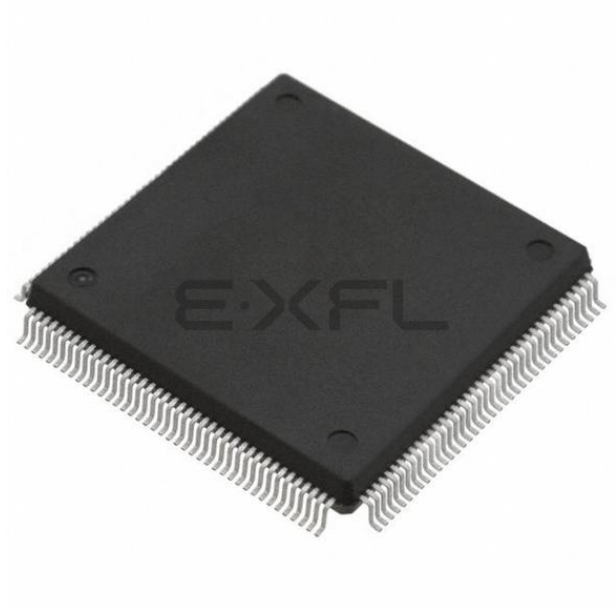

Welcome to E-XFL.COM

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	33MHz
Co-Processors/DSP	-
RAM Controllers	DRAM
Graphics Acceleration	-
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5V
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=mcf5206ab33a

TABLE OF CONTENTS (Continued)

devices. The number of devices that can be connected is limited by bus capacitance and the number of unique addresses.

1.3.8 System Interface

The MCF5206 processor provides a glueless interface to 8-, 16-, and 32-bit port size SRAM, ROM, and peripheral devices with independent programmable control of the assertion and negation of chip-selects and write-enables. Programmable address and data-hold times can be extended for a compatible interface to external devices and memory. The MCF5206 also supports bursting ROMs.

1.3.8.1 EXTERNAL BUS INTERFACE. The bus interface controller transfers information between the ColdFire core and memory, peripherals, or other masters on the external bus. The external bus interface provides as many as 28 bits of address bus space, a 32-bit data bus, and all associated control signals. This interface implements an extended synchronous protocol that supports bursting operations. For nonsynchronous external memory and peripherals, the MCF5206 processor provides an alternate asynchronous bus transfer acknowledgment signal.

Simple two-wire request/acknowledge bus arbitration between the MCF5206 processor and another bus master, such as a DMA device, is glueless with arbitration handled internal to the MCF5206 processor. Alternately, an external bus arbiter can control more complex three-wire (request, grant, busy) multiple-master bus arbitration, allowing overlapped bus arbitration with one clock-bus handovers.

1.3.8.2 CHIP-SELECTS . Eight programmable chip-select outputs provide signals that enable external memory and peripheral circuits for automatic wait-state insertion. These signals also interface to 8-, 16-, or 32-bit ports. In addition, other external bus masters can access chip-selects. The upper four chip-selects are multiplexed with A[27:24] of the address bus and the four write-enable signals. The base address, access permissions, and timing waveforms are all programmable with configuration registers.

1.3.9 8-Bit Parallel Port (General-Purpose I/O)

An 8-bit general-purpose programmable parallel port serves as either an input or an output on a bit-by-bit basis. The parallel port is multiplexed with PST[3:0] and DDATA[3:0] debug signals.

1.3.10 Interrupt Controller

The interrupt controller provides user-programmable control of three or seven external interrupt and five internal peripheral interrupts. You can program each internal interrupt to any one of seven interrupt levels and four priority levels within each of these levels. You can configure the three external interrupt signals as either fixed interrupt levels 1, 4, and 7, or as a seven-level encoded interrupt. You can program the external interrupts to any one of the four priority levels within the respective interrupt levels.

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

This three-state bidirectional active-low synchronous signal indicates the completion of a requested data transfer operation. During transfers initiated by the MCF5206, transfer acknowledge ($\overline{TA}$) is an input signal from the referenced slave device indicating completion of the transfer.

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

When an alternate master is controlling the bus, $\overline{TA}$ may be driven as an output by the MCF5206 or may be driven by the referenced slave device to indicate the completion of the requested data transfer. If the alternate master requested transfer is to a chip select or default memory, the assertion of $\overline{TA}$ is controlled by the number of wait states and the setting of the Alternate Master Automatic Acknowledge (EMAA) bit in the Chip Select Control Registers (CSCRs) or the Default Memory Control Register (DMCR). If the alternate master requested transfer is a DRAM access, $\overline{TA}$ is driven by the MCF5206 as an output and asserted at the completion of the transfer.

2.5.7 Asynchronous Transfer Acknowledge ($\overline{ATA}$)

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

NOTE

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

$\overline{ATA}$ must be driven for one full clock to ensure that the MCF5206 properly synchronizes the signal. $\overline{ATA}$ is not used for termination during DRAM accesses.

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

This active-low input signal is asserted by the external slave to indicate an error condition for the current transfer. The assertion of transfer error acknowledge ($\overline{TEA}$) causes the MCF5206 to immediately abort the bus cycle. The assertion of $\overline{TEA}$ has precedence over the assertion of $\overline{ATA}$ and $\overline{TA}$.

NOTE

$\overline{TEA}$ can be asserted to a maximum of one clock after the assertion of $\overline{ATA}$ and still be recognized.

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

2.6 BUS ARBITRATION SIGNALS

2.6.1 Bus Request ($\overline{BR}$)

This active-low output signal indicates to an external arbiter that the MCF5206 needs use of the bus for one or more bus cycles. $\overline{BR}$ is negated when the MCF5206 begins an access to the external bus, and remains negated until another internal request occurs with $\overline{BG}$ negated.

2.6.2 Bus Grant ($\overline{BG}$)

An external arbiter asserts this active-low input signal to indicate that the MCF5206 can become master of the external bus at the next rising edge of CLK. When the arbiter negates $\overline{BG}$, the MCF5206 relinquishes the bus as soon as the current transfer is complete, provided the bus lock bit in the SIMR is not set. If the bus lock bit is set, the MCF5206 retains bus mastership until the bus lock bit is cleared. The external arbiter must not grant the bus to any other master until the MCF5206 negates $\overline{BD}$.

2.6.3 Bus Driven ($\overline{BD}$)

The MCF5206 asserts this active-low output signal to indicate it has assumed explicit mastership of the external bus. The MCF5206 asserts $\overline{BD}$ if $\overline{BG}$ is asserted and either the MCF5206 has a pending bus transfer or the bus lock bit in the SIMR is set to 1. If the MCF5206 is granted mastership of the external bus, but does not have a pending bus transfer and the bus lock bit in the SIMR is cleared, the $\overline{BD}$ signal is not asserted (implicit mastership of the bus is assumed).

If $\overline{BG}$ is negated to the MCF5206 during a bus transfer and the bus lock bit in the SIMR is cleared, the MCF5206 completes the last transfer of the current access, negates $\overline{BD}$, and three-states all bus signals on the rising edge of CLK. If the MCF5206 loses bus ownership during an idle bus period with $\overline{BD}$ asserted and the bus lock bit in the SIMR cleared, the MCF5206 negates $\overline{BD}$ and three-states all bus signals on the next rising edge of CLK. If the MCF5206 loses bus ownership during an idle bus period with $\overline{BD}$ asserted and the bus lock bit in the SIMR set to 1, the MCF5206 continues to assert $\overline{BD}$ and maintains explicit ownership of the external bus until the bus lock bit in the SIMR is cleared.

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

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

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

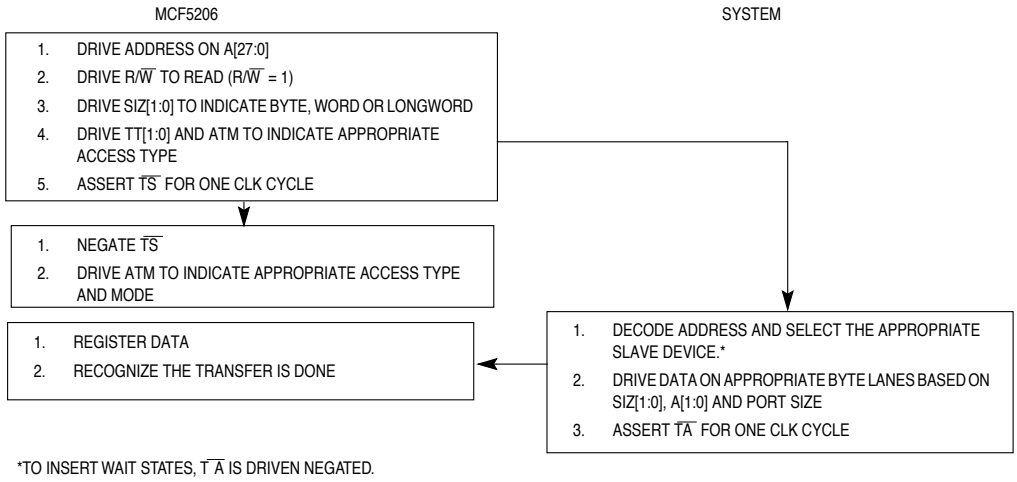

Figure 6-4. Byte-, Word-, and Longword-Read Transfer Flowchart

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

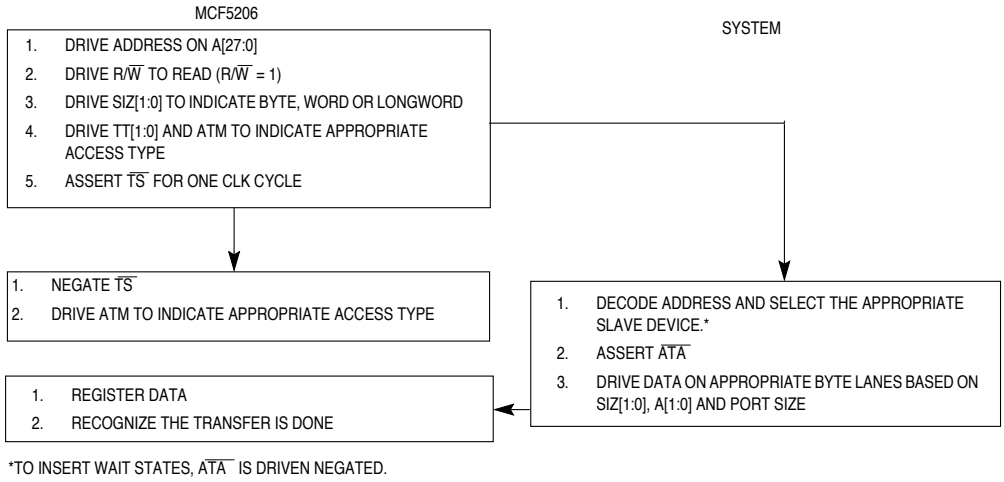

Figure 6-16. Byte-, Word-, and Longword-Read Transfer with Asynchronous Termination Flowchart (One Wait State)

NOTE

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

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

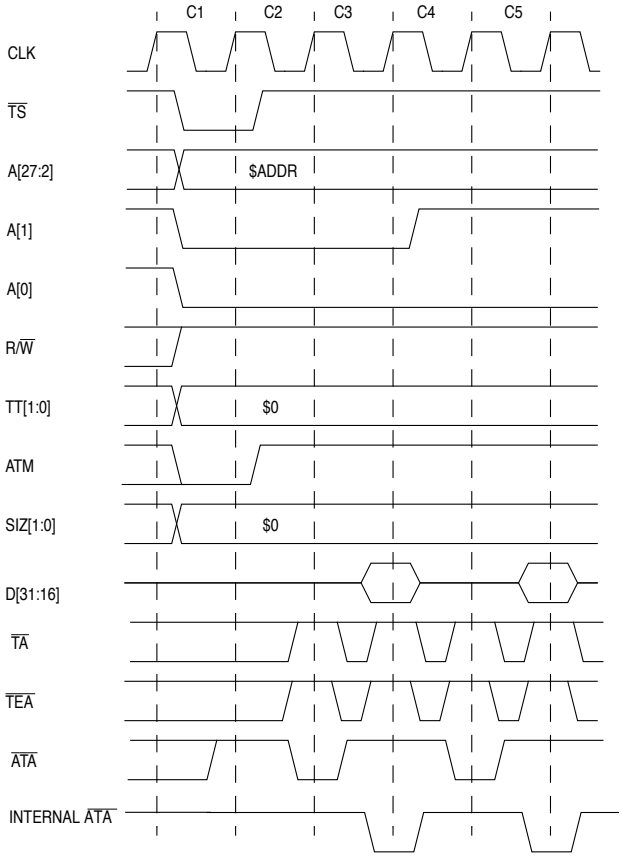

Figure 6-21. Bursting Longword-Read from 16-Bit Port Using Asynchronous Termination (One Wait State)

Clock 1 (C1)

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

Clock 2 (C2)

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

Clock 3 (C3)

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

Clock 4 (C4)

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

Clock 5 (C5)

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

Clock 6 (C6)

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

6.5.12 Termination Tied to GND

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

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

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

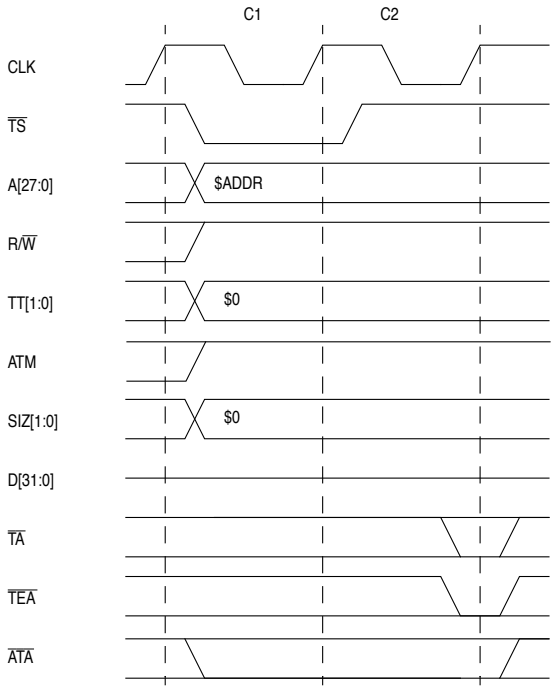

Figure 6-32. Bursting Longword-Read Access from 16-Bit Port Terminated with TEA Timing

Clock 1 (C1)

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

Clock 2 (C2)

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

Table 8-2. Maximum Memory Bank Sizes

AVAILABLE ADDRESS SIGNALS	MAXIMUM CS BANK SIZE
A[23:0]	16 Mbyte
A[24:0]	32 Mbyte
A[25:0]	64 Mbyte
A[26:0]	128 Mbyte
A[27:0]	256 Mbyte

connected to external memory. The MCF5206 does not output the address during alternate master initiated transfers to chip select memory.

8.2.1.4 DATA BUS. You can configure the chip select and default memory spaces to be 8-, 16-, or 32-bits wide. A 32-bit port must reside on data bus bits D[31:0], a 16-bit port must reside on data bus bits D[31:16], and an 8-bit port must reside on data bus bits D[31:24]. This ensures that the MCF5206 correctly transfers valid data to 8-, 16-, and 32-bit ports. Figure 8-1 illustrates the connection of the data bus to 8-, 16-, and 32-bit ports.

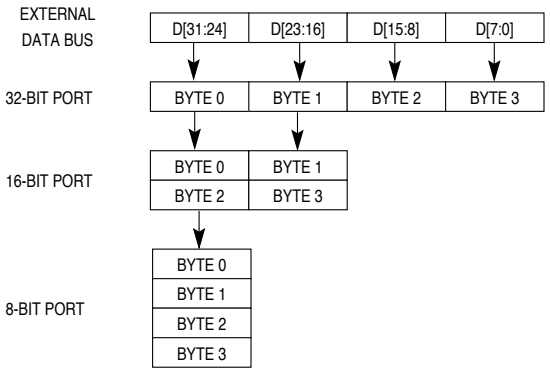

Figure 8-1. MCF5206 Interface to Various Port Sizes

8.2.1.5 TRANSFER ACKNOWLEDGE ($\overline{TA}$). Transfer acknowledge is a bidirectional signal that indicates the current data transfer has been successfully completed. You can program $\overline{TA}$ to be output after a programmed number of wait states during alternate master-initiated transfers that hit in chip select or default memory address space. $\overline{TA}$ is an input during ColdFire core-initiated transfers that hit in chip select or default memory address space.

8.3 CHIP SELECT OPERATION

The chip select controller provides a glueless interface to certain types of external memory including PROM and peripherals and external control signals for an easy interface to SRAM, EPROM, EEPROM and peripherals. Each of the eight chip select

Clock H4

The MCF5206 registers the read data driven by the DRAM and negates the internal transfer acknowledge, $\overline{RAS}$ and $\overline{CAS}$[0], ending the first byte-read transfer. This begins the $\overline{RAS}$ precharge. Once $\overline{CAS}$[0] is negated the DRAM disables its output drivers, and the data bus is three-stated.

Clock H6

Clock H6 is the earliest the next transfer initiated by the ColdFire core can start. The second DRAM byte-read transfer starts in H6. During H6, the MCF5206 drives the row address on the A[27:9], drives $\overline{DRAMW}$ high indicating a DRAM-read transfer, drives SIZ[1:0] to $1 indicating a byte transfer, and asserts $\overline{TS}$.

Clock L6

Clock L6 is the same as Clock L1.

Clock H7

Clock H7 is the same as Clock H2.

Clock L7

Clock L7 is the same as Clock L2.

Clock H8

Clock H8 is the same as Clock H3.

Clock H9

Clock H9 is the same as Clock H4.

10.3.3.2 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 all DRAM transfers, the MCF5206 asserts $\overline{TS}$ only once. The start of the secondary transfers of a burst is delayed by the DRAMC until the programmed $\overline{RAS}$ precharge time is reached.

The timing of 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 MCF5206 drives data on writes.

The fastest possible burst transfer in normal mode requires 3 clocks for the first transfer of the burst and 4 clocks for the secondary transfers (including a 1.5 clock $\overline{RAS}$ precharge time). You can program the DCTR to generate slower normal mode transfers.

You can use this feature for flow control to prevent overrun in the receiver by using the $\overline{RTS}$ output to control the $\overline{CTS}$ input of the transmitting device. If both the receiver and transmitter are programmed for $\overline{RTS}$ control, $\overline{RTS}$ control is disabled for both because such a configuration is incorrect. See **Section 11.4.1.2 Mode Register 2 (UMR2)** for information on programming the transmitter $\overline{RTS}$ control. On UART 2, $\overline{RTS}$ is muxed.

RxIRQ — Receiver Interrupt Select

1 = FFULL is the source that generates IRQ
0 = RxRDY is the source that generates IRQ

ERR — Error Mode

This bit controls the meaning of the three FIFO status bits (RB, FE, and PE) in the USR.

1 = Block mode—The values in the channel USR are the accumulation (i.e., the logical OR) of the status for all characters coming to the top of the FIFO since the last reset error status command for the channel was issued. Refer to **Section 11.4.1.5 Command Register (UCR)** for more information on UART module commands.
0 = Character mode—The values in the channel USR reflect the status of the character at the top of the FIFO.

NOTE

You must use ERR = 0 to obtain the correct A/$\overline{D}$ flag information when in multidrop mode.

PM1–PM0 — Parity Mode

These bits encode the type of parity used for the channel (see Table 11-3). The parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. These bits can alternatively select multidrop mode for the channel.

PT — Parity Type

This bit selects the parity type if parity is programmed by the parity mode bits; if multidrop mode is selected, it configures the transmitter for data character transmission or address character transmission. Table 11-3 lists the parity mode and type or the multidrop mode for each combination of the parity mode and the parity type bits.

Table 11-3. PMx and PT Control Bits

PM1	PM0	PARITY MODE	PT	PARITY TYPE
0	0	With Parity	0	Even Parity
0	0	With Parity	1	Odd Parity
0	1	Force Parity	0	Low Parity
0	1	Force Parity	1	High Parity
1	0	No Parity	X	No Parity
1	1	Multidrop Mode	0	Data Character
1	1	Multidrop Mode	1	Address Character

NOTE

For further information on M-Bus system configuration, protocol, and restrictions please refer to the Philip's I^2C standard

12.4 M-BUS PROTOCOL

Normally, a standard communication is composed of four parts: (1) START signal, (2) slave address transmission, (3) data transfer, and (4) STOP signal. They are described briefly in the following sections and illustrated in Figure 12-2.

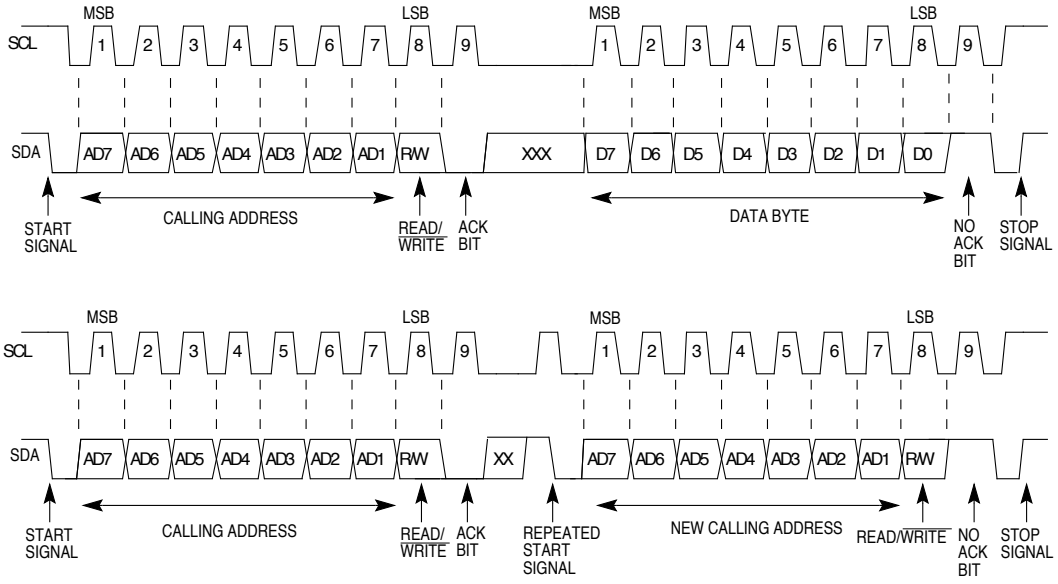

Figure 12-2. M-Bus Standard Communication Protocol

12.4.1 START Signal

When the bus is free, i.e., no master device is engaging the bus (both SCL and SDA lines are at logic high), a master can initiate communication by sending a START signal. As shown in Figure 12-2, a START signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer (each data transfer can contain several bytes of data) and awakens all slaves.

12.4.2 Slave Address Transmission

The first byte of data transfered by the master immediately after the START signal is the slave address. This is a seven-bit calling address followed by a R/$\overline{W}$ bit. The R/$\overline{W}$ bit tells the slave data transfer direction. No two slaves in the system can have the same address.

TXAK — Transmit Acknowledge Enable

This bit specifies the value driven onto SDA during acknowledge cycles for both master and slave receivers. Note that writing this bit only applies when the M-Bus is a receiver, not a transmitter.

1 = No acknowledge signal response is sent (i.e., acknowledge bit = 1)
0 = An acknowledge signal is sent out to the bus at the 9th clock bit after receiving one byte data

RSTA — Repeat Start

Writing a 1 to this bit generates a repeated START condition on the bus, provided it is the current bus master. This bit is always read as a low. Attempting a repeated start at the wrong time, if the bus is owned by another master, results in loss of arbitration. Note that this bit is not readable.

1 = Generate repeat start cycle

12.5.4 M-Bus Status Register (MBSR)

This status register is read-only with the exception of bit 1 (MIF) and bit 4 (MAL), which can be cleared by software. All bits are cleared on reset except bit 7 (MCF) and bit 0 (RXAK), which are set (=1) at reset.

M-Bus Status Register (MBSR) Address MBAR+$1EC

	7	6	5	4	3	2	1	0
	MCF	MAAS	MBB	MAL	-	SRW	MIF	RXAK
RESET	1	0	0	0	0	0	0	1
	Read/Write					Supervisor or User Mode		

MCF — Data Transferring Bit

While one byte of data is being transferred, this bit is cleared. It is set by the falling edge of the 9th clock of a byte transfer.

1 = Transfer complete
0 = Transfer in progress

MAAS — Addressed as a Slave Bit

When its own specific address (M-Bus Address Register) is matched with the calling address, this bit is set. The CPU is interrupted provided the MIEN is set. Next, the CPU must check the SRW bit and set its TX/RX mode accordingly.

Writing to the M-Bus Control Register clears this bit.

1 = Addressed as a slave
0 = Not addressed

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.

Figure14-3 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.

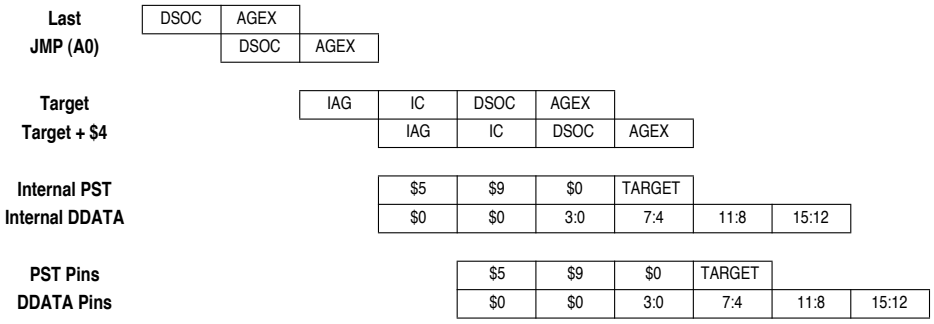

Figure 14-2. Pipeline Timing Example (Debug Output)

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.

Table 14-3. BDM Command Summary (Continued)

COMMAND	MNEMONIC	DESCRIPTION	CPUIMPACT[1]

NOTE:

1. General command effect and/or requirements on CPU operation:
   Halted - The CPU must be halted to perform this command
   Steal - Command generates a bus cycle which is interleaved with CPU accesses
   Parallel - Command is executed in parallel with CPU activity
   Refer to command summaries for detailed operation descriptions.

14.2.3.2 COLDFIRE BDM COMMANDS. All ColdFire Family BDM commands include a 16-bit operation word followed by an optional set of one or more extension words.

15 10	9	8	7 6	5	4	3	2 0
OPERATION	0	R/$\overline{W}$	OP SIZE	0	0	A/$\overline{D}$	REGISTER
EXTENSION WORD(S)							

Operation Field

The operation field specifies the command.

R/$\overline{W}$ Field

The R/$\overline{W}$ field specifies the direction of operand transfer. When the bit is set, the transfer is from the CPU to the development system. When the bit is cleared, data is written to the CPU or to memory from the development system.

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 14-4.

Table 14-4. BDM Size Field Encoding

ENCODING	OPERAND SIZE
00	Byte
01	Word
10	Long
11	Reserved

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

The A/$\overline{D}$ 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.

REVISION NO.: 1.1
REVISION DATE: 6/17/98, 8/28/98
PAGES AFFECTED: SEE RED CHANGE BARS

SECTION 16
ELECTRICAL CHARACTERISTICS

16.1 MAXIMUM RATINGS

16.1.1 Supply, Input Voltage and Storage Temperature

RATING	SYMBOL	VALUE	UNIT
Supply voltage	V_{DD}	-0.3 to +7.0	V
Input voltage	V_{in}	-0.5 to V_{DD} +0.5	V
Storage temperature range	T_{stg}	-55 to 150	°C

The ratings in the above table define maximum conditions under which the MCF5206 device may be subjected without being damaged. However, the device cannot operate normally while being exposed to these electrical extremes.

This device contains circuitry that protects against damage from high static voltages or electrical fields; however, you should take normal precautions to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Operational reliability improves when unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD}).

16.1.2 Operating Temperature

CHARACTERISTIC	SYMBOL	VALUE	UNIT
Maximum operating junction temperature	T_J	TBD	°C
Maximum operating ambient temperature	T_{Amax}	70[a]	°C
Minimum operating ambient temperature	T_{Amin}	0	°C

[a] This published maximum operating ambient temperature should be used only as a system design guideline. All device operating parameters are guaranteed only when the junction temperature lies within the specified range.

NOTE

At press time power dissipation figures were not available. Refer to the World Wide Web site at http://www.mot.com/ColdFire for the latest accurate power dissipation information for the MCF5206 processor.

16.3.12 M-BUS Module AC Timing Specifications

16.3.12.1 INPUT TIMING SPECIFICATIONS BETWEEN SCL AND SDA.

NAME	CHARACTERISTIC	16.67 MHz		25 MHz		33.33 MHz		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
M1[1]	Start condition hold time	2	–	2	–	2	–	CLKs
M2[1]	Clock low period	8	–	8	–	8	–	CLKs
M3	SCL/SDA rise time (from V_l=0.5V to V_h=2.4V)	–	1	–	1	–	1	µs
M4	Data hold time	0	–	0	–	0	–	ns
M5	SCL/SDA fall time (from V_h=2.4V to V_l=0.5V)	–	1	–	1	–	1	µs
M6[1]	Clock high time	4	–	4	–	4	–	CLKs
M7	Data setup time	0	–	0	–	0	–	ns
M8[1]	Start condition setup time (for repeated start condition only)	2	–	2	–	2	–	CLKs
M9[1]	Stop condition setup time	2	–	2	–	2	–	CLKs

[1] Note: Units for these specifications are in processor CLK units.

16.3.12.2 OUTPUT TIMING SPECIFICATIONS BETWEEN SCL AND SDA.

NAME	CHARACTERISTIC	16.67 MHz		25 MHz		33.33 MHz		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
M1[1,2]	Start condition hold time	6	–	6	–	6	–	CLKs
M2[1,2]	Clock low period	10	–	10	–	10	–	CLKs
M3[3]	SCL/SDA rise time (from V_l=0.5V to V_h=2.4V)	–	–	–	–	–	–	ms
M4[1,2]	Data hold time	7	–	7	–	7	–	CLKs
M5[4]	SCL/SDA fall time (from V_h=2.4V to V_l=0.5V)	–	TBD	–	TBD	–	TBD	ns
M6[1,2]	Clock high time	10	–	10	–	10	–	CLKs
M7[1,2]	Data setup time	2	–	2	–	2	–	CLKs
M8[1,2]	Start condition setup time (for repeated start condition only)	20	–	20	–	20	–	CLKs
M9[1,2]	Stop condition setup time	10	–	10	–	10	–	CLKs

[1] Note: Units for these specifications are in processor CLK units.
[2] Note: Output numbers are dependent on the value programmed into the MFDR; an MFDR programmed with the maximum frequency (MFDR = 0x20) will result in minimum output timings as shown in the above table. The MBUS interface is designed to

Appendix A

ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE	ACCESS
MBAR + $05C	DCMR1	32	DRAM Controller Mask Register - Bank 1	Master Reset: uninitialized Normal Reset: uninitialized	R/W
MBAR + $063	DCCR1	8	DRAM Controller Control Register - Bank 1	Master Reset: $00 Normal Reset: $00	R/W
MBAR + 64	CSAR0	16	Chip-Select Address Register - Bank 0	0000	R/W
MBAR + 68	CSMR0	32	Chip-Select Mask Register - Bank 0	00000000	R/W
MBAR + $06E	CSCR0	16	Chip-Select Control Register - Bank 0	3C1F, 3C5F, 3C9F, 3CDF, 3D1F, 3D5F, 3D9F, or 3DDF AA set by IRQ7 at reset PS1 set by IRQ4 at reset PS0 set by IRQ1 at reset	R/W
MBAR + $070	CSAR1	16	Chip-Select Address Register - Bank 1	uninitialized	R/W
MBAR + $074	CSMR1	32	Chip-Select Mask Register - Bank 1	uninitialized	R/W
MBAR + $07A	CSCR1	16	Chip-Select Control Register - Bank 1	uninitialized (except BRST=ASET=WRAH=RDAH=WR=RD=0)	R/W
MBAR + $07C	CSAR2	16	Chip-Select Address Register - Bank 2	uninitialized	R/W
MBAR + $080	CSMR2	32	Chip-Select Mask Register - Bank 2	uninitialized	R/W
MBAR + $086	CSCR2	16	Chip-Select Control Register - Bank 2	uninitialized (except BRST=ASET=WRAH=RDAH=WR=RD=0)	R/W
MBAR + $088	CSAR3	16	Chip-Select Address Register - Bank 3	uninitialized	R/W
MBAR + $08C	CSMR3	32	Chip-Select Mask Register - Bank 3	uninitialized	R/W
MBAR + $092	CSCR3	16	Chip-Select Control Register - Bank 3	uninitialized (except BRST=ASET=WRAH=RDAH=WR=RD=0)	R/W
MBAR + $094	CSAR4	16	Chip-Select Address Register - Bank 4	uninitialized	R/W
MBAR + $098	CSMR4	32	Chip-Select Mask Register - Bank 4	uninitialized	R/W
MBAR + $09E	CSCR4	16	Chip-Select Control Register - Bank 4	uninitialized (except BRST=ASET=WRAH=RDAH=WR=RD=0)	R/W
MBAR + $0A0	CSAR5	16	Chip-Select Address Register - Bank 5	uninitialized	R/W
MBAR + $0A4	CSMR5	32	Chip-Select Mask Register - Bank 5	uninitialized	R/W
MBAR + $0AA	CSCR5	16	Chip-Select Control Register - Bank 5	uninitialized (except BRST=ASET=WRAH=RDAH=WR=RD=0)	R/W
MBAR + $0AC	CSAR6	16	Chip-Select Address Register - Bank 6	uninitialized	R/W
MBAR + $0B0	CSMR6	32	Chip-Select Mask Register - Bank 6	uninitialized	R/W
MBAR + $0B6	CSCR6	16	Chip-Select Control Register - Bank 6	uninitialized (except BRST=ASET=WRAH=RDAH=WR=RD=0)	R/W
MBAR + $0B8	CSAR7	16	Chip-Select Address Register - Bank 7	uninitialized	R/W
MBAR + $0BC	CSMR7	32	Chip-Select Mask Register - Bank 7	uninitialized	R/W
MBAR + $0C2	CSCR7	16	Chip-Select Control Register - Bank 7	uninitialized (except BRST=ASET=WRAH=RDAH=WR=RD=0)	R/W
MBAR + $0C6	DMCR	16	Default Memory Control Register	0000	R/W

APPENDIX B
PORTING FROM M68K ARCHITECTURE

This section is an overview of the issues encountered when porting embedded development tools to work with the ColdFire processor when starting with the M68K architecture.

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 M68K architecture can also be used without modification with ColdFire.

Although you can use an existing M68K assembler and disassembler with ColdFire, Motorola recommends modifying the assembler so that nonColdFire 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 will run on existing M68K hardware to test the conversion from M68K code to the proper ColdFire subset. This step verifies that the process of code conversion does not introduce new errors.