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	Active
Core Processor	Coldfire V2
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	40MHz
Co-Processors/DSP	-
RAM Controllers	DRAM
Graphics Acceleration	-
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 85°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=mcf5206ecab40

- DMA Controller
  - Two fully programmable channels
  - Supports dual-address and single-address transfers, with 32-bit capability
  - Two address pointers per channel that can increment or remain constant
  - 16-bit transfer counter per channel
  - Operand packing and unpacking supported
  - Auto-alignment transfers supported for efficient block movement
  - Supports bursting and cycle steal
  - Provides two clock-cycle internal access
  - Three request mechanisms: Software via register bits; External $\overline{\text{DREQ}}$; UART interrupts.

- Two Universal Synchronous/Asynchronous Receiver/Transmitter (UART) Modules
  - Full duplex operation
  - Baud-rate generator
  - Modem control signals available ($\overline{\text{CTS}}$, $\overline{\text{RTS}}$)
  - Processor-interrupt capability

- Dual 16-Bit General-Purpose Multimode Timers
  - 8-bit prescaler
  - Timer input and output pins
  - 30 ns resolution with 33 MHz system clock
  - Processor-interrupt capability

- Motorola Bus (M-Bus) Module
  - Interchip bus interface for EEPROMs, LCD controllers, A/D converters, keypads
  - Compatible with industry-standard I^2C Bus
  - Master or slave modes support multiple masters
  - Automatic interrupt generation with programmable level
- System Interface
  - Glueless bus interface to 8 bit, 16 bit, and 32 bit DRAM, SRAM, ROM, and I/O devices
  - 32-bit internal address bus with 28 bit external bus; chip select and DRAM
  - 8 programmable chip selects
  - Programmable wait states and port sizes
  - Allows external bus masters to access chip selects
  - System protection
    - 16-bit software watchdog timer with prescaler
    - Double bus fault monitor
    - Bus timeout monitor
    - Spurious interrupt monitor
  - Programmable interrupt controller
    - Low interrupt latency
    - 3 external interrupt inputs
    - Programmable interrupt priority and autovector generator
  - IEEE 1149.1 test (JTAG) support

The processor employs the user mode and the user programming model when it is in normal processing. During exception processing, the processor changes from user to supervisor mode. Exception processing saves the current SR value on the stack and then sets the S bit, forcing the processor into the supervisor mode. To return to the user mode, a system routine must execute a MOVE to SR, or an RTE, which operate in the supervisor mode, modifying the S bit of the SR. After these instructions execute, the instruction fetch pipeline flushes and is refilled from the appropriate address space.

The registers depicted in the programming model (see Figure 1-2) provide operand storage and control for the ColdFire processor core. The registers are also partitioned into user and supervisor privilege modes. The user programming model consists of 16 general-purpose, 32 bit registers and two control registers. The supervisor model consists of five more registers that can be accessed only by code running in supervisor mode.

Only system programmers can use the supervisor programming model to implement operating system functions and I/O control. This supervisor/user distinction allows for the coding of application software that will run without modification on any ColdFire Family processor. The supervisor programming model contains the control features that system designers would not want user code to erroneously access as this might effect normal system operation. Furthermore, the supervisor programming model may need to change slightly from ColdFire generation to generation to add features or improve performance as the architecture evolves.

2.9.2 Transmit Data (TxD[1], TxD[2])

The UART modules transmit serial data on these outputs. TxD[1] corresponds to UART 1 and TxD[2] corresponds to UART 2. Data is transmitted on the falling edge of the serial clock source, with the least significant bit transmitted (LSB) first. When no data is being transmitted or the transmitter is disabled, these two signals are held high. TxD[1] and TxD[2] are also held high in local loopback mode.

2.9.3 Request To Send ($\overline{\text{RTS}}$[1], $\overline{\text{RTS}}$[2]/$\overline{\text{RSTO}}$)

$\overline{\text{RTS}}$[2] is multiplexed with the $\overline{\text{RSTO}}$ signal. Programming the Pin Assignment Register (PAR) in the SIM determines the function of this pin. During reset, this pin is configured to be $\overline{\text{RSTO}}$.

The request-to-send output indicates to the peripheral device that the UART module is ready to receive data. $\overline{\text{RTS}}$[1] corresponds to UART 1 and $\overline{\text{RTS}}$[2] corresponds to UART 2.

2.9.4 Clear To Send ($\overline{\text{CTS}}$[1], $\overline{\text{CTS}}$[2])

Peripherals drive these inputs to indicate to the UART module that it can begin data transmission. $\overline{\text{CTS}}$[1] corresponds to UART 1 and $\overline{\text{CTS}}$[2] corresponds to UART 2.

2.10 TIMER MODULE SIGNALS

The signal descriptions that follow are the external interface to the two general purpose timer modules (Timer 1 and Timer 2).

2.10.1 Timer Input (TIN[2], TIN[1])

You can program the timer input to be the clock for the timer module. You can also program the timer module to trigger a capture on the rising edge, falling edge, or both edges of the timer input. TIN[1] corresponds to Timer 1 and TIN[2] corresponds to Timer 2. TIN[1] is muxed with DREQ[0]. The reset state of the dual function TIN[1] pin is for timer operation.

2.10.2 Timer Output (TOUT[2], TOUT[1])

The programmable timer output pulses or toggles when the timer reaches the programmed count value. TOUT[1] corresponds to Timer 1 and TOUT[2] corresponds to Timer 2. The reset state of the dual function TOUT[1] pin is for timer operation.

2.11 DMA MODULE SIGNALS

The signal descriptions that follow are the external interface to the two DMA channels (DREQ[0] and DREQ[1]).

2.11.1 DMA Request (DREQ[0], DREQ[1])

DREQ[0] is multiplexed with the TIN[1] pin and DREQ[1] is multiplexed with the TOUT[1] pin. Refer to Figure 2.1. The reset state of the timer/DMA dual function pins is for the timer

These outputs indicate the MCF5206e processor status. During debug mode, the timing is synchronous with the processor clock (CLK) and the status is not related to the current bus transfer. Table 2-11 shows the encodings of PST[3:0].

Table 2-11. Processor Status Encodings

PST[3:0]	DEFINITION
0000	Continue execution
0001	Begin execution of an instruction
0010	Reserved
0011	Entry into user mode
0100	Begin execution of **PULSE** instruction
0101	Begin execution of taken branch
0110	Reserved
0111	Begin execution of **RTE** instruction
1000	Reserved
1001	Reserved
1010	Reserved
1011	Reserved
1100	† Exception processing
1101	† Emulator mode entry exception processing
1110	† Processor is stopped, waiting for interrupt
1111	† Processor is halted
† These encodings are asserted for multiple cycles.	

2.14.2 Debug Data (PP[3:0]/DDATA[3:0])

The debug data signals are multiplexed with general purpose I/O signals. Programming the Pin Assignment Register (PAR) in the SIM determines the function of these pins. During reset, these pins are configured as general purpose inputs.

The DDATA[3:0] outputs display captured processor data and breakpoint status. See **Section 15: Debug Support** section for additional information on this bus.

2.14.3 Development Serial Clock ($\overline{\text{TRST}}$/DSCLK)

The MTMOD signal determines the function of this dual-purpose pin. If MTMOD= 0, the $\overline{\text{TRST}}$ function is selected. If MTMOD=1, the DSCLK function is selected. MTMOD should not be changed while $\overline{\text{RSTI}}$ = 1.

The DSCLK input signal is used as the development serial clock for the serial interface to the debug module.The maximum frequency for the DSCLK signal is 1/2 the CLK frequency. See **Section 15: Debug Support** section for additional information on this signal.

First, the processor makes an internal copy of the SR and then enters supervisor mode by asserting the S bit and disabling trace mode by negating the T bit. The occurrence of an interrupt exception also forces the M bit to be cleared and the interrupt priority mask to be set to the level of the current interrupt request.

Second, the processor determines the exception vector number. For all faults *except* interrupts, the processor performs this calculation based on the exception type. For interrupts, the processor performs an interrupt-acknowledge (IACK) bus cycle to obtain the vector number from a peripheral device. The IACK cycle is mapped to a special acknowledge address space with the interrupt level encoded in the address.

Third, the processor saves the current context by creating an exception stack frame on the system stack. The V2 Core supports a single stack pointer in the A7 address register; therefore, there is no notion of separate supervisor or user stack pointers. As a result, the exception stack frame is created at a 0-modulo-4 address on the top of the current system stack. Additionally, the processor uses a simplified fixed-length stack frame for all exceptions. The exception type determines whether the program counter placed in the exception stack frame defines the location of the faulting instruction (fault) or the address of the next instruction to be executed (next).

Fourth, the processor calculates the address of the first instruction of the exception handler. By definition, the exception vector table is aligned on a 1 Mbyte boundary. This instruction address is generated by fetching an exception vector from the table located at the address defined in the vector base register. The index into the exception table is calculated as (4 x vector number). Once the exception vector has been fetched, the contents of the vector determine the address of the first instruction of the desired handler. After the instruction fetch for the first opcode of the handler has been initiated, exception processing terminates and normal instruction processing continues in the handler.

ColdFire 5200 processors support a 1024-byte vector table aligned on any 1 Mbyte address boundary (see Table 3-1). The table contains 256 exception vectors where the first 64 are defined by Motorola and the remaining 192 are user-defined interrupt vectors.

Clock 2 (C2)

During C2, the MCF5206e 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, which is recognized by the MCF5206e on the next falling clock edge.

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 MCF5206e 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 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 $\overline{ATA}$ is asserted prior to the falling edge of C2, the internal asynchronous transfer acknowledge is asserted by the rising edge of C3.

Clock 4 (C4)

The MCF5206e 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 MCF5206e 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 MCF5206e continues to sample internal $\overline{ATA}$ and inserts wait states instead of terminating the transfer. The MCF5206e continues to sample internal $\overline{ATA}$ on successive falling edges of the 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.

NOTE

$\overline{TA}$ cannot be tied to GND if the MCF5206e is not the only bus master in the system. Damage to the part could occur if $\overline{TA}$ is tied to GND and external master accesses using 5206e generated termination.

6.6 MISALIGNED OPERANDS

All MCF5206e 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 MCF5206e 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 occurs with any attempt to prefetch an instruction word at an odd address.

The MCF5206e 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 MCF5206e 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 MCF5206e 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

Figure 6-48 illustrates $\overline{TA}$ assertion by the MCF5206e during external master bursting write transfers.

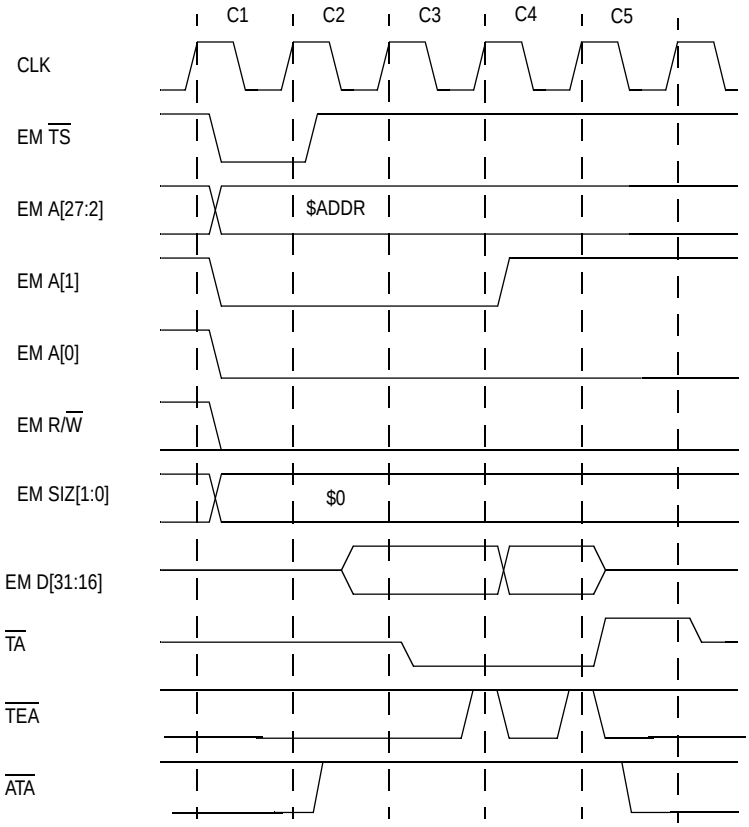


Figure 6-48. External Master Bursting Longword Write Transfer to a 16-Bit Port Using MCF5206e Transfer Acknowledge Timing (No Wait States)

Clock 1 (C1)

The write cycle starts in C1. During C1, the external 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 external master asserts $\overline{TS}$ to indicate the beginning of a bus cycle.

Clock 2 (C2)

At the start of C2, the MCF5206e registers and decodes the external 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 MCF5206e selects the indicated number of wait states for loading into the internal wait state counter. During C2, the external 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.

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 ($\overline{RTS2}$/$\overline{RSTO}$) is asserted by the MCF5206e. $\overline{RSTO}$ is 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 MCF5206e 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 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 Reset Status Register (RSR) is cleared and the software reset (SRST) bit in the Reset Status Register (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 $\overline{IPLx}$/$\overline{IRQx}$ pins to be resampled.

Table 8-8. PAR3 - PAR0 Pin Assignment (Continued)

PAR[3:0]	A27/CS7/WE0	A26/CS6/WE1	A25/CS5/WE2	A24/CS4/WE3
0110	WE0	CS6	A25	A24
0111	WE0	A26	A25	A24
1000	CS7	CS6	CS5	CS4
1001	CS7	CS6	CS5	A24
1010	CS7	CS6	A25	A24
1011	CS7	A26	A25	A24
1100	A27	A26	A25	A24
1101	Reserved			
1110	Reserved			
1111	Reserved			

8.9 BUS ARBITRATION CONTROL

8.9.1 Bus Master Arbitration Control (MARB)

The MARB determines the internal master bus arbitration. It selects the highest master, and can also disable arbitration.

The MPARK is an 8-bit read-write register:

Bus Master Arbitration Control (MARB) Address MBAR + $07

7	6	5	4	3	2	1	0
-	-	-	-	NOARB	ARBCTRL	-	-
RESET: 0	0	0	0	0	0	0	0

Read/Write

Default Bus Master Register (MPARK)

The internal bus master arbiter uses a very simple algorithm for arbitration; arbitration is based solely on priority. For as long as the highest priority master continues to request the internal bus, it receives control of the bus. When the highest priority master relinquishes control of the bus, another master can take full control of the bus, but it would have to relinquish control once a higher priority request was received. No master can preempt an active bus-transaction, however. Once a bus cycle is started, arbitration does not change until that cycle is complete.

For the most part, the operation of the arbiter is invisible. In fact, code written for any V2 ColdFire microprocessor with no other internal masters will work without modification, since the default configuration places the core at the highest priority. Priority must be taken, however, when prioritizing non-core master devices, such as the internal DMA, which must arbitrate for the bus. The arbitration algorithm used could effectively starve any master not

SECTION 9
CHIP SELECT MODULE

9.1 INTRODUCTION

The chip select module provides user-programmable control of the eight chip select and four write enable outputs. This subsection describes the operation and programming model of the chip select registers, including the chip select address, mask, and control registers.

9.1.1 Features

The following list summarizes the key chip select features:

- Eight programmable chip selects
- Address masking for memory block sizes from 64 K- to 2 GBytes
- Programmable wait states and port sizes
- Programmable address setup
- Programmable address hold for read and write
- Programmable wait states and port sizes for default memory
- External master access to chip selects

9.2 CHIP SELECT MODULE I/O

9.2.1 Control Signals

The chip select controller outputs eight chip select and four write enables. The chip select controller activates these signals for ColdFire core-initiated transfers as well as during external master-initiated transfers. The chip select controller can also output transfer acknowledge ($\overline{TA}$) during external master transfers.

9.2.1.1 CHIP SELECT ($\overline{CS}$[7:0]). These active-low output signals provide control for peripherals and memory. $\overline{CS}$[7:4] are multiplexed with upper address signals (A[27:24]) and the write enable ($\overline{WE}$[3:0]) signals. $\overline{CS}$[0] provides the special function of global chip select to let you relocate boot ROM at any defined address space. $\overline{CS}$[1] provides the special function of asserting during CPU space accesses including interrupt acknowledge cycles.

9.2.1.2 WRITE ENABLE ($\overline{WE}$[3:0]). These active-low output signals provide control for peripherals and memory during write transfers. $\overline{WE}$[3:0] are multiplexed with upper address and upper chip selects.

C2, the MCF5206e samples the level of $\overline{TA}$ and if $\overline{TA}$ is asserted, latches the current value of D[31:16]. If $\overline{TA}$ is asserted, the transfer of the first word of the longword is complete. If $\overline{TA}$ is negated, the MCF5206e continues to sample $\overline{TA}$ and inserts wait states instead of terminating the transfer. The MCF5206e continues to sample $\overline{TA}$ on successive rising edge 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 is terminated with an internal bus error.

Clock 3 (C3)

During C3, the MCF5206e increments A[1:0] to indicate the second word in the longword transfer. The selected device(s) places the addressed data onto D[31:16] and asserts the transfer acknowledge ($\overline{TA}$). At the end of C3, the MCF5206e samples the level of $\overline{TA}$ and if $\overline{TA}$ is asserted, latches the current value of D[31:16]. If $\overline{TA}$ is asserted, the transfer of the second word of the longword is complete and the transfer terminates and the chip select ($\overline{CS}$) is negated. If $\overline{TA}$ is negated, the MCF5206e continues to sample $\overline{TA}$ and inserts wait states instead of terminating the transfer. The MCF5206e continues to sample $\overline{TA}$ on successive rising edge 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 is terminated with an internal bus error.

9.3.3.5 BURST TRANSFER WITH ADDRESS SETUP. Figure 9-6 illustrates a longword user code read from a 16-bit port with address setup enabled and read address hold disabled.

Table 11-8. 32-bit Port Size Address Multiplexing Configurations

MCF5206E ADDRESS PIN	PS = 32-BIT BPS = 512 BYTE		MCF5206E ADDRESS PIN	PS = 32-BIT BPS = 1 KBYTE		MCF5206E ADDRESS PIN	PS = 32-BIT BPS = 2 KBYTE	
	ROW ADDRESS	CAS		ROW ADDRESS	CAS		ROW ADDRESS	CAS
A[27]	IA[27]	IA[26]	A[27]	IA[27]	IA[26]	A[27]	IA[27]	IA[26]
A[26]	IA[26]	IA[26]	A[26]	IA[26]	IA[26]	A[26]	IA[26]	IA[26]
A[25]	IA[25]	IA[24]	A[25]	IA[25]	IA[24]	A[25]	IA[25]	IA[24]
A[24]	IA[24]	IA[24]	A[24]	IA[24]	IA[24]	A[24]	IA[24]	IA[24]
A[23]	IA[23]	IA[22]	A[23]	IA[23]	IA[22]	A[23]	IA[23]	IA[22]
A[22]	IA[22]	IA[22]	A[22]	IA[22]	IA[22]	A[22]	IA[22]	IA[22]
A[21]	IA[21]	IA[20]	A[21]	IA[21]	IA[20]	A[21]	IA[21]	IA[20]
A[20]	IA[20]	IA[20]	A[20]	IA[20]	IA[20]	A[20]	IA[20]	IA[20]
A[19]	IA[19]	IA[18]	A[19]	IA[19]	IA[18]	A[19]	IA[19]	IA[10]
A[18]	IA[18]	IA[18]	A[18]	IA[18]	IA[18]	A[18]	IA[18]	IA[9]
A[17]	IA[17]	IA[16]	A[17]	IA[17]	IA[9]	A[17]	IA[17]	IA[8]
A[16]	IA[16]	IA[16]	A[16]	IA[16]	IA[8]	A[16]	IA[16]	IA[7]
A[15]	IA[15]	IA[8]	A[15]	IA[15]	IA[7]	A[15]	IA[15]	IA[6]
A[14]	IA[14]	IA[7]	A[14]	IA[14]	IA[6]	A[14]	IA[14]	IA[5]
A[13]	IA[13]	IA[6]	A[13]	IA[13]	IA[5]	A[13]	IA[13]	IA[4]
A[12]	IA[12]	IA[5]	A[12]	IA[12]	IA[4]	A[12]	IA[12]	IA[3]
A[11]	IA[11]	IA[4]	A[11]	IA[11]	IA[3]	A[11]	IA[11]	IA[2]
A[10]	IA[10]	IA[3]	A[10]	IA[10]	IA[2]	A[10]	IA[10]	IA[10]
A[9]	IA[9]	IA[2]	A[9]	IA[9]	IA[9]	A[9]	IA[9]	IA[9]

The BPS field in each DCCR defines the DRAMC internal page size. The internal page size is used by the DRAMC to determine whether a transfer is a page hit or a page miss. The page size of the DRAM used in the bank is not always the same as the DRAMC internal page size. For example, if a 2 KByte page size is selected and an 8-bit wide DRAM is used, 11 address bits and 1 $\overline{\text{CAS}}$ signal are needed to define the page. However, if a 2 KByte page size is selected and a 32-bit wide DRAM is used, only 9 address signals and 4 $\overline{\text{CAS}}$ signals are needed to define the page. Using a DRAM which has a larger page size than is listed in the Actual DRAM Page Size column of Table 11-9 for a given internal page size and port size gives no performance advantage.

To allow for future upgrades to larger DRAMs without requiring multiple printed circuit board layouts, the page size must remain constant. After the page size has been selected, use the tables to determine which address pins to use for the maximum DRAM size. These traces can then be routed to the DRAM socket on the printed circuit board.

Figure 11-13 shows the timing of a word read in Fast Page Mode followed by a page miss word read using 8-bit wide EDO DRAM (the EDO bit in the DCTR is set).

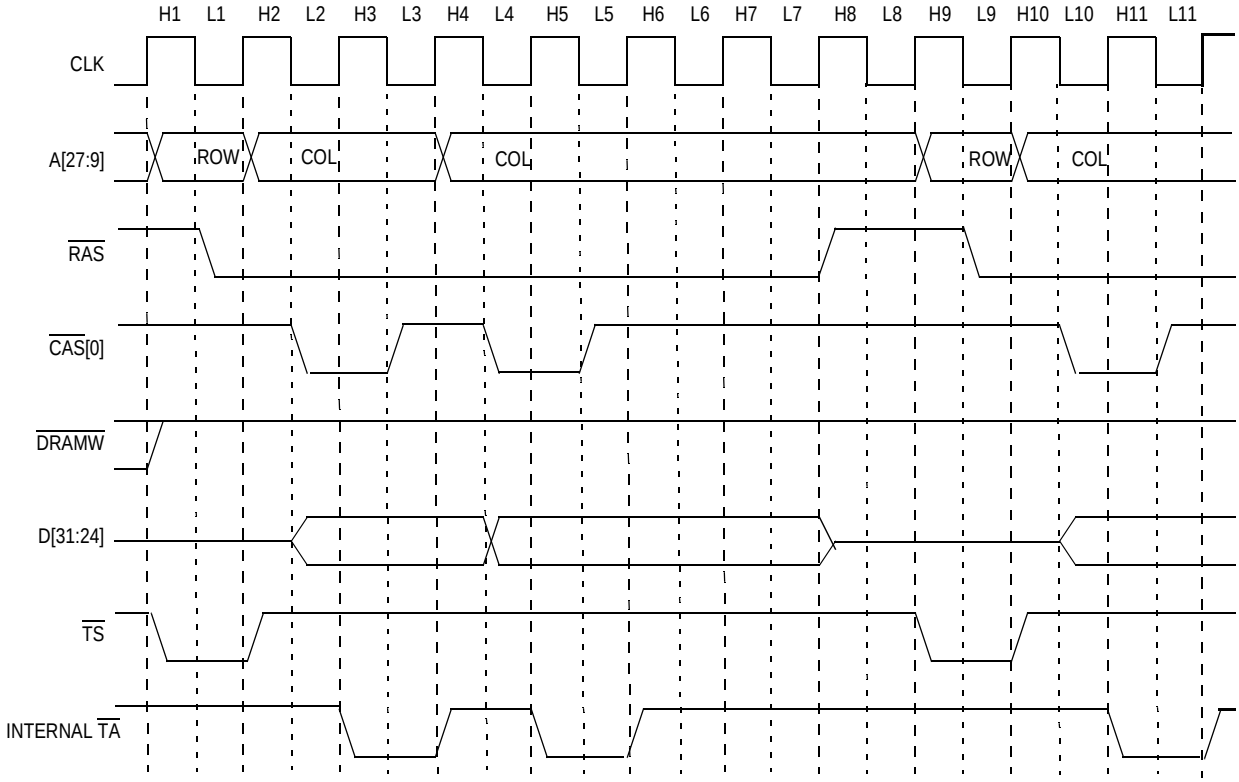


Figure 11-13. Word Read Transfer Followed by a Page Miss Byte Read Transfer in Fast Page Mode with 8-Bit EDO DRAM

Clock H1

The first byte read transfer of the burst word transfer starts in H1. During H1, the MCF5206e drives the row address on A[27:9], drives DRAMW high indicating a DRAM write transfer, drives SIZ[1:0] to $2 indicating a byte transfer, and asserts TS.

Clock L1

The MCF5206e asserts RAS to indicate the row address is valid on A[27:9].

Clock H2

The MCF5206e negates TS, drives the column address on A[27:9].

Clock L2

The MCF5206e asserts CAS[0] to indicate the column address is valid on A[27:9]. At this point the EDO DRAM drives data on D[31:24].

in the correct byte lanes based on the data size and the port size of the DRAM, and must drive the data to meet the timing specifications of the DRAM. In this case, the external master should drive the write data on D[31:16].

Clock H2

On the rising edge of CLK when $\overline{TS}$ is asserted, the MCF5206e registers the address and attribute signals. It internally decodes these signals and determines if the external master transfer is a DRAM access. The external master negates $\overline{TS}$ and must three-state A[27:0] after the rising edge of CLK H2, if the internal address multiplexing is to be used.

Clock H3

The MCF5206e has determined that the external master transfer is a DRAM access, so the MCF5206e drives A[27:0] with the same value as was registered on the rising edge of H2. A[27:9] contain the DRAM row address. The MCF5206e also drives $\overline{DRAMW}$ low indicating a DRAM write cycle.

Clock L3

The MCF5206e asserts $\overline{RAS}$ to indicate the row address is valid on A[27:9].

Clock H4

The MCF5206e internally multiplexes the address and drives out the column address on A[27:9]. The MCF5206e also actively drives $\overline{TA}$ negated.

Clock L4

The MCF5206e asserts $\overline{CAS}$[1:0] to indicate the column address is valid on A[27:9]. The external master must set up and hold the first word of data on D[31:16] with respect to the falling edge of $\overline{CAS}$[1:0] based on the DRAM specifications.

Clock H5

The MCF5206e asserts the $\overline{TA}$ signal to indicate that the first word write transfer of the longword burst will be completed on the next rising edge of CLK.

Clock H6

The MCF5206e negates $\overline{RAS}$, $\overline{CAS}$[1:0], and $\overline{TA}$, ending the first word transfer of the longword burst. The negation of $\overline{RAS}$ starts the $\overline{RAS}$ precharge. The MCF5206e drives the row address again for second word transfer of the longword burst write.

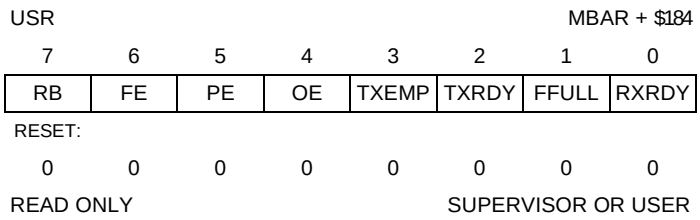
Clock L7

Once the $\overline{RAS}$ precharge time has been met, the MCF5206e asserts $\overline{RAS}$ to indicate the row address is valid on A[27:9].

Table 12-5. SBx Control Bits

SB3	SB2	SB1	SB0	LENGTH 6-8 BITS	LENGTH 5 BITS
0	0	0	0	0.563	1.063
0	0	0	1	0.625	1.125
0	0	1	0	0.688	1.188
0	0	1	1	0.750	1.250
0	1	0	0	0.813	1.313
0	1	0	1	0.875	1.375
0	1	1	0	0.938	1.438
0	1	1	1	1.000	1.500
1	0	0	0	1.563	1.563
1	0	0	1	1.625	1.625
1	0	1	0	1.688	1.688
1	0	1	1	1.750	1.750
1	1	0	0	1.813	1.813
1	1	0	1	1.875	1.875
1	1	1	0	1.938	1.938
1	1	1	1	2.000	2.000

12.4.1.3 STATUS REGISTER (USR). The USR indicates the status of the characters in the receive FIFO and the status of the transmitter and receiver. The RB, FE, and PE bits



are cleared by the Reset Error Status command in the UCR if the RB bit has not been read. Also, RB, FE, PE and OE can also be cleared by reading the Receive buffer (RE).

RB — Received Break

1 = An all-zero character of the programmed length has been received without a stop bit. The RB bit is valid only when the RxRDY bit is set. A single FIFO position is occupied when a break is received. Additional entries into the FIFO are inhibited until RxD returns to the high state for at least one-half bit time, which is equal to two successive edges of the internal or external clock x 1 or 16 successive edges of the external clock x 16. The received break circuit detects breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until the end of the next detected character time.

0 = No break has been received.

a configurable parameter (2, 3, or 4 bytes).

The nibble-wide DDATA port includes two 32-bit storage elements for capturing the CPU core bus information. These two elements effectively form a FIFO buffer connecting the core bus to the external development system. The FIFO buffer captures variant branch target addresses along with certain operand read/write data for eventual display on the DDATA output port. The execution speed of the ColdFire processor is affected only when both storage elements contain valid data waiting to be dumped onto the DDATA port. In this case, the processor core stalls until one FIFO entry is available. In all other cases, data output on the DDATA port does not impact execution speed.

From the processor core perspective, the PST outputs signal the first AGEX cycle of an instruction's execution. Most single-cycle instructions begin and complete their execution within a given machine cycle.

Because the processor status (PST[3:0]) values of $C, $D, $E, and $F define a multicycle mode or a special operation, the PST outputs are driven with these values until the mode is exited or the operation completed. All the remaining fields specify information that is updated each machine cycle.

The status values of $8, $9, $A, and $B qualify the contents of the DDATA output bus. These encodings are driven onto the PST port one machine cycle before the actual data is displayed on DDATA.

Figure15-2 shows the execution of an indirect JMP instruction with the lower 16 bits of the target address being displayed on the DDATA output. In this diagram, the indirect JMP branches to address "target." The processor internally forms the PST marker ($9) one cycle before the address begins to appear on the DDATA port. The target address is displayed on DDATA for four consecutive clocks, starting with the least-significant nibble. The processor continues execution, unaffected by the DDATA bus activity.

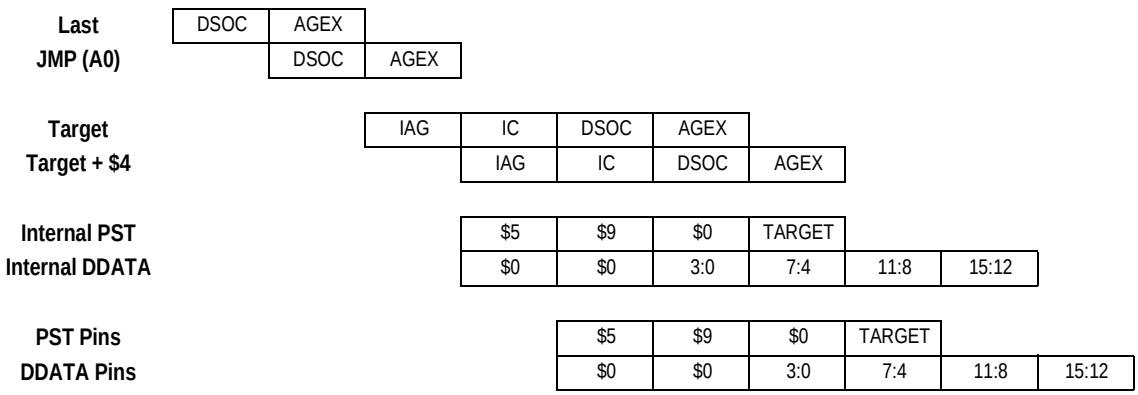


Figure 15-2. Pipeline Timing Example (Debug Output)

The ColdFire instruction set architecture (ISA) includes a PULSE opcode. This opcode generates a unique PST encoding when executed (PST = $4). This instruction can define logic analyzer triggers for debug and/or performance analysis.

Operand Size

For sized operations, this field specifies the operand data size. All addresses are expressed as 32-bit absolute values. The size field is encoded as listed in Table 15-4.

Table 15-4. BDM Size Field Encoding

ENCODING	OPERAND SIZE	BIT VALUES
00	Byte	8 bits
01	Word	16 bits
10	Long	32 bits
11	Reserved	

Address / $\overline{\text{Data}}$ (A/$\overline{\text{D}}$) Field

The A/$\overline{\text{D}}$ field is used in commands that operate on address and data registers in the processor. It determines whether the register field specifies a data or address register. A one indicates an address register; zero, a data register.

Register Field

In commands that operate on processor registers, this field specifies which register is selected. The field value contains the register number.

Extension Word(s) (as required):

Certain commands require extension words for addresses and/or immediate data. Addresses require two extension words because only absolute long addressing is permitted. Immediate data can be either one or two words in length; byte and word data each require a single extension word; longword data requires two words. Both operands and addresses are transferred by most significant word first. In the following descriptions of the BDM command set, the optional set of extension words are defined as the “Operand Data.”

15.2.3.3 Command Sequence Diagram. A command sequence diagram (see Figure 14-4) illustrates the serial bus traffic for each command. Each bubble in the diagram represents a single 16-bit transfer across the bus. The top half in each diagram corresponds to the data transmitted by the development system to the debug module; the bottom half corresponds to the data returned by the debug module in response to the development system commands. Command and result transactions are overlapped to minimize latency.

The cycle in which the command is issued contains the development system command mnemonic (in this example, “read memory location”). During the same cycle, the debug module responds with either the lowest order results of the previous command or with a command complete status (if no results were required).

During the second cycle, the development system supplies the high-order 16 bits of the memory address. The Debug module returns a “not ready” response ($10000) unless the received command was decoded as unimplemented, in which case the response data is the illegal command ($1FFFF) encoding. If an illegal command response occurs, the development system should retransmit the command.

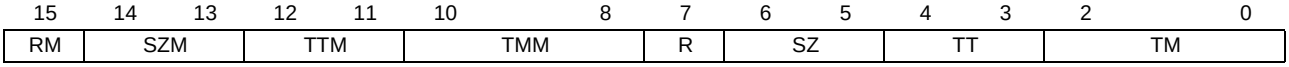
NOTE

Control Registers (DRc), are addressed using a 4-bit value as part of two new BDM commands (WDREG, RDREG).

These registers are also accessible from the processor's supervisor programming model through the execution of the WDEBUG instruction (Figure 15-5 illustrates the debug module programming model). Thus, the breakpoint hardware within the Debug module can be accessed by the external development system using the serial interface, or by the operating system running on the processor core. It is the responsibility of the software to guarantee that all accesses to these resources are serialized and logically consistent. The hardware provides a locking mechanism in the CSR to allow the external development system to disable any attempted writes by the processor to the Breakpoint Registers (setting IPW =1).

15.3.3.1 ADDRESS BREAKPOINT REGISTERS (ABLR, ABHR). The Address Breakpoint Registers define an upper (ABHR) and a lower (ABLR) boundary for a region in the operand logical address space of the processor that can be used as part of the trigger. The ABLR and ABHR values are compared with the ColdFire CPU core address signals, as defined by the setting of the TDR.

15.3.3.2 ADDRESS ATTRIBUTE BREAKPOINT REGISTER (AATR). The AATR defines the address attributes and a mask to be matched in the trigger. The AATR value is compared with the ColdFire CPU core address attribute signals, as defined by the setting of the TDR. The AATR is accessible in supervisor mode as debug control register $6 using the WDEBUG instruction and via the BDM port using the WDMREG command. The lower five bits of the AATR are also used for BDM command definition to define the address space for memory references as described in subsection **15.3.2.1 Reuse of the Debug Module Hardware.**



AATR Bit Definitions

RM[15]–Read/Write Mask

This field corresponds to the R-field. Setting this bit causes R to be ignored in address comparisons.

SZM[14:13]–Size Mask

This field corresponds to the SZ field. Setting a bit in this field causes the corresponding bit in SZ to be ignored in address comparisons.

TTM[12:11]–Transfer Type Mask

This field corresponds to the TT field. Setting a bit in this field causes the corresponding bit in TT to be ignored in address comparisons.

TMM[10:8]–Transfer Modifier Mask

This field corresponds to the TM field. Setting a bit in this field causes the corresponding bit in TM to be ignored in address comparisons.

System calls are often implemented by using the TRAP instruction. For trap exceptions, parameter passing is performed through data and address registers—rarely, if ever, directly through the stack. In addition, a system call typically does not need to know the stacked SR or PC information.

Breakpoints are usually implemented with the TRAP instruction or an illegal instruction such as an $A-line exception. If so, the stacked SR and PC are typically used. Other items in the stack may also need to be queried, especially if the breakpoint displays a stack trace. If so, you should examine the format closely for stack misalignments at the time of the breakpoint. This stack misalignment check would be useful in applications where stack alignment is a software design goal. These same concerns for the breakpoint implementation are applicable to trace exceptions as well.

A generalized exception handler can be implemented to catch unexpected exceptions. In addition to the SR and PC information, it is often necessary to obtain the vector information in the stack. Otherwise, the issues are similar to those found on breakpoints and tracing.

To port ColdFire processor access error exception, it is best to start with an MC68000 bus error handler. The ColdFire device access error recovery sequence has many similarities to the procedure recommended for the MC68000. However, you should be aware that a read bus error on the ColdFire processor will not advance the program counter to the next instruction. In addition, a write bus error may be taken long after an instruction has been executed and the stacked program counter may not point to the offending instruction. The main cause of an address error exception in the M68000 Family is that program flow is forced to continue at an odd address boundary. In addition, an MC68000 reports an address error if a data byte access is initiated on an odd address.

On a ROM monitor, it is often necessary to provide a means by which a user program is executed given a certain starting address. This is often implemented by placing an exception stack frame and then performing an RTE. If this is the case, the header files that define what a stack frame looks like would require modification to reflect the ColdFire stack frame format.

B.5 SUPERVISOR REGISTERS

The target software would eventually need to communicate the contents of registers to the host software. Both the host portions and target portions of a debugger must be modified to reflect the single stack pointer architecture of the ColdFire Family. In addition, the target debugger must keep a copy of the MOVEC register images in memory so that it can provide the host software register contents when asked to do so. A UNIX *grep* utility can find all instances of the MOVEC instruction and perform the appropriate modifications to accommodate the unidirectional MOVEC instruction.

The ColdFire architecture does not distinguish between a supervisor stack and a user stack. There is only a single stack pointer, A7. One way of dealing with this issue is to emulate the dual stacks by placing some code at the beginning and end portions of exception handlers to change the stack pointer contents, if necessary, during exceptions. This approach has the disadvantage that interrupt latency would be degraded because interrupts would have to be disabled during the stack-swapping process, but enable full flexibility of the 68K stack model.