Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	Coldfire V4
Core Size	32-Bit Single-Core
Speed	220MHz
Connectivity	EBI/EMI, I²C, UART/USART
Peripherals	DMA, WDT
Number of I/O	16
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	208-BFQFP
Supplier Device Package	208-FQFP (28x28)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mcf5407cai220

CONTENTS

Chapter 14
UART Modules

2.5 Addressing Mode Summary

Addressing modes are categorized by how they are used. Data addressing modes refer to data operands. Memory addressing modes refer to memory operands. Alterable addressing modes refer to alterable (writable) data operands. Control addressing modes refer to memory operands without an associated size.

These categories sometimes combine to form more restrictive categories. Two combined classifications are alterable memory (both alterable and memory) and data alterable (both alterable and data). Twelve of the most commonly used effective addressing modes from the M68000 Family are available on ColdFire microprocessors. Table 2-5 summarizes these modes and their categories.

Table 2-5. ColdFire Effective Addressing Modes

Addressing Modes	Syntax	Mode Field	Reg. Field	Category			
				Data	Memory	Control	Alterable
Register direct Data Address	 Dn An	 000 001	 reg. no. reg. no.	 X —	 — —	 — —	 X X
Register indirect Address Address with Postincrement Address with Predecrement Address with Displacement	 (An) (An)+ –(An) (d_{16}, An)	 010 011 100 101	 reg. no. reg. no. reg. no. reg. no.	 X X X X	 X X X X	 X — — X	 X X X X
Address register indirect with scaled index 8-bit displacement	(d_8, An, Xi*SF)	110	reg. no.	X	X	X	X
Program counter indirect with displacement	(d_{16}, PC)	111	010	X	X	X	—
Program counter indirect with scaled index 8-bit displacement	(d_8, PC, Xi*SF)	111	011	X	X	X	—
Absolute data addressing Short Long	 (xxx).W (xxx).L	 111 111	 000 001	 X X	 X X	 X X	 — —
Immediate	#<xxx>	111	100	X	X	—	—

2.6 Instruction Set Summary

The ColdFire instruction set is a simplified version of the M68000 instruction set. The removed instructions include BCD, bit field, logical rotate, decrement and branch, and integer multiply with a 64-bit result. Nine new MAC instructions have been added.

Table 2-6 lists notational conventions used throughout this manual.

MVS — Move with Sign Extend — MVS

Operation: (Source with sign extension) → Destination

Assembler Syntax: MVS <ea>y,Dx

Attributes: Size = byte, word

Description: Sign-extend the source operand and move to the destination register. For the byte operation, bit 7 of the source is copied to bits 31–8 of the destination. For the word operation, bit 15 of the source is copied to bits 31-16 of the destination.

Condition Codes:

X	N	Z	V	C
—	*	*	0	0

X Not affected
N Set if the result is negative; cleared otherwise
Z Set if the result is zero; cleared otherwise
V Always cleared
C Always cleared

Instruction Format:

15	14	13	12	11 10 9	8	7	6	5 4 3	2 1 0
0	1	1	1	REGISTER	1	0	SIZE	EFFECTIVE ADDRESS	
								MODE	REGISTER

Instruction Fields:

- Size field—specifies the size of the operation
  0 byte operation
  1 word operation
- Register field—specifies a data register as the destination.
- Effective address field—specifies the source operand; use only data addressing modes from the following table:

Addressing Mode	Mode	Register		Addressing Mode	Mode	Register
Dy	000	reg. number:Dy		(d_8,Ay,Xi)	110	reg. number:Ay
Ay	001	reg. number:Ay		(xxx).W	111	000
(Ay)	010	reg. number:Ay		(xxx).L	111	001
(Ay) +	011	reg. number:Ay		#<data>	111	100
– (Ay)	100	reg. number:Ay		(d_{16},PC)	111	010
(d_{16},Ay)	101	reg. number:Ay		(d_8,PC,Xi)	111	011

MVS	V2, V3 Core	V4 Core
Opcode present	No	Yes
Operand sizes supported	—	.b, .w

SATS Signed Saturate SATS

Operation:

```
If CCR.V == 1,
then if Dx[31] == 0,
        then Dx[31:0] = 0x80000000
        else Dx[31:0] = 0x7FFFFFFF
else Dx[31:0] is unchanged
```

Assembler Syntax: SATS Dx

Attributes: Size = long

Description: Update the destination register only if the overflow bit of the CCR is set. If the operand is negative, then set the result to greatest positive number, otherwise set the result to the largest negative value. The condition codes are set according to the result.

Condition Codes:

X	N	Z	V	C
—	*	*	0	0

X Not affected
N Set if the result is negative; cleared otherwise
Z Set if the result is zero; cleared otherwise
V Always cleared
C Always cleared

Instruction Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	0	0	1	0	0	0	0	REGISTER		

Instruction Fields:

- Register field—Specifies the destination data register.

SATS	V2, V3 Core	V4 Core
Opcode present	No	Yes
Operand sizes supported	—	.l

```
                                ;instruction cache
        lea     16(a0),a0       ;increment address to next line
        subq.l  #1,d0           ;decrement loop counter
        bne.b   instCacheLoop   ;if done, then exit, else continue
; A 8K region was loaded into levels 0 and 1 of the 16-Kbyte instruction cache.
; lock it!

        move.l  #0xa2088800,d0  ;set the instruction cache lock bit
        movec   d0,cacr         ;in the CACR
        rts
```

4.12 Cache Operation Summary

This section gives operational details for the cache and presents instruction and data cache-line state diagrams.

4.12.1 Instruction Cache State Transitions

Because the instruction cache does not support writes, it supports fewer operations than the data cache. As Figure 4-12 shows, an instruction cache line can be in one of two states, valid or invalid. Modified state is not supported. Transitions are labeled with a capital letter indicating the previous state and with a number indicating the specific case listed in Table 4-6. These numbers correspond to the equivalent operations on data caches, described in Section 4.12.2, “Data Cache State Transitions.”

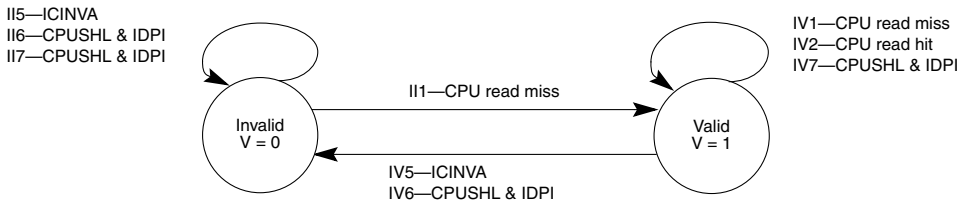

Figure 4-12. Instruction Cache Line State Diagram

Table 4-6 describes the instruction cache state transitions shown in Figure 4-12.

Table 4-6. Instruction Cache Line State Transitions

Access	Current State			
	Invalid (V = 0)		Valid (V = 1)	
Read miss	II1	Read line from memory and update cache; supply data to processor; go to valid state.	IV1	Read new line from memory and update cache; supply data to processor; stay in valid state.
Read hit	II2	Not possible	IV2	Supply data to processor; stay in valid state.
Write miss	II3	Not possible	IV3	Not possible
Write hit	II4	Not possible	IV4	Not possible

The Version 2 ColdFire core implemented the original debug architecture, now called Revision A. Based on feedback from customers and third-party developers, enhancements have been added to succeeding generations of ColdFire cores. The Version 3 core implements the Revision B of the debug architecture, providing more flexibility for configuring the hardware breakpoint trigger registers and removing the restrictions involving concurrent BDM processing while hardware breakpoint registers are active.

The MCF5407 core implements Revision C of the debug architecture, which more than doubles the on-chip breakpoint registers and provides an ability to interrupt debug service routines. For Revision C, the revision level bit, CSR[HRL], is 2. See Section 5.4.4, “Configuration/Status Register (CSR).”

5.2 Signal Descriptions

Table 5-1 describes debug module signals. All ColdFire debug signals are unidirectional and related to a rising edge of the processor core’s clock signal. The standard 26-pin debug connector is shown in Section 5.7, “Motorola-Recommended BDM Pinout.”

Table 5-1. Debug Module Signals

Signal	Description
Development Serial Clock (DSCLK)	Internally synchronized input that clocks the serial communication port to the debug module. Maximum frequency is 1/5 the processor CLK speed. At the synchronized rising edge of DSCLK, the data input on DSI is sampled and DSO changes state. The logic level on DSCLK is validated if it has the same value on two consecutive rising CLKIN edges.
Development Serial Input (DSI)	Internally synchronized input that provides data input for the serial communication port to the debug module.
Development Serial Output (DSO)	Provides serial output communication for debug module responses. DSO is registered internally.
Breakpoint ($\overline{\text{BKPT}}$)	Used to request a manual breakpoint. Assertion of $\overline{\text{BKPT}}$ puts the processor into a halted state after the current instruction completes. Halt status is reflected on processor status/debug data signals (PSTDDATA[7:0]) as the value 0xF. If CSR[BKD] is set (disabling normal $\overline{\text{BKPT}}$ functionality), asserting $\overline{\text{BKPT}}$ generates a debug interrupt exception in the processor.
Processor Status Clock (PSTCLK)	Half-speed version of the processor clock. Its rising edge appears in the center of the two processor-cycle window of valid PSTDDATA output. See Figure 5-2. Because debug trace port signals change on alternate processor cycles and are unrelated to external bus frequency, PSTCLK helps the development system sample PSTDDATA values. If real-time trace is not used, setting CSR[PCD] keeps PSTCLK and PSTDDATA outputs from toggling without disabling triggers. Non-quiescent operation can be reenabled by clearing CSR[PCD], although the emulator must resynchronize with the PSTDDATA output. PSTCLK starts clocking only when the first non-zero PST value (0xC, 0xD, or 0xF) occurs during system reset exception processing. Table 5-4 describes PST values. Chapter 7, “Phase-Locked Loop (PLL),” describes PSTCLK generation.
Processor Status/Debug Data (PSTDDATA[7:0])	These outputs indicate both processor status and captured address and data values and are discussed more thoroughly in Section 5.2.1, “Processor Status/Debug Data (PSTDDATA[7:0]).

Table 5-11. CSR Field Descriptions (Continued)

Bit	Name	Description
12–11	DDC	Debug data control. Controls operand data capture for PSTDDATA, which displays the number of bytes defined by the operand reference size before the actual data; byte displays 8 bits, word displays 16 bits, and long displays 32 bits (one nibble at a time across multiple clock cycles). See Table 5-4. 00 No operand data is displayed. 01 Capture all write data. 10 Capture all read data. 11 Capture all read and write data.
10	UHE	User halt enable. Selects the CPU privilege level required to execute the HALT instruction. 0 HALT is a supervisor-only instruction. 1 HALT is a supervisor/user instruction.
9–8	BTB	Branch target bytes. Defines the number of bytes of branch target address PSTDDATA displays. 00 0 bytes 01 Lower 2 bytes of the target address 10 Lower 3 bytes of the target address 11 Entire 4-byte target address See Section 5.3.1, “Begin Execution of Taken Branch (PST = 0x5).”
7	—	Reserved, should be cleared.
6	NPL	Non-pipelined mode. Determines whether the core operates in pipelined or mode. 0 Pipelined mode 1 Nonpipelined mode. The processor effectively executes one instruction at a time with no overlap. This adds at least 5 cycles to the execution time of each instruction. Instruction folding is disabled. Given an average execution latency of 1.6, throughput in non-pipeline mode would be 6.6, approximately 25% or less compared to pipelined performance. Regardless of the NPL state, a triggered PC breakpoint is always reported before the triggering instruction executes. In normal pipeline operation, the occurrence of an address and/or data breakpoint trigger is imprecise. In non-pipeline mode, triggers are always reported before the next instruction begins execution and trigger reporting can be considered precise. An address or data breakpoint should always occur before the next instruction begins execution. Therefore the occurrence of the address/data breakpoints should be guaranteed.
5	—	Reserved, should be cleared.
4	SSM	Single-step mode. Setting SSM puts the processor in single-step mode. 0 Normal mode. 1 Single-step mode. The processor halts after execution of each instruction. While halted, any BDM command can be executed. On receipt of the GO command, the processor executes the next instruction and halts again. This process continues until SSM is cleared.
3–0	—	Reserved, should be cleared.

5.4.5 Data Breakpoint/Mask Registers (DBR/DBR1, DBMR/DBMR1)

The data breakpoint registers (DBR/DBR1), Figure 5-10, specify data patterns used as part of the trigger into debug mode. Only DBR*n* bits not masked with a corresponding zero in DBMR*n* are compared with the data from the processor’s local bus, as defined in TDR.

5.5.3.3.11 Write Control Register (WCREG)

The operand (longword) data is written to the specified control register. The write alters all 32 register bits.

Command/Result Formats:

	15–12	11–8	7–4	3–0
Command	0x2	0x8	0x8	0x0
	0x0	0x0	0x0	0x0
	0x0	Rc		
Result	D[31:16]			
	D[15:0]			

Figure 5-40. WCREG Command/Result Formats

Command Sequence:

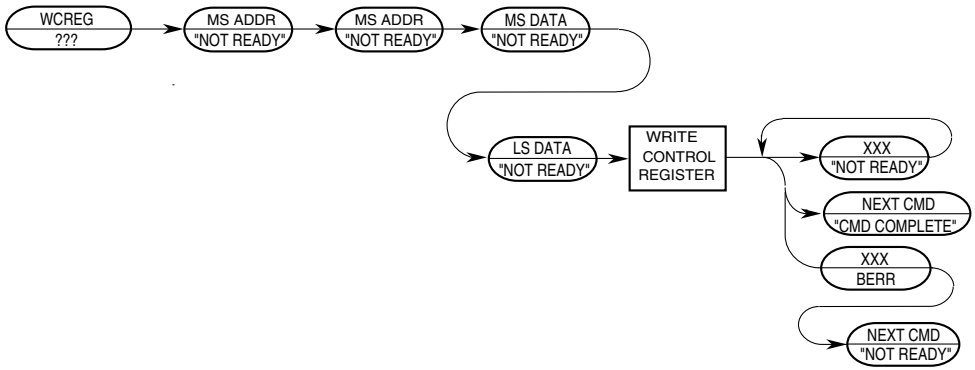

Figure 5-41. WCREG Command Sequence

Operand Data: This instruction requires two longword operands. The first selects the register to which the operand data is to be written; the second contains the data.

Result Data: Successful write operations return 0xFFFF. Bus errors on the write cycle are indicated by the setting of bit 16 in the status message and by a data pattern of 0x0001.

Table 5-28. PSTDDATA Specification for Supervisor-Mode Instructions

Instruction	Syntax	PSTDDATA
cpushl		PSTDDATA = 1
halt		PSTDDATA = 1, PSTDDATA = F
intouch		PSTDDATA = 1
move.w	SR,Dx	PSTDDATA = 1
move.w	{Dy,#imm},SR	PSTDDATA = 1, {3}
movec	Ry,Rc	PSTDDATA = 1
rte		PSTDDATA = 7, {B, source operand}, {3},{B, source operand}, {DD}, PSTDDATA = 5, {[9AB], target address}
stop	#imm	PSTDDATA = 1, PSTDDATA = E
wdebug	<ea>y	PSTDDATA = 1, {B, source, B, source}

The move-to-SR and RTE instructions include an optional PSTDDATA = 0x3 value, indicating an entry into user mode. Additionally, if the execution of a RTE instruction returns the processor to emulator mode, a multiple-cycle status of 0xD is signaled.

Similar to the exception processing mode, the stopped state (PSTDDATA = 0xE) and the halted state (PSTDDATA = 0xF) display this status throughout the entire time the ColdFire processor is in the given mode.

Table 10-9. CSMRn Field Descriptions (Continued)

Bits	Name	Description
5–1	C/I, SC, SD, UC, UD	Address space mask bits. These bits determine whether the specified accesses can occur to the address space defined by the BAM for this chip select. C/I CPU space and 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 0 The address space assigned to this chip select. is available to the specified access type. 1 The address space assigned to this chip select. is not available (masked) to the specified access type. If this address space is accessed, chip select is not activated and a regular external bus cycle occurs. Note that if if AM = 0, SC, SD, UC, and UD are ignored in the chip select decode on external master or DMA access.
0	V	Valid bit. Indicates whether the corresponding CSAR, CSMR, and CSCR contents are valid. Programmed chip selects do not assert until V is set (except for $\overline{CS0}$, which acts as the global chip select). Reset clears each CSMRn[V]. 0 Chip select invalid 1 Chip select valid

10.4.1.3 Chip-Select Control Registers (CSCR0–CSCR7)

Each chip-select control register, Figure 10-4, controls the auto acknowledge, external master support, port size, burst capability, and activation of each chip select. Note that to support the global chip select, $\overline{CS0}$, the CSCR0 reset values differ from the other CSCRs. $\overline{CS0}$ allows address decoding for boot ROM before system initialization.

	15	14	13	10	9	8	7	6	5	4	3	2	0
Field	—		WS		—	AA	PS1	PS0	BEM	BSTR	BSTW	—	
Reset: CSCR0	—		11_11		—	D7	D6	D5	D3	—			
Reset: Other CSCRs	Unitialized												
R/W	R/W												
Address	0x08A (CSCR0); 0x096 (CSCR1); 0x0A2 (CSCR2); 0x0AE (CSCR3); 0x0BA (CSCR4); 0x0C6 (CSCR5); 0x0D2 (CSCR6); 0x0DE (CSCR7)												

Figure 10-4. Chip-Select Control Registers (CSCR0–CSCR7)

Table 10-10 describes CSCR*n* fields.

Table 10-10. CSCRn Field Descriptions

Bits	Name	Description
15–14	—	Reserved, should be cleared.
13–10	WS	Wait states. The number of wait states inserted before an internal transfer acknowledge is generated (WS = 0 inserts zero wait states, WS = 0xF inserts 15 wait states). If AA = 0, $\overline{TA}$ must be asserted by the external system regardless of the number of wait states generated. In that case, the external transfer acknowledge ends the cycle. An external $\overline{TA}$ supersedes the generation of an internal $\overline{TA}$.
9	—	Reserved, should be cleared.

```
CSAR5 EQU MBARx+0x0BC   ;Chip select 5 address register
CSMR5 EQU MBARx+0x0C0   ;Chip select 5 mask register
CSCR5 EQU MBARx+0x0C6   ;Chip select 5 control register

CSAR6 EQU MBARx+0x0C8   ;Chip select 6 address register
CSMR6 EQU MBARx+0x0CC   ;Chip select 6 mask register
CSCR6 EQU MBARx+0x0D2   ;Chip select 6 control register

CSAR7 EQU MBARx+0x0D4   ;Chip select 7 address register
CSMR7 EQU MBARx+0x0D8   ;Chip select 7 mask register
CSCR7 EQU MBARx+0x0DE   ;Chip select 7 control register

; All other chip selects should be programmed and made valid before global
; chip select is de-activated by validating CS0

; Program Chip Select 3 Registers
move.w  #0x0040,D0      ;CSAR3 base address 0x00400000
move.w  D0,CSAR3

move.w  #0x00A0,D0      ;CSCR3 = no wait states, AA=0, PS=16-bit, BEM=1,
move.w  D0,CSCR3        ;BSTR=0, BSTW=0

move.l #0x001F016B,D0   ;Address range from 0x00400000 to 0x005FFFFF
move.l D0,CSMR3         ;WP,EM,C/I,SD,UD,V=1; SC,UC=0

; Program Chip Select 2 Registers

move.w  #0x0020,D0      ;CSAR2 base address 0x00200000 (to 0x003FFFFF)
move.w  D0,CSAR2

move.w  #0x0538,D0      ;CSCR2 = 1 wait state, AA=1, PS=32-bit, BEM=1,
move.w  D0,CSCR2        ;BSTR=1, BSTW=1

move.l  #0x001F0001,D0  ;Address range from 0x00200000 to 0x003FFFFF
move.l  D0,CSMR2        ;WP,EM,C/I,SC,SD,UC,UD=0; V=1

; Program Chip Select 1 Registers

move.w  #0x0000,D0      ;CSAR1 base addresses 0x00000000 (to 0x001FFFFF)
move.w  D0,CSAR1        ;and 0x80000000 (to 0x801FFFFF)

move.w  #0x09B0,D0      ;CSCR1 = 2 wait states, AA=1, PS=16-bit, BEM=1,
move.w  D0,CSCR1        ;BSTR=1, BSTW=0

move.l  #0x801F0001,D0  ;Address range from 0x00000000 to 0x001FFFFF and
move.l  D0,CSMR1        ;0x80000000 to 0x801FFFFF
                        ;WP, EM, C/I, SC, SD, UC, UD=0, V=1

; Program Chip Select 0 Registers

move.w  #0x0080,D0      ;CSAR0 base address 0x00800000 (to 0x009FFFFF)
move.w  D0,CSAR0

move.w  #0x0D80,D0      ;CSCR0 = three wait states, AA=1, PS=16-bit, BEM=0,
move.w  D0,CSCR0        ;BSTR=0, BSTW=0

; Program Chip Select 0 Mask Register (validate chip selects)

move.l  #0x001F0001,D0  ;Address range from 0x00800000 to 0x009FFFFF
move.l  D0,CSMR0        ;WP,EM,C/I,SC,SD,UC,UD=0; V=1
```

Table 11-9. DRAM Addressing for 32-Bit Wide Memories

MCF5407 Address Pin	MCF5407 Address Bit Driven for $\overline{RAS}$	MCF5407 Address Bit Driven when CAS is Asserted	Memory Size
15	15	2	Base Memory Size of 64 Kbytes
14	14	3	
13	13	4	
12	12	5	
11	11	6	
10	10	7	
9	9	8	
17	17	16	256 Kbytes
19	19	18	1 Mbyte
21	21	20	4 Mbytes
23	23	22	16 Mbytes
25	25	24	64 Mbytes

11.3.3.1 Non-Page-Mode Operation

In non-page mode, the simplest mode, the DRAM controller provides termination and runs a separate bus cycle for each data transfer. Figure 11-5 shows a non-page-mode access in which a DRAM read is followed by a write. Addresses for a new bus cycle are driven at the rising clock edge.

For a DRAM block hit, the associated $\overline{RAS}$ is driven at the next falling edge. Here DACR*n*[RCD] = 0, so the address is multiplexed at the next rising edge to provide the column address. The required $\overline{CAS}$ signals are then driven at the next falling edge and remain asserted for the period programmed in DACR*n*[CAS]. Here, DACR*n*[RNCN] = 1, so it is precharged one clock before $\overline{CAS}$ is negated. On a read, data is sampled on the last rising edge of the clock that $\overline{CAS}$ is valid.

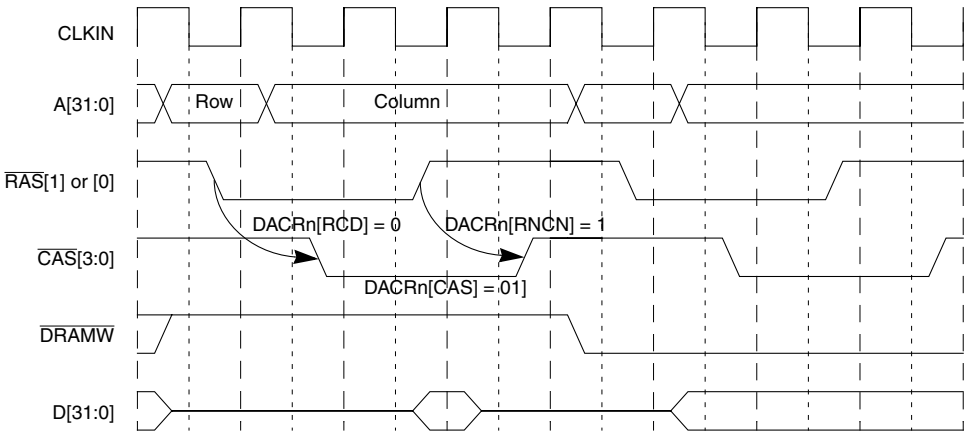

Figure 11-5. Basic Non-Page-Mode Operation RCD = 0, RNCN = 1 (4-4-4-4)

Table 11-14. DMR0/DMR1 Field Descriptions

<table>
<tr><th>Bits</th><th>Name</th><th>Description</th></tr>
<tr><td>31–18</td><td>BAM</td><td>Base address mask. Masks the associated DACRn[BA]. Lets the DRAM controller connect to various DRAM sizes. Mask bits need not be contiguous (see Section 11.5, “SDRAM Example.”)
0 The associated address bit is used in decoding the DRAM hit to a memory block.
1 The associated address bit is not used in the DRAM hit decode.</td></tr>
<tr><td>17–9</td><td>—</td><td>Reserved, should be cleared.</td></tr>
<tr><td>8</td><td>WP</td><td>Write protect. Determines whether the associated block of DRAM is write protected.
0 Allow write accesses
1 Ignore write accesses. The DRAM controller ignores write accesses to the memory block and an address exception occurs. Write accesses to a write-protected DRAM region are compared in the chip select module for a hit. If no hit occurs, an external bus cycle is generated. If this external bus cycle is not acknowledged, an access exception occurs.</td></tr>
<tr><td>7</td><td>—</td><td>Reserved, should be cleared.</td></tr>
<tr><td>6–1</td><td>AMx</td><td>Address modifier masks. Determine which accesses can occur in a given DRAM block.
0 Allow access type to hit in DRAM
1 Do not allow access type to hit in DRAM
<table>
<tr><th>Bit</th><th>Associated Access Type</th><th>Access Definition</th></tr>
<tr><td>C/I</td><td>CPU space/interrupt acknowledge</td><td>MOVEC instruction or interrupt acknowledge cycle</td></tr>
<tr><td>AM</td><td>Alternate master</td><td>External or DMA master</td></tr>
<tr><td>SC</td><td>Supervisor code</td><td>Any supervisor-only instruction access</td></tr>
<tr><td>SD</td><td>Supervisor data</td><td>Any data fetched during the instruction access</td></tr>
<tr><td>UC</td><td>User code</td><td>Any user instruction</td></tr>
<tr><td>UD</td><td>User data</td><td>Any user data</td></tr>
</table></td></tr>
<tr><td>0</td><td>V</td><td>Valid. Cleared at reset to ensure that the DRAM block is not erroneously decoded.
0 Do not decode DRAM accesses.
1 Registers controlling the DRAM block are initialized; DRAM accesses can be decoded.</td></tr>
</table>

11.4.4 General Synchronous Operation Guidelines

To reduce system logic and to support a variety of SDRAM sizes, the DRAM controller provides SDRAM control signals as well as a multiplexed row address and column address to the SDRAM.

When SDRAM blocks are accessed, the DRAM controller can operate in either burst or continuous page mode. The following sections describe the DRAM controller interface to SDRAM, the supported bus transfers, and initialization.

11.4.4.1 Address Multiplexing

Table 11-6 shows the generic address multiplexing scheme for SDRAM configurations. All possible address connection configurations can be derived from this table.

The following tables provide a more comprehensive, step-by-step way to determine the correct address line connections for interfacing the MCF5407 to SDRAM. To use the

- Single-address transfers—An external device can initiate a single-address transfer by asserting $\overline{\text{DREQ}}$. The MCF5407 provides address and control signals for single-address transfers. The external device reads to or writes from the specified address, as Figure 12-4 shows. External logic is required.

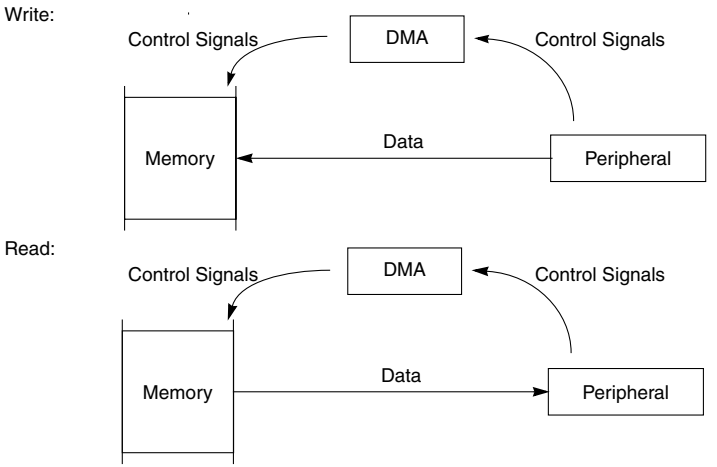

Figure 12-4. Single-Address Transfers

Any operation involving the DMA module follows the same three steps:

1. Channel initialization—Channel registers are loaded with control information, address pointers, and a byte-transfer count.
2. Data transfer—The DMA accepts requests for operand transfers and provides addressing and bus control for the transfers.
3. Channel termination—Occurs after the operation is finished, either successfully or due to an error. The channel indicates the operation status in the channel's DSR, described in Section 12.4.5, "DMA Status Registers (DSR0–DSR3)."

12.4 DMA Controller Module Programming Model

This section describes each internal register and its bit assignment. Note that there is no way to prevent a write to a control register during a DMA transfer. Table 12-3 shows the mapping of DMA controller registers.

Table 12-4. DCRn Field Descriptions (Continued)

Bits	Name	Description
15	AT	DMA acknowledge type. Controls whether acknowledge information is provided for the entire transfer or only the final transfer. 0 Entire transfer. DMA acknowledge information is displayed anytime the channel is selected as the result of an external request. 1 Final transfer (when BCR reaches zero). For dual-address transfer, the acknowledge information is displayed for both the read and write cycles.
14–0	—	Reserved, should be cleared.

12.4.5 DMA Status Registers (DSR0–DSR3)

In response to an event, the DMA controller writes to the appropriate DSR*n* bit, Figure 12-9. Only a write to DSR*n*[DONE] results in action.

	7	6	5	4	3	2	1	0
Field	—	CE	BES	BED	—	REQ	BSY	DONE
Reset	—	0	0	0	—	0	0	0
R/W	R/W							
Address	MBAR + 0x310, 0x350, 0x390, 0x3D0							

Figure 12-9. DMA Status Registers (DSRn)

Table 12-5 describes DSR*n* fields.

Table 12-5. DSRn Field Descriptions

Bits	Name	Description
7	—	Reserved, should be cleared.
6	CE	Configuration error. Occurs when BCR, SAR, or DAR does not match the requested transfer size, or if BCR = 0 when the DMA receives a start condition. CE is cleared at hardware reset or by writing a 1 to DSR[DONE]. 0 No configuration error exists. 1 A configuration error has occurred.
5	BES	Bus error on source 0 No bus error occurred. 1 The DMA channel terminated with a bus error either during the read portion of a transfer or during an access in single-address mode (SAA = 1).
4	BED	Bus error on destination 0 No bus error occurred. 1 The DMA channel terminated with a bus error during the write portion of a transfer.
3	—	Reserved, should be cleared.
2	REQ	Request 0 No request is pending or the channel is currently active. Cleared when the channel is selected. 1 The DMA channel has a transfer remaining and the channel is not selected.

Chapter 15
Parallel Port (General-Purpose I/O)

This chapter describes the operation and programming model of the parallel port pin assignment, direction-control, and data registers. It includes a code example for setting up the parallel port.

15.1 Parallel Port Operation

The MCF5407 parallel port module has 16 signals, which are programmed as follows:

- The pin assignment register (PAR) selects the function of the 16 multiplexed pins.
- Port A data direction register (PADDR) determines whether pins configured as parallel port signals are inputs or outputs.
- The Port A data register (PADAT) shows the status of the parallel port signals.

The operations of the PAR, PADDR, and PADAT are described in the following sections.

15.1.1 Pin Assignment Register (PAR)

The pin assignment register (PAR), which is part of the system integration module (SIM), defines how each PAR bit determines each pin function, as shown in Figure 15-1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	PAR15	PAR14	PAR13	PAR12	PAR11	PAR10	PAR9	PAR8	PAR7	PAR6	PAR5	PAR4	PAR3	PAR2	PAR1	PAR0
PAR[n] = 0	PP15	PP14	PP13	PP12	PP11	PP10	PP9	PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0
PAR[n] = 1	A31	A30	A29	A28	A27	A26	A25	A24	$\overline{\text{TIP}}$	$\overline{\text{DREQ0}}$	$\overline{\text{DREQ1}}$	TM2	TM1/ DACK1	TM0/ DACK0	TT1	TT0
Reset	Determined by driving D4/ADDR_CONFIG with a 1 or 0 when $\overline{\text{RSTI}}$ negates. The system is configured as PP[15:0] if D4 is low; otherwise alternate pin functions selected by PAR[n] = 1 are used.															
R/W	R/W															
Address	Address MBAR + 0x004															

Figure 15-1. Parallel Port Pin Assignment Register (PAR)

If PP[9:8]/A[25:24] are unavailable because A[25:0] are needed for external addressing, PP[15:10]/A[31:26] can be configured as general-purpose I/O. Table 15-1 summarizes MCF5407 parallel port pins, described in detail in Chapter 17, “Signal Descriptions.”

shown in Figure 18-25, the MCF5407 continues to assert $\overline{BD}$ until the completion of the bus cycle. If $\overline{BG}$ is negated by the end of the bus cycle, the MCF5407 negates $\overline{BD}$. While $\overline{BG}$ is asserted, $\overline{BD}$ remains asserted to indicate the MCF5407 is master, and it continuously drives the address bus, attributes, and control signals.

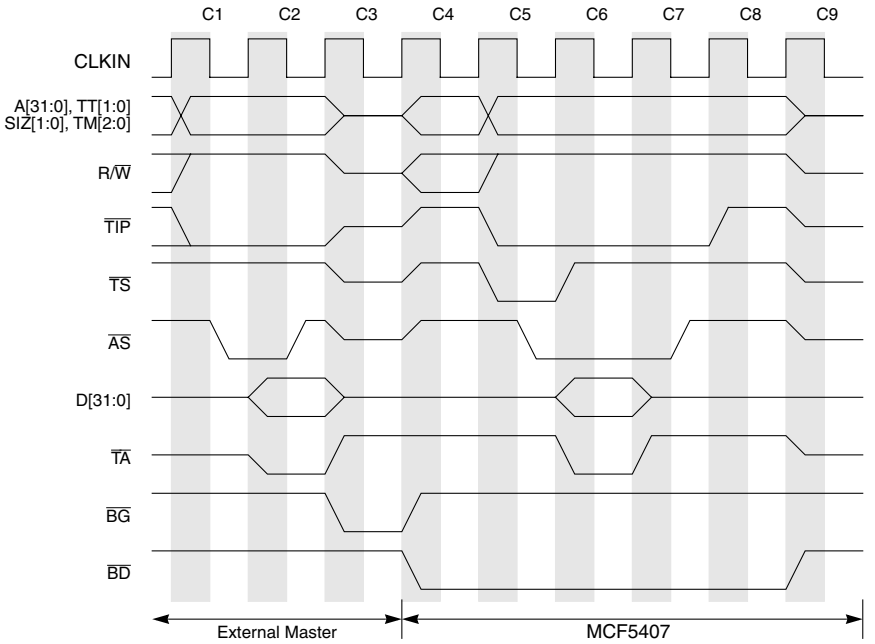

Figure 18-27. Two-Wire Bus Arbitration with Bus Request Asserted

In the second situation, the bus is granted to the MCF5407, but it does not have an internal bus request pending, so it takes implicit bus mastership. The MCF5407 does not drive the bus and does not assert $\overline{BD}$ if the bus has an implicit master. If an internal bus request is generated, the MCF5407 assumes explicit bus mastership. If explicit mastership was assumed because an internal request was generated, the MCF5407 immediately begins an access and asserts $\overline{BD}$.

In Figure 18-28, the external device is bus master during C1 and C2. During C3 the external device releases control of the bus by asserting $\overline{BG}$ to the MCF5407. At this point, there is an internal access pending so the MCF5407 asserts $\overline{BD}$ during C4 and begins the access. Thus, the MCF5407 becomes the explicit external bus master. Also during C4, the external device removes the grant from the MCF5407 by negating $\overline{BG}$. As the current bus master, the MCF5407 continues to assert $\overline{BD}$ until the current transfer completes. Because $\overline{BG}$ is negated, the MCF5407 negates $\overline{BD}$ during C9 and three-states the external bus, thereby returning external bus mastership to the external device.

that this logic does not affect system or debug operation.

Figure 19-1 is a block diagram of the MCF5407 implementation of the 1149.1 IEEE standard. The test logic includes several test data registers, an instruction register, instruction register control decode, and a 16-state dedicated TAP controller.

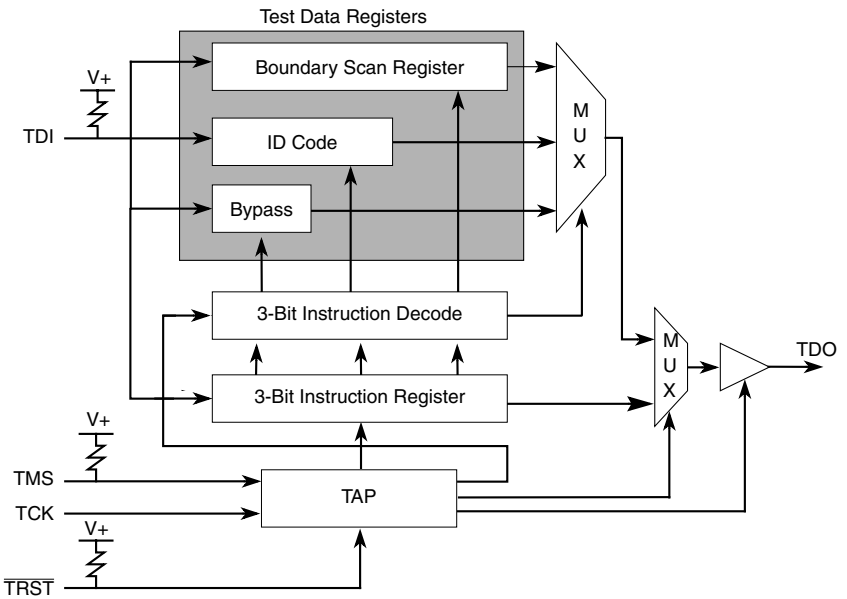

Figure 19-1. JTAG Test Logic Block Diagram

19.2 JTAG Signal Descriptions

JTAG operation on the MCF5407 is enabled when MTMOD0 is high (logic 1), as described in Table 19-1. Otherwise, JTAG TAP signals, TCK, TMS, TDI, TDO, and $\overline{\text{TRST}}$, are interpreted as the debug port pins. MTMOD0 should not be changed while $\overline{\text{RSTI}}$ is asserted.

INDEX

T