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	Not For New Designs
Core Processor	Coldfire V2
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	I²C, IDE, Memory Card, SPI, UART/USART
Peripherals	DMA, I²S, POR, Serial Audio, WDT
Number of I/O	34
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	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcf5249lag120

LIST OF FIGURES

be generated on the rising or falling edge of the pin. All interrupts are autovectored and interrupt levels are programmable.

1.6.23 JTAG

To help with system diagnostics and manufacturing testing, the MCF5249 includes dedicated user-accessible test logic that complies with the IEEE 1149.1A standard for boundary scan testability, often referred to as Joint Test Action Group, or JTAG. For more information, refer to the IEEE 1149.1A standard. Motorola provides BSDL files for JTAG testing.

1.6.24 SYSTEM DEBUG INTERFACE

The ColdFire processor core debug interface supports real-time instruction trace and debug, plus background-debug mode. A background-debug mode (BDM) interface provides system debug.

In real-time instruction trace, four status lines provide information on processor activity in real time (PST pins). A four-bit wide debug data bus (DDATA) displays operand data and change-of-flow addresses, which helps track the machine's dynamic execution path.

1.6.25 CRYSTAL AND ON-CHIP PLL

Typically, an external 16.92 Mhz or 33.86 Mhz clock input is used for CD R/W applications, while an 11.2896 MHz clock is more practical for Portable CD player applications. However, the on-chip programmable PLL, which generates the processor clock, allows the use of almost any low frequency external clock (5-35 Mhz).

Two clock outputs (MCLK1 and MCLK2) are provided for use as Audio Master Clock. The output frequencies of both outputs are programmable to Fxtal, Fxtal/2, Fxtal/3, and Fxtal/4. The Fxtal/3 option is only available when the 33.86 Mhz crystal is connected.

The MCF5249 supports VCO operation of the oscillator by means of a 16-bit pulse density modulation output. Using this mode, it is possible to lock the oscillator to the frequency of an incoming IEC958 or IIS signal. The maximum trim depends on the type and design of the oscillator. Typically a trim of +/- 100 ppm can be achieved with a crystal oscillator and over +/- 1000 ppm with an LC oscillator.

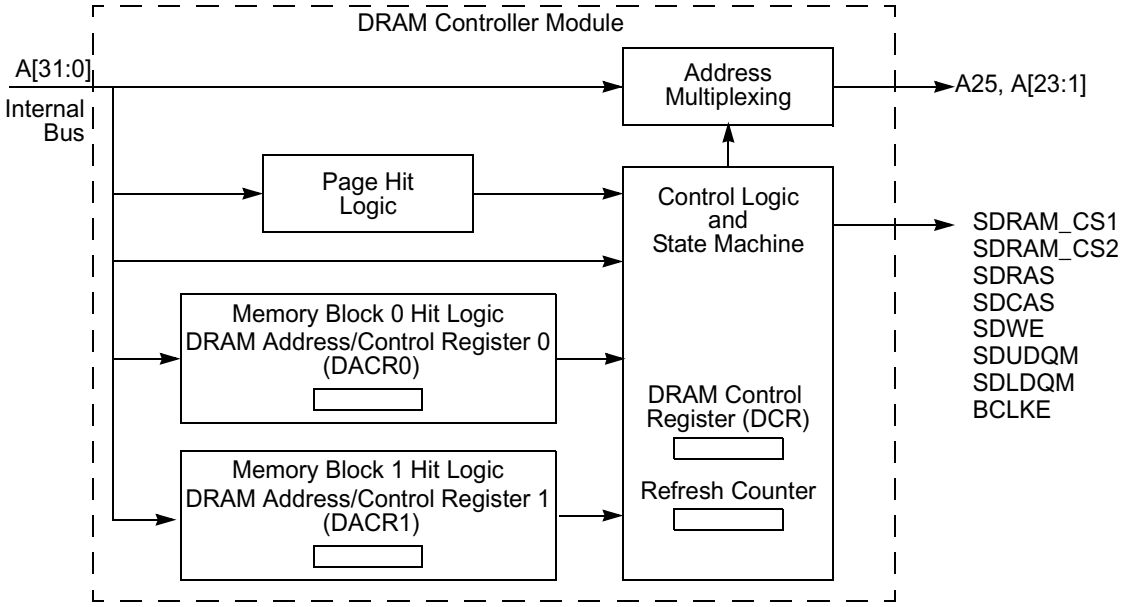

Figure 7-1 Synchronous DRAM Controller Block Diagram

The DRAM controller's major components, shown in Figure 7-1, are described as follows:

- DRAM address and control registers (DACR0 and DACR1)—The DRAM controller consists of two configuration register units, one for each supported memory block. DACR0 is accessed at MBAR + 0x0108; DACR1 is accessed at 0x0110. The register information is passed on to the hit logic.
- Control logic and state machine—Generates all DRAM signals, taking bus cycle characteristic data from the block logic, along with hit information to generate DRAM accesses. Handles refresh requests from the refresh counter.
  - DRAM control register (DCR)—Contains data to control refresh operation of the DRAM controller. Both memory blocks are refreshed concurrently as controlled by DCR[RC].
  - Refresh counter—Determines when refresh should occur, determined by the value of DCR[RC]. It generates a refresh request to the control block.
- Hit logic—Compares address and attribute signals of a current DRAM bus cycle to both DACRs to determine if a DRAM block is being accessed. Hits are passed to the control logic along with characteristics of the bus cycle to be generated.
- Page hit logic—Determines if the next DRAM access is in the same DRAM page as the previous one. This information is passed on to the control logic.
- Address multiplexing—Multiplexes addresses to allow column and row addresses to share pins. This allows glueless interface to DRAMs.

7.2 DRAM CONTROLLER OPERATION

7.2.1 DRAM CONTROLLER REGISTERS

The DRAM controller registers memory map is shown in Table 7-1

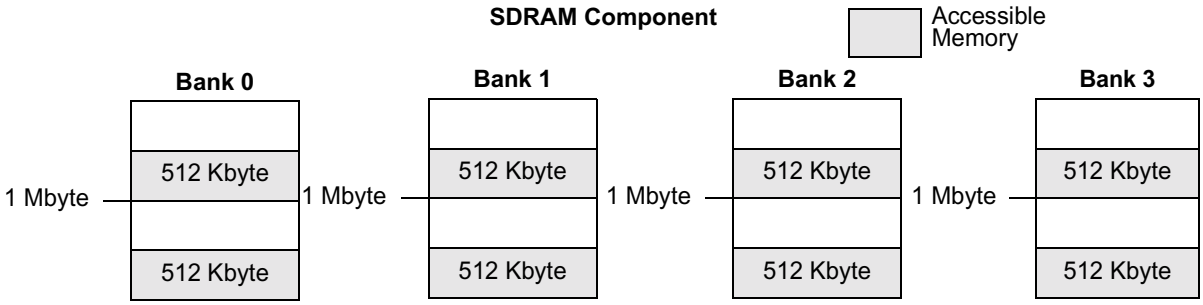

Figure 7-14 SDRAM Configuration

The DACRs should be programmed as shown in Figure 7-15.

	31 … 18	17	16
Field	BA	—	
Setting	1111_1111_1000_10	xx	
(hex)	15	15	8	8

	15	14	13	12	11	10 … 8	7	6	5	4	3	2	1	0
Field	RE	—	CASL		—	CBM	—	IMRS	PS		IP	PM	—	
Setting	0	X	01		X	010	X	0	10		0	1	xx	
(hex)	1				2		2				4			

Figure 7-15 DACR Register Configuration

This configuration results in a value of DACR0 = 0xFF88_1224, as described in Table 7-18. DACR1 initialization is not needed because there is only one block. Subsequently, DACR1[RE,IMRS,IP] should be cleared; everything else is a don't care.

Table 7-18 DACR Initialization Values

BITS	NAME	SETTING	DESCRIPTION
31–18	BA		Base address. So DACR0[31–16] = 0xFF88, which places the starting address of the SDRAM accessible memory at 0xFF88_0000.
17–16	—		Reserved. Don't care.
15	RE	0	0, which keeps auto-refresh disabled because registers are being set up at this time.
14	—		Reserved. Don't care.
13–12	CASL	01	Indicates a delay of data 1 cycle after $\overline{\text{CAS}}$ is asserted
11	—		Reserved. Don't care.
10–8	CBM	010	Command bit is pin 19 and bank selects are 20 and up.
7	—		Reserved. Don't care.
6	IMRS	0	Indicates MRS command has not been initiated.
5–4	PS	10	16-bit port.

bug fixes. For example, developers may want to distribute a single code image or library for use on different revisions of the silicon.

Table 9-7 DeviceID Register (DeviceID)

BITS	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	PART NUMBER															
RESET	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	READ ONLY															
BITS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	PART NUMBER								MASK REVISION							
RESET	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
R/W	READ ONLY															
ADDR	MBAR+2 0xAC															

9.3.3 INTERRUPT CONTROLLER

For legacy reasons, there are two interrupt controllers on the MCF5249.

1. The *primary* interrupt controller offers the same functionality as the MCF5307 interrupt controller.
2. The *secondary* interrupt controller offers additional interrupts for on-chip peripheral devices that are not present in the MCF5307.

The primary interrupt controller is centralized, and services the following:

- Software watchdog timer
- Timer modules
- I^2C 1 module
- UART modules
- DMA module
- QSPI module

The secondary interrupt controller is decentralized, and services the following:

- gpio interrupts
- Audio interface module
- MemoryStick/SD module
- AD convertor module
- I^2C 2 module

9.4 INTERRUPT INTERFACE

9.4.1 PRIMARY CONTROLLER INTERRUPT REGISTERS

Primary internal interrupt sources have their own interrupt control registers ICR[11:0], IPR, and IMR. Table 9-7 gives the location and description of each ICR.

Table 9-29 SWT Timeout Period (Continued)

SWP	SWT[1:0]	SWT TIMEOUT PERIOD
0	10	2^{13} / System Frequency
0	11	2^{15}/ System Frequency
1	00	2^{22} / System Frequency
1	01	2^{24} / System Frequency
1	10	2^{26} / System Frequency
1	11	2^{28}/ System Frequency

Note: If the SWP and SWT bits are modified to select a new software timeout, users must peform the software service sequence ($55 followed by $AA written to the SWSR) before the new timeout period takes effect.

Table 9-30 SWP and SWT Bit Descriptions

BIT NAME	DESCRIPTION
SWTA	Software Watchdog Transfer Acknowledge Enable 0 = SWTA Transfer Acknowledge disabled 1 = SWTA Assert Transfer Acknowledge enabled. After 1 SWT timeout period of the unacknowledged assertion of the SWT interrupt, the Software Watchdog Transfer Acknowledge will assert, which allows SWT to terminate a bus cycle and allow the IACK to occur
SWTAVAL	Software Watchdog Transfer Acknowledge Valid 0 = SWTA Transfer Acknowledge has NOT occurred 1 = SWTA Transfer Acknowledge has occurred. Write a 1 to clear this flag bit

9.5.2.2 Software Watchdog Interrupt Vector Register

The SWIVR contains the 8-bit interrupt vector the SIM returns during an interrupt- acknowledge cycle in response to a SWT-generated interrupt. The following register illustrates the SWIVR programming model.

The SWIVR is an 8-bit supervisor write-only register. This register is set to the uninitialized vector $0F at system reset

.

Table 9-31 Software Watchdog Interrupt Vector Register (SWIVR)

BITS	7	6	5	4	3	2	1	0
FIELD	SWIV7	SWIV6	SWIV5	SWIV4	SWIV3	SWIV2	SWIV1	SWIV0
RESET	0	0	0	0	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	MBAR + $(0X02)							

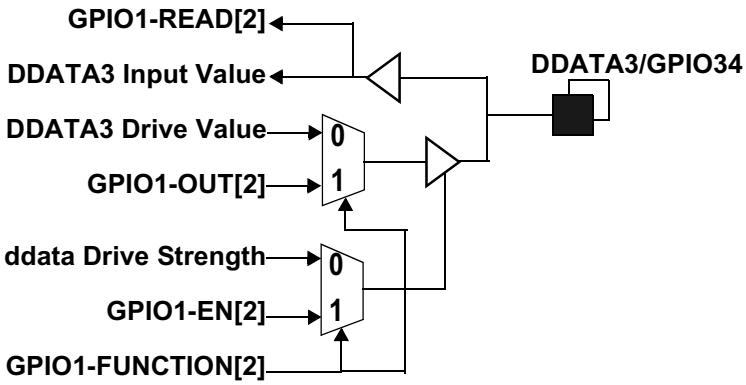

Figure 9-2 General-Purpose Pin Logic for Pin ddata3/gpio34

Table 9-42 General-Purpose Output Register Bits to Pins Mapping

GPIO-FUNCTION GPIO-EN GPIO-OUT BIT NUMBER	ASSOCIATED PIN	PIN TYPE	GPIO1-FUNCTION GPIO1-EN GPIO1-OUT BIT NUMBER	ASSOCIATED PIN	PIN TYPE
31	RTS2_B/GPO31	O	63	PSTCLK/GPO63	O
30	RTS1_B/GPO30	O	62	PST3/GPIO62	I/O
29	QSPI_CS0/GPIO29	I/O	61	PST2/GPIO61	I/O
28	TXD2/GPO28	O	60	PST1/GPIO60	I/O
27	TXD1/GPO27	O	59	PST0/GPIO59	I/O
26	QSPI_DOUT/GPIO26	I/O	58	CS1/GPIO58	I/O
25	SDATAO1/GPIO25	I/O	57	BUFENB1/GPIO57	I/O
24	QSPI_CS1/GPIO24	I/O	56	SDATA3/GPIO56	I/O
23	TIN1/GPIO23	I/O	55	SDA2/GPIO55	I/O
22	QSPI_CS3/GPIO22	I/O	54	SDATA0_SDIO1/GPIO54	I/O
21	QSPI_CS2/GPIO21	I/O	53	SUBR/GPIO53	I/O
20	TA/GPIO20	I/O	52	SFSY/GPIO52	I/O
19	EF/GPIO19	I/O	51	RCK/GPIO51	I/O
18	CFLG/GPIO18	I/O	50	SCLK4/GPIO50	I/O
17	BUFENB2/GPIO17	I/O	49	SCLK3/GPIO49	I/O
16	IDE_IORDY/GPIO16	I/O	48	SCLK2/GPIO48	I/O
15	SCLK_OUT/GPIO15	I/O	47		
14	IDE_DIOW/GPIO14	I/O	46	LRCK4/GPIO46	I/O
13	CS2/IDE_DIOR/GPIO13	I/O	45	LRCK3/GPIO45	I/O
12	SWE/GPIO12	I/O	44	LRCK2/GPIO44	I/O

Table 12-2 ADconfig (ADconfig) Register

BITS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	-	-	-	-	-	ADOUT_SEL	SOURCE SELECT		INT INTCLEAR	INTERRUPT ENABLE	ADOUT_DRIVE		ADCLK_SEL			
RESET	-	-	-	-	-	0	000		-	-	-	-	-	-	-	-
R/W	R/W															
Address MBAR2BAS + 0x402																

Table 12-3 ADconfig Register Bit Descriptions

FIELD	FIELD NAME	DESCRIPTION	RESET
10	ADOUT_SEL	1: TOUT1/GPO35/ADOUT pin function is ADOUT 0: TOUT1/GPO35/ADOUT pin function is not ADOUT	0
9:8	SOURCE SELECT	00: in0 01: in1 10: in2 11: in3	000
7	INT INTCLEAR	(On read): '1' indicates interrupt pending '0' no interrupt pending (On write): '1' clear interrupt '0' no action	
6	INTERRUPT ENABLE	0: interrupt disabled 1: interrupt enabled	
5:4	ADOUT_DRIVE	01: ADout tri-state 00: ADout drives +Vdd for Hi, GND for lo 11: ADout drives +Vdd for Hi, Hi-Z for lo 10: ADout drives Hi-Z for Hi, GND for lo	
3:0	ADCLK_SEL	0: adclk = busclk 1: adclk = busclk / 2 2: adclk = busclk / 4 3: adclk = busclk / 8 4: adclk = busclk / 16 5: adclk = busclk / 32 6: adclk = busclk / 64 7: adclk = busclk / 128 8: adclk = busclk / 256	

Note: 1. Measurement frequency and interrupt frequency is adclk / 4096
Note: 2. For the circuit shown Figure 12-1, the adout_drive should be set to 00.
Other circuits can use settings 10 or 11.
Note: 3. AD resolution is 12 bits. AD precision depends on many factors, and is TBD.
Note: 4. Only one channel can be measured simultaneously.

13.4.5.2 Writing Data to the MemoryStick

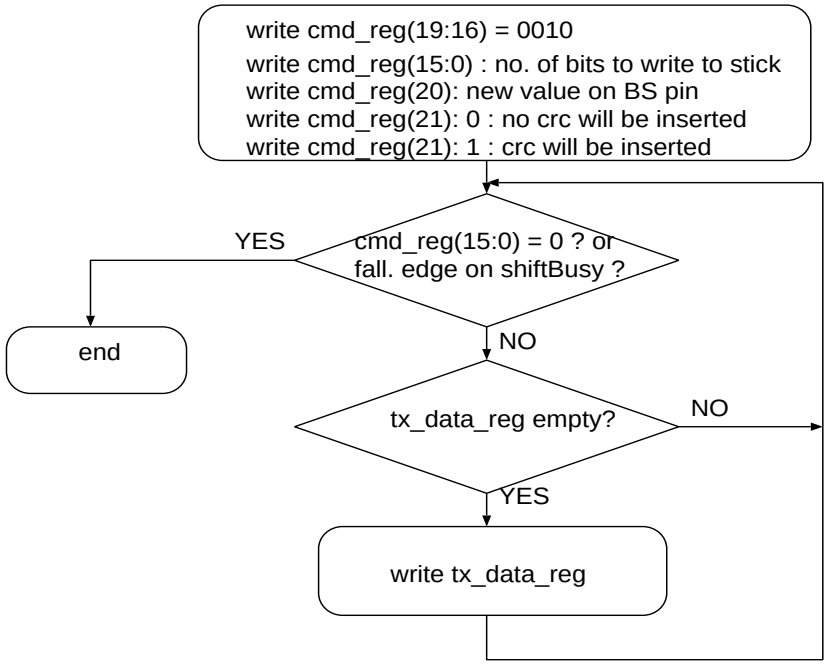

Figure 13-12 Writing Data To MemoryStick

A timing diagram is also given. In this timing diagram, the assumption is made the processor writes the empty transmit buffer register before the next 32 bits are transmitted. If this is not the case, the FlashMedia interface will stop the outgoing sclk clock, and in this way prevent data underrun.

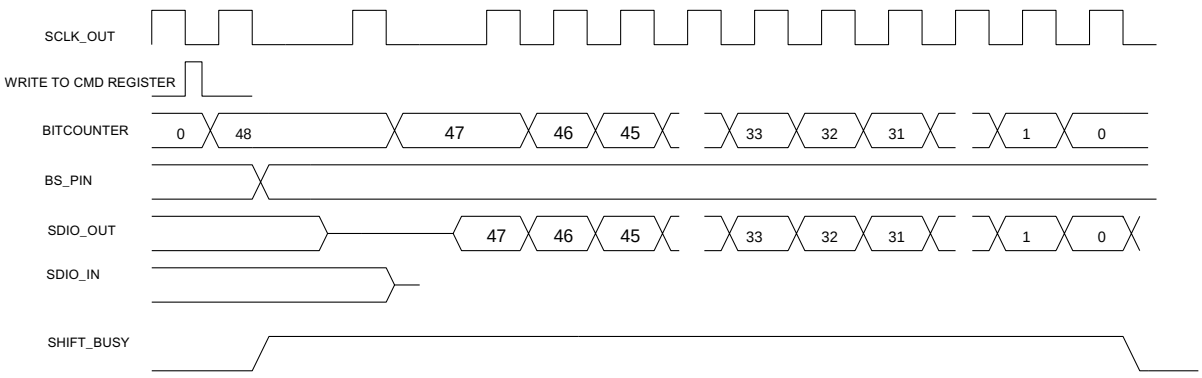

Figure 13-13 Writing Data to MemoryStick Timing

Section 14
DMA Controller Module

The direct memory access (DMA) controller module quickly and efficiently moves blocks of data with minimal processor overhead. The DMA module, shown in Figure 14-1, provides four channels that allow byte, word, or longword operand transfers. These transfers should be dual address to on-chip devices; such as the UART, SDRAM controller, and audio module.

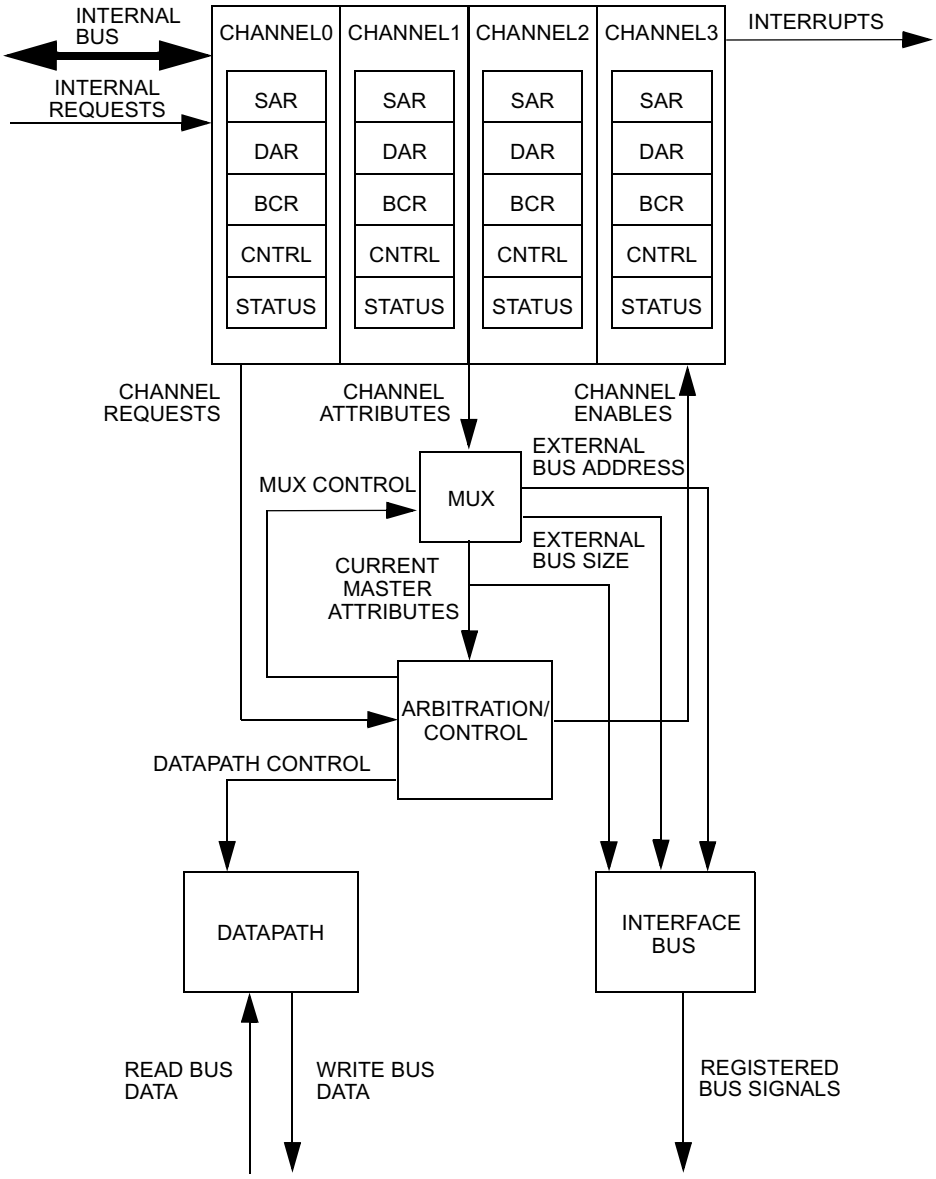

Figure 14-1 DMA Signal Diagram

break detection operate normally. The A/D bit takes the place of the parity bit; therefore, parity is neither calculated nor checked. Messages in this mode can still contain error detection and correction information. One way to provide error detection, if 8-bit characters are not required, is to use software to calculate parity and append it to the 5-, 6-, or 7-bit character.

15.3.5 BUS OPERATION

This section describes the operation of the bus during read, write, and interrupt- acknowledge cycles to the UART module. All UART module registers must be accessed as bytes.

15.3.5.1 Read Cycles

The CPU accesses the UART module with 1 to 2 wait states because the core system clock is divided by 2 for the UART module. The UART module responds to reads with byte data on D[7:0]. Reserved registers return logic zero during reads.

15.3.5.2 Write Cycles

The CPU with zero wait states accesses the UART module. The UART module accepts write data on D[7:0]. Write cycles to read-only registers and reserved registers complete in a normal manner without exception processing; however, the data is ignored.

15.3.5.3 Interrupt Acknowledge Cycles

The UART module can arbitrate for interrupt servicing and supply the interrupt vector when it has successfully won arbitration. The vector number must be provided if interrupt servicing is necessary; thus, the interrupt vector register (UIVR) must be initialized. The interrupt vector number generated by the IVR is used if the autovector is not enabled in the SIM Interrupt Control Register (ICR). If the UIVR is not initialized and the ICR is not programmed for autovector, a spurious interrupt exception is taken if interrupts are generated. This works in conjunction with the MCF5249 interrupt controller, which allows a programmable Interrupt Priority Level (IPL) for the interrupt.

15.4 REGISTER DESCRIPTION AND PROGRAMMING

This section contains a detailed description of each register and its specific function as well as flowcharts of basic UART module programming.

15.4.1 REGISTER DESCRIPTION

Writing control bytes into the appropriate registers controls the UART operation. A list of UART module registers and their associated addresses is shown in Table 15-1.

Note: All UART module registers are accessible only as bytes. The contents of the mode registers (UMR1 and UMR2), clock-select register (UCSR), and the auxiliary control register (UACR) bit 7 should be changed only after the receiver/transmitter is issued a software RESET command—i.e., channel operation must be disabled. Be careful if the register contents are changed during receiver/transmitter operations because unpredictable results can occur.

For the registers described in this section, the numbers above the register description represent the bit position in the register. The register description contains the mnemonic for the bit. The values as shown in the following tables are the values of those register bits after a hardware reset. A value of U indicates that the bit value is unaffected by reset. The read/write status is shown in the last line.

Table 17-7 IIS Configuration Bit Descriptions (Continued)

BIT NAME	BITS	DESCRIPTION
SIZE	7,6	See notes 3, 4, and 8 following bit these descriptions. 00: 16 bits 01: 18 bits 10: 20 bits 11: zero
MODE	5	1 = Sony, EIAJ mode 0 = Philips IIS mode
LRCK FREQUENCY	4,.3,2	100: 64 bit clocks / word clock 010: 48 bit clocks / word clock 000: 32 bit clocks / word clock Other settings: reserved, undefined
LRCK INVERT	1	See note 5 following bit these descriptions. 1 = Invert on word clock 0 = No invert on word clock
SCLK INVERT	0	See note 6following bit these descriptions. 1 = Invert on bit clock 0 = No invert on bit clock

Note: 1. Audio Clk is normally 16.93 MHz. Actual value given Table 4-4 on page 4-5. When divided SYSCLOCK is selected as SCLK output, LRCK will be output too, and is divided from SCLK, using division factor defined by field *4,3,2 - LRCK frequency*.
Note: 2. When bit 11 is set, FIFO is in reset condition. The FIFO is always re-set to “1 sample remaining”. The value of the remaining one sample will be all-zero.
Note: 3. When Philips IIS mode is selected, 16-18-20 bits should yield same result.
Note: 4. Internal interface in MCF5249 is 40 bits / sample (20 left + 20 right). 16, 18 bit words are padded with zeros
Note: 5. LRCK invert will invert the incoming LRCK signal between the pin and the serial data receiver and transmitter
Note: 6. SCLK invert will invert the incoming SCLK signal between the pin and the serial data receiver and transmitter.
Note: 7. Reset to one sample remaining is used to synchronize the data transfer from one input interface to another output interface running at the same frequency.
Note: 8. “zero” means data is transferred at the sampling frequency, with all data cleared down to digital zero.
Note: 9. PDOR1, PDOR2, PDOR3: ColdFire data out registers.
Note: 10. Serial data transmit / receive interfaces have no limit on minimum incoming or outgoing sampling frequency. The maximum SCLK frequency is limited to 1/3 of the internal system clock (CPUclk/2). Mark/space ratio should be equal or better than 38/62.
Note: 11. Reprogramming bits 15-12 during functional operation is not allowed. Reprogramming only allowed while FIFO is in reset condition (bit 11 set ‘1’)
Note: 12. When “digital zero” is selected as source, the FIFO outputs “zero” on its outgoing data bus, regardless of the input side and content of the FIFO. No FIFO related exceptions are generated.
Note: 13. When the FIFO leaves the reset state, because the user writes a “normal operation” state into the control register, while previous state was reset state, the **FIFO is kept in reset until the first long-word is written to it.** As a result, the “start” of the normal operation is synchronized with the writing of the first data into the fifo.
Note: 14. When IIS/Sony interface LRCK/SCLK is set in “follow IIS” mode, the bit clock and word clock become exactly identical to bit and word clock of followed interface. If e.g. LRCK/SCLK for IIS interface 2 is set in “follow IIS1”, the DAC or AD connected to IIS2 can use bit clock and word clock of IIS1. Bit and word clock for IIS2 can be used as gpio.
Note: 15. Bit 16 extends the Tx FIFO control bit and the bit order becomes 16, 10, 9, 8.
Note: 16. These bits should be programmed zero for normal operation. For interface1 receiver, it is possible to use the special EF/CFLG insertion mode, by setting bit 18 = 1. This mode is intended to interface with Philips CD decoders (SAA7345 and successors). When this mode is used, IIS1CONFIG must be programmed to “Sony” mode, 16 bits. The SAA7345 must also be programmed to “Sony” mode, 16 bits. The CFLG flag coming from SAA7345 must be connected with CFLG input. The EF flag coming from SAA7345 must be connected with EF input. If all this is done correctly, the device will receive the 16 MSB ‘s of the incoming data in bits [17:2] of the received serial data. Bit [1] of the received data is the EF flag of the corresponding word, as output by SAA7345. Bit [1] will be set if the MSB or the LSB or both are flagged. Bit [0] of the received data is the CFLG flag of the corresponding word, as output by SAA7345. These flags can be used for implementing an electronic shock protection FIFO.

Table 17-31 Interrupt Register Description (0x94, 0x98) (Continued)

BIT	INTERRUPT NAME	DESCRIPTION	HOW TO CLEAR
4	iis1TxEmpty	IIS 1 transmit fifo empty	write to FIFO
3	iis2TxEmpty	IIS 2 transmit fifo empty	write to FIFO
2	ebuTxEmpty	IEC958 transmit fifo empty	write to FIFO
1	PDIR2 full	Processor data input full	read from PDIR2
0	PDIR1 full	Processor data input full	read from PDIR1

17.4.6.4 Audio Interrupt Routines and Timing

Usually, the MCF5249 processor will run an audio interrupt routine. Every time the audio interrupt routine runs, it will process 2, 3, or 4 audio samples, and send this many samples to one or more PDOR output registers. Also, the audio interrupt routine will read one or more PDIR registers until empty.

In the audio interrupt routine, typically at the beginning, the PDIR registers are read until empty, while the PDOR registers are written at the end of the routine when all calculations are completed. Due to this calculation latency, there is a delay between entering the audio interrupt routine and the filling of the transmit FIFOs.

Due to this delay, it is difficult to “fire” the audio interrupt routine on a transmit FIFO empty interrupt. Because of the extra delay before the data is written, the transmit fifo will underrun before any data is written.

To make it easy for the programmer, the audioTick interrupt was added. To start the audio interrupt routine, use the following sequence:

1. Reset the transmit FIFOs
2. Program the transmit FIFOs to correct source, de-assert reset on transmit FIFOs
3. Reset the PDIR FIFOs
4. Load audio interrupt routine in on-chip SRAM
5. Release reset for the PDIR FIFOs and enable audioTick interrupt

The transmit FIFOs have a special feature. After the software releases the reset to them, they will stay in reset until the audio Interrupt Routine writes data to them for the first time. So, during Step 2 of above mentioned start-up procedure, all transmit data out FIFOs are set in reset, with one sample remaining. They will stay in this state, until the audio Interrupt Routine writes data to them. At this point in time, they are then filled up with extra 2,3, or 4 samples to a total of 3,4, or 5 samples. Also, the first data write to the FIFOs releases the reset, and starts transmission of FIFO data on the corresponding transmit output. (IIS1, IIS2 or IEC958). The next time that data is written to the FIFOs in the audioTick interrupt routine, 2,3, or 4 samples have been transmitted and the FIFO is ready to accept new data.

To work properly, the jitter from one audioTick write point to the next is important. Jitter should be lower than 1 sample period if data is written in groups of 2 or 3 samples to the transmit FIFOs, and lower than 1/2 sample period if data is written in groups of 4 samples to the transmit FIFOs.

The receive FIFOs (PDIR) don’t have an auto-reset-de-assert mechanism, and should be released out of reset just before enabling audioTick interrupt.

Figure 17-8 shows the timing (relative to the Word Clock) of the Empty, Under-run, and Audio Tick interrupts. Each FIFO holds up to six audio samples (left and right).

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

configuration/status register (CSR). Additionally, execution of the WDDATA instruction by the processor captures operands which are displayed on DDATA. These signals are updated each processor cycle.

19.1.3 DEVELOPMENT SERIAL CLOCK (DSCLK)

This input signal is synchronized internally and provides the clock for the serial communication port to the debug module. The maximum frequency is 1/5 the speed of the processor's clock (CLK). At the synchronized rising edge of DSCLK, the data input on DSI is sampled, and the DSO output changes state. See Figure 19-3 for more information,

19.1.4 DEVELOPMENT SERIAL INPUT (DSI)

The input signal is synchronized internally and provides the data input for the serial communication port to the debug module.

19.1.5 DEVELOPMENT SERIAL OUTPUT (DSO)

This signal provides serial output communication for the debug module responses.

19.1.6 PROCESSOR STATUS (PST[3:0])

These output signals report the processor status. Table 19-1 shows the encoding of these signals. These outputs indicate the current status of the processor pipeline and are not related to the current bus transfer. The PST value is updated each processor cycle.

Table 19-30 Access and Operand Data Location

ADDRESS[1:0]	ACCESS SIZE	OPERAND LOCATION
00	Byte	Data[31:24]
01	Byte	Data[23:16]
10	Byte	Data[15:8]
11	Byte	Data[7:0]
0x	Word	Data[31:16]
1x	Word	Data[15:0]
xx	Long	Data[31:0]

19.4.2.5 Trigger Definition Register (TDR)

The TDR configures the operation of the hardware breakpoint logic within the debug module and controls the actions taken under the defined conditions. The breakpoint logic may be configured as a one- or two-level trigger, where bits [31:16] of the TDR define the 2nd level trigger and bits [15:0] define the first level trigger. The TDR is accessible in supervisor mode as debug control register $7 using the WDEBUG instruction and through the BDM port using the WDMREG command.

Section 20
IEEE 1149.1 Test Access Port (JTAG)

The MCF5249 JTAG test architecture implementation currently supports circuit board test strategies that are based on the IEEE standard. This architecture provides access to all of the data and chip control pins from the board edge connector through the standard four-pin test access port (TAP) and the active-low JTAG reset pin, $\overline{\text{TRST}}$. The test logic uses static design and is wholly independent of the system logic, except where the JTAG is subordinate to other complimentary test modes (see the DEBUG section, for more information). When in subordinate mode, the JTAG test logic is placed in reset and the TAP pins can be used for other purposes in accordance with the rules and restrictions set forth using a JTAG compliance-enable pin.

The MCF5249 JTAG implementation can do the following:

- Perform boundary-scan operations to test circuit board electrical continuity
- Bypass the MCF5249 by reducing the shift register path to a single cell
- Sample the MCF5249 system pins during operation and transparently shift out the result
- Set the MCF5249 output drive pins to fixed logic values while reducing the shift register path to a single cell
- Protect the MCF5249 system output and input pins from backdriving and random toggling (such as during in-circuit testing) by placing all system signal pins to high- impedance state

Note: The IEEE Standard 1149.1 test logic cannot be considered completely benign to those planning not to use JTAG capability. Users must observe certain precautions to ensure that this logic does not interfere with system or debug operation. Refer to Section **20.6 Disabling IEEE 1149.1A Standard Operation** .

20.1 JTAG OVERVIEW

Figure 20-1 is a block diagram of the MCF5249 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.

```
“184 (BC_2, cs0_pin,            output3,     X,   183,  0,    Z   ) ,” &
“185 (BC_2, *,                  control,     0   ) ,” &
“186 (BC_7, scl_qspiclk_pin,    bidir,       X,   185,  0,    Z   ) ,” &
“187 (BC_2, *,                  control,     0   ) ,” &
“188 (BC_2, a12_pin,            output3,     X,   187,  0,    Z   ) ,” &
“189 (BC_2, *,                  control,     0   ) ,” &
“190 (BC_2, a7_pin,             output3,     X,   189,  0,    Z   ) ,” &
“191 (BC_2, *,                  control,     0   ) ,” &
“192 (BC_2, a8_pin,             output3,     X,   191,  0,    Z   ) ,” &
“193 (BC_2, *,                  control,     0   ) ,” &
“194 (BC_2, a5_pin,             output3,     X,   193,  0,    Z   ) ,” &
“195 (BC_2, *,                  control,     0   ) ,” &
“196 (BC_2, a6_pin,             output3,     X,   195,  0,    Z   ) ,” &
“197 (BC_2, *,                  control,     0   ) ,” &
“198 (BC_2, a4_pin,             output3,     X,   197,  0,    Z   ) ,” &
“199 (BC_2, *,                  control,     0   ) ,” &
“200 (BC_2, a3_pin,             output3,     X,   199,  0,    Z   ) ,” &
“201 (BC_2, *,                  control,     0   ) ,” &
“202 (BC_2, a2_pin,             output3,     X,   201,  0,    Z   ) ,” &
“203 (BC_2, *,                  control,     0   ) ,” &
“204 (BC_7, tin1_gp23_pin,      bidir,       X,   203,  0,    Z   ) ,” &
“205 (BC_2, *,                  control,     0   ) ,” &
“206 (BC_2, a1_pin,             output3,     X,   205,  0,    Z   ) ,” &
“207 (BC_2, *,                  control,     0   ) ,” &
“208 (BC_7, cs1_gp58_pin,       bidir,       X,   207,  0,    Z   ) ,” &
“209 (BC_4, rxd1_gpi27_pin,     input,       X   ) ,” &
“210 (BC_2, *,                  control,     0   ) ,” &
“211 (BC_7, sclk3_gp49_pin,     bidir,       X,   210,  0,    Z   ) ,” &
“212 (BC_2, *,                  control,     0   ) ,” &
“213 (BC_2, txd1_gpo27_pin,     output3,     X,   212,  0,    Z   ) ,” &
“214 (BC_2, *,                  control,     0   ) ,” &
“215 (BC_7, swe_gp12_pin,       bidir,       X,   214,  0,    Z   ) ,” &
“216 (BC_2, *,                  control,     0   ) ,” &
“217 (BC_7, lrck3_gp45_pin,     bidir,       X,   216,  0,    Z   ) ,” &
“218 (BC_2, *,                  control,     0   ) ,” &
“219 (BC_7, sre_gp11_pin,       bidir,       X,   218,  0,    Z   ) ,” &
“220 (BC_4, ebuin4_adin1_gpi39_pin, input,   X   ) ,” &
“221 (BC_2, *,                  control,     0   ) ,” &
“222 (BC_2, rts1_gpo30_pin,     output3,     X,   221,  0,    Z   ) ,” &
“223 (BC_2, *,                  control,     0   ) ,” &
“224 (BC_2, txd2_gpo28_pin,     output3,     X,   223,  0,    Z   ) ,” &
“225 (BC_4, cts1_gpi30_pin,     input,       X   ) ,” &
“226 (BC_4, sdatai3_gpi41_pin,  input,       X   ) ,” &
“227 (BC_2, *,                  control,     0   ) ,” &
“228 (BC_2, rts2_gpo31_pin,     output3,     X,   227,  0,    Z   ) ,” &
“229 (BC_4, rxd2_adin2_gpi28_pin, input,     X   ) ,” &
“230 (BC_2, *,                  control,     0   ) ,” &
“231 (BC_7, dbdcddata0_gp0_pin, bidir,       X,   230,  0,    Z   ) ,” &
“232 (BC_4, cts2_adin3_gpi31_pin, input,     X   ) ,” &
“233 (BC_2, *,                  control,     0   ) ,” &
“234 (BC_7, dbdcddata2_gp2_pin, bidir,       X,   233,  0,    Z   ) ,” &
“235 (BC_2, *,                  control,     0   ) ,” &
“236 (BC_7, dbdcddata1_gp1_pin, bidir,       X,   235,  0,    Z   ) ,” &
“237 (BC_2, *,                  control,     0   ) ,” &
```

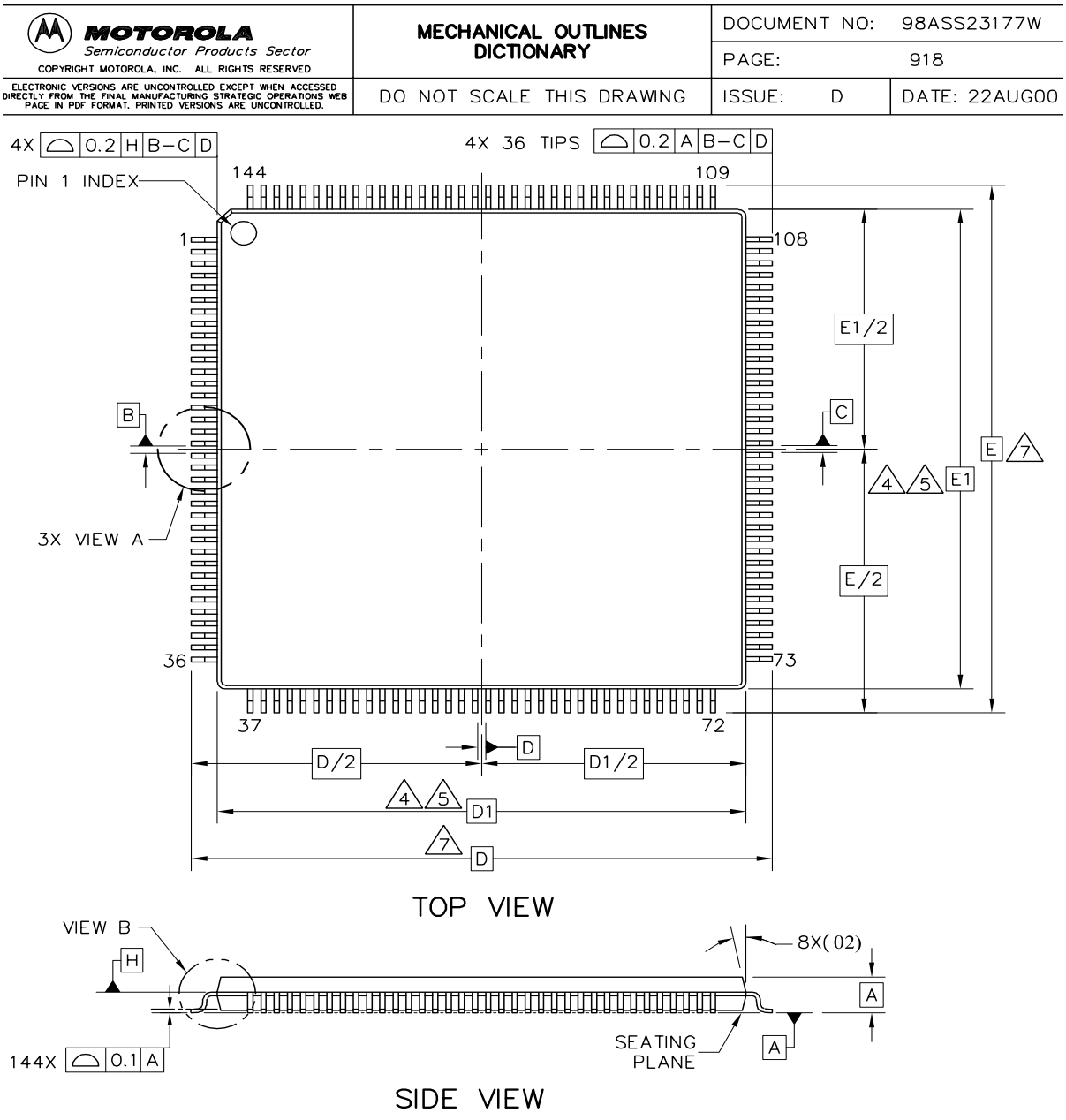

Figure 22-1 144 QFP Package (1 of 3)

Table A-2 MBAR Address Space Memory Map

ADDRESS	BYTE 0	BYTE 1	BYTE 2	BYTE 3	DESCRIPTION
MBAR + 21Ch	UBG21				UART baud rate generator LSB
MBAR + 230	UIVR1				UART interrupt vector reg 1
MBAR + 234h	UIP1				UART interrupt port 1
MBAR + 238h	UOP11				UART RTS Output Port 1
MBAR + 23Ch	UOP01				UART Output Port 1
MBAR + 280h	MADR				Mbus address reg
MBAR + 284h	MFDR				Mbus frequency reg
MBAR + 288h	MBCR				Mbus control reg
MBAR + 28Ch	MBSR				Mbus status reg
MBAR + 290h	MBDR				Mbus data reg
MBAR + 300h	SAR0				DMA source address reg 0
MBAR + 304h	DAR0				DMA destination addr reg 0
MBAR + 308h	DCR0				DMA control reg 0
MBAR + 30Ch	BCR0				DMA byte count reg 0
MBAR + 310h	DSR0				DMA status reg 0
MBAR + 314h	DIVR0				DMA vector reg 0
MBAR + 340h	SAR1				DMA source address reg 1
MBAR + 344h	DAR1				DMA destination addr reg 1
MBAR + 348h	DCR1				DMA control reg 1
MBAR + 34Ch	BCR1				DMA byte count reg 1
MBAR + 350h	DSR1				DMA status reg 1
MBAR + 354h	DIVR1				DMA vector reg 1
MBAR + 380h	SAR2				DMA source address reg 2
MBAR + 384h	DAR2				DMA destination addr reg 2
MBAR + 388h	DCR2				DMA control reg 2
MBAR + 38Ch	BCR2				DMA byte count reg 2
MBAR + 390h	DSR2				DMA status reg 2
MBAR + 394h	DIVR2				DMA vector reg 2
MBAR + 3C0h	SAR3				DMA source address reg 3
MBAR + 3C4h	DAR3				DMA destination addr reg 3
MBAR + 3C8h	DCR3				DMA control reg 3
MBAR + 3CCh	BCR3				DMA byte count reg 3
MBAR + 3D0h	DSR3				DMA status reg 3
MBAR + 3D4h	DIVR3				DMA vector reg 3
MBAR + 400	QIR				QSPI mode register
MBAR + 404	QSPIQDLYR				QSPI delay register
MBAR + 408	QSPIQWR				QSPI Wrap register
MBAR + 40C	QSPIQIR				QSPI Interrupt register
MBAR + 410	QSPIQAR				QSPI address register