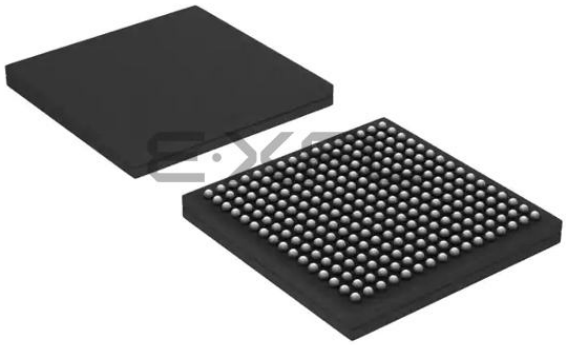


Welcome to E-XFL.COM

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	Obsolete
Core Processor	Coldfire V2
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	CANbus, EBI/EMI, Ethernet, I²C, SPI, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	150
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	256-LBGA
Supplier Device Package	256-MAPBGA (17x17)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mcf5281cvm66j

Table 2-9. D0 Hardware Configuration Info Field Description (continued)

Field	Description
11–8	Reserved.
7–4 ISA	ISA revision. Defines the instruction-set architecture (ISA) revision level implemented in ColdFire processor core. 0000 ISA_A 0001 ISA_B 0010 ISA_C 1000 ISA_A+ (This is the value used for this device.) Else Reserved
3–0 DEBUG	Debug module revision number. Defines revision level of the debug module used in the ColdFire processor core. 0000 DEBUG_A 0001 DEBUG_B 0010 DEBUG_C 0011 DEBUG_D 0100 DEBUG_E 1001 DEBUG_B+ 1011 DEBUG_D+ 1111 DEBUG_D+PST Buffer Else Reserved

Information loaded into D1 defines the local memory hardware configuration as shown in the figure below.

BDM: Load: 0x1 (D1)
Store: 0x1 (D1)

Access: User read-only
BDM read-only

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CLSZ		CCAS		CCSZ				FLASHSZ					0	0	0
W																
Reset	0	0	0	1	0	0	1	1	1	0	1	1	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MBSZ		UCAS		0	0	0	0	SRAMSZ					0	0	0
W																
Reset	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0

Figure 2-19. D1 Hardware Configuration Info

Table 2-10. D1 Hardware Configuration Information Field Description

Field	Description
31–30 CLSZ	Cache line size. This field is fixed to a hex value of 0x0 indicating a 16-byte cache line size.
29–28 CCAS	Configurable cache associativity. 00 Four-way 01 Direct mapped (This is the value used for this device) Else Reserved for future use

Table 6-3. CFM Register Address Map

IPSBAR Offset	Bits 31–24	Bits 23–16	Bits 15–8	Bits 7–0	Access[1]
0x1D_0014	CFMSACC				S
0x1D_0018	CFMDACC				S
0x1D_001C	Reserved[2]				S
0x1D_0020	CFMUSTAT	Reserved[2]			S
0x1D_0024	CFMCMD	Reserved[2]			S

1 S = Supervisor access only. User mode accesses to supervisor only addresses have no effect and result in a cycle termination transfer error.

2 Addresses not assigned to a register and undefined register bits are reserved for expansion. Write accesses to these reserved address spaces and reserved register bits have no effect.

6.3.4 Register Descriptions

The Flash registers are described in this subsection.

6.3.4.1 CFM Configuration Register (CFMCR)

The CFMCR is used to configure and control the operation of the CFM array.

	15–11	10	9	8	7	6	5	4–0
Field	—	LOCK	PVIE	AEIE	CBEIE	CCIE	KEYACC	—
Reset	0000_0000_0000_0000							
R/W	R/W							
Address	IPSBAR + 0x1D_0000							

Figure 6-4. CFM Module Configuration Register (CFMCR)

Bits 10 -5 in the CFMCR register are readable and writable with restrictions.

Table 6-4. CFMCR Field Descriptions

Bits	Name	Description
15–11	—	Reserved, should be cleared.
10	LOCK	Write lock control. The LOCK bit is always readable and is set once. 1 CFMPROT, CMFSACC, and CFMDACC register are write-locked. 0 CFMPROT, CMFSACC, and CFMDACC register are writable.
9	PVIE	Protection violation interrupt enable. The PVIE bit is readable and writable. The PVIE bit enables an interrupt in case the protection violation flag, PVIOL, is set. 1 An interrupt will be requested whenever the PVIOL flag is set. 0 PVIOL interrupts disabled.
8	AEIE	Access error interrupt enable. The AEIE bit is readable and writable. The AEIE bit enables an interrupt in case the access error flag, ACCERR, is set. 1 An interrupt will be requested whenever the ACCERR flag is set. 0 ACCERR interrupts disabled.

7.2.3.2 Low-Power Control Register (LPCR)

The LPCR controls chip operation and module operation during low-power modes.

	7	6	5	4	3	2	1	0
Field	LPMD		—	STPMD		—	LVDSE	—
Reset	0000_0010							
R/W	R/W							
Address	IPSBAR + 0x0011_0007							

Figure 7-2. Low-Power Control Register (LPCR)

Table 7-4. LPCR Field Descriptions

Bits	Name	Description
7–6	LPMD	Low-power mode select. Used to select the low-power mode the chip enters once the ColdFire CPU executes the STOP instruction. These bits must be written prior to instruction execution for them to take effect. The LPMD[1:0] bits are readable and writable in all modes. Table 7-5 illustrates the four different power modes that can be configured with the LPMD bit field.
5	—	Reserved, should be cleared.
4–3	STPMD	PLL/CLKOUT stop mode. Controls PLL and CLKOUT operation in stop mode as shown in Table 7-6
2	—	Reserved, should be cleared.
1	LVDSE	LDV standby enable. Controls whether the PMM enters VREG Standby Mode (LVD disabled) or VREG Pseudo-Standby (LVD enabled) mode when the PMM receives a power down request. This bit has no effect if the RCR[LVDE] bit is a logic 0. 1 VREG Pseudo-Standby mode (LVD enabled on power down request). 0 VREG Standby mode (LVD disabled on power down request).
0	—	Reserved, should be cleared.

Table 7-5. Low-Power Modes

LPMD[1:0]	Mode
11	STOP
10	WAIT
01	DOZE
00	RUN

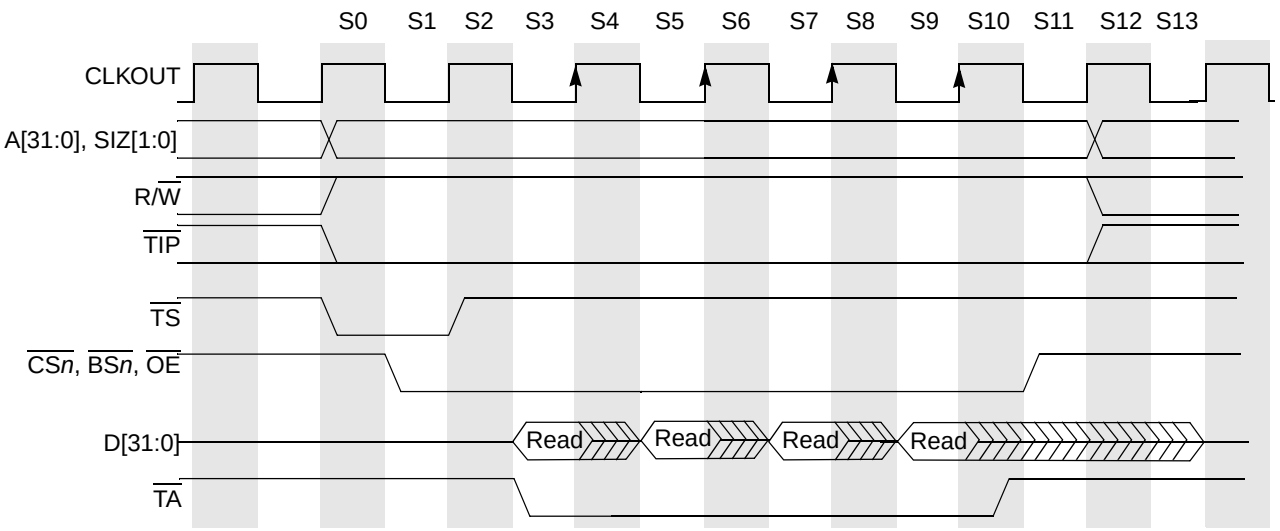


Figure 13-12. Line Read Burst (2-1-1-1), External Termination

Figure 13-13 shows timing when internal termination is used.

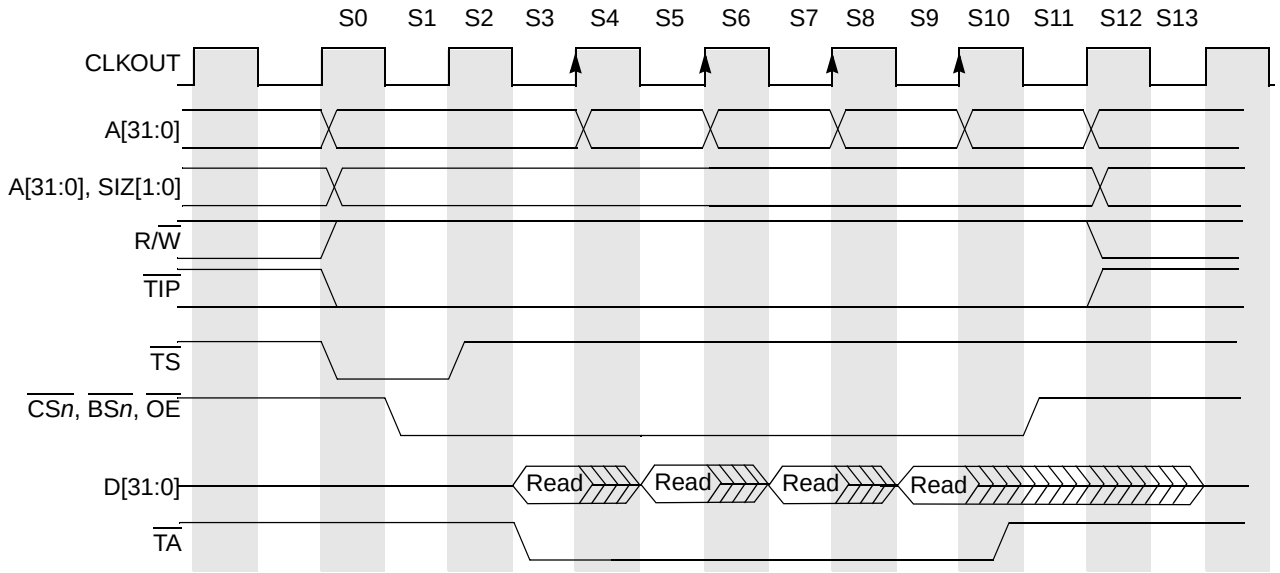


Figure 13-13. Line Read Burst (2-1-1-1), Internal Termination

Figure 13-14 shows a line access read with one wait state programmed in CSCR*n* to give the peripheral or memory more time to return read data. This figure follows the same execution as a zero-wait state read burst with the exception of an added wait state.

Table 15-5. DACR*n* Field Descriptions (continued)

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>10–8</td><td>CBM</td><td>Command and bank MUX [2:0]. Because different SDRAM configurations cause the command and bank select lines to correspond to different addresses, these resources are programmable. CBM determines the addresses onto which these functions are multiplexed.
Note: It is important to set CBM according to the location of the command bit.

<table>
<tr><th>CBM</th><th>Command Bit</th><th>Bank Select Bits</th></tr>
<tr><td>000</td><td>17</td><td>18 and up</td></tr>
<tr><td>001</td><td>18</td><td>19 and up</td></tr>
<tr><td>010</td><td>19</td><td>20 and up</td></tr>
<tr><td>011</td><td>20</td><td>21 and up</td></tr>
<tr><td>100</td><td>21</td><td>22 and up</td></tr>
<tr><td>101</td><td>22</td><td>23 and up</td></tr>
<tr><td>110</td><td>23</td><td>24 and up</td></tr>
<tr><td>111</td><td>24</td><td>25 and up</td></tr>
</table>
This encoding and the address multiplexing scheme handle common SDRAM organizations. Bank select bits include a base bit and all address bits above for SDRAMs with multiple bank select bits.</td></tr>
<tr><td>7</td><td>—</td><td>Reserved, should be cleared.</td></tr>
<tr><td>6</td><td>IMRS</td><td>Initiate mode register set (MRS) command. Setting IMRS generates a MRS command to the associated SDRAMs. In initialization, IMRS should be set only after all DRAM controller registers are initialized and PALL and REFRESH commands have been issued. After IMRS is set, the next access to an SDRAM block programs the SDRAM’s mode register. Thus, the address of the access should be programmed to place the correct mode information on the SDRAM address pins. Because the SDRAM does not register this information, it doesn’t matter if the IMRS access is a read or a write or what, if any, data is put onto the data bus. The DRAM controller clears IMRS after the MRS command finishes.
0 Take no action
1 Initiate MRS command</td></tr>
<tr><td>5–4</td><td>PS</td><td>Port size. Indicates the port size of the associated block of SDRAM, which allows for dynamic sizing of associated SDRAM accesses. PS functions the same in asynchronous operation.
00 32-bit port
01 8-bit port
1x 16-bit port</td></tr>
<tr><td>3</td><td>IP</td><td>Initiate precharge all (PALL) command. The DRAM controller clears IP after the PALL command is finished. Accesses via IP should be no wider than the port size programmed in PS.
0 Take no action.
1 A PALL command is sent to the associated SDRAM block. During initialization, this command is executed after all DRAM controller registers are programmed. After IP is set, the next write to an appropriate SDRAM address generates the PALL command to the SDRAM block.</td></tr>
<tr><td>2–0</td><td>—</td><td>Reserved, should be cleared.</td></tr>
</table>

Table 15-28. DACR Initialization Values (continued)

Bits	Name	Setting	Description
3	IP	0	Indicates precharge has not been initiated.
2–0	—		Reserved. Don't care.

15.3.4 DMR Initialization

Again, in this example only the second 512-Kbyte block of each 1-Mbyte space is accessed in each bank. In addition, the SDRAM component is mapped only to readable and writable supervisor and user data. The DMRs have the following configuration.

	31 … 18	17	16
Field	BAM	—	
Setting	0000_0000_0111_01xx		
(hex)	0 074		

	15 … 9	8	7	6	5	4	3	2	1	0
Field	—	WP	—	C/I	AM	SC	SD	UC	UD	V
Setting	xxxx_xxx0_x111_0101									
(hex)	0075									

Figure 15-14. DMR0 Register

With this configuration, the DMR0 = 0x0074_0075, as described in Table 15-29.

Table 15-29. DMR0 Initialization Values

Bits	Name	Setting	Description
31–18	BAM		With bits 17 and 16 as don't cares, BAM = 0x0074, which leaves bank select bits and upper 512K select bits unmasked. Note that bits 22 and 21 are set because they are used as bank selects; bit 20 is set because it controls the 1-Mbyte boundary address.
17–16	—		Reserved. Don't care.
15–9	—		Reserved. Don't care.
8	WP	0	Allow reads and writes
7	—		Reserved. Don't care.
6	C/I	1	Disable CPU space access.
5	AM	1	Disable alternate master access.
4	SC	1	Disable supervisor code accesses.
3	SD	0	Enable supervisor data accesses.
2	UC	1	Disable user code accesses.
1	UD	0	Enable user data accesses.
0	V	1	Enable accesses.

Chapter 17
Fast Ethernet Controller (FEC)

17.1 Introduction

This chapter provides a feature-set overview, a functional block diagram, and transceiver connection information for the 10 and 100 Mbps MII (media independent interface), as well as the 7-wire serial interface. Additionally, detailed descriptions of operation and the programming model are included.

NOTE

The MCF5214 and MCF5216 do NOT contain an FEC module.

17.1.1 Overview

The Ethernet media access controller (MAC) supports 10 and 100 Mbps Ethernet/IEEE 802.3 networks. An external transceiver interface and transceiver function are required to complete the interface to the media. The FEC supports three different standard MAC-PHY (physical) interfaces for connection to an external Ethernet transceiver. The FEC supports the 10/100 Mbps MII and the 10 Mbps-only 7-wire interface.

NOTE

The GPIO module must be configured to enable the peripheral function of the appropriate pins (refer to Chapter 26, “General Purpose I/O Module”) prior to configuring the FEC.

17.1.2 Block Diagram

Figure 17-1 shows the block diagram of the FEC. The FEC is implemented with a combination of hardware and microcode. The off-chip (Ethernet) interfaces are compliant with industry and IEEE 802.3 standards.

Table 17-10. MMFR Field Descriptions

Field	Description
31–30 ST	Start of frame delimiter. These bits must be programmed to 0b01 for a valid MII management frame.
29–28 OP	Operation code. 00 Write frame operation, but not MII compliant. 01 Write frame operation for a valid MII management frame. 10 Read frame operation for a valid MII management frame. 11 Read frame operation, but not MII compliant.
27–23 PA	PHY address. This field specifies one of up to 32 attached PHY devices.
22–18 RA	Register address. This field specifies one of up to 32 registers within the specified PHY device.
17–16 TA	Turn around. This field must be programmed to 10 to generate a valid MII management frame.
15–0 DATA	Management frame data. This is the field for data to be written to or read from the PHY register.

To perform a read or write operation on the MII Management Interface, write the MMFR register. To generate a valid read or write management frame, ST field must be written with a 01 pattern, and the TA field must be written with a 10. If other patterns are written to these fields, a frame is generated, but does not comply with the IEEE 802.3 MII definition.

To generate an IEEE 802.3-compliant MII Management Interface write frame (write to a PHY register), the user must write {01 01 PHYAD REGAD 10 DATA} to the MMFR register. Writing this pattern causes the control logic to shift out the data in the MMFR register following a preamble generated by the control state machine. During this time, contents of the MMFR register are altered as the contents are serially shifted and are unpredictable if read by the user. After the write management frame operation completes, the MII interrupt is generated. At this time, contents of the MMFR register match the original value written.

To generate an MII management interface read frame (read a PHY register), the user must write {01 10 PHYAD REGAD 10 XXXX} to the MMFR register (the content of the DATA field is a don’t care). Writing this pattern causes the control logic to shift out the data in the MMFR register following a preamble generated by the control state machine. During this time, contents of the MMFR register are altered as the contents are serially shifted and are unpredictable if read by the user. After the read management frame operation completes, the MII interrupt is generated. At this time, the contents of the MMFR register match the original value written except for the DATA field whose contents are replaced by the value read from the PHY register.

If the MMFR register is written while frame generation is in progress, the frame contents are altered. Software must use the MII interrupt to avoid writing to the MMFR register while frame generation is in progress.

Table 18-2. Watchdog Timer Module Memory Map

IPSBAR Offset	Bits 15–8	Bits 7–0	Access[1]
0x0014_0000	Watchdog Control Register (WCR)		S
0x0014_0002	Watchdog Modulus Register (WMR)		S
0x0014_0004	Watchdog Count Register (WCNTR)		S/U
0x0014_0006	Watchdog Service Register (WSR)		S/U

1 S = CPU supervisor mode access only. S/U = CPU supervisor or user mode access. User mode accesses to supervisor only addresses have no effect and result in a cycle termination transfer error.

18.5.2 Registers

The watchdog timer programming model consists of these registers:

- Watchdog control register (WCR), which configures watchdog timer operation
- Watchdog modulus register (WMR), which determines the timer modulus reload value
- Watchdog count register (WCNTR), which provides visibility to the watchdog counter value
- Watchdog service register (WSR), which requires a service sequence to prevent reset

18.5.2.1 Watchdog Control Register (WCR)

The 16-bit WCR configures watchdog timer operation.

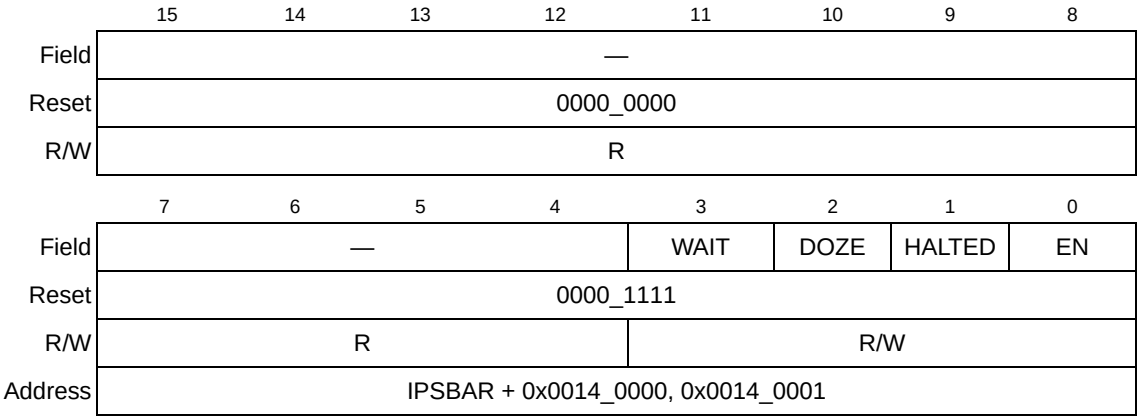
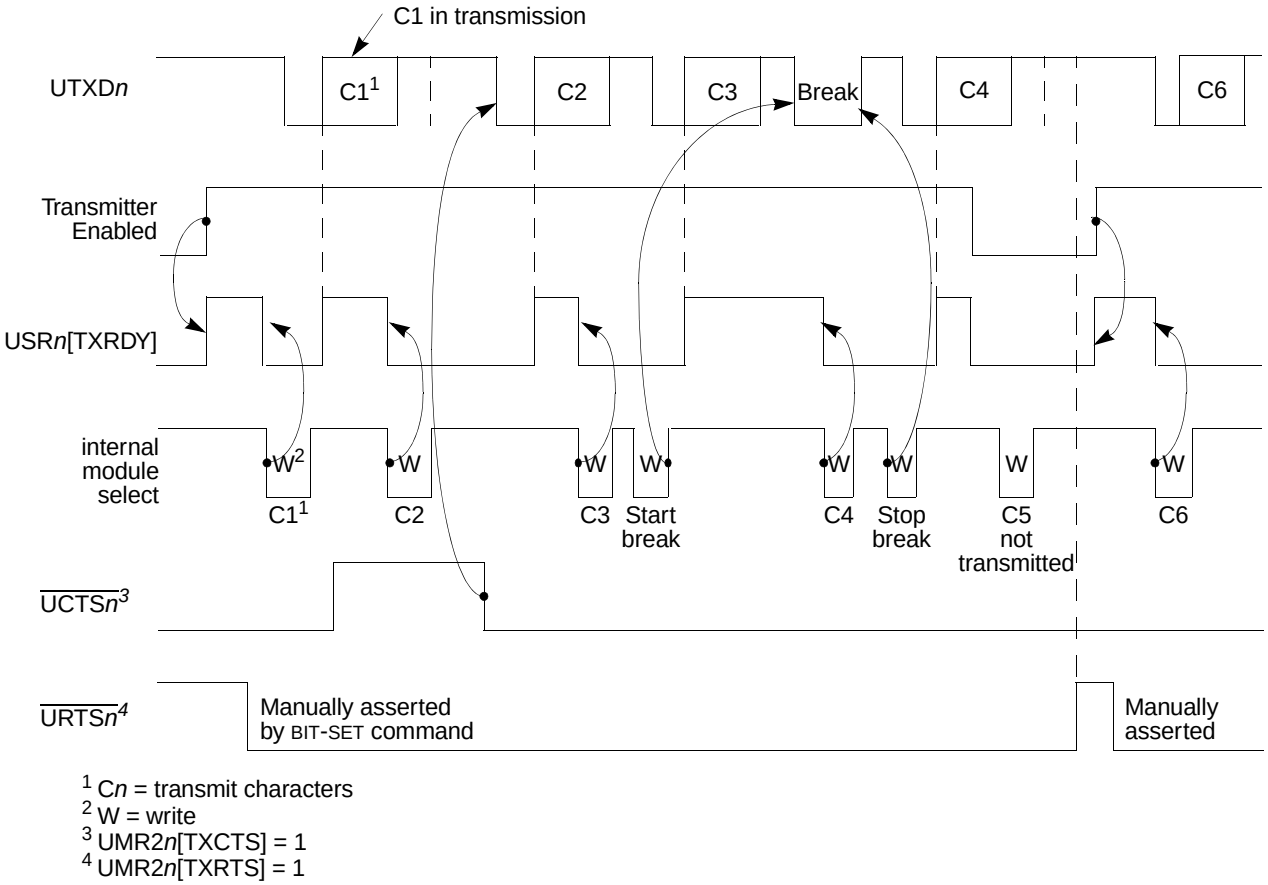


Figure 18-2. Watchdog Control Register (WCR)



[1] C*n* = transmit characters
[2] W = write
[3] UMR2*n*[TXCTS] = 1
[4] UMR2*n*[TXRTS] = 1

Figure 23-19. Transmitter Timing Diagram

23.4.2.2 Receiver

The receiver is enabled through its UCR*n*, as described in Section 23.3.5, “UART Command Registers (UCRn).”

When the receiver detects a high-to-low (mark-to-space) transition of the start bit on URXD*n*, the state of URXD*n* is sampled eight times on the edge of the bit time clock starting one-half clock after the transition (asynchronous operation) or at the next rising edge of the bit time clock (synchronous operation). If URXD*n* is sampled high, start bit is invalid and the search for the valid start bit begins again.

If URXD*n* remains low, a valid start bit is assumed. The receiver continues sampling the input at one-bit time intervals at the theoretical center of the bit until the proper number of data bits and parity, if any, is assembled and one stop bit is detected. Data on the URXD*n* input is sampled on the rising edge of the programmed clock source. The lsb is received first. The data then transfers to a receiver holding register and USR*n*[RXRDY] is set. If the character is less than 8 bits, the most significant unused bits in the receiver holding register are cleared.

After the stop bit is detected, receiver immediately looks for the next start bit. However, if a non-zero character is received without a stop bit (framing error) and URXD*n* remains low for one-half of the bit period after the stop bit is sampled, receiver operates as if a new start bit were detected. Parity error,

Table 25-15. Mask examples for Normal/Extended Messages

	Base ID ID28.....ID18	IDE	Extended ID ID17.....ID0	Match
MB2-Id	1 1 1 1 1 1 1 1 0 0 0	0		
MB3-Id	1 1 1 1 1 1 1 1 0 0 0	1	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	
MB4-Id	0 0 0 0 0 0 0 1 1 1 1	0		
MB5-Id	0 0 0 0 0 0 0 1 1 1 0 1	1	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	
MB14-Id	1 1 1 1 1 1 1 1 0 0 0	1	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	
Rx_Global_Mask	1 1 1 1 1 1 1 1 1 1 0		1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1	
Rx_Msg in	1 1 1 1 1 1 1 1 0 0 1	1	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	3[1]
Rx_Msg in	1 1 1 1 1 1 1 1 0 0 1	0		2[2]
Rx_Msg in	1 1 1 1 1 1 1 1 0 0 1	1	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0	3
Rx_Msg in	0 1 1 1 1 1 1 1 0 0 0	0		4
Rx_Msg in	0 1 1 1 1 1 1 1 0 0 0	1	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	5
Rx_14_Mask	0 1 1 1 1 1 1 1 1 1 1		1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0	
Rx_Msg in	1 0 1 1 1 1 1 1 0 0 0	1	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	6
Rx_Msg in	0 1 1 1 1 1 1 1 0 0 0	1	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	14[7]

1 Match for Extended Format (MB3).
2 Match for Standard Format. (MB2).
3 Un-Match for MB3 because of ID0.
4 Un-Match for MB2 because of ID28.
5 Un-Match for MB3 because of ID28, Match for MB14.
6 Un-Match for MB14 because of ID27.
7 Match for MB14.

25.5.7.1 Receive Mask Registers (RXGMASK, RX14MASK, RX15MASK)

The Rx global mask register (RXGMASK) is composed of 4 bytes. The mask bits are applied to all Rx-Identifiers excluding Rx-buffers 14-15, that have their specific Rx-mask registers (RX14MASK and RX15MASK).

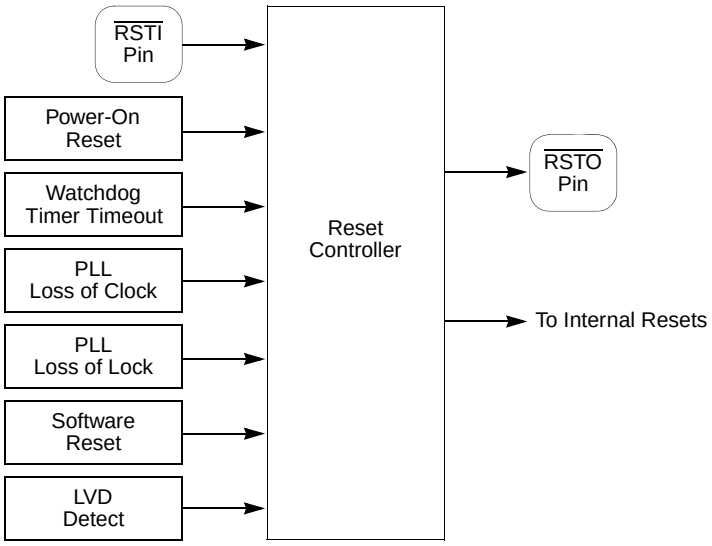


Figure 29-1. Reset Controller Block Diagram

29.3 Signals

Table 29-1 provides a summary of the reset controller signal properties. The signals are described in the following paragraphs.

Table 29-1. Reset Controller Signal Properties

Name	Direction	Input Hysteresis	Input Synchronization
$\overline{\text{RSTI}}$	I	Y	Y[1]
$\overline{\text{RSTO}}$	O	—	—

1 $\overline{\text{RSTI}}$ is always synchronized except when in low-power stop mode.

29.3.1 $\overline{\text{RSTI}}$

Asserting the external $\overline{\text{RSTI}}$ for at least four rising CLKOUT edges causes the external reset request to be recognized and latched.

29.3.2 $\overline{\text{RSTO}}$

This active-low output signal is driven low when the internal reset controller module resets the chip. When $\overline{\text{RSTO}}$ is active, the user can drive override options on the data bus.

29.4 Memory Map and Registers

The reset controller programming model consists of these registers:

- Reset control register (RCR), which selects reset controller functions
- Reset status register (RSR), which reflects the state of the last reset source

See Table 29-2 for the memory map and the following paragraphs for a description of the registers.

Table 30-5. AATR Field Descriptions (continued)

Bits	Name	Description
2–0	TM	Transfer modifier. Compared with the local bus transfer modifier signals, which give supplemental information for each transfer type. TT = 00 (normal mode): 000 Explicit cache line push 001 User data access 010 User code access 011 Reserved 100 Reserved 101 Supervisor data access 110 Supervisor code access 111 Reserved TT = 10 (emulator mode): 0xx–100 Reserved 101 Emulator mode data access 110 Emulator mode code access 111 Reserved TT = 11 (acknowledge/CPU space transfers): 000 CPU space access 001–111 Interrupt acknowledge levels 1–7 These bits also define the TM encoding for BDM memory commands (for backward compatibility).

30.4.3 Address Breakpoint Registers (ABLR, ABHR)

The ABLR and ABHR, shown in Figure 30-6, define regions in the processor's data address space that can be used as part of the trigger. These register values are compared with the address for each transfer on the processor's high-speed local bus. The trigger definition register (TDR) identifies the trigger as one of three cases:

1. Identical to the value in ABLR
2. Inside the range bound by ABLR and ABHR inclusive
3. Outside that same range

	31 … 0
Field	Address
Reset	—
R/W	Write only. ABHR is accessible in supervisor mode as debug control register 0x0C using the WDEBUG instruction and via the BDM port using the RDMREG and WDMREG commands. ABLR is accessible in supervisor mode as debug control register 0x0D using the WDEBUG instruction and via the BDM port using the WDMREG command.
DRc[4–0]	0x0D (ABLR); 0x0C (ABHR)

Figure 30-6. Address Breakpoint Registers (ABLR, ABHR)

NOTE

The debug module has no hardware interlocks, so to prevent spurious breakpoint triggers while the breakpoint registers are being loaded, disable TDR (by clearing TDR[29,13]) before defining triggers.

A write to TDR clears the CSR trigger status bits, CSR[BSTAT].

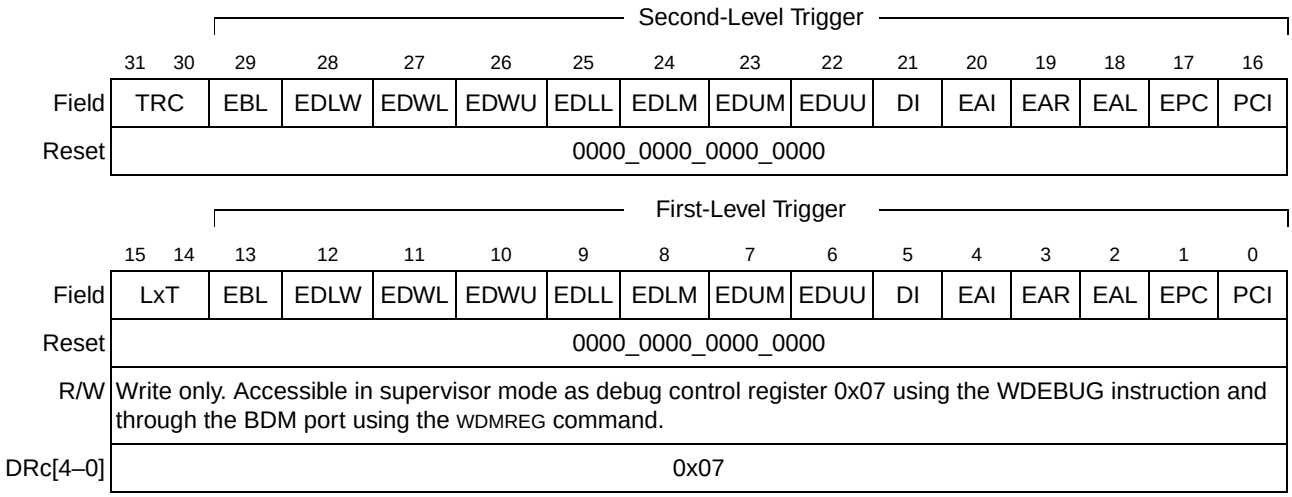


Figure 30-11. Trigger Definition Register (TDR)

Table 30-14 describes TDR fields.

Table 30-14. TDR Field Descriptions

Bits	Name	Description
31–30	TRC	Trigger response control. Determines how the processor responds to a completed trigger condition. The trigger response is always displayed on DDATA. 00 Display on DDATA only 01 Processor halt 10 Debug interrupt 11 Reserved
15–14	LxT	Level-x trigger. This is a Rev. B function only. The Level-x Trigger bit determines the logic operation for the trigger between the PC_condition and the (Address_range & Data_condition) where the inclusion of a Data condition is optional. The ColdFire debug architecture supports the creation of single or double-level triggers. TDR[15] 0 Level-2 trigger = PC_condition & Address_range & Data_condition 1 Level-2 trigger = PC_condition | (Address_range & Data_condition) TDR[14] 0 Level-1 trigger = PC_condition & Address_range & Data_condition 1 Level-1 trigger = PC_condition | (Address_range & Data_condition)
29/13	EBL	Enable breakpoint. Global enable for the breakpoint trigger. Setting TDR[EBL] enables a breakpoint trigger. Clearing it disables all breakpoints at that level.

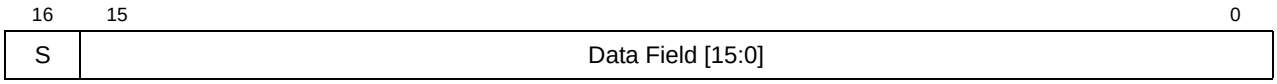


Figure 30-13. Receive BDM Packet

Table 30-15 describes receive BDM packet fields.

Table 30-15. Receive BDM Packet Field Description

Bits	Name	Description
16	S	Status. Indicates the status of CPU-generated messages listed below. The not-ready response can be ignored unless a memory-referencing cycle is in progress. Otherwise, the debug module can accept a new serial transfer after 32 processor clock periods. S DataMessage 0 xxxxValid data transfer 0 0xFFFFStatus OK 1 0x0000Not ready with response; come again 1 0x0001Error—Terminated bus cycle; data invalid 1 0xFFFFIllegal command
15–0	D	Data. Contains the message to be sent from the debug module to the development system. The response message is always a single word, with the data field encoded as shown above.

30.5.2.2 Transmit Packet Format

The basic transmit packet, Figure 30-14, consists of 16 data bits and 1 control bit.

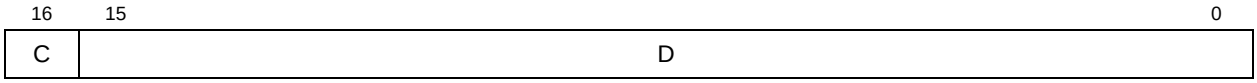


Figure 30-14. Transmit BDM Packet

Table 30-16 describes transmit BDM packet fields.

Table 30-16. Transmit BDM Packet Field Description

Bits	Name	Description
16	C	Control. This bit is reserved. Command and data transfers initiated by the development system should clear C.
15–0	D	Data bits 15–0. Contains the data to be sent from the development system to the debug module.

30.5.3 BDM Command Set

Table 30-17 summarizes the BDM command set. Subsequent paragraphs contain detailed descriptions of each command. Issuing a BDM command when the processor is accessing debug module registers using the WDEBUG instruction causes undefined behavior.

The DSCLK pin clocks the serial communication port to the debug module. Maximum frequency is 1/5 the processor clock speed. At the rising edge of DSCLK, the data input on DSI is sampled and DSO changes state.

31.3.1.6 TDO/DSO — Test Data Output / Development Serial Output

The TDO pin is the LSB-first data output. Data is clocked out of TDO on the falling edge of TCLK. TDO is tri-stateable and is actively driven in the shift-IR and shift-DR controller states.

The DSO pin provides serial output data in BDM mode.

31.4 Memory Map/Register Definition

31.4.1 Memory Map

The JTAG module registers are not memory mapped and are only accessible through the TDO/DSO pin.

31.4.2 Register Descriptions

All registers are shift-in and parallel load.

31.4.2.1 Instruction Shift Register (IR)

The JTAG module uses a 4-bit shift register with no parity. The IR transfers its value to a parallel hold register and applies an instruction on the falling edge of TCLK when the TAP state machine is in the update-IR state. To load an instruction into the shift portion of the IR, place the serial data on the TDI pin before each rising edge of TCLK. The MSB of the IR is the bit closest to the TDI pin, and the LSB is the bit closest to the TDO pin.

31.4.2.2 IDCODE Register

The IDCODE is a read-only register; its value is chip dependent. For more information, see Section 31.5.3.2, “IDCODE Instruction.”

	31–28	27–22	21–16
Field	PRN[[3:0]	DC[5:0]	PIN[9:0]
Reset	PRN[3] PRN[2] PRN[1] PRN[0]	0111_01	PIN[9] PIN[8] PIN[7] PIN[6] PIN[5] PIN[4]
R/W	Read only		

	15–12	11–1	0
Field	PIN[9:0]	JEDEC[10]	ID
Reset	0000_0000_0000_0000		
R/W	Read only		

Figure 31-2. IDCODE Register

S