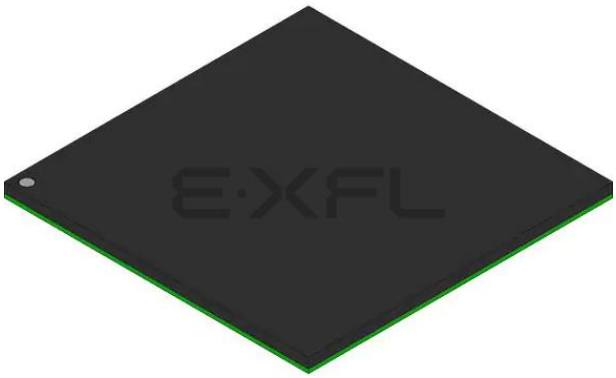

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcf5280cvm66

Chapter 4
Cache

Chapter 5
Static RAM (SRAM)

Chapter 6
ColdFire Flash Module (CFM)

Chapter 16
DMA Controller Module

Chapter 31
IEEE 1149.1 Test Access Port (JTAG)

Table 2-8. ColdFire Opword Line Definition (continued)

Opword[Line]	Instruction Class
0xA	EMAC, Move 3-bit Quick (MOV3Q)
0xB	Compare (CMP), Exclusive-OR (EOR)
0xC	Logical AND (AND), Multiply Word (MUL)
0xD	Add (ADD), Add Extended (ADDX)
0xE	Arithmetic and logical shifts (ASL, ASR, LSL, LSR)
0xF	Cache Push (CPUSHL), Write DDATA (WDDATA), Write Debug (WDEBUG)

In the original M68000 ISA definition, lines A and F were effectively reserved for user-defined operations (line A) and co-processor instructions (line F). Accordingly, there are two unique exception vectors associated with illegal opwords in these two lines.

Any attempted execution of an illegal 16-bit opcode (except for line-A and line-F opcodes) generates an illegal instruction exception (vector 4). Additionally, any attempted execution of any non-MAC line-A and most line-F opcodes generate their unique exception types, vector numbers 10 and 11, respectively. ColdFire cores do not provide illegal instruction detection on the extension words on any instruction, including MOVEC.

2.3.4.4 Divide-By-Zero

Attempting to divide by zero causes an exception (vector 5, offset equal 0x014).

2.3.4.5 Privilege Violation

The attempted execution of a supervisor mode instruction while in user mode generates a privilege violation exception. See *ColdFire Programmer's Reference Manual* for a list of supervisor-mode instructions.

There is one special case involving the HALT instruction. Normally, this opcode is a supervisor mode instruction, but if the debug module's CSR[UHE] is set, then this instruction can be also be executed in user mode for debugging purposes.

2.3.4.6 Trace Exception

To aid in program development, all ColdFire processors provide an instruction-by-instruction tracing capability. While in trace mode, indicated by setting of the SR[T] bit, the completion of an instruction execution (for all but the stop instruction) signals a trace exception. This functionality allows a debugger to monitor program execution.

The stop instruction has the following effects:

1. The instruction before the stop executes and then generates a trace exception. In the exception stack frame, the PC points to the stop opcode.
2. When the trace handler is exited, the stop instruction executes, loading the SR with the immediate operand from the instruction.

of these registers. The CACR and ACRs can only be accessed in supervisor mode using the MOVEC instruction with an Rc value of 0x002, 0x004 and 0x005, respectively.

Table 4-1. Cache Memory Map

BDM[1]	Register	Width (bits)	Access[2]	Reset Value	Section/Page
0x002	Cache Control Register (CACR)	32	W	0x0000_0000	4.2.1/4-3
0x004	Access Control Register 0 (ACR0)	32	W	See Section	4.2.2/4-6
0x005	Access Control Register 1 (ACR1)	32	W	See Section	4.2.2/4-6

1 The values listed in this column represent the Rc field used when accessing the core registers via the BDM port. For more information see Chapter 30, “Debug Support.”

2 Readable through debug.

4.2.1 Cache Control Register (CACR)

The CACR controls the operation of the cache. The CACR provides a set of default memory access attributes used when a reference address does not map into the spaces defined by the ACRs.

The CACR is a 32-bit, write-only supervisor control register. It is accessed in the CPU address space via the MOVEC instruction with an Rc encoding of 0x002. The CACR can be read when in background debug mode (BDM). Therefore, the register diagram, Figure 4-2, is shown as read/write. At system reset, the entire register is cleared.

BDM: 0x002 (CACR)

Access: Supervisor write-only
Debug read/write

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CENB	0	0	CPD	CFRZ	0	0	CINV	DISI	DISD	INVI	INVD	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CEIB	DCM	DBWE	0	0	DWP	EUSP	0	0	CLNF	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4-2. Cache Control Register (CACR)

Table 4-2. CACR Field Descriptions

Field	Description
31 CENB	Cache enable. The memory array of the cache is enabled only if CENB is asserted. This bit, along with the DISI (disable instruction caching) and DISD (disable data caching) bits, control the cache configuration. 0 Cache disabled 1 Cache enabled Table 4-3 describes cache configuration.
30–29	Reserved, must be cleared.

Chapter 5
Static RAM (SRAM)

5.1 SRAM Features

- One 64-Kbyte SRAM
- Single-cycle access
- Physically located on processor's high-speed local bus
- Memory location programmable on any 0-modulo-64 Kbyte address
- Byte, word, longword address capabilities

5.2 SRAM Operation

The SRAM module provides a general-purpose memory block that the ColdFire processor can access in a single cycle. The location of the memory block can be specified to any 0-modulo-64K address within the 4-GByte address space. The memory is ideal for storing critical code or data structures or for use as the system stack. Because the SRAM module is physically connected to the processor's high-speed local bus, it can service processor-initiated access or memory-referencing commands from the debug module.

Depending on configuration information, instruction fetches may be sent to both the cache and the SRAM block simultaneously. If the reference is mapped into the region defined by the SRAM, the SRAM provides the data back to the processor, and the cache data discarded. Accesses from the SRAM module are not cached.

The SRAM is dual-ported to provide DMA access. The SRAM is partitioned into two physical memory arrays to allow simultaneous access to both arrays by the processor core and another bus master. See Chapter 8, “System Control Module (SCM)” for more information.

5.3 SRAM Programming Model

The SRAM programming model includes a description of the SRAM base address register (RAMBAR), SRAM initialization, and power management.

5.3.1 SRAM Base Address Register (RAMBAR)

The configuration information in the SRAM base address register (RAMBAR) controls the operation of the SRAM module.

- The RAMBAR holds the base address of the SRAM. The MOVEC instruction provides write-only access to this register.
- The RAMBAR can be read or written from the debug module in a similar manner.
- All undefined bits in the register are reserved. These bits are ignored during writes to the RAMBAR, and return zeroes when read from the debug module.
- The RAMBAR valid bit is cleared by reset, disabling the SRAM module. All other bits are unaffected.

NOTE

CFMCLKD must be written with an appropriate value before programming or erasing the Flash array. Refer to Section 6.4.3.1, “Setting the CFMCLKD Register.”

6.3.4.3 CFM Security Register (CFMSEC)

The CFMSEC controls the Flash security features.

NOTE

Enabling Flash security will disable BDM communications.

NOTE

When Flash security is enabled, the chip will boot in single-chip mode regardless of the external reset configuration.

	31	30	29 … 16
Field	KEYEN	SECSTAT	—
Reset	See Note		
R/W	R		

	15 … 0
Field	SEC
Reset	See Note
R/W	R
Address	IPSBAR + 0x1D_0008

Note: The SECSTAT bit reset value is determined by the security state of the Flash. All other bits in the register are loaded at reset from the Flash Security longword stored at the array base address + 0x0000_0414.

Figure 6-6. CFM Security Register (CFMSEC)

Table 6-6. CFMSEC Field Descriptions

Bits	Name	Description
31	KEYEN	Enable back door key to security 1 Back door to Flash is enabled. 0 Back door to Flash is disabled.
30	SECSTAT	Flash security status 1 Flash security is enabled 0 Flash security is disabled

Table 7-6. PLL/CLKOUT Stop Mode Operation

STPMD[1:0]	Operation During Stop Mode				
	System Clocks	CLKOUT	PLL	OSC	PMM
00	Disabled	Enabled	Enabled	Enabled	Enabled
01	Disabled	Disabled	Enabled	Enabled	Enabled
10	Disabled	Disabled	Disabled	Enabled	Enabled
11	Disabled	Disabled	Disabled	Disabled	Low-Power Option

NOTE

If LPCR[LPMD] is cleared, then the device stops executing code upon issue of a STOP instruction. However, no clocks are disabled.

7.3 Functional Description

The functions and characteristics of the low-power modes, and how each module is affected by, or affects, these modes are discussed in this section.

7.3.1 Low-Power Modes

The system enters a low-power mode by executing a STOP instruction. Which mode the device actually enters (either stop, wait, or doze) depends on what is programmed in LPCR[LPMD]. Entry into any of these modes idles the CPU with no cycles active, powers down the system and stops all internal clocks appropriately. During stop mode, the system clock is stopped low.

For entry into stop mode, the LPICR[ENBSTOP] bit must be set before a STOP instruction is issued.

A wakeup event is required to exit a low-power mode and return to run mode. Wakeup events consist of any of these conditions:

- Any type of reset
- Any valid, enabled interrupt request

The latter method of exiting from low-power mode, by a valid and enabled interrupt request, requires several things:

- An interrupt request whose priority is higher than the value programmed in the XLPM_IPL field of the LPICR
- An interrupt request whose priority higher than the value programmed in the interrupt priority mask (I) field of the core's status register
- An interrupt request from a source which is not masked in the interrupt controller's interrupt mask register
- An interrupt request which has been enabled at the module of the interrupt's origin

7.3.1.1 Run Mode

Run mode is the normal system operating mode. Current consumption in this mode is related directly to the system clock frequency.

7.3.2.17 Clock Module

In wait and doze modes, the clocks to the CPU, Flash, and SRAM will be stopped and the system clocks to the peripherals are enabled. Each module may disable the module clocks locally at the module level. In stop mode, all clocks to the system will be stopped.

During stop mode, there are several options for enabling/disabling the PLL and/or crystal oscillator (OSC); each of these options requires a compromise between wakeup recovery time and stop mode power. The PLL may be disabled during stop mode. A wakeup time of up to 200 μs is required for the PLL to re-lock. The OSC may also be disabled during stop mode. The time required for the OSC to restart is dependent upon the startup time of the crystal used. Power consumption can be reduced in stop mode by disabling either or both of these functions via the SYNCR[STMPD] bits.

The external CLKOUT signal may be enabled during low-power stop (if the PLL is still enabled) to support systems using this signal as the clock source.

The system clocks may be enabled during wakeup from stop mode without waiting for the PLL to lock. This eliminates the wakeup recovery time, but at the risk of sending a potentially unstable clock to the system. It is recommended, if this option is used, that the PLL frequency divider is set so that the targeted system frequency is no more than half the maximum allowed. This will allow for any frequency overshoot of the PLL while still keeping the system clock within specification.

In external clock mode, there are no wait times for the OSC startup or PLL lock.

During wakeup from stop mode, the Flash clock will always clock through 16 cycles before the system clocks are enabled. This allows the Flash module time to recover from the low-power mode. Thus, software may immediately continue to fetch instructions from the Flash memory.

The external CLKOUT output pin may be disabled in the low state to lower power consumption via the DISCLK bit in the SYNCR. The external CLKOUT pin function is enabled by default at reset.

7.3.2.18 Edge Port

In wait and doze modes, the edge port continues to operate normally and may be configured to generate interrupts (either an edge transition or low level on an external pin) to exit the low-power modes.

In stop mode, there is no system clock available to perform the edge detect function. Thus, only the level detect logic is active (if configured) to allow any low level on the external interrupt pin to generate an interrupt (if enabled) to exit the stop mode.

7.3.2.19 Watchdog Timer

In stop mode (or in wait/doze mode, if so programmed), the watchdog ceases operation and freezes at the current value. When exiting these modes, the watchdog resumes operation from the stopped value. It is the responsibility of software to avoid erroneous operation.

When not stopped, the watchdog may generate a reset to exit the low-power modes.

7.3.2.20 Programmable Interrupt Timers (PIT0, PIT1, PIT2 and PIT3)

In stop mode (or in doze mode, if so programmed), the programmable interrupt timer (PIT) ceases operation, and freezes at the current value. When exiting these modes, the PIT resumes operation from the stopped value. It is the responsibility of software to avoid erroneous operation.

When not stopped, the PIT may generate an interrupt to exit the low-power modes.

The write cycle timing diagram is shown in Figure 13-8.

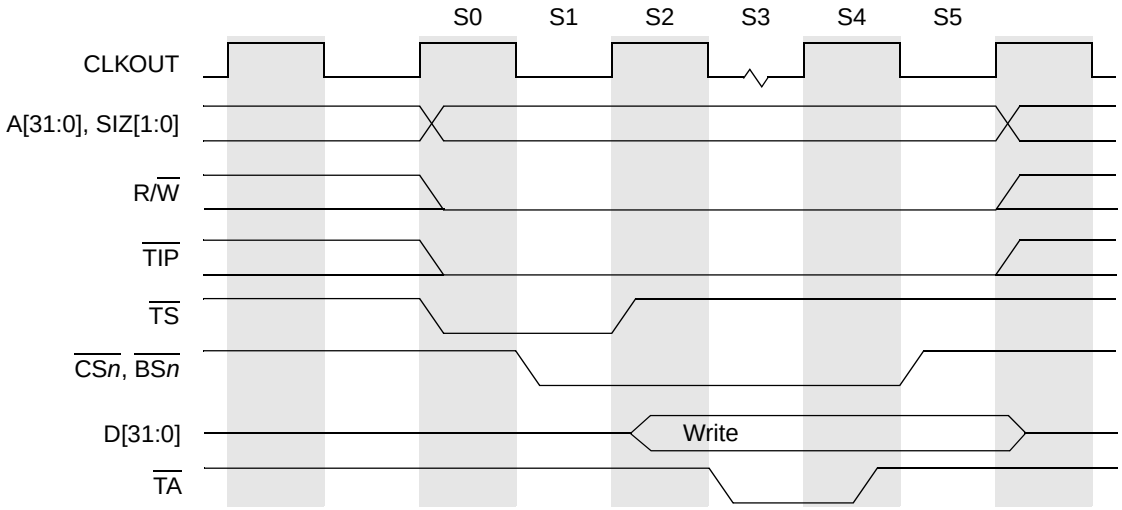

Figure 13-8. Basic Write Bus Cycle

Table 13-3 describes the six states of a basic write cycle.

13.4.5 Fast Termination Cycles

Two clock cycle transfers are supported on the external bus. In most cases, this is impractical to use in a system because the termination must take place in the same half-clock during which TS is asserted. As this is atypical, it is not referred to as the zero-wait-state case but is called the fast-termination case. Fast termination cycles occur when the external device or memory asserts TA less than one clock after TS is asserted. This means that the processor samples TA on the rising edge of the second cycle of the bus transfer. Figure 13-9 shows a read cycle with fast termination. Note that fast termination cannot be used with internal termination.

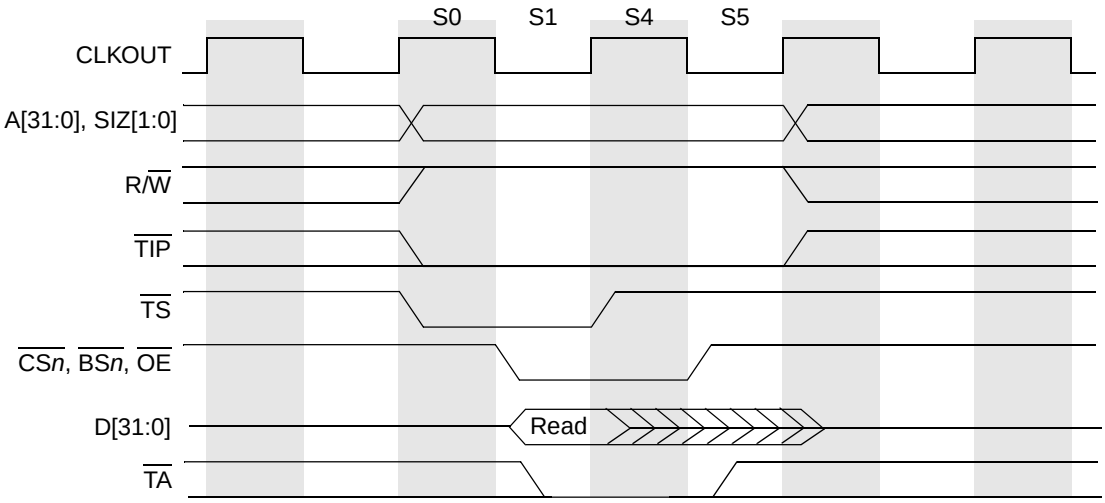

Figure 13-9. Read Cycle with Fast Termination

Figure 13-10 shows a write cycle with fast termination.

Table 14-3. MCF5282 Signals and Pin Numbers Sorted by Function (continued)

MAPBGA Pin	Pin Functions			Description	Primary I/O	Internal Pull-up[1]
	Primary[2]	Secondary	Tertiary			
External Memory Interface and Ports						
C6:B6:A5	A[23:21]	PF[7:5]	$\overline{CS}$[6:4]	Address bus	O	Yes
C4:B4:A4:B3:A3	A[20:16]	PF[4:0]	—	Address bus	O	Yes
A2:B1:B2:C1: C2:C3:D1:D2	A[15:8]	PG[7:0]	—	Address bus	O	Yes
D3:D4:E1:E2: E3:E4:F1:F2	A[7:0]	PH[7:0]	—	Address bus	O	Yes
F3:G1:G2:G3: G4:H1:H2:H3	D[31:24]	PA[7:0]	—	Data bus	I/O	—
H4:J1:J2:J3: J4:K1:K2:K3	D[23:16]	PB[7:0]	—	Data bus	I/O	—
L1:L2:L3:L4: M1:M2:M3:M4	D[15:8]	PC[7:0]	—	Data bus	I/O	—
N1:N2:N3:P1: N5:T6:R6:P6	D[7:0]	PD[7:0]	—	Data bus	I/O	—
P14:T15:R15:R16	$\overline{BS}$[3:0]	PJ[7:4]	—	Byte strobe	I/O	Yes
N16	$\overline{OE}$	PE7	—	Output enable	I/O	—
P16	$\overline{TA}$	PE6	—	Transfer acknowledge	I/O	Yes
P15	$\overline{TEA}$	PE5	—	Transfer error acknowledge	I/O	Yes
N15	R/$\overline{W}$	PE4	—	Read/write	I/O	Yes
N14	SIZ1	PE3	SYNCA	Transfer size	I/O	Yes[3]
M16	SIZ0	PE2	SYNCB	Transfer size	I/O	Yes[4]
M15	$\overline{TS}$	PE1	SYNCA	Transfer start	I/O	Yes
M14	$\overline{TIP}$	PE0	SYNCB	Transfer in progress	I/O	Yes
Chip Selects						
L16:L15:L14:L13	$\overline{CS}$[3:0]	PJ[3:0]	—	Chip selects 3-0	I/O	Yes
C6:B6:A5	A[23:21]	PF[7:5]	$\overline{CS}$[6:4]	Chip selects 6-4	O	Yes
SDRAM Controller						
H15	$\overline{SRAS}$	PSD5	—	SDRAM row address strobe	I/O	—
H16	$\overline{SCAS}$	PSD4	—	SDRAM column address strobe	I/O	—
G15	$\overline{DRAMW}$	PSD3	—	SDRAM write enable	I/O	—
H13:G16	$\overline{SDRAM\_CS}$[1:0]	PSD[2:1]	—	SDRAM chip selects	I/O	—
H14	SCKE	PSD0	—	SDRAM clock enable	I/O	—

14.2.1.10 Transfer In Progress (TIP)

The TIP output is asserted indicating a bus transfer is in progress. It is negated during idle bus cycles. Note that TIP is held asserted on back-to-back cycles.

NOTE

TIP is not asserted during SDRAM accesses.

This pin can also be configured as GPIO PE0 or SYNCB.

14.2.1.11 Chip Selects (CS[6:0])

Each chip select can be programmed for a base address location and for masking addresses, port size and burst-capability indication, wait-state generation, and internal/external termination.

Reset clears all chip select programming; CS0 is the only chip select initialized out of reset. CS0 is also unique because it can function at reset as a global chip select that allows boot ROM to be selected at any defined address space. The port size for boot CS0 is set during chip configuration by the levels on D[19:18] on the rising edge of RSTI, as described in Chapter 27, “Chip Configuration Module (CCM).” The chip-select implementation is described in Chapter 12, “Chip Select Module.”

These pins can also be configured as A[23:21] and GPIO PJ[3:0].

14.2.2 SDRAM Controller Signals

These signals are used for SDRAM accesses.

14.2.2.1 SDRAM Row Address Strobe (SRAS)

This output is the SDRAM synchronous row address strobe.

This pin is configured as GPIO PSD5 in single-chip mode.

14.2.2.2 SDRAM Column Address Strobe (SCAS)

This output is the SDRAM synchronous column address strobe.

This pin is configured as GPIO PSD4 in single-chip mode.

14.2.2.3 SDRAM Write Enable (DRAMW)

The DRAM write signal (DRAMW) is asserted to signify that a DRAM write cycle is underway. A read cycle is indicated by the negation of DRAMW.

This pin is configured as GPIO PSD3 in single-chip mode.

14.2.2.4 SDRAM Bank Selects (SDRAM_CS[1:0])

These signals interface to the chip-select lines of the SDRAMs within a memory block. Thus, there is one SDRAM_CS line for each memory block (the processor supports two SDRAM memory blocks).

These pins is configured as GPIO PSD[2:1] in single-chip mode.

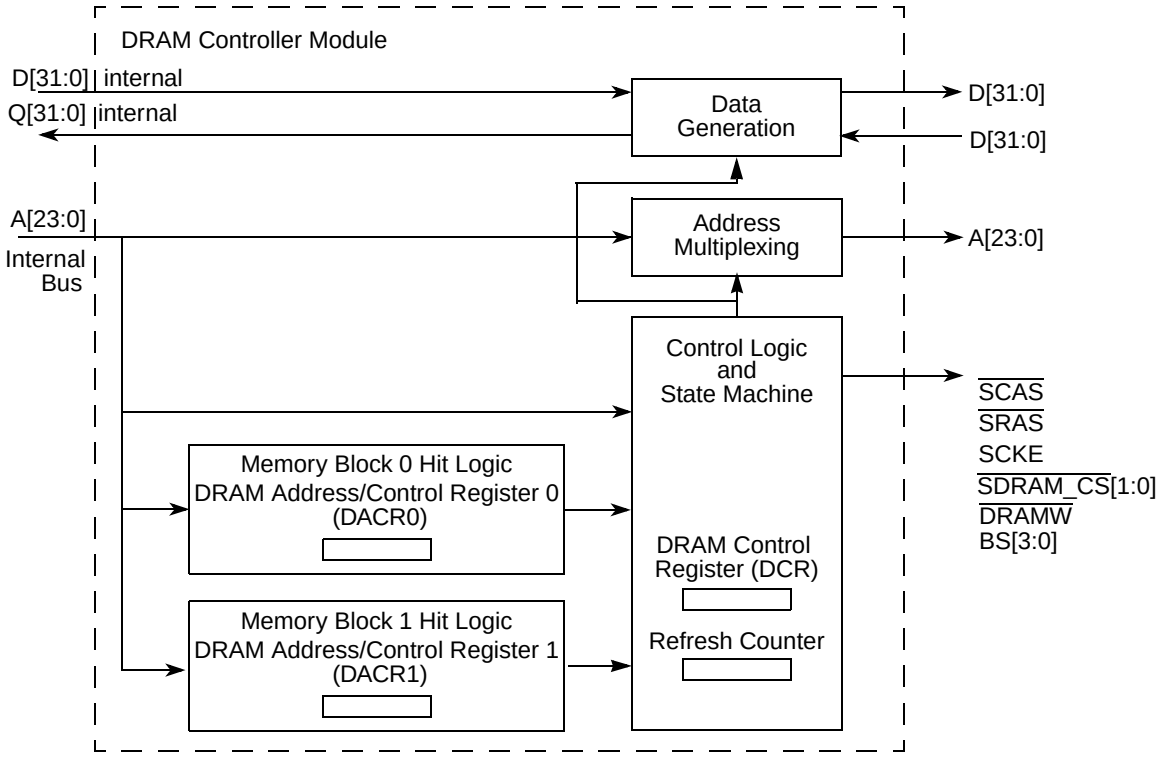

Figure 15-1. Synchronous DRAM Controller Block Diagram

The DRAM controller's major components are 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 IPSBAR + 0x048; DACR1 is accessed at IPSBAR + 0x050. The register information is passed on to the hit logic.
- Control logic and state machine—Generates all SDRAM signals, taking hit information and bus-cycle characteristic data from the block logic in order to generate SDRAM 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; controlled 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 SDRAM bus cycle to both DACRs to determine if an SDRAM block is being accessed. Hits are passed to the control logic along with characteristics of the bus cycle to be generated.
- Address multiplexing—Multiplexes addresses to allow column and row addresses to share pins. This allows glueless interface to SDRAMs.
- Data Generation—Controls the data input and data output transmission between the on-platform and off-platform data buses.

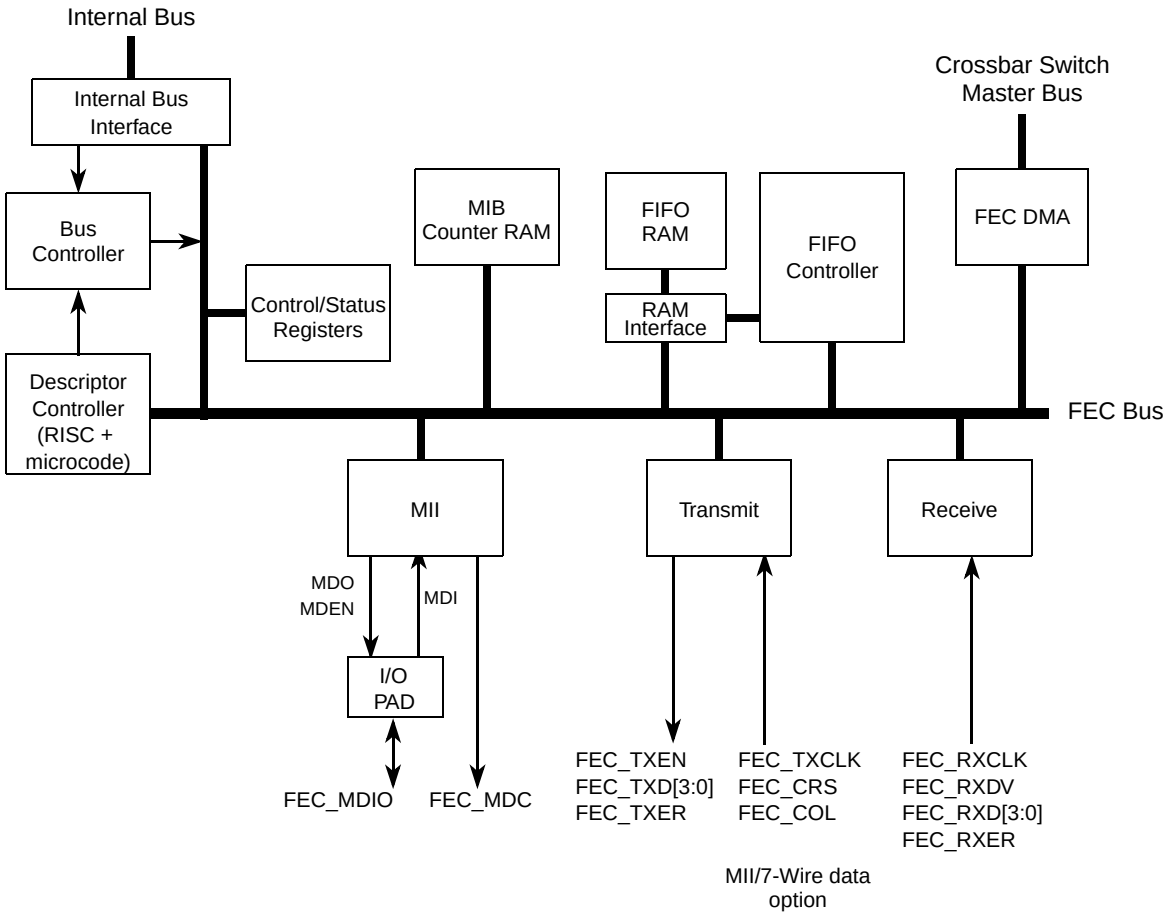

Figure 17-1. FEC Block Diagram

The descriptor controller is a RISC-based controller providing these functions in the FEC:

- Initialization (those internal registers not initialized by you or hardware)
- High level control of the DMA channels (initiating DMA transfers)
- Interpreting buffer descriptors
- Address recognition for receive frames
- Random number generation for transmit collision backoff timer

NOTE

DMA references in this section refer to the FEC's DMA engine. This DMA engine transfers FEC data only and is not related to the eDMA controller described in Chapter 16, "DMA Controller Module," nor to the DMA timers described in Chapter 21, "DMA Timers (DTIM0–DTIM3)."

The RAM is the focal point of all data flow in the Fast Ethernet controller and divides into transmit and receive FIFOs. The FIFO boundaries are programmable using the FRSR register. User data flows to/from the DMA block from/to the receive/transmit FIFOs. Transmit data flows from the transmit FIFO into the transmit block, and receive data flows from the receive block into the receive FIFO.

framing error, overrun error, and received break conditions set the respective PE, FE, OE, and RB error and break flags in the USR*n* at the received character boundary. They are valid only if USR*n*[RXRDY] is set.

If a break condition is detected (URXD*n* is low for the entire character including the stop bit), a character of all 0s loads into the receiver holding register and USR*n*[RB,RXRDY] are set. URXD*n* must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The receiver detects the beginning of a break in the middle of a character if the break persists through the next character time. The receiver places the damaged character in the Rx FIFO and sets the corresponding USR*n* error bits and USR*n*[RXRDY]. Then, if the break lasts until the next character time, the receiver places an all-zero character into the Rx FIFO and sets USR*n*[RB,RXRDY].

Figure 23-20 shows receiver functional timing.

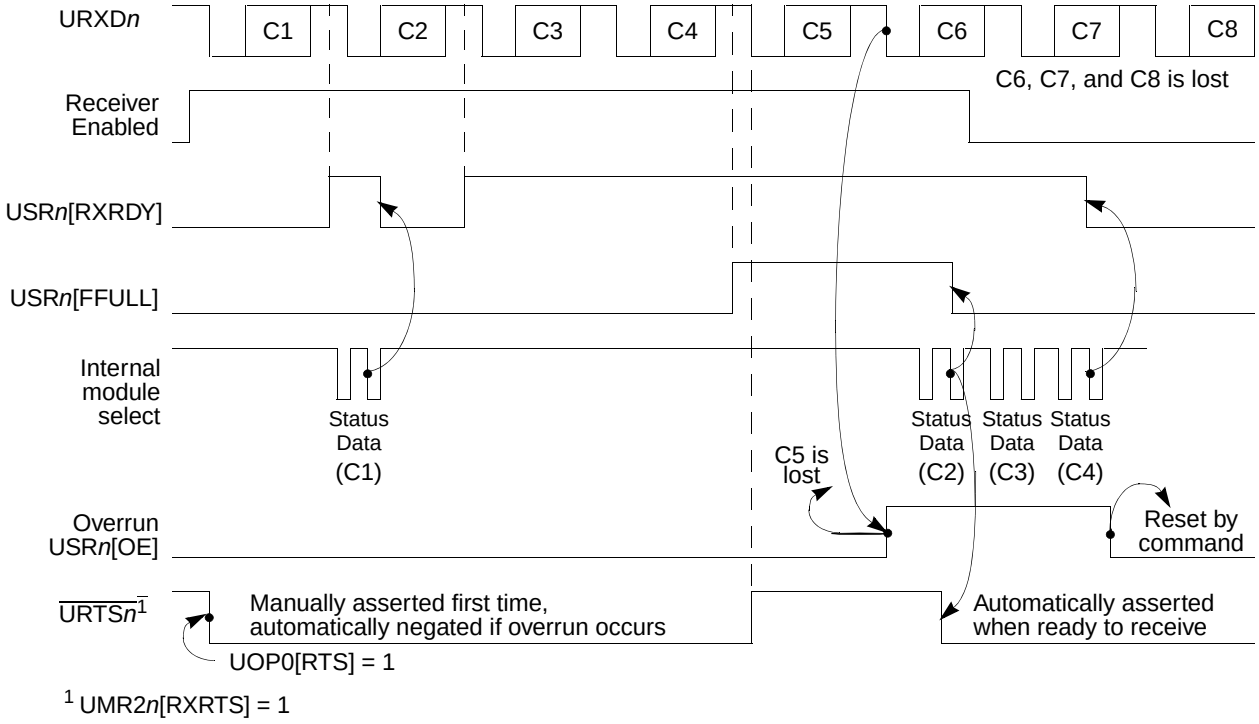

Figure 23-20. Receiver Timing Diagram

23.4.2.3 FIFO

The FIFO is used in the UART's receive buffer logic. The FIFO consists of three receiver holding registers. The receive buffer consists of the FIFO and a receiver shift register connected to the URXD*n* (see Figure 23-18). Data is assembled in the receiver shift register and loaded into the top empty receiver holding register position of the FIFO. Therefore, data flowing from the receiver to the CPU is quadruple-buffered.

In addition to the data byte, three status bits—parity error (PE), framing error (FE), and received break (RB)—are appended to each data character in the FIFO; overrun error (OE) is not appended. By

Chapter 28
Queued Analog-to-Digital Converter (QADC)

The queued analog-to-digital converter (QADC) is a 10-bit, unipolar, successive approximation converter. Up to eight analog input channels can be supported using internal multiplexing. A maximum of 18 input channels can be supported in the expanded, externally multiplexed mode.

The QADC consists of an analog front-end and a digital control subsystem. The analog section includes input pins, an analog multiplexer, and sample and hold analog circuits.

The digital control section contains queue control logic to sequence the conversion process and interrupt generation logic. Also included are the periodic/interval timer, control and status registers, the conversion command word (CCW) table, random-access memory (RAM), and the result table RAM.

The bus interface unit (BIU) provides access to registers that configure the QADC, control the analog-to-digital converter and queue mechanism, and present formatted conversion results.

28.1 Features

Features of the QADC module include:

- Internal sample and hold
- Up to eight analog input channels using internal multiplexing
- Up to four external analog multiplexers directly supported
- Up to 18 total input channels with internal and external multiplexing
- Programmable input sample time for various source impedances
- Two conversion command word (CCW) queues with a total of 64 entries for setting conversion parameters of each A/D conversion
- Subqueues possible using pause mechanism
- Queue complete and pause interrupts available on both queues
- Queue pointers indicating current location for each queue
- Automated queue modes initiated by:
  - External edge trigger and gated trigger
  - Periodic/interval timer, within QADC module (queues 1 and 2)
  - Software command
- Single scan or continuous scan of queues
- 64 result registers
- Output data readable in three formats:
  - Right-justified unsigned
  - Left-justified signed
  - Left-justified unsigned
- Unused analog channels can be used as discrete input/output pins.

for queue 1 and a trigger event occurs for queue 1 with BQ2 set to 0. Queue 1 execution starts momentarily, but is terminated after CCW0 is read. No conversions occur.

The BQ2[6:0] pointer may be changed dynamically to alternate between queue 2 scan sequences. A change in BQ2 after queue 2 has begun or when queue 2 has a trigger pending does not affect queue 2 until it is started again. For example, two scan sequences could be defined as follows: The first sequence starts at CCW10, with a pause after CCW11 and an end of queue (EOQ) programmed in CCW15; the second sequence starts at CCW16, with a pause after CCW17 and an EOQ programmed in CCW39.

With BQ2[6:0] set to CCW10 and the continuous-scan mode selected, queue execution begins. When the pause is encountered in CCW11, an interrupt service routine can retarget BQ2[6:0] to CCW16. When the end-of-queue is recognized in CCW15, an internal retrigger event is generated and execution restarts at CCW16. When the pause software interrupt occurs again, BQ2 can be changed back to CCW10. After the end-of-queue is recognized in CCW39, an internal retrigger event is created and execution now restarts at CCW10.

If BQ2[6:0] is changed while queue 1 is active, the effect of BQ2[6:0] as an end-of-queue indication for queue 1 is immediate. However, beware of the risk of losing the end-of-queue 1 when changing BQ2[6:0]. Using EOQ (channel 63) to end queue 1 is recommended.

NOTE

If BQ2[6:0] was assigned to the CCW that queue 1 is currently working on, then that conversion is completed before the change to BQ2[6:0] takes effect.

Each time a CCW is read for queue 1, the CCW location is compared with the current value of the BQ2[6:0] pointer to detect a possible end-of-queue condition. For example, if BQ2[6:0] is changed to CCW3 while queue 1 is converting CCW2, queue 1 is terminated after the conversion is completed. However, if BQ2[6:0] is changed to CCW1 while queue 1 is converting CCW2, the QADC would not recognize a BQ2[6:0] end-of-queue condition until queue 1 execution reached CCW1 again, presumably on the next pass through the queue.

Stop mode resets this register (0x007f)

	15	14	13	12	11	10	9	8
Field	CIE2	PIE2	SSE2	MQ212	MQ211	MQ210	MQ29	MQ28
Reset	0000_0000							
R/W:	R/W							

	7	6	5	4	3	2	1	0
Field	RESUME	BQ26	BQ25	BQ24	BQ23	BQ22	BQ21	BQ20
Reset	0111_1111							
R/W:	R/W							
Address	IPSBAR + 0x19_000e, 0x19_000f							

Figure 28-10. QADC Control Register 2 (QACR2)

NOTE

The BDM status bit (S) is 0 for normally completed commands; S = 1 for illegal commands, not-ready responses, and transfers with bus-errors. Section 30.5.2, “BDM Serial Interface,” describes the receive packet format.

Freescale reserves unassigned command opcodes for future expansion. Unused command formats in any revision level perform a NOP and return an illegal command response.

30.5.3.3.1 Read A/D Register (RAREG/RDREG)

Read the selected address or data register and return the 32-bit result. A bus error response is returned if the CPU core is not halted.

Command/Result Formats:

	15–12	11–8	7–4	3	2–0
Command	0x2	0x1	0x8	A/D	Register
Result	D[31:16]				
	D[15:0]				

Figure 30-17. RAREG/RDREG Command Format

Command Sequence:

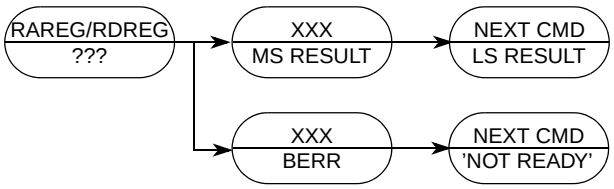

Figure 30-18. RAREG/RDREG Command Sequence

Operand Data: None

Result Data: The contents of the selected register are returned as a longword value, most-significant word first.

30.5.3.3.2 Write A/D Register (WAREG/WDREG)

The operand longword data is written to the specified address or data register. A write alters all 32 register bits. A bus error response is returned if the CPU core is not halted.

Command Format:

15–12	11–8	7–4	3	2–0
0x2	0x0	0x8	A/D	Register
D[31:16]				
D[15:0]				

Figure 30-19. WAREG/WDREG Command Format

Command Sequence

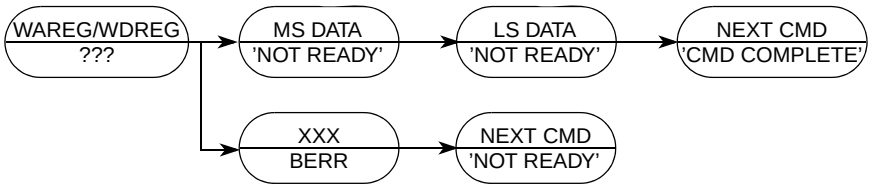

Figure 30-20. WAREG/WDREG Command Sequence

Operand Data	Longword data is written into the specified address or data register. The data is supplied most-significant word first.
Result Data	Command complete status is indicated by returning 0xFFFF (with S cleared) when the register write is complete.

30.5.3.3.3 Read Memory Location (READ)

Read data at the longword address. Address space is defined by BAAR[TT,TM]. Hardware forces low-order address bits to zeros for word and longword accesses to ensure that word addresses are word-aligned and longword addresses are longword-aligned.

Table A-3. Register Memory Map (continued)

Address	Name	Mnemonic	Size
Edge Port Registers			
IPSBAR + 0x13_0000	EPORT Pin Assignment Register	EPPAR	16
IPSBAR + 0x13_0002	EPORT Data Direction Register	EPDDR	8
IPSBAR + 0x13_0003	EPORT Interrupt Enable Register	EPIER	8
0x0013_0004	EPORT Data Register	EPDR	8
IPSBAR + 0x13_0005	EPORT Pin Data Register	EPPDR	8
0x0013_0006	EPORT Flag Register	EPFR	8
Watchdog Timer Registers			
IPSBAR + 0x14_0000	Watchdog Control Register	WCR	16
IPSBAR + 0x14_0002	Watchdog Modulus Register	WMR	16
IPSBAR + 0x14_0004	Watchdog Count Register	WCNTR	16
IPSBAR + 0x14_0006	Watchdog Service Register	WSR	16
Programmable Interrupt Timer 0 Registers			
IPSBAR + 0x15_0000	PIT Control and Status Register 0	PCSR 0	16
IPSBAR + 0x15_0002	PIT Modulus Register 0	PMR 0	16
IPSBAR + 0x15_0004	PIT Count Register 0	PCNTR 0	16
Programmable Interrupt Timer 1 Registers			
IPSBAR + 0x16_0000	PIT Control and Status Register 1	PCSR 1	16
IPSBAR + 0x16_0002	PIT Modulus Register 1	PMR 1	16
IPSBAR + 0x16_0004	PIT Count Register 1	PCNTR 1	16
Programmable Interrupt Timer 2 Registers			
IPSBAR + 0x17_0000	PIT Control and Status Register 2	PCSR 2	16