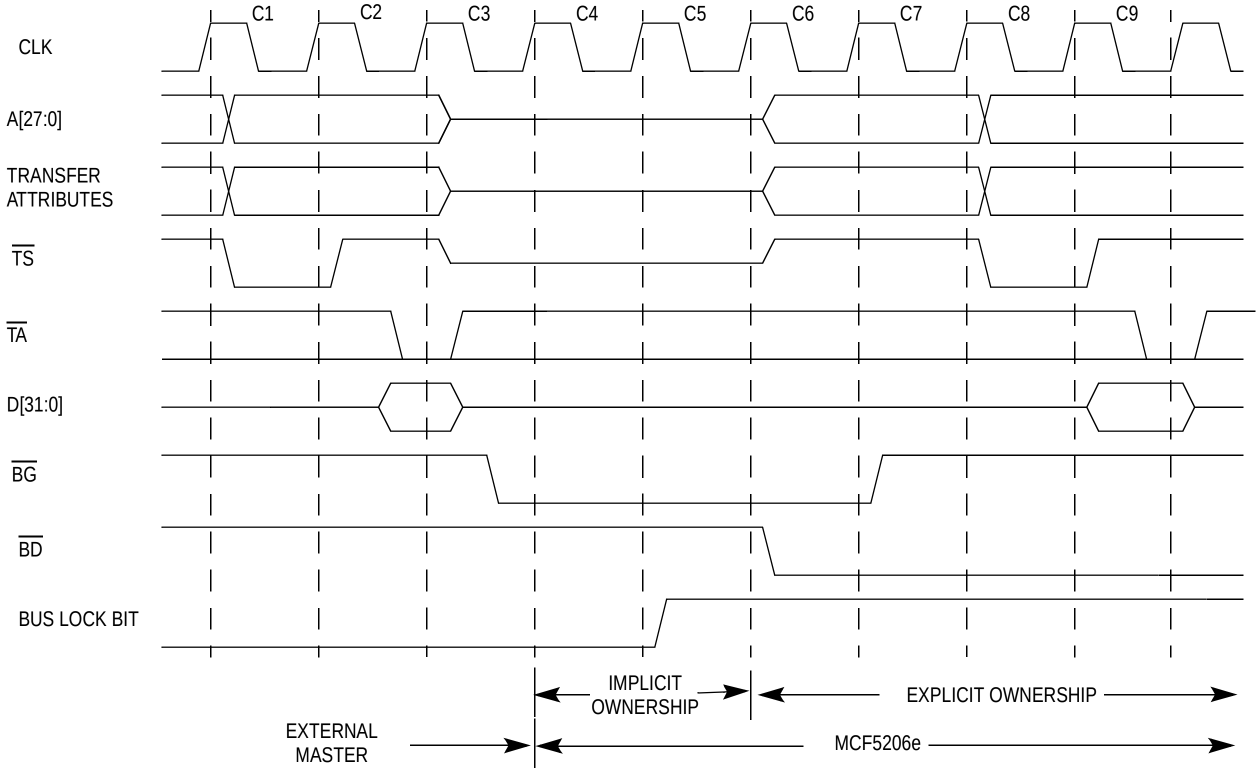
This clock is identical to C2.

Clock 6 (C6)

This clock is identical to C3 except once internal $\overline{\text{ATA}}$ 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.



then an internal bus request being generated.

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

In Figure 6-34, the external master has ownership of the external bus during Clock 1 (C1) and Clock 2 (C2). In Clock 3 (C3) the external master releases control of the bus by asserting bus grant ($\overline{BG}$) to the MCF5206e. During Clock 4 (C4) and Clock 5 (C5) the MCF5206e 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 MCF5206e to take explicit ownership of the bus in Clock 6 (C6) by asserting $\overline{BD}$. In Clock 7 (C7) the external master removes the bus grant to the MCF5206e. Because the bus lock bit is set to 1, the MCF5206e does not relinquish the bus (the MCF5206e continues to assert $\overline{BD}$).

NOTE

> The MCF5206e can start a transfer during the CLK cycle after $\overline{BG}$ is asserted. The external master should not assert $\overline{BG}$ to the MCF5206e until it has stopped driving the bus. $\overline{BG}$ cannot be asserted while the external master transfer is still in progress or damage to the part could occur.

NOTES
1)“N” means negated; “A” means asserted; “EM” means external 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 6-11. MCF5206e 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
EM Own	No	Not Driven	Negated

The MCF5206e can be in any one of four arbitration states during bus operation: reset, external master ownership, implicit ownership, and explicit ownership.

The MCF5206e 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 MCF5206e proceeds to the implicit ownership state or external master ownership state, depending on $\overline{\text{BG}}$.

The external master ownership state denotes the MCF5206e does not have ownership ($\overline{\text{BG}}$ negated) of the bus and the MCF5206e does not drive the bus. The MCF5206e 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 MCF5206e owns the bus because $\overline{\text{BG}}$ is asserted to it. The MCF5206e, 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 MCF5206e keeps the bus three-stated until an internal bus request occurs or the bus lock bit in the SIMR is set to 1.

The MCF5206e 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 MCF5206e asserts $\overline{\text{BD}}$ in this state to indicate the MCF5206e has explicit ownership of the bus. Until $\overline{\text{BG}}$ is negated, the MCF5206e 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 MCF5206e relinquishes the bus at the end of the current bus cycle. When the MCF5206e 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 MCF5206e 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

Table 6-12. MCF5206e Three-Wire Bus Arbitration Protocol Transition Conditions

PRESENT STATE	CONDITION LABEL	RSTI	SOFTWARE WATCHDOG RESET	BG	BUS LOCK BIT	INTERNAL BUS REQUEST (IBR)	TRANSFER IN PROGRESS	END OF CYCLE	NEXT STATE
RESET	A1	A	-	-	-	-	-	-	Reset
	A2	N	A	-	-	-	-	-	Reset
	A3	N	N	N	-	-	-	-	EM Own
	A4	N	N	A	-	-	-	-	Implicit Own
IMPLICIT OWN	B1	N	N	N	-	-	-	-	EM Own
	B2	N	N	A	A	-	-	-	Explicit Own
	B3	N	N	A	N	N	-	-	Implicit Own
	B4	N	N	A	-	A	-	-	Explicit Own
EXPLICIT OWN	C1	N	N	A	-	-	-	-	Explicit Own
	C2	N	N	N	Y	-	-	-	Explicit Own
	C3	N	N	N	N	-	N	-	EM Own
	C4	N	N	N	-	-	Y	N	Explicit Own
	C5	N	N	N	N	-	Y	Y	EM Own
EM OWN	D1	N	N	N	-	-	-	-	EM 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; “EM” means external 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)$\overline{\text{IBR}}$ refers to an internal bus request.The output signals $\overline{\text{BR}}$ is a registered version of $\overline{\text{IBR}}$ when $\overline{\text{BG}}$ is negated and $\overline{\text{BD}}$ is negated. There is an internal bus request when the Coldfire core requires the external bus for an operand transfer.

Table 6-13. MCF5206e 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
EM Own	No	Not Driven	Negated

The MCF5206e can be in any one of four arbitration states during bus operation: reset, external master own, implicit ownership, and explicit ownership.

The reset state is entered 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 MCF5206e proceeds to the implicit ownership state or external master ownership state, depending on bus grant ($\overline{\text{BG}}$).

The external master ownership state denotes the MCF5206e does not have ownership (bus grant ($\overline{\text{BG}}$) negated) of the bus and the MCF5206e does not drive the bus. The

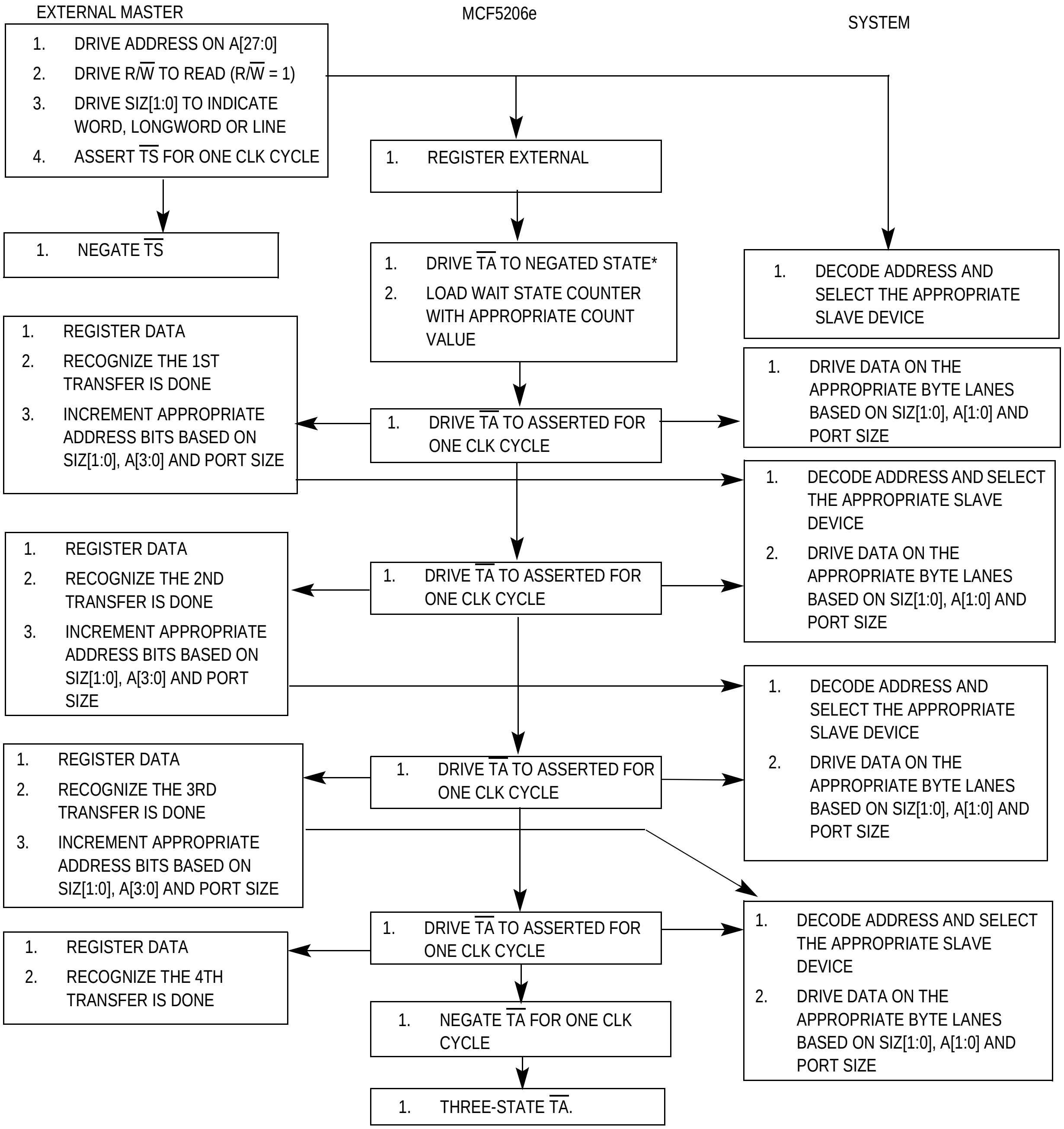

EXTERNAL MASTER
MCF5206e
SYSTEM
1. DRIVE ADDRESS ON A[27:0]
2. DRIVE R/W TO READ (R/W = 1)
3. DRIVE SIZ[1:0] TO INDICATE WORD, LONGWORD OR LINE
4. ASSERT TS FOR ONE CLK CYCLE
1. REGISTER EXTERNAL
1. NEGATE TS
1. DRIVE TA TO NEGATED STATE*
2. LOAD WAIT STATE COUNTER WITH APPROPRIATE COUNT VALUE
1. DECODE ADDRESS AND SELECT THE APPROPRIATE SLAVE DEVICE
1. REGISTER DATA
2. RECOGNIZE THE 1ST TRANSFER IS DONE
3. INCREMENT APPROPRIATE ADDRESS BITS BASED ON SIZ[1:0], A[3:0] AND PORT SIZE
1. DRIVE TA TO ASSERTED FOR ONE CLK CYCLE
1. DRIVE DATA ON THE APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
1. DECODE ADDRESS AND SELECT THE APPROPRIATE SLAVE DEVICE
2. DRIVE DATA ON THE APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
1. REGISTER DATA
2. RECOGNIZE THE 2ND TRANSFER IS DONE
3. INCREMENT APPROPRIATE ADDRESS BITS BASED ON SIZ[1:0], A[3:0] AND PORT SIZE
1. DRIVE TA TO ASSERTED FOR ONE CLK CYCLE
1. DECODE ADDRESS AND SELECT THE APPROPRIATE SLAVE DEVICE
2. DRIVE DATA ON THE APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
1. REGISTER DATA
2. RECOGNIZE THE 3RD TRANSFER IS DONE
3. INCREMENT APPROPRIATE ADDRESS BITS BASED ON SIZ[1:0], A[3:0] AND PORT SIZE
1. DRIVE TA TO ASSERTED FOR ONE CLK CYCLE
1. DECODE ADDRESS AND SELECT THE APPROPRIATE SLAVE DEVICE
2. DRIVE DATA ON THE APPROPRIATE BYTE LANES BASED ON SIZ[1:0], A[1:0] AND PORT SIZE
1. REGISTER DATA
2. RECOGNIZE THE 4TH TRANSFER IS DONE
1. DRIVE TA TO ASSERTED FOR ONE CLK CYCLE
1. NEGATE TA FOR ONE CLK CYCLE
1. THREE-STATE TA.

Figure 6-45. External Master Bursting Read Transfer Using MCF5206e-Generated Transfer-Acknowledge Flowchart

6-49 is a functional timing diagram of the master reset operation, illustrating relationships among Vcc, RSTI, HIZ, 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 MCF5206e caused by the random manner in which internal flip-flops power up. RSTI and HIZ are internally synchronized on consecutive rising and falling clocks before being used and must meet the specified setup and hold times to the falling edge of he clock only if recognition by a specific CLK falling edge is required.

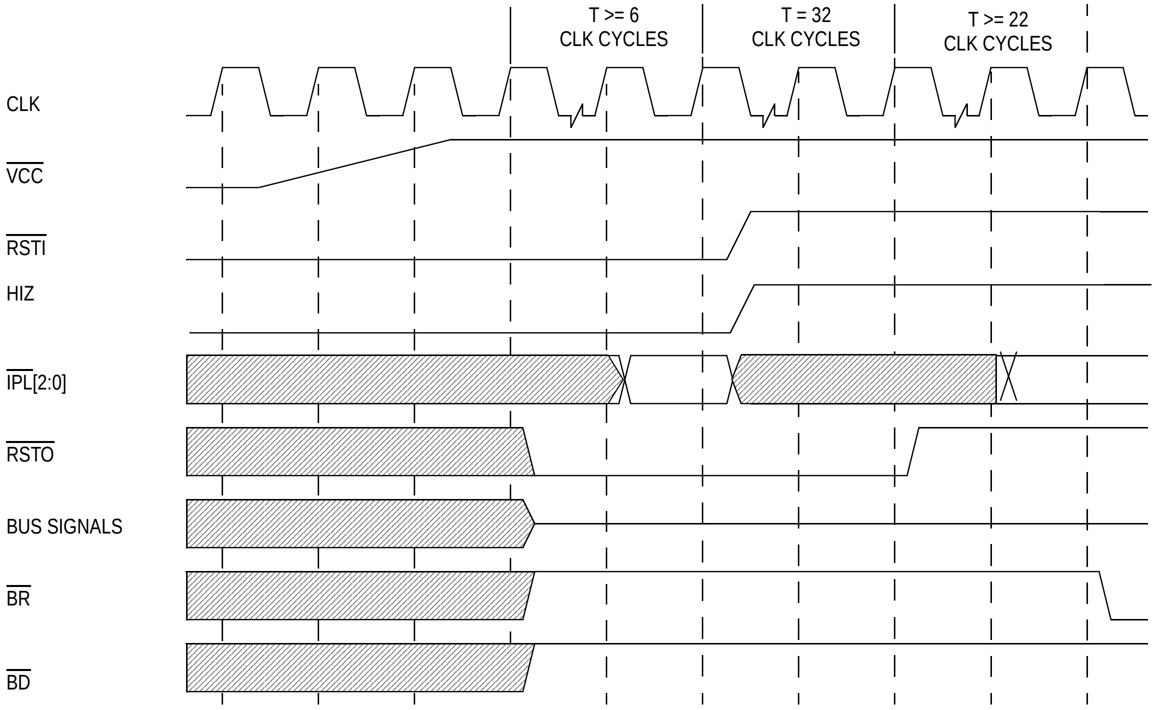


Figure 6-49. Master Reset Timing

TS must be pulled up or negated during master reset. When the assertion of RSTI is recognized internally, the MCF5206e asserts the reset out pin (RSTO). RSTO is asserted as long as RSTI is asserted and remains asserted for 32 CLK cycles after RSTI is negated. For proper master reset operation, 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 RSTI negates, all bus signals continue to remain in a high-impedance state until the MCF5206e 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 Reset Status Register (RSR) is cleared to indicate that an external hardware reset caused the previous reset.

Bit 7—Reserved

CE—Configuration Error

A configuration error results when either the number of bytes represented by the BCR is not consistent with the requested source or destination transfer size, or the SAR or DAR contains an address that does not match the requested transfer size for the source or destination, respectively. The bit is cleared during a hardware reset, or by writing a logic one to the DONE bit of the DSR.

1 = A configuration error has occurred.
0 = No configuration error exists.

BES—Bus Error on Source

1 = The DMA channel has terminated with a bus error either during the read portion of a transfer or during an access in single address mode (SAA = 1).
0 = No bus error has occurred.

BED—Bus Error on Destination

1 = The DMA channel has terminated with a bus error during the write portion of a transfer.
0 = No bus error has occurred.

Bit 3 —Reserved

REQ—Request

1 = The DMA channel has transfers remaining and the channel is not selected
0 = There is no request pending or the channel is currently active. The bit is cleared when the channel is selected.

BSY—Busy

1 = This bit is set the first time the channel is enabled after a transfer is initiated.
0 = DMA channel is not active. This bit is cleared to 0 when the DMA has finished the last transaction.

DONE—Transaction Done

This bit may be read or written and is set when all DMA Controller Module transactions have completed normally, as determined by the transfer count or error conditions. When the BCR reaches zero, DONE is asserted by the DMA or a peripheral device at the successful conclusion of the final DMA Controller Module transfer.

This bit is written with a 1 to reset the DMA control/state bits and can be used in an interrupt handler to clear the DMA interrupt and the DONE and error bits. It can also be used to kill a transfer in progress by resetting state bits. Writing a 0 to this location has no affect.

1= DMA transfer is complete.
0= Writing or reading a 0 at this bit location has no affect.

loaded with the address of the peripheral data register. This address may be any byte address. In the single-address mode, this register is not used.

The manner in which the SAR and DAR change after each cycle depends on the values in the DCR SSIZE and DSIZE fields and the SINC and DINC bits, and the starting address in the SAR and DAR. If programmed to increment, the increment value is 1, 2, 4, or 16 for byte, word, longword, or line operands, respectively. If the address register is programmed to remain unchanged (no count), the register is not incremented after the operand transfer.

The BCR must be loaded with the number of byte transfers that are to occur. This register is decremented by 1, 2, 4, or 16 at the end of each transfer. The DSR must be cleared for channel startup.

Once the channel has been initialized, it is started by writing a one to the START bit in the DCR or asserting the $\overline{\text{DREQ}}$ signal, depending on the status of the EEXT bit in the DCR. Programming the channel for internal request causes the channel to request the bus and start transferring data immediately. If the channel is programmed for external request, $\overline{\text{DREQ}}$ must be asserted before the channel requests the bus.

If any fields in the DCR are modified while the channel is active, that change is effective immediately. To avoid any problems with changing the setup for the DMA channel, a 1 should be written to the DONE bit in the DSR to stop the DMA channel.

7.7.2 Data Transfers

7.7.2.1 EXTERNAL DMA REQUEST OPERATION. Each channel has the feature of interfacing to an external device to initiate transfers to the device. If the EEXT bit is set, when the $\overline{\text{DREQ}}$ signal asserts, the DMA initiates a transfer provided the channel is idle. If the CS (cycle steal) bit is set, then a single read/write transfer occurs. If the CS bit is clear, multiple read/write transfers occur as programmed. The $\overline{\text{DREQ}}$ signal is not required to be negated until the DONE bit of the DSR asserts. In cycle-steal mode, the maximum length of $\overline{\text{DREQ}}$ assertion to maintain a single transfer depends on configuration. In the worst case of a single-address access, byte accesses, and idle channels, $\overline{\text{DREQ}}$ may be asserted for no more than five rising clock edges (see Figure 7-5).

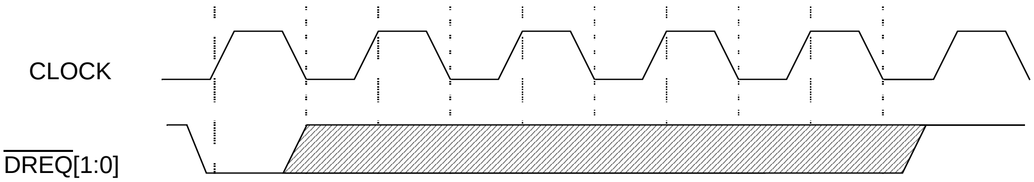


Figure 7-5. External Request Timing - Cycle Steal Mode, Single-Address Mode

See Figure 7-7 for timing relationships for a dual-address transfer using cycle-steal mode. The maximum assertion time for $\overline{\text{DREQ}}$ in this configuration is eight clocks.

NOTE

The UIMR does not mask reading of the UISR. True status is provided regardless of the contents of UIMR. A UART module reset clears the contents of UISR.

UISR MBAR + $194

7	6	5	4	3	2	1	0
COS	—	—	—	—	DB	RXRDY	TXRDY
RESET:							
0	0	0	0	0	0	0	0

READ ONLY SUPERVISOR OR USER

COS — Change-of-State

1 = A change-of-state has occurred at the $\overline{\text{CTS}}$ input and has been selected to cause an interrupt by programming bit 0 of the UACR.

0 = COS bit in the UIPCR is not selected.

DB — Delta Break

1 = The receiver has detected the beginning or end of a received break.

0 = No new break-change condition to report. Refer to **Section 12.4.1.5 Command Register (UCR)** for more information on the reset break-change interrupt command.

RxRDY — Receiver Ready or FIFO Full

UMR1 bit 6 programs the function of this bit. It is a duplicate of either the FFULL or RxRDY bit of USR.

TxRDY — Transmitter Ready

This bit is the duplication of the TxRDY bit in USR.

1 = The transmitter holding register is empty and ready to be loaded with a character.

0 = The CPU loads the transmitter-holding register or the transmitter is disabled. Characters loaded into the transmitter-holding register when TxRDY=0 are not transmitted.

12.4.1.11 INTERRUPT MASK REGISTER (UIMR). The UIMR selects the corresponding bits in the UISR that cause an interrupt. By setting the bit, the interrupt is enabled. If one of the bits in the UISR is set and the corresponding bit in the UIMR is also set, the internal interrupt output is asserted. If the corresponding bit in the UIMR is zero,

Command Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$2				$C				$8				DRC			
DATA [31:16]															
DATA [15:0]															

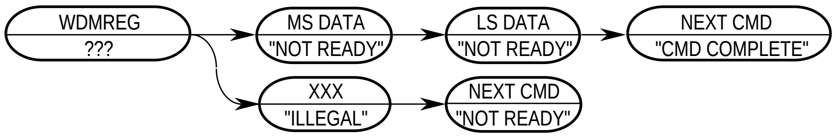
WDMREG BDM Command

DRc encoding:

Table 15-7. Definition of DRc Encoding - Write

DRc[3:0]	DEBUG REGISTER DEFINITION	MNEMONIC	INITIAL STATE
$0	Configuration/Status	CSR	$0
$1-$5	Reserved	-	–
$6	Bus Attributes And Mask	AATR	$0005
$7	Trigger Definition	TDR	$0
$8	PC Breakpoint	PBR	–
$9	PC Breakpoint Mask	PBMR	–
$A-$B	Reserved	–	–
$C	Operand Address High Breakpoint	ABHR	–
$D	Operand Address Low Breakpoint	ABLR	–
$E	Data Breakpoint	DBR	–
$F	Data Breakpoint Mask	DBMR	–

Command Sequence:



Operand Data:

Longword data is written into the specified debug register. The data is supplied most significant word first.

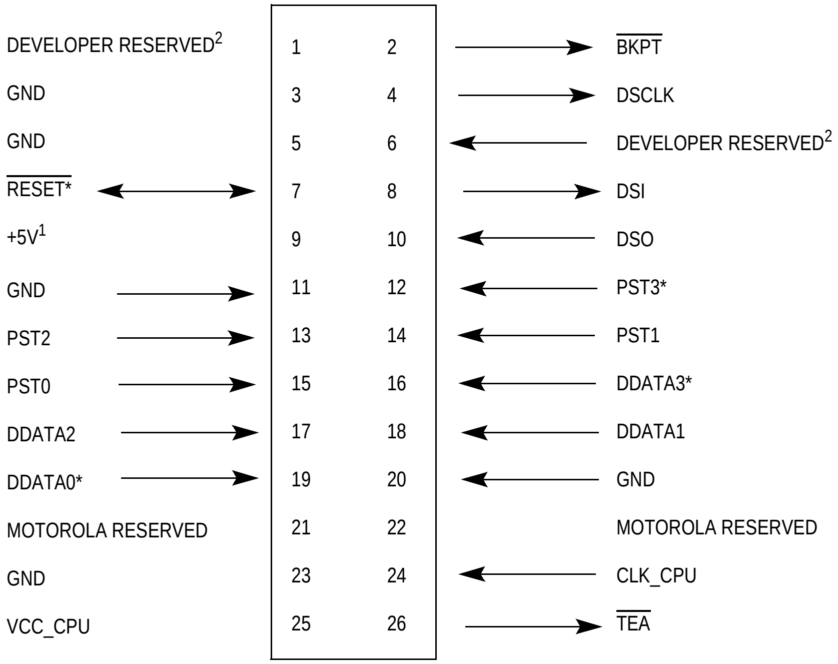
Result Data:

Command complete status ($0FFFF) is returned when register write is complete.

15.2.3.4.13 Unassigned Opcodes. Motorola reserves unassigned command opcodes. All unused command formats within any revision level perform a NOP and return the ILLEGAL command response*.*

15.4 MOTOROLA RECOMMENDED BDM PINOUT

The ColdFire BDM connector is a 26-pin Berg connector arranged 2x13, shown in Figure 15-6.



DEVELOPER RESERVED[2]		1	2	→	$\overline{\text{BKPT}}$
GND		3	4	→	DSCLK
GND		5	6	←	DEVELOPER RESERVED[2]
$\overline{\text{RESET}}$*	↔	7	8	→	DSI
+5V[1]		9	10	←	DSO
GND	→	11	12	←	PST3*
PST2	→	13	14	←	PST1
PST0	→	15	16	←	DDATA3*
DDATA2	→	17	18	←	DDATA1
DDATA0*	→	19	20	←	GND
MOTOROLA RESERVED		21	22		MOTOROLA RESERVED
GND		23	24	←	CLK_CPU
VCC_CPU		25	26	→	$\overline{\text{TEA}}$

NOTES:
1. Supplied by target
2. Pins reserved for BDM developer use. Contact developer.
* Denotes a vectored signal

Figure 15-7. 26-Pin Berg Connector Arranged 2x13

15.4.1 Differences Between the ColdFire BDM and a CPU32 BDM

1. DSCLK, BKPT, and DSDI must meet the setup and hold times relative to the rising edge of the processor clock to prevent the processor from propagating metastable states.
2. DSO transitions relative to the rising edge of DSCLK only. In the CPU32 BDM, DSO transitions between serial transfers to indicate to the development system that a command has successfully completed. The ColdFire BDM does not support this feature.
3. The development system must note that the DSO is not valid during the first rising edge of DSCLK. Instead, the first rising edge of DSCLK causes DSO to transmit the MSB of DSO. A serial transfer is illustrated in Figure 15-8.

also configure the direction of bidirectional pins and establish high-impedance states on some pins. The EXTEST instruction becomes active on the falling edge of TCK in the update-IR state when the data held in the instruction-shift register is equivalent to octal 0.

16.3.1.2 IDCODE. The IDCODE instruction selects the 32-bit IDcode register for connection as a shift path between the TDI pin and the TDO pin. This instruction lets you interrogate the MCF5206e to determine its version number and other part identification data. The IDcode register has been implemented in accordance with IEEE 1149.1 so that the least significant bit of the shift register stage is set to logic 1 on the rising edge of TCK following entry into the capture-DR state. Therefore, the first bit to be shifted out after selecting the IDcode register is always a logic 1. The remaining 31-bits are also set to fixed values (see 16.3.2 IDCode Register) on the rising edge of TCK following entry into the capture-DR state.

The IDCODE instruction is the default value placed in the instruction register when a JTAG reset is accomplished by either asserting $\overline{\text{TRST}}$ or holding TMS high while clocking TCK through at least five rising edges and the falling edge after the fifth rising edge. A JTAG reset causes the TAP state machine to enter the test-logic-reset state (normal operation of the TAP state machine into the test-logic-reset state also results in placing the default value of octal 1 into the instruction register). The shift register portion of the instruction register is loaded with the default value of octal 1 when in the Capture-IR state and a rising edge of TCK occurs.

16.3.1.3 SAMPLE/PRELOAD INSTRUCTION. The SAMPLE/PRELOAD instruction provides two separate functions. First, it obtains a sample of the system data and control signals present at the MCF5206e input pins and just prior to the boundary scan cell at the output pins. This sampling occurs on the rising edge of TCK in the capture-DR state when an instruction encoding of octal 4 is resident in the instruction register. You can observe this sampled data by shifting it through the boundary-scan register to the output TDO by using the shift-DR state. Both the data capture and the shift operation are transparent to system operation. You are responsible for providing some form of external synchronization to achieve meaningful results because there is no internal synchronization between TCK and the system clock, CLK.

The second function of the SAMPLE/PRELOAD instruction is to initialize the boundary scan register update cells before selecting EXTEST or CLAMP. This is achieved by ignoring the data being shifted out of the TDO pin while shifting in initialization data. The update-DR state in conjunction with the falling edge of TCK can then transfer this data to the update cells. This data will be applied to the external output pins when one of the instructions listed above is applied.

16.3.1.4 HIGHZ INSTRUCTION. The HIGHZ instruction anticipates the need to backdrive the output pins and protect the input pins from random toggling during circuit board testing. The HIGHZ instruction selects the bypass register, forcing all output and bidirectional pins to the high-impedance state.

Introduction 1

Signal Description 2

ColdFire Core 3

Instruction Cache 4

SRAM 5

Bus Operation 6

DMA Controller Module 7

System Integration Module (SIM) 8

Chip Select Module 9

Parallel Port (General-Purpose I/O) 10

DRAM Controller 11

UART Modules 12

MBus Module 13

Timer Module 14

Debug Support 15

IEEE 1149.1 JTAG 16

Electrical Characteristics 17

Mechanical Characteristics 18

Appendix A: MCF5206e Memory Map A

Appendix B: Porting from M68000 B