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	140MHz
Connectivity	I²C, IDE, Memory Card, SPI, UART/USART
Peripherals	DMA, I²S, POR, Serial Audio, WDT
Number of I/O	47
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	96K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 4x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	160-BGA
Supplier Device Package	160-MAPBGA (15x15)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mcf5249cvf140

Paragraph Number | Page Number

19.1.5 Development Serial Output (DSO) 19-2
19.1.6 Processor Status (PST[3:0]) 19-2
19.1.7 Processor Status Clock (PSTCLK) 19-3
19.2 Real-Time Trace Support 19-4
19.2.1 Processor Status Signal Encoding 19-4
19.2.1.1 Continue Execution (PST = $0) 19-4
19.2.1.2 Begin Execution of an Instruction (PST = $1) 19-4
19.2.1.3 Entry into User Mode (PST = $3) 19-4
19.2.1.4 Begin Execution of PULSE or WDDATA instructions (PST = $4) 19-4
19.2.1.5 Begin Execution of Taken Branch (PST = $5) 19-5
19.2.1.6 Begin Execution of RTE Instruction (PST = $7) 19-6
19.2.1.7 Begin Data Transfer (PST = $8–$B) 19-6
19.2.1.8 Exception Processing (PST = $C) 19-6
19.2.1.9 Emulator Mode Exception Processing (PST = $D) 19-6
19.2.1.10 Processor Stopped (PST = $E) 19-6
19.2.1.11 Processor Halted (PST = $F) 19-6
19.3 Background-Debug Mode (BDM) 19-6
19.3.1 CPU Halt 19-7
19.3.2 BDM Serial Interface 19-7
19.3.2.1 Receive Packet Format 19-8
19.3.2.2 Transmit Packet Format 19-9
19.3.3 BDM Command Set 19-9
19.3.3.1 BDM Command Set Summary 19-9
19.3.3.2 ColdFire BDM Commands 19-9
19.3.3.3 Command Sequence Diagram 19-11
19.3.3.4 Command Set Descriptions 19-13
19.3.3.4.1 Read Address/Data Register (RAREG/RDREG) 19-13
19.3.3.4.2 Write Address/Data Register (WAREG and WDREG) 19-13
19.3.3.4.3 Read Memory Location (READ) 19-14
19.3.3.4.4 Write Memory Location (WRITE) 19-16
19.3.3.4.5 Dump Memory Block (DUMP) 19-17
19.3.3.4.6 Fill Memory Block (FILL) 19-18
19.3.3.4.7 Resume Execution (GO) 19-20
19.3.3.4.8 No Operation (NOP) 19-20
19.3.3.4.9 Read Control Register (RCREG) 19-21
19.3.3.4.10 Write Control Register (WCREG) 19-22
19.3.3.4.11 Read Debug Module Register (RDMREG) 19-22
19.3.3.4.12 Write Debug Module Register (WDMREG) 19-23
19.3.3.4.13 Unassigned Opcodes 19-24
19.3.3.5 BDM Accesses of the EMAC Registers 19-24
19.4 Real-Time Debug Support 19-25
19.4.1 Theory of Operation 19-26
19.4.1.1 Emulator Mode 19-27
19.4.1.2 Debug Module Hardware 19-27
19.4.1.2.1 Reuse of Debug Module Hardware (Rev. A) 19-27
19.4.2 Programming Model 19-28
19.4.2.1 Address Breakpoint Registers 19-28
19.4.2.2 Address Attribute Trigger Register 19-29
19.4.2.3 Program Counter Breakpoint Register (PBR, PBMR) 19-31
19.4.2.4 Data Breakpoint Registers (DBR, DBMR) 19-32
19.4.2.5 Trigger Definition Register (TDR) 19-34
19.4.2.6 Configuration/Status Register (CSR) 19-36

LIST OF FIGURES

		Page Number
Figure 1-1	MCF5249 Block Diagram	1-2
Figure 3-1	V2 ColdFire Processor Core Pipelines	2-1
Figure 3-2	User Programming Model	2-3
Figure 3-3	Supervisor Programming Model	2-5
Figure 3-4	Vector Base Register (VBR)	2-6
Figure 3-5	Exception Stack Frame Form	2-9
Figure 4-1	Phase-Locked Loop Module Block Diagram	4-1
Figure 5-1	Instruction Cache Block Diagram	5-2
Figure 7-1	Synchronous DRAM Controller Block Diagram	7-2
Figure 7-2	MCF5249 SDRAM Interface	7-5
Figure 7-3	DRAM Control Register (DCR) (Synchronous Mode)	7-5
Figure 7-4	DACR0 and DACR1 (Synchronous Mode)	7-7
Figure 7-5	DRAM Controller Mask Registers (DMR0 and DMR1)	7-9
Figure 7-6	Burst Read SDRAM Access	7-12
Figure 7-7	Burst Write SDRAM Access	7-13
Figure 7-8	Synchronous, Continuous Page-Mode Access—Consecutive Reads	7-14
Figure 7-9	Synchronous, Continuous Page-Mode Access—Read after Write	7-15
Figure 7-10	Auto-Refresh Operation	7-16
Figure 7-11	Self-Refresh Operation	7-16
Figure 7-12	Mode Register Set (mrs) Command	7-18
Figure 7-13	Initialization Values for DCR	7-19
Figure 7-14	SDRAM Configuration	7-20
Figure 7-15	DACR Register Configuration	7-20
Figure 7-16	DMR0 Register	7-21
Figure 7-17	Mode Register Mapping to MCF5249 A[31:0]	7-22
Figure 8-1	Connections for External Memory Port Sizes	8-3
Figure 8-2	Signal Relationship to BCLK for Non-DRAM Access	8-5
Figure 8-3	Read Cycle Flowchart	8-7
Figure 8-4	Basic Read Bus Cycle	8-7
Figure 8-5	Write Cycle Flowchart	8-9
Figure 8-6	Basic Write Bus Cycle	8-9
Figure 8-7	Back-to-Back Bus Cycles	8-10
Figure 8-8	Line Read Burst (one wait cycle)	8-12
Figure 8-9	Line Read Burst (no wait cycles)	8-12
Figure 8-10	Line Write Burst (no wait cycles)	8-13
Figure 8-11	Line Read Burst-Inhibited	8-13
Figure 8-12	Line Write Burst with One Wait State	8-14
Figure 8-13	Line Write Burst-Inhibited	8-14
Figure 8-14	Misaligned Longword Transfer	8-15
Figure 8-15	Misaligned Word Transfer	8-15
Figure 8-16	Master Reset Timing	8-16
Figure 8-17	Software Watchdog Reset Timing	8-17
Figure 9-1	MCF5249 Unterminated Access Recovery	9-17
Figure 9-2	General-Purpose Pin Logic for Pin ddata3/gpio34	9-27
Figure 11-1	Timer Block Diagram Module Operation	11-2
Figure 12-1	ADC with On-chip and External Parts	12-2
Figure 13-1	Bus Setup with IDE and SmartMedia Interface	13-1
Figure 13-2	Buffer Enables (BUFENB1 and BUFENB2)	13-2
Figure 13-3	DIOR and SRE Timing Diagram	13-5
Figure 13-4	Non-IORDY Controlled IDE/SmartMedia TA Timing	13-6
Figure 13-5	CS2 (DIOR, DIOW) and CS3 (SRE, SWE) Cycle Timing	13-6

3.2.3.1 STATUS REGISTER (SR)

The SR stores the processor status and includes the CCR, the interrupt priority mask, and other control bits. In the supervisor mode, software can access the entire SR. In user mode, only the lower 8 bits are accessible (CCR). The control bits indicate the following states for the processor: trace mode (T-bit), supervisor or user mode (S bit), and master or interrupt state (M).

Table 3-4 Status Register

SYSTEM BYTE								CONDITION CODE REGISTER (CCR)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T	0	S	M	0	I[2:0]			0	0	0	X	N	Z	V	C

Table 3-5 Status Bit Descriptions

BIT NAME	DESCRIPTION
T	When set, the trace enable allows the processor to perform a trace exception after every instruction.
S	The supervisor / user state bit denotes whether the processor is in supervisor mode (S=1) or user mode (S=0).
M	The master / interrupt state bit is cleared by an interrupt exception, and can be set by software during execution of the RTE or move to SR instructions.
I [2:0]	The interrupt priority mask defines the current interrupt priority. Interrupt requests are inhibited for all priority levels less than or equal to the current priority, except the edge-sensitive level 7 request, which cannot be masked.

3.2.3.2 VECTOR BASE REGISTER (VBR)

The VBR contains the base address of the exception vector table in memory. The displacement of an exception vector is added to the value in this register to access the vector table. The lower 20 bits of the VBR are not implemented by ColdFire processors; they are zero, forcing the table to be aligned on a 1 MByte boundary.

	30 — 21	19 — 0
Field	Exception vector table base address	—
Reset	0000_0000_0000_0000_0000_0000_0000_0000	
R/W	Written from a BDM serial command or from the CPU using the MOVEC instruction. VBR can be read from the debug module only. The upper 12 bits are returned, the low-order 20 bits are undefined.	
Rc [11-0]	0x801	

Figure 3-4 Vector Base Register (VBR)

4.2 PLL PROGRAMMING

The different settings for the PLL/clock module are summarized in Table 4-1.

Table 4-1 PLLCR Register

BITS	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	LOCK	CLSEL			N/A	CPUDIV			CRSEL	AUDIO SEL	DEBUG SEL	VCODIV				
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W														

BITS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	VCODIV				QSPI SEL	RST SEL	PLL POWER DOWN	PLLDIV					VCOOUT		N/A	PLL BYPASS
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W															
ADDR	ADDRESS MBAR2BAS + 0 x 180															

Note: Bits marked N/A are reserved bits; program these bits to 0.

Table 4-2 PLLCR Bit Descriptions

BIT NAME	DESCRIPTION
LOCK	Read-only bit, 1 if PLL is locked. (See Note 2 following these bit descriptions.)
CLSEL	(See Note 12) MCLK1,MCLK2 select bit.
CPUDIV	(See Notes 8 and 9) CPU clock divider
CRSEL	(See Note 3) 0 = Fin = Fxtal 1 = Fin = Fxtal/2
AUDIOSEL	(See Note 4) 1 = FXTAL 0 = FXTAL/2
DEBUGSEL	(See Note 11) 1 = Secondary functions on aux debug port. 0 = Aux debug port active.
VCODIV	(See Notes 5 and 10) PLL compare frequency is VCO frequency divided by (VCODIV + 2)
QSPISEL	(See Note 7) 1 = QSPI functions active on pins. (qspi_clk, qspi_din) 0 = IIC functions active on pins. (scl, sda)
VCOOUT	(See Note 6) VCO output divider
PLLBYPASS	(See Notes 1 and 2 following these bit descriptions) 1 = switch to PLL after PLL is locked 0 = Bypass PLL and dividers
RSTSEL	(See Note 7) 1 = SDATA2BS2 function active on pin 0 = RST function active on pin

4.4 REDUCED POWER MODE

To save power, it is recommended that users reduce the frequency of the CPU clocks. This is done by reprogramming the PLLCR register.

The PLL is also configured with a power down bit. This bit, when set to ‘1’, allows the PLL to enter “sleep” mode. In “sleep” mode, the VCO and charge pump are turned off.

Note: The PLL must go through the re-locking procedure when it is re-enabled.

4.5 RECOMMENDED SETTINGS

Many valid PLL settings exist. However, in many cases some limitations apply so that only a few typical settings will be used. In a typical system, the following limitations may exist:

- Users want to run the processor at 120, 96, 64, 84, or 72 Mhz clock frequency
- MCLK2 must be one of the following: 16.9344, 11.2896, or 8.4672 Mhz see Table 4-4 in this section for further definition.
- MCLK1 must be one of the following: 16.9344, 11.2896, or 8.4672 Mhz see Table 4-4 in this section for further definition.

As a result of these limitations, users may select a 33.8688 Mhz X-TAL and use the settings shown in Table 4-5.

A utility that calculates PLL frequencies from PLL register settings is available at the following URL:

http://e-www.motorola.com/webapp/sps/library/prod_lib.jsp
(Select 32-Bit Embedded Processors, 68K/ColdFire, ColdFire MC5XXX, MCF5249).

Table 4-5 Recommended PLL Settings

X-TAL FREQ MHZ	CPU DIV	CRSEL	VCO DIV	PLL DIV	VCO OUT	CPU CLOCK MHZ
33.8688	4	1	0x1AD	0x11	0	96
33.8688	6	1	0x1AD	0x011	0	64
33.8688	4	1	0x100	0x0B	0	84

NOTES

Figure 7-2 shows a typical signal configuration for synchronous mode.

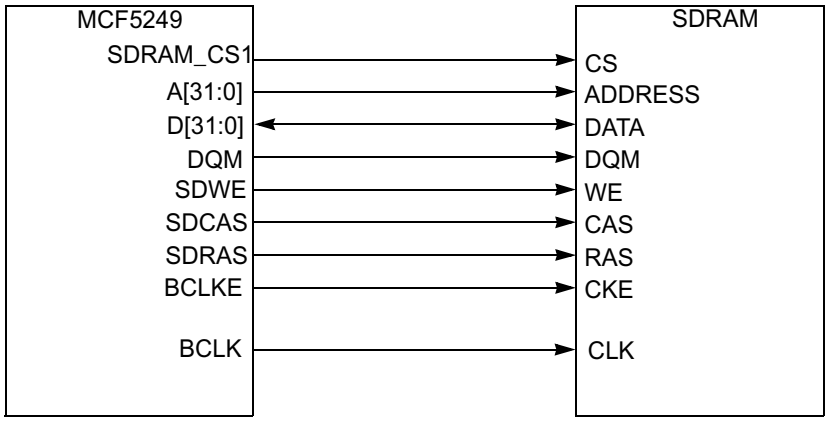

Figure 7-2 MCF5249 SDRAM Interface

7.3.2 SYNCHRONOUS REGISTER SET

The memory map is shown in Table 7-1. Bit descriptions are shown in the following sections.

7.3.2.1 DRAM CONTROL REGISTER (DCR) (SYNCHRONOUS MODE)

The DRAM control register (DCR), Figure 7-3, controls refresh logic.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SO	—	NAM	COC	IS	RTIM					RC					
Reset	0	Uninitialized														
R/W	R/W															
Addr	MBAR + 0x100															

Figure 7-3 DRAM Control Register (DCR) (Synchronous Mode)

Table 7-4 describes DCR fields.

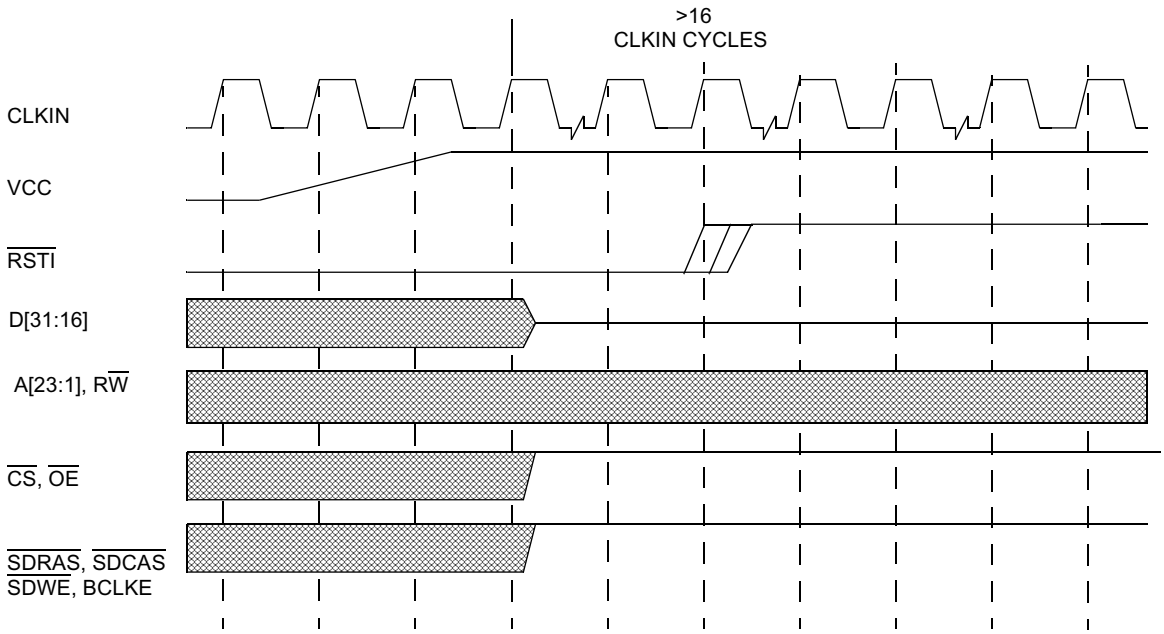

Figure 8-16 Master Reset Timing

During the master reset period, the data bus is being three-stated, the address bus is driven to any value, and all other bus signals are driven to their negated state. Once RSTI negates, the bus stays in this state until the ColdFire core begins the first bus cycle for reset exception processing. A master reset causes any bus cycle (including DRAM refresh cycle) to terminate. In addition, master reset initializes registers appropriately for a reset exception.

At power-on reset, CS0 is configured to address the boot ROM. Boot ROM configuration is hard-wired inside the MCF5249 Configuration is summarized in table Table 8-8.

Table 8-8 Power-on Reset Configuration for CS0

PORT SIZE	16 BITS
Cycle type	Internal termination, 15 wait cycles burst inhibit asserted for both read and write cycles

8.7.1 SOFTWARE WATCHDOG RESET

The software watchdog reset is performed anytime the executing software does not provide the correct write data sequence with the enable-control bit set. This reset helps prevent runaway software or nonterminated bus cycles. Figure 8-17 is a functional timing diagram of the software watchdog reset operation, illustrating relationships among RSTO and bus signals.

NOTES

9.5.2.3 Software Watchdog Service Register

The SWSR is where the SWT servicing sequence should be written. To prevent an SWT timeout, users should write a $55 followed by a $AA to this register. Both writes must be performed in the order listed prior to the SWT timeout, but any number of instructions or accesses to the SWSR can be executed between the two writes. If the SWT has already timed out, writing to this register will have no effect in negating the SWT interrupt. The following register illustrates the SWSR programming model.

The SWSR is an 8-bit write-only register. At system reset, the contents of SWSR are uninitialized.

Table 9-32 Software Watchdog Service Register (SWSR)

BITS	7	6	5	4	3	2	1	0
FIELD	SWSR7	SWSR6	SWSR5	SWSR4	SWSR3	SWSR2	SWSR1	SWSR0
RESET	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	MBAR + $(0X03)							

9.6 CPU STOP INSTRUCTION

Executing the CPU STOP instruction does not stop any of the clocks.

9.7 MCF5249 BUS ARBITRATION CONTROL

9.7.1 DEFAULT BUS MASTER PARK REGISTER

The MPARK determines the default bus master arbitration between internal transfers. This arbitration is needed because there are two bus masters inside the MCF5249. One is the CPU, the other is the DMA unit. Both can access internal registers within the MCF5249 peripherals. Table 9-33 shows the MPARK register bit encoding.

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

Table 9-33 Default Bus Master Register (MPARK)

BITS	7	6	5	4	3	2	1	0
FIELD	PARK[1]	PARK[0]	IARBCTRL	EARBCTRL	SHOWDATA	-	-	BCR24BIT
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W			R/W
ADDR	MBAR + $(0X0C)							

9.7.1.1 Internal Arbitration Operation

The PARK[1:0] bits are programmed to indicate the priority of internal transfers within MCF5249 resources. The possible masters that can initiate internal transfers internal to the MCF5249 are the core and the on-chip DMAs. Since the priority between DMAs is resolved by their relative priority amongst each other and by programming the BWC bits in their respective DMA control registers (see 14.4.5 DMA Control Register), the MPARK bits need only arbitrate priority between the core and the DMA module (which contains all four DMA channels) for internally generated transfers.

Section 10
Chip-Select Module

10.1 INTRODUCTION

The Chip Select Module provides user-programmable control of the four chip select outputs, two buffer enable outputs and one output-enable signal.

This section describes the operation and programming model of the chip-select (CS) registers, including the chip select address, mask, and control registers.

10.1.1 CHIP SELECT FEATURES

- Four programmable chip select signals
- IORDY and TA handshake pins
- Two programmable buffers enable signals for glueless interface to bus buffers
- Address masking for memory block sizes from 64KBytes to 4GBytes
- Programmable wait states
- Port size is 16 bits

10.2 CHIP-SELECT SIGNALS

The MCF5249 provides four programmable chip selects that can directly interface with SRAM, EPROM, EEPROM, and peripherals. Two of these chip selects are usable for AT-bus peripherals that need separate read and write strobe, and use IORDY signalling to insert wait states.

10.2.1 CHIP SELECTS

10.2.1.1 $\overline{CS0}$

CS0 is the first chip select and it addresses the boot memory. (A ROM or flash memory device.) At power-on reset, all bus cycles are mapped to the CS0. This allows the boot memory to be defined at any address space. CS0 is the only chip select initialized at reset.

10.2.1.2 $\overline{CS1}$/GPIO1

CS1 is the second chip select and it can be programmed for an address location as well as for masking, port size and burst capability indication, wait state generation, and internal/external termination. A reset clears all chip select programming

10.2.1.3 CS2/IDE-$\overline{DIOR}$/GPIO13 AND IDE-$\overline{DIOW}$/GPIO14

These two signals go active during $\overline{CS2}$ cycles. IDE-$\overline{DIOR}$ can be programmed to go active on read and write cycles, or IDE-$\overline{DIOR}$ can be programmed to go active only on read cycles, and IDE-$\overline{DIOW}$ only on write cycles. It has identical features as the normal $\overline{CS2}$. It can be programmed for an address location as well as for masking, port size and burst capability indication, wait state generation, and internal/external termination.

Table 13-2 IDECONFIG1 Bits (Continued)

IDE CONFIG1 BITS	FIELD NAME	MEANING	RES
17	BUFEN1CS2 EN	0: bufen1 inactive on DIOR, DIOW cycles 1: bufen1 active on DIOR, DIOW cycles	0
18	BUFEN1CS3 EN	0: bufen1 inactive on SRE, SWE cycles 1: bufen1 active on SRE, SWE cycles	0
19	BUFEN2CS1 EN	0: bufen2 inactive on CS1 cycles 1: bufen2 active on CS1 cycles	0
20	BUFEN2CS2 EN	0: bufen2 inactive on DIOR, DIOW cycles 1: bufen2 active on DIOR, DIOW cycles	0
21	BUFEN2CS3 EN	0: bufen2 inactive on SRE, SWE cycles 1: bufen2 active on SRE, SWE cycles	0
24:22	CS3PRE	pre-drive for SRE, SWE 000: no predrive 001: 1 clock 010: 2 clocks 011: 3 clocks 100: 4 clocks 101: 5 clocks	0
26:25	CS3POST	post-drive for CS1 00: no post-drive 01: 1 clock post-drive 10: 2 clock post drive 11: 3 clock post drive	0
27	DIOR on write	0: DIOR not active during write cycles 1: DIOR active during write cycles	0
28	SRE active during write	0: SRE not active during write 1: SRE active during write	0

Note: The SWE and SRE signals are only used on the 160 MAPBGA package.

13.1.2 GENERATION OF IDE-DIOR, IDE-DIOW, SRE, SWE

These four signals are generated internally by gating the CS2_pin and CS3_pin signals with RWb. DIOR and DIOW are created by gating CS2_pin with RWb. SRE and SWE are created by gating CS3_pin with RWb.

Note: The SWE and SRE signals are only used on the 160 MAPBGA package.

DIOR and SRE are programmable if these signals go active on write cycles. If these signals are programmed to go active during write cycles, they can be used as extra chip enables. (CS2 and CS3).

The BCR must be loaded with the number of byte transfers that are to occur. This register is decremented by 1, 2, 4, or 16 at the end of each transfer. The DSR must be cleared for channel startup.

Once the channel has been initialized, it is started by writing a one to the START bit in the DCR or asserting the REQUEST signal, depending on the status of the EEXT bit in the DCR. Programming the channel for processor request causes the channel to request the bus and start transferring data immediately. If the channel is programmed for periphery request, REQUEST must be asserted before the channel requests the bus.

If any fields in the DCR are modified while the channel is active, that change is effective immediately. To avoid any problems with changing the setup for the DMA channel, a 1 should be written to the DONE bit in the DSR to stop the DMA channel.

14.7.2 DATA TRANSFER

14.7.2.1 Periphery Request Operation

All channels can initiate transfers to/from a periphery module by means of REQUEST[3:0]. Source where REQUEST is coming from is programmed in register DMAROUTE. If the EEXT bit (DCR[30]) is set, when a REQUEST is asserted, the DMA initiates a transfer provided the channel is idle. If the CS (cycle steal) bit is set, the read/write transaction on the bus is limited to a single transfer. If the CS bit is clear, multiple read/write transfers can occur on the bus as programmed. REQUEST does not need to be negated until the DONE bit (DSR[0]) is set.

14.7.2.2 Auto Alignment

This feature allows for block transfers to occur at the optimum size based on the address, byte count, and programmed size. To use this feature, AA in the DCR must be set. The source is auto-aligned when the SSIZE bits indicate a larger transfer size compared to DSIZE. Source alignment takes precedence over the destination when the source and destination sizes are equal. Otherwise, the destination is auto-aligned. The address register that is chosen for alignment increments regardless of the value of the increment bit. Configuration error checking is performed on the registers that are not chosen for alignment.

If the BCR contains a value greater than 16, the address will determine the size of the transfer. Single byte, word or longword transfers will occur until the address is aligned to the programmed size boundary, at which time the programmed size accesses begin. When the BCR is less than 16 at the beginning of a read/write transfer, the number of bytes remaining will dictate the transfer size, longword, word or byte.

For example:

AA = 1, SAR = $0001, BCR = $00f0, SSIZE = 00 (longword) and DSIZE = 01 (byte),

Because the SSIZE > DSIZE, the source is auto-aligned. Error checking is performed on the destination registers. The sequence of accesses is as follows:

1. Read byte from $0001—write byte, increment SAR
2. Read word from $0002—write 2 bytes, increment SAR
3. Read long word from $0004—write 4 bytes, increment SAR
4. Repeat longwords until SAR = $00f0
5. Read byte from $00f0—write byte, increment SAR.

If DSIZE is set to another size, then the data writes are optimized to write the largest size allowed based on the address, but not exceeding the configured size.

indirectly through the QSPI address register (QAR) and the QSPI data register (QDR). The RAM is divided into three segments with 16 addresses each:

- Receive data RAM, the initial destination for all incoming data
- Transmit data RAM, a buffer for all out-bound data
- Command RAM, where commands are loaded

The transmit and command RAM are write-only by the user. The receive RAM is read-only by the user. Figure 16-2 shows the RAM configuration. The RAM contents are undefined immediately after a reset.

The command and data RAM in the QSPI is indirectly accessible with QDR and QAR as 48 separate locations that comprise 16 words of transmit data, 16 words of receive data and 16 bytes of commands.

A write to QDR causes data to be written to the RAM entry specified by QAR[ADDR] and causes the value in QAR to increment. Correspondingly, a read at QDR returns the data in the RAM at the address specified by QAR[ADDR]. This also causes QAR to increment. A read access requires a single wait state.

RELATIVE ADDRESS	REGISTER	FUNCTION
0x00	QTR0	Transmit RAM
0x01	QTR1	
. . .	. . .	16 bits wide
0x0F	QTR15	

0x10	QRR0	Receive RAM
0x11	QRR1	
. . .	. . .	16 bits wide
0x1F	QRR15	

0x20	QCR0	Command RAM
0x21	QCR1	
. . .	. . .	8 bits wide
0x2F	QCR15	

Figure 16-2 QSPI RAM Model

There are four serial audio interface blocks labeled as follows:

1. IIS1: Capable of transmitting and receiving audio data.
2. IIS2: Transmit only.
3. IIS3: Receive only.
4. IIS4: Receive only.

As shown in Figure 17-1, there two IEC958 receivers. The source selector (**18)** and the receiver block itself (**19)**. The receiver is capable of taking its input signal from four possible EBU inputs:

1. EBUIN1
2. EBUIN2
3. EBUIN3
4. EBUIN4

There is one IEC958 transmitter (**30**) with two outputs. One carries the “professional” C-channel, and the other carries the “consumer” C-channel.

Five audio interface receivers (IIS1, IIS3, IIS4, and two EBU receivers) send their received data on an internal 40-bit wide bus, the Internal Audio Data Bus. Every transmitter sources its data to be transmitted from this same internal bus. Every transmitter has a multiplexer to select the data source. Possible sources are (IIS1 receiver, IIS3 receiver, IIS4 receiver, two EBU receivers, processor data output1, processor data output2, processor data output 3). Every transmitter also has a FIFO after the multiplexer. This FIFO gives the data source some freedom when data is generated. The FIFOs compensate for phase shifts when a transmitter takes data from another receiver. In the case that the transmitter sends out processor-generated data, the FIFO allows the processor to send several audio words in one burst to the audio transmitter.

To allow the MCF5249 processor to receive and transmit audio data, an interface is present between the internal Audio Data Bus and the ColdFire memory space. As shown in Figure 17-1, this interface is seen in the MCF5249 memory map as Processor Data Interface Registers. Three of these are Processor Data Out registers, PDOR1, PDOR2 and PDOR3. When the processor writes to one of these registers, the data is sent directly to the Internal Audio Data Bus, and depending on the setting of the multiplexers (**13, 15,** and **24)** it will end up in one or several of the transmit FIFOs **(12, 14,** and **25)**. There are three Processor Data In registers, PDIR1, PDIR2, and PDIR3. When the processor reads from one of these address locations, it actually reads data from one of the FIFOs (**17**, **17a or 17b)**. These FIFOs receive data from the Internal Audio Data Bus using multiplexers **(16**, **16a and 16b)**. Depending on the setting of the multiplexers, data from one of the audio data receivers will end in the FIFOs. Possible receivers for the three PDIR channels are IIS1 receiver, IIS3 receiver, IIS4 receiver and the two IEC958 receivers.

Besides the mechanism to let the MCF5249 processor access the audio data, there are several interrupts and control registers to allow the MCF5249 to determine when it should read or write data to the appropriate processor data interface register.

The IEC958 receiver and transmitter handle the main data audio stream in the same way as the IIS receivers and transmitters. This is done using the internal Audio Data Bus. Additionally, they support the IEC958 “C” and “U” channels. IEC958 “C” and “U” channel data is interfaced directly to memory-mapped registers (**22,26,27** and **28)**.

17.1.1.1 Audio Interrupt Mask and Interrupt Status Registers

The interrupts of the audio interface feed into vectors 0-31 of the interrupt controller. There are two sets of registers associated with interrupt operation.

Operation is as follows:

- If PDIR2 is full and DMAConfig(1) is set to ‘0’, Dma1Req is activated.
- If PDIR2 is full and DMAConfig(0) is set to ‘0’, DMA0req is activated.
- If the FIFO connected to PDOR3 is empty, and DMAConfig(1) is set ‘1’, Dma1Req is activated.
- If the FIFO connected to PDOR3 is empty, and DMAConfig(0) is set to ‘1’, DMA0req is activated.

Both DMA1req and DMA0req can be routed to DMA channel 0 or DMA channel 1. For details, see description of ColdFire DMA controller.

Table 17-35 DMA Config Register Address

BITS	7	6	5	4	3	2	1	0
FIELD							DMA1REQ	DMA0REQ
RESET							0	0
R/W							R/W	R/W
ADDR	MBAR2 + 0X9F							

Table 17-36 DMA Config Bit Descriptions

BIT NAME	DESCRIPTION	NOTES
DMA1REQ	0 = PDIR2 1 = PDOR3	1, 2, 3
DMA0REQ	0 = PDIR2 1 = PDOR3	1, 2, 3

17.6 PHASE/FREQUENCY DETERMINATION AND XTRIM FUNCTION

These features are necessary so that users can determine when a software sample rate convertor should be enabled and provide the necessary control to steer the sample rate convertor clock (when the incoming sample rate is other then 44.1 Khz).

In addition, users can also utilize this function to determine when the incoming IEC958 clock does not match the phase of the Xtal oscillator and use the XTRIM function to trim the external oscillator to match (within a 150ppm range). Typically, when the IEC958 input is being used, the xtal requires trimming to match but this is only when the source is completely external to the application.

When the source is internal to the application, such as from a CD player controlled by the MCF5249, then the input sample rate does not need to match the output sample rate, so they can be asynchronous. When FIFO under-run or over-run occurs, request that the data lost is re-read by the system.

17.6.1 INCOMING SOURCE FREQUENCY MEASUREMENT

The PLL maintaining phase relation between incoming source signal and internal signal is mainly digital. It is necessary, however, to measure phase/frequency of the incoming signal in relationship with the crystal

DRc encoding

:

Table 19-19 Definition of DRc Encoding--Write

DRc[3:0]	DEBUG REGISTER DEFINITION	MNEMONIC	INITIAL STATE
$0	Configuration/Status	CSR	$0
$1-$4	Reserved	-	–
$5	BDM Address Attribute	BAAR	$5
$6	Bus Attributes and Mask	AATR	$5
$7	Trigger Definition	TDR	$0
$8	PC Breakpoint	PBR	–
$9	PC Breakpoint Mask	PBMR	–
$A-$B	Reserved	–	–
$C	Operand Address High Breakpoint	ABHR	–
$D	Operand Address Low Breakpoint	ABLR	–
$E	Data Breakpoint	DBR	–
$F	Data Breakpoint Mask	DBMR	–

Command Sequence:

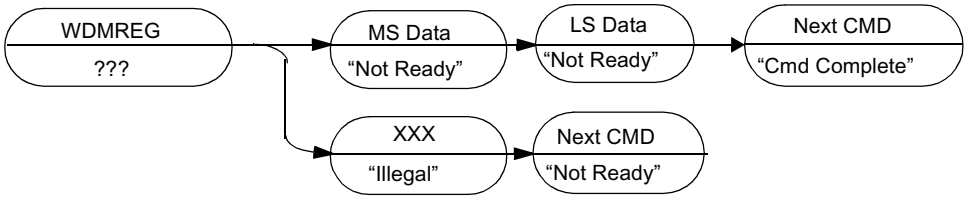

Figure 19-23 Write Debug Module Register Command Sequence

Operand Data:

Longword data is written into the specified debug register. The data is supplied most significant word first.

Result Data:

Command complete status ($0FFFF) is returned when register write is complete.

19.3.3.4.13 Unassigned Opcodes

Unassigned command opcodes are reserved by Motorola. All unused command formats within any revision level perform a NOP and return the ILLEGAL command response.

19.3.3.5 BDM Accesses of the EMAC Registers

The presence of rounding logic in the output data path of the EMAC requires special care for BDM-initiated reads and writes of its programming model. In particular, any result rounding modes must be disabled during the read/write process so the exact bit-wise EMAC register contents are accessed. For example, a BDM read of an accumulator (ACCx) requires the following sequence:

Table 21-19 SCLK INPUT, SDATAI INPUT Timing Specifications

NAME	CHARACTERISTIC			UNIT
		MIN	MAX	
TSU	SDATAI IN to SCLKn	-5	—	ns
TH	SCLK rise to SDATAI	3	—	ns

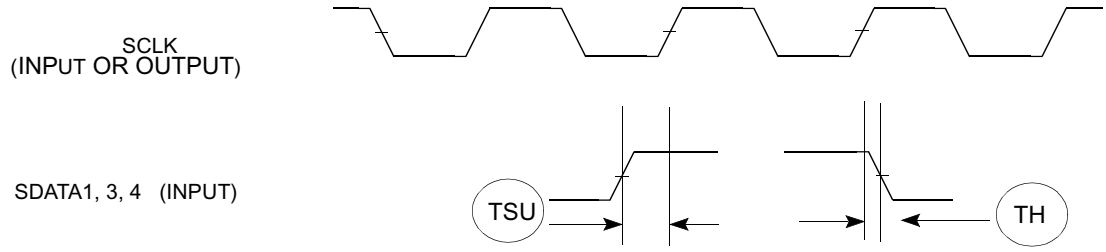

Figure 21-18 SCLK Input/Output, SDATAI Input Timing Diagram

Section 22
Mechanical Data

Visit the URL [http://www.motorola.com/coldfire] and choose the documentation library to obtain information on the mechanical characteristics of the MCF5249 integrated microprocessor.

22.1 PACKAGE

The MCF5249 can be assembled in either a 160-pin MAP BGA or 144-pin QFP package. Thermal characteristics are not available at this time.

22.2 PIN ASSIGNMENT

The MCF5249 is available in 160 pin MAPBGA package and 144 pin QFP package options.

Table 22-3 160 MAPBGA Pin Assignments

PIN BGA	NAME	TYPE	DESCRIPTION
E9	CORE-VDD		CORE-VDD
E9	CORE-VDD		CORE-VDD
E14	RW_B	o	bus write enable
F12	RCK / GPIO 51	io	subcode clock
E13	TMS/BKPT	i	Jtag
E10	CORE-GND		CORE-GND
E10	CORE-GND		CORE-GND
E12	TCK	i	Jtag
E11	PAD-GND		PAD-GND
E11	PAD-GND		PAD-GND
D14	PST3/GPIO 62	io	coldFire debug port
D13	CNPSTCLK / GPO 63	o	coldfire debug clock
C14	PST1/GPIO 60	io	coldFire debug port
D12	PAD-VDD		PAD-VDD
D12	PAD-VDD		PAD-VDD
C13	PST2/GPIO 61	io	coldFire debug port
B14	PST0/GPIO 59	io	coldFire debug port
D11	TDI/DSI	i	Jtag
C12	TEST0	i	structural test
B13	TIN0 / GPI33	i	timer input 0
C11	HI-Z	i	Jtag
A14	DDATA3/GPIO 4	io	coldFire debug port
A13	TOUT0 / GPO33	o	timer output 0
B12	DDATA1/GPIO 1	io	coldFire debug port
A12	DDATA2/GPIO 2	io	coldFire debug port
B11	CTS2_B / ADIN3/GPI31	i	Second UART clear to send, AD input 3
A11	DDATA0/GPIO 0	io	coldFire debug port
B10	RXD2 / GPI28/ADIN2	i	Second UART receive data input AD input 2
D10	TDSO	o	Jtag
A10	RTS2_B / GPO31	o	Second UART request to send
B9	SDATAI3 / GPI 41	i	audio interfaces serial data 3 in
D9	CTS1_B / GPI30	i	First UART clear to send
A9	TXD2 / GPO28	o	Second UART transmit data output
D8	RTS1_B / GPO30	o	First UART request to send
A8	EBUIN4 / ADIN1/GPI 39	i	audio interfaces EBU in 4/ AD convertor input 1
E8	SRE/GPIO11	io	SmartMedia read enable
B8	LRCK3 / GPIO 45	io	audio interfaces serial word clock 3
E7	SWE/GPIO12	io	SmartMedia write enable