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	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=mcf5206eab54

TABLE OF CONTENTS

1.3.1.1 PROCESSOR STATES. The processor is always in one of four states: normal processing, exception processing, stopped, or halted. It is in the normal processing state when executing instructions, fetching instructions and operands, and storing instruction results.

Exception processing is the transition from program processing to system, interrupt, and exception handling. Exception processing includes fetching the exception vector, stacking operations, and refilling the instruction fetch pipe after an exception. The processor enters exception processing when an exceptional internal condition arises, such as tracing an instruction, an instruction resulting in a trap, or executing specific instructions. External conditions, such as interrupts and access errors, also cause exceptions. Exception processing ends when the first instruction of the exception handler enters the operand execution pipeline.

Stopped mode is a reduced power operation mode that causes the processor to remain quiescent until either a reset or nonmasked interrupt occurs. The STOP instruction is used to enter this operation mode.

The processor halts when it receives an access error or generates an address error while in the exception processing state. For example, if during exception processing of one access error another access error occurs, the MCF5206e processor cannot complete the transition to normal processing nor can it save the internal machine state. The processor assumes that the system is not operational and halts. Only an external reset can restart a halted processor. When the processor executes a STOP instruction, it is in a special type of normal processing state, e.g., one without bus cycles. The processor stops but it does not halt.

The processor can also halt in a restart mode because of background debug mode events.

1.3.1.2 PROGRAMMING MODEL. The ColdFire programming model is separated into two privilege modes: supervisor and user. The S bit in the status register (SR) indicates the current privilege mode. The processor identifies a logical address by accessing either the supervisor or user address space, which differentiates between supervisor and user modes.

Programs access registers based on the indicated mode. User programs can access only registers specific to the user mode. System software executing in the supervisor mode can access all registers using the control registers to perform supervisory functions. User programs are thus restricted from accessing privileged information. The operating system performs management and service tasks for user programs by coordinating their activities. This difference allows the supervisor mode to protect system resources from uncontrolled accesses.

Most instructions execute in either mode but some instructions that have important system effects are privileged and can execute only in the supervisor mode. For instance, user programs cannot execute the STOP instructions. To prevent a program executing in user mode from entering the supervisor mode, instructions that can alter the S bit in the SR are privileged. The TRAP instructions provide controlled access to operating system services for user programs.

NOTE

The terms *assert* and *negate* are used throughout this section to avoid confusion when dealing with a mixture of active-low and active-high signals. The term *assert* or *assertion* indicates that a signal is active or true, independent of the level represented by a high or low voltage. The term *negate* or *negation* indicates that a signal is inactive or false.

Table 2-1. MCF5206e Signal Index

SIGNAL NAME	MNEMONIC	FUNCTION	INPUT/ OUTPUT
Address[27:24]/ Chip Select[7:4]/ Write Enable[3:0]	A[27:24]/ CS[7:4]/ WE[3:0]	Upper four bits of the address bus/ Upper four chip selects enable peripherals at programmed addresses/ Write enables select individual bytes in memory	In,Out/ Out/ Out
Address	A[23:0]	Lower 24 bits of the address bus. A[4:2] indicate the interrupt level during an IACK cycle	In,Out
Data	D[31:0]	Data bus used to transfer byte, word, or longword data	In,Out
Chip Select[3:0]	CS[3:0]	Enables peripherals at programmed addresses. CS[1] can indicate IACK during an interrupt acknowledge cycle. CS[0] provides relocatable boot ROM capability	Out
Interrupt Priority Level/ Interrupt Request	IPL[2]/IRQ[7] IPL[1]/IRQ[4] IPL[0]/IRQ[1]	Provides encoded interrupt priority level to processor/ Three individual external interrupts set to levels 7, 4, 1	In/ In
Read/Write	R/W	Identifies read and write data transfers	In,Out
Size	SIZ[1:0]	Indicates the data transfer size	In,Out
Transfer Type	TT[1:0]	Indicates the transfer type: normal, CPU space/Interrupt acknowledge or emulator mode	Out
Access Type & Mode	ATM	Time-multiplexed output signal indicating access type (instruction or data) and access mode (supervisor or user)	Out
Transfer Start	TS	Indicates the beginning of a bus cycle	In,Out
Transfer Acknowledge	TA	Synchronous transfer acknowledge. Asserted to indicate the successful completion of a bus transfer.	In,Out
Asynchronous Transfer Acknowledge	ATA	Asynchronous transfer acknowledge. Asserted to indicate the successful completion of a bus transfer	In
Transfer Error Acknowledge	TEA	Asserted to indicate an error condition exists for a bus transfer	In
Bus Request	BR	Asserted by the MCF5206e to request bus mastership	Out
Bus Grant	BG	Asserted by bus arbiter to grant bus mastership privileges to the MCF5206e	In
Bus Driven	BD	Indicates the MCF5206e has assumed explicit bus mastership of the external bus	Out
Clock Input	CLK	Input used to clock internal logic	In
Reset	RSTI	Processor reset	In
Row Address Strobe	RAS[1:0]	Row address strobe for external DRAM	Out
Column Address Strobe	CAS[3:0]	Column address strobe for external DRAM	Out
DRAM Write	DRAMW	Asserted on DRAM write cycles and negated on DRAM read cycles	Out
Receive Data	RxD[1], RxD[2]	Receive serial data input for UART 1 and UART 2	In
Transmit Data	TxD[1],TxD[2]	Transmit serial data output for UART 1 and UART 2	Out
Request-To-Send	RTS[1]	Indicates UART 1 is ready to receive data	Out
Request-To-Send/ Reset Out	RTS[2]/RSTO	RTSindicates UART 2 is ready to receive data/ RSTO is the reset out signal	Out/ Out
Clear-To-Send	CTS[1], CTS[2]	Indicates can transmit serial data for UART 1 and UART 2	In

C/I, SC, SD, UC, UD - Address Space Masks

This field allows specific address spaces to be enabled or disabled, placing the internal modules in a specific address space. If an address space is disabled, an access to the register location in that address space becomes an external bus access, and the module resource is not accessed. The address space mask bits are:

C/I = CPU Space/Interrupt Acknowledge Cycle mask
SC = Supervisor Code address space mask
SD = Supervisor Data address space mask
UC = User Code address space mask
UD = User Data address space mask

For each address space bit:

0= An access to the internal module can occur for this address space.
1= Disable this address space from the internal module selection. If this address space is used, no accesses occur to the internal peripheral, instead an external bus cycle will be generated.

V - Valid

This bit indicates when the contents of the RAMBAR are valid. The base address value is not used, and the SRAM module is not accessible until the V bit is set. An external bus cycle is generated if the base address field matches the internal core address, and the V bit is clear.

0 = Contents of RAMBAR are not valid.
1 = Contents of RAMBAR are valid.

The mapping of a given access into the SRAM uses the following algorithm to determine if the access “hits” in the memory:

```
if (RAMBAR[0] = 1)
    if (requested address[31:9] = RAMBAR[31:9])
        if (address space mask of the requested type = 0)
            Access is mapped to the SRAM module
            if (access = read)
                Read the SRAM and return the data
            if (access = write)
                if (RAMBAR[8] = 0)
                     Write the data into the SRAM
                else Signal a write-protect access error
```

5.3.3 SRAM Initialization

After a hardware reset, the contents of the SRAM module are undefined. The valid bit of the RAMBAR is cleared, disabling the module. If the SRAM needs to be initialized with instructions or data, you should perform the following steps:

1. Load the RAMBAR mapping the SRAM module to the desired location within the

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

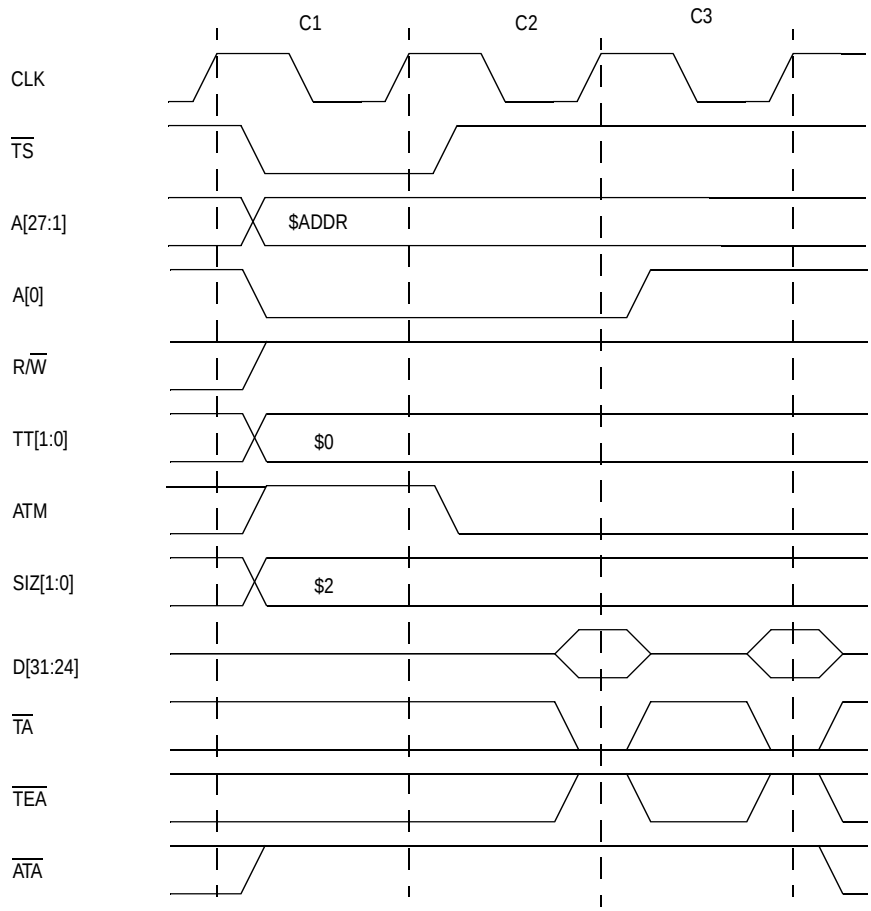

Figure 6-9. Bursting Word-Read From an 8-Bit Port (No Wait States)

Clock 1 (C1)

The read cycle starts in C1. During C1, the MCF5206e 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/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 MCF5206e asserts transfer start (TS) to indicate the beginning of a bus cycle.

Clock 2 (C2)

During C2, the MCF5206e negates 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 (TA). At the end of C2, the MCF5206e samples the level of TA and if TA is asserted, latches the current value of D[31:24]. If TA is asserted, the transfer of the first byte of the word read is complete. If TA is negated, the MCF5206e continues to sample TA and inserts wait states instead of terminating the transfer. The MCF5206e continues to sample TA on successive rising edges of CLK until it is asserted.

When the bus has been removed from the MCF5206e, one of two situations can occur. In the first case, the bus lock bit in the SIMR is cleared and the MCF5206e has explicit ownership of the bus. When the external bus master negates BG, the MCF5206e completes the current transfer, then negates BD and three-states the address, data, TS, R/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 MCF5206e has explicit ownership of the bus. In this case, when the external bus master negates BG, the MCF5206e continues to assert BD and continues to drive address, attributes, and control signals. The MCF5206e retains mastership of the bus until the bus lock bit in the SIMR is cleared. By asserting the bus lock bit, you can select the MCF5206e to be the highest priority master, even when mastership of the bus is controlled by an external arbiter. In this fashion, the MCF5206e 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.

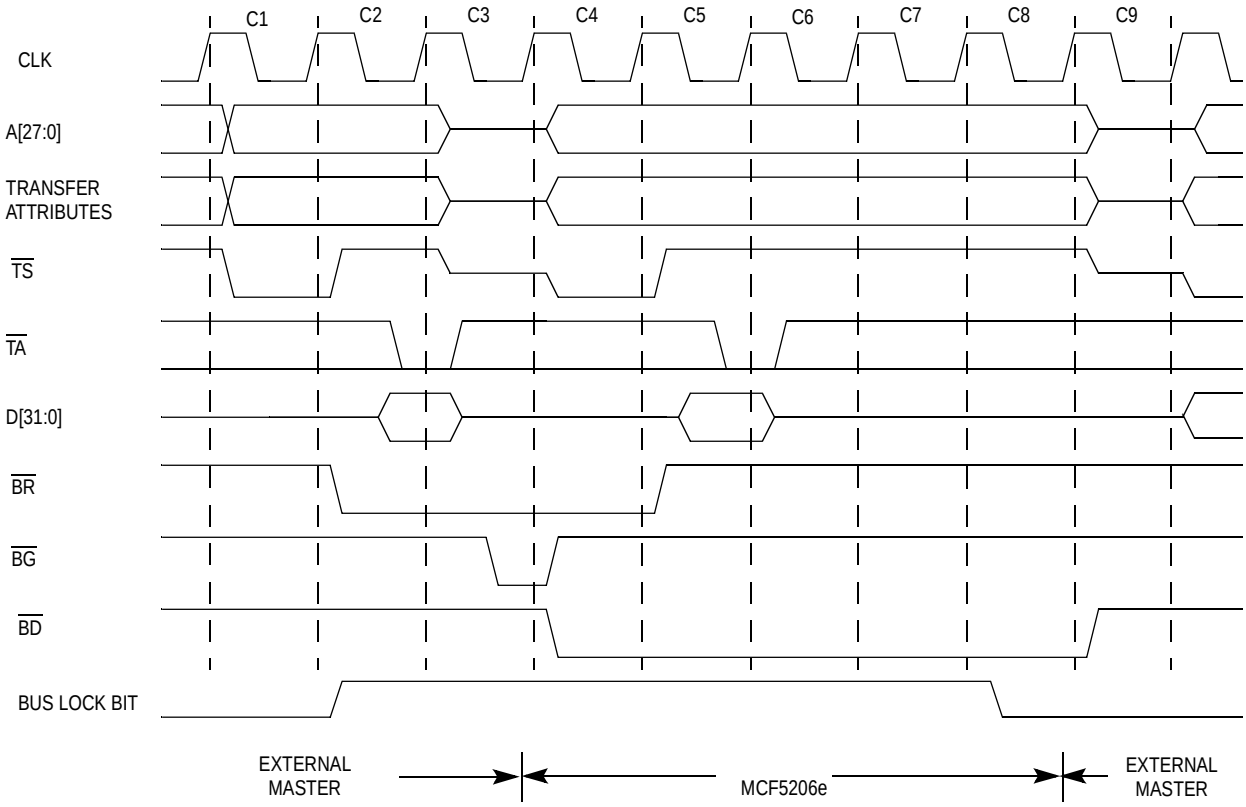

Figure 6-39. Three-Wire Bus Arbitration with Bus Lock Bit Asserted

In Figure 6-39, the external master is owner of the external bus during C1 and C2. During C2, the MCF5206e requests the external bus due to a pending internal transfer. On Clock C3, the external master relinquishes control of the bus and the external arbiter grants the bus to the MCF5206e by asserting bus grant (BG). At this point the bus lock bit is set to 1, and there is an internal access pending so the MCF5206e asserts bus driven (BD) during Clock C4, and begins the access. Thus, the MCF5206e becomes the explicit

last transfer before the boundary programmed in the BWC field. After the transfer is complete, it then asserts its internal bus request again to regain mastership at the earliest possible time as determined by the internal bus arbiter. The minimum amount of time that the DMA does not have the bus is one bus cycle.

7.6 DATA TRANSFER MODES

Each DMA channel supports single- and dual-address transfers. The single-address transfer mode consists of one DMA bus cycle, which allows either a read or a write cycle to occur. The dual-address transfer mode consists of a source operand read and a destination operand write.

7.6.1 Single Address Transactions

The DMA Controller Module begins a single address transfer sequence when the SAA bit is set while a DMA request is made. If no error conditions exist, the REQ bit is set. When the channel is enabled, the BSY bit is set and the REQ bit is cleared. The SAR contents are then driven onto the address bus and the value of the S_RW bit is driven onto the R/$\overline{W}$ signal. The BCR decrements on successful address phase accesses until it reaches 0, and the DONE bit is set.

In the event of a termination error, the BES and DONE bit of the DSR are set, and no further DMA Controller Module transactions are attempted.

7.6.2 Dual Address Transactions

The DMA Controller Module begins a dual-address transfer sequence when the SAA bit is cleared while a DMA request is made. If no error condition exists, the REQ bit of the DSR is set.

7.6.2.1 DUAL ADDRESS READS. The DMA Controller Module drives the value in the SAR onto the address bus. If the SINC bit of the DCR is set, then the SAR increments by the appropriate number of bytes upon a successful read cycle. When the appropriate number of read cycles completes successfully, the DMA initiates the write portion of the transfer.

In the event of a termination error, the BES and DONE bit of the DSR are set, and no further DMA Controller Module transactions are attempted.

7.6.2.2 DUAL ADDRESS WRITES. The DMA Controller Module drives the value in the DAR onto the address bus. If the DINC bit of the DCR is set, the DAR increments by the appropriate number of bytes at the completion of a successful write cycle. The BCR decrements by the appropriate number of bytes. If the BCR equals zero, the DONE bit is set. If the BCR is greater than 0, then another read/write transfer is initiated. If the BCR is a multiple of the programmed BWC, then the DMA request signal is negated until termination of the bus cycle, to allow the internal arbiter to return control to the ColdFire core. There is an idle clock before the next assertion of $\overline{MTS}$.

In the event of a termination error, the BED and DONE bit of the DSR are set, and no further DMA Controller Module transactions are attempted.

7.7 DMA CONTROLLER MODULE FUNCTIONAL DESCRIPTION

In the following descriptions, “DMA request” implies that the START bit is set or the $\overline{\text{DREQ}}$ signal is asserted while the EEXT bit is set. The START bit is cleared when the channel begins an internal access. Before initiating a transfer request, the DMA Controller Module first verifies that the source size and destination size (dual address only) as configured in the DCR, are consistent with the source address and destination address. If a misalignment is detected, no transfer occurs, and the CE bit of the DSR is set. The CE bit is also set if the BCR contains a value inconsistent with both the destination size (dual address only) and the source size. Depending on the configuration of the DCR, an interrupt event may be issued when the CE bit is set. Note that if the AA bit is set, error checking is performed only on the appropriate registers.

A “read/write” transfer refers to a dual-address access in which a number of bytes are read from the source address and written to the destination address. The number of bytes transferred is determined by the larger of the sizes specified by the source and destination size encodings.

The source and destination address registers (SAR and DAR) increment at the completion of a successful address phase. The BCR decrements at the completion of a successful address phase write when SAA=0 or any successful address phase when SAA=1. A successful address phase occurs when a valid address request is not held by the arbiter.

7.7.1 Channel Initialization and Startup

Before starting a block transfer operation, the channel registers must be initialized with information describing the channel configuration, request generation method, and data block. This initialization is accomplished by programming the appropriate information into the channel registers.

7.7.1.1 CHANNEL PRIORITIZATION. Channel 0 has priority over Channel 1 unless overwritten by the BWC bits in channel 1’s DCR. If the BWC bits for DMA channel 1 are set to 000, then it will have priority over channel 0, unless channel 0 also has the BWC bits set to 000.

7.7.1.2 PROGRAMMING THE DMA CONTROLLER MODULE. Some general comments on programming the DMA follow:

- No mechanism exists for preventing writes to control registers during DMA accesses
- If the BWC of sequential channels are equivalent, channel priority is in ascending order

The SAR is loaded with the source (read) address. If the transfer is from a peripheral device to memory, the source address is the location of the peripheral data register. If the transfer is from memory to a peripheral device or memory to memory, the source address is the starting address of the data block. This address may be any byte address. In the single-address mode, this register is used regardless of the direction of the transfer.

The DAR should contain the destination (write) address. If the transfer is from a peripheral device to memory or memory to memory, the DAR is loaded with the starting address of the data block to be written. If the transfer is from memory to a peripheral device, the DAR is

You can assign as many as four interrupts to the same interrupt level, but you must assign unique interrupt priorities. The interrupt controller uses the interrupt priorities during an interrupt acknowledge cycle to determine which interrupt is being acknowledged. The interrupt priority bits determine the appropriate interrupt being acknowledged when multiple interrupts are assigned to the same level and are pending when the interrupt-acknowledge cycle is generated.

NOTE

You should not program interrupts to have the same level and priority. Interrupts can have the same level but different priorities. All level and priority combinations must be unique.

If an external interrupt request is being acknowledged and the AVEC bit in the corresponding ICR is not set, an external interrupt acknowledge cycle occurs. During an external interrupt acknowledge cycle, TT[1:0] and A[27:5] are driven high; A[4:2] are set to the interrupt level being acknowledged; and A[1:0] are driven low. Additionally, ATM is asserted when $\overline{TS}$ is asserted and ATM is negated when $\overline{TS}$ is negated. For nonautovector responses, the external device places the vector number on D[31:24]. For autovector responses, the autovector is generated internally and no external interrupt acknowledge cycle is run.

An interrupt request from the SWT does not require an external interrupt acknowledge cycle because SWIVR stores its interrupt vector number.

If an internal peripheral interrupt source is being acknowledged and the AVEC bit in the corresponding ICR is cleared, an internal interrupt-acknowledge cycle occurs with the internal peripheral supplying the interrupt vector number. If the corresponding AVEC bit is set, an internal interrupt-acknowledge cycle run but the SIM internally generates an autovector. No external interrupt acknowledge cycle is run.

8.3 PROGRAMMING MODEL

8.3.1 SIM Registers Memory Map

Table 8-2 shows the memory map of all the SIM registers. The internal registers in the SIM are memory-mapped registers offset from the MBAR address pointer. The following list addresses several key notes regarding the programming model table:

- The Module Base Address Register can only be accessed in supervisor mode using the MOVEC instruction with an Rc value of $C0F.
- Underlined registers are status or event registers.
- Addresses not assigned to a register 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 returns zeros.
- The reset value column indicates the register initial value at reset. Certain registers may be uninitialized at reset.

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

During C2, the MCF5206e negates transfer start (TS), drives access type and mode (ATM) high to identify if the transfer as supervisor, drives data onto D[31:16] and asserts the appropriate chip select (CS). At the end of C2, the MCF5206e samples the level of TA. If TA was asserted the transfer of the word would be complete. Since TA is negated, the MCF5206e continues to output the data and inserts a wait state instead of terminating the transfer.

Clock 3 (C3)

The MCF5206e asserts the write enable (WE[1:0]) signals. If the selected device(s) is ready to latch the data, it latches D[31:0] and asserts the transfer acknowledge (TA). At the end of C3, the MCF5206e samples the level of TA. If TA is asserted, the transfer of the word is complete. If TA is negated, the MCF5206e continues to sample TA and inserts wait states instead of terminating the transfer. The MCF5206e continues to sample TA on successive rising edge of CLK until it is asserted. If the bus monitor timer is enabled and TA is not asserted before the programmed bus monitor time is reached, the cycle is terminated with an internal bus error.

Clock 4 (C4)

The MCF5206e negates the chip select (CS) and write enable (WE[1:0]) signals and continues to drive the address, data and attribute signals until after the next rising edge of CLK.

NOTE

When address setup is enabled (ASET=1), write enables (WE[3:0]) do not assert on zero wait state write transfers.

9.3.3.4 BURST TRANSFER AND CHIP SELECTS. If the burst enable bit in the appropriate control register (Chip Select Control Register or Default Memory Control Register) is set to 1, and the operand size is larger than the port size of the memory being accessed, the MCF5206e performs word, longword and line transfers in burst mode. When burst mode is selected, the size of the transfer (indicated by SIZ[1:0]) reflects the size of the operand being read - not the size of the port being accessed (i.e. a line transfer is indicated by SIZ[1:0] = $3 and a longword transfer is indicated by SIZ[1:0] = $0, regardless of the size of the port or the number of transfers required to access the data).

The MCF5206e supports burst-inhibited transfers for memory devices that are unable to support bursting. For this type of bus cycle, the burst enable bit in the Chip Select Control Registers (CSCRs) or Default Memory Control Register (DMCR) must be cleared.

Figure 9-5 illustrates a supervisor code longword read transfer to a 16-bit port with address setup and address hold disabled.

DCAR1). The bits that are masked is determined by the value programmed in the BAM field in the DRAMC Mask Registers (DCMR0 - DCMR1).

The masking of address bits is used to define the address space of the DRAM bank. Address bits that are masked are not used in the comparison with the transfer address. The base address field (BA31-BA17) in the DCARs and the base address mask field (BAM31-BAM17) in the DCMRs correspond to transfer address bits 31-17. Clearing (unmasking) all bits in the BAM field makes the address space 128 KBytes. For the address space of a DRAM bank to be contiguous, address bits should be masked (BAM bits set to a 1) in ascending order starting with A[17].

For example, if the DCARs and DCMRs are programmed as shown in Table 11-3, DRAM bank 0 would have a 16 MByte address space starting at address $04000000, while DRAM bank 1 would have a 1 MByte address space starting at address $05000000. A transfer with A[31:24] = $04 accesses DRAM bank 0, and a transfer address with A[31:20] = $050 accesses DRAM bank 1.

Table 11-3. DRAM Bank Programming Example 1

DRAM BANK	DCAR	DCMR	DRAM ADDRESS SPACE	ADDRESS MATCH
0	$0400	$00FE0000	16 Mbyte	$04xxxxxx
1	$0500	$000E0000	1 Mbyte	$050xxxxx

Refer to **Section 11.4.2.3 DRAM Controller Address Register (DCAR0 - DCAR1)** and **Section 11.4.2.4 DRAM Controller Mask Register (DCMR0 - DCMR1)** for further details.

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 MCF5206e. The output of A[27:24] are dependent on the setting of PAR3-PAR0 in the Pin Assignment Register (PAR) in the SIM.

NOTE

The MCF5206e compares the address for the current bus transfer with the address and mask bits in the Chip Select Address Registers (CSARs), DRAM Controller Address Registers (DCARs) and the Chip Select Mask Registers (CSMRs) and DRAM Controller Mask Register (DCMRs),

Clock H8

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

Clock L8

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

Clock H9

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

Clock L9

The MCF5206e asserts CAS[0] to indicate the column address is valid on A[27:9]. The external master must set up and hold the data with respect to the falling edge of CAS[0] based on the DRAM specifications.

Clock H10

The MCF5206e asserts the TA signal to indicate that the byte write transfer will be completed on the next rising edge of CLK.

Clock H11

The MCF5206e then negates RAS, CAS[0], and TA, and three-state the address bus, ending the byte write transfer. The negation of RAS starts the RAS precharge. Once A[27:0] has three-stated, the external master can start another transfer.

Clock H12

The MCF5206e three-states TA.

11.3.8.2 EXTERNAL MASTER BURST TRANSFER IN NORMAL MODE. A burst transfer to DRAM is generated when the operand size is larger than the DRAM bank port size (e.g., line transfer to a 32-bit port, longword transfer to an 8-bit port). On DRAM burst transfers, the external master should assert TS only once. The start of the secondary transfers of a burst is delayed by the DRAMC until the programmed RAS precharge time is met.

The timing of external master burst reads and burst writes is identical in normal page mode, with the exception of when the DRAM drives data on reads and when the external master drives data on writes.

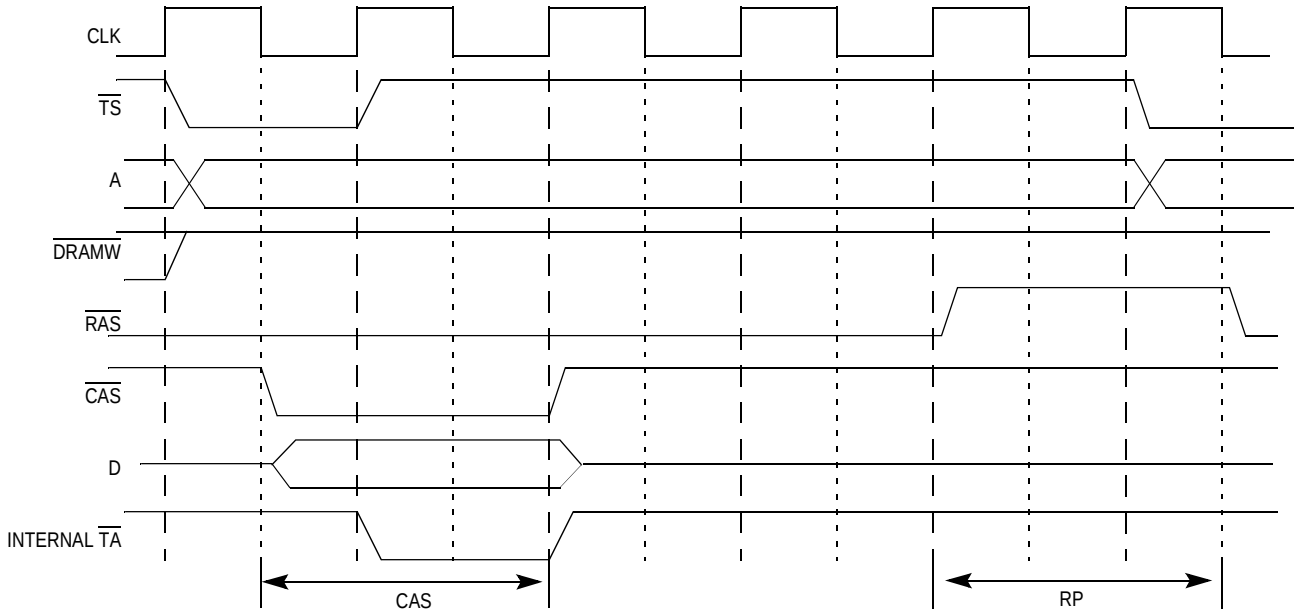

Figure 11-20. Fast Page Mode Page Hit and Page Miss DRAM Transfer Timing

CAS - Column Address Strobe Time

This field, together with the EDO field, controls the number of system clocks that $\overline{CAS}$ asserts on transfers once a page is open in fast page mode and burst page mode. Refer to Figure 11-18 for timing diagrams of fast-page-mode or burst-page- mode transfers to standard DRAMs and Figure 11-21 for fast-page-mode or burst-page-mode transfers to EDO DRAMs.

For EDO = 0:

0 = $\overline{CAS}$ is asserted for 1.5 system clocks
1 = $\overline{CAS}$ is asserted for 2.5 system clocks

For EDO = 1:

0 = $\overline{CAS}$ is asserted for 1.0 system clock
1 = $\overline{CAS}$ is asserted for 2.0 system clocks

the failed attempt to engage the bus. When considering these cases, the slave service routine should test the MAL first and the software should clear the MAL bit if it is set.

SECTION 14
TIMER MODULE

14.1 OVERVIEW OF THE TIMER MODULE

The MCF5206e contains two general-purpose16-bit timers. This section of the manual documents how the 16-bit timers operate.

The output of an 8-bit prescaler clocks each 16-bit timer. The prescaler input can be the system clock, the system clock divided by 16, or the timer input (TIN) pin. Figure 14-1 is a block diagram of the timer module.

Both timer pins are multiplexed with the DMA request function. Their function is defined by programming bits 8 and 9 of the PAR (Pin Assignment Register).

14.2 OVERVIEW OF KEY FEATURES

The general-purpose 16-bit timer unit has the following features:

- Maximum period of 5 seconds at 54 MHz & 6.7 seconds at 40 MHz.
- 18.5 ns resolution at 54 MHz & 25 ns at 40 MHz
- Programmable sources for the clock input, including external clock
- Input-capture capability with programmable trigger edge on input pin
- Output-compare with programmable mode for the output pin
- Free run and restart modes
- Maskable interrupts on input capture or reference-compare

17.3.14 General-Purpose I/O Port AC Timing Specifications

Table 17-14. General-Purpose I/O Port AC Timing Specifications

NAME	CHARACTERISTIC	40 MHz		54 MHz		UNIT
		MIN	MAX	MIN	MAX	
P1	PP[7:0]input setup time to CLK (rising)	3	—	2	—	ns
P2	PP[7:0] input hold time from CLK (rising)	4.5	—	4.5	—	ns
P3	CLK to PP[7:0] Output Valid	3	19.5	3	17	ns
P4	CLK to PP[7:0] Output Invalid (Output Hold)	3	—	3	—	ns

17.3.15 General-Purpose I/O Port Timing Diagram

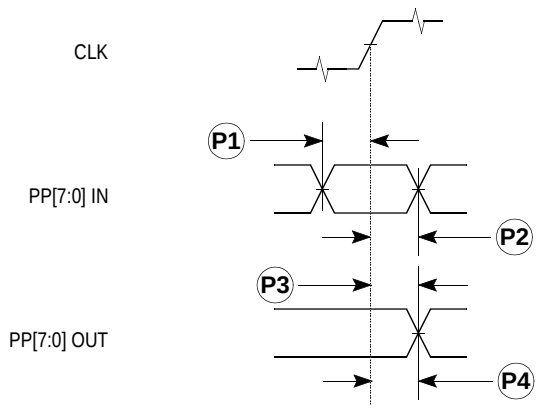

Figure 17-14. General-Purpose I/O Port Timing

18.1 PACKAGE DIAGRAM & PINOUT

The MCF5206e is supplied in a 160-pin plastic quad flat pack package as shown:

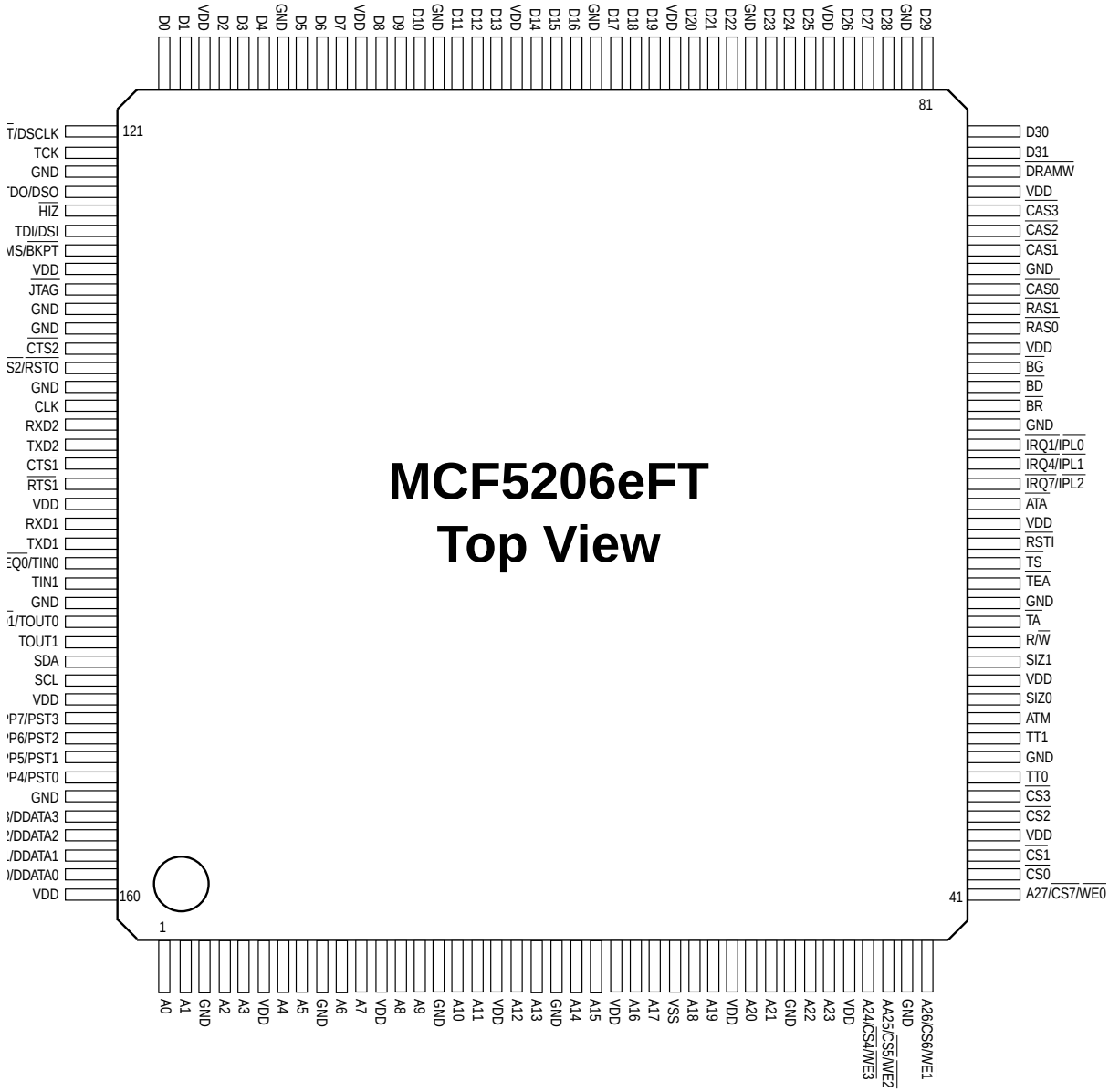

Figure 18-1. MCF5206e Package Diagram & Pinout

18.1.1 Package/Frequency Availability

The following table identifies the packages and operating frequencies available for the MCF5206e processor.

MBAR OFFSET ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE	ACCESS
$1C9	PPDAT	8	Port A Data Register	$00	R/W
$1E0	MADR	8	M-Bus Address Register	$00	R/W
$1E4	MFDR	8	M-Bus Frequency Divider Register	$00	R/W
$1E8	MBCR	8	M-Bus Control Register	$00	R/W
$1EC	MBSR	8	M-Bus Status Register	$00	R/W
$1F0	MBDR	8	M-Bus Data I/O Register	$00	R/W
$200	DMASAR0	32	Source Address Register 0	$00	R/W
$204	DMADAR0	32	Destination Address Register 0	$00	R/W
$208	DCR0	16	DMA Control Register 0	$00	R/W
$20C	BCR0	16	Byte Count Register 0	$00	R/W
$210	DSR0	8	Status Register 0	$00	R/W
$214	DIVR0	8	Interrupt Vector Register 0	$0F	R/W
$240	DMASAR1	32	Source Address Register 1	$00	R/W
$244	DMADAR1	32	Destination Address Register 1	$00	R/W
$248	DCR1	16	DMA Control Register 1	$00	R/W
$24C	BCR1	16	Byte Count Register 1	$00	R/W
$250	DSR1	8	Status Register 1	$00	R/W
$254	DIVR1	8	Interrupt Vector Register 1	$0F	R/W

APPENDIX B
PORTING FROM M68000 FAMILY DEVICES

This section is an overview of the issues encountered when porting embedded development tools to ColdFire® devices when starting with the M68000 Family devices.

B.1 C COMPILERS AND HOST SOFTWARE

For the purpose of this discussion, it is assumed that an embedded software development tool chain consists of a “host” portion and a “target” portion. The host portion consists of tool chain parts that execute on a desktop computer or workstation. The target portion of the tool chain runs ColdFire executables on a physical ColdFire target board.

Compilers, assemblers, linkers, loaders, instruction set simulators, and the host portion of debuggers are examples of host tools. Many host tools such as linkers and loaders that work with the M68000 Family devices can also be used without modification with ColdFire processors.

Although you can use an existing M68000 Family assembler and disassembler with ColdFire devices, Motorola recommends modifying the assembler so that non-ColdFire assembly code cannot be put together in the executable. This is especially true if the assembler assembles handwritten code. Porting the disassembler is for convenience and can be performed later.

Debuggers usually are comprised of two parts. A host portion of the debugger typically issues higher level commands for the target portion of debugger target. The target portion of the debugger typically handles the exact details of the implementation of tracing, breakpoints, and other lower level details. The debugger host portion requires little modification. Most likely, the only architectural items of concern are the following:

- Differences in the designed supervisor registers and stack pointers (for displaying registers)
- Interpretation of exception stack frames (if not already performed by the target portion)

B.2 TARGET SOFTWARE PORT

Porting ROM monitors and operating systems can begin after the compiler and assembler have been ported. For example, consider the steps involved in porting a ROM debugger. Similar issues are encountered when porting an RTOS and target applications.

It is assumed that target software consists of C and assembly source code. The first step is to create executables that runs on existing M68000 Family hardware to test the conversion from M68000 Family code to the proper ColdFire subset. This step verifies that the process of code conversion does not introduce new errors.