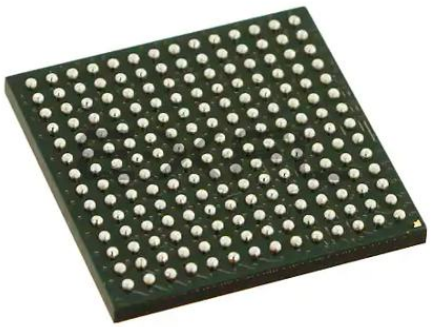

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	Coldfire V2
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	EBI/EMI, Ethernet, I²C, SPI, UART/USART, USB
Peripherals	DMA, WDT
Number of I/O	32
Program Memory Size	16KB (4K x 32)
Program Memory Type	ROM
EEPROM Size	-
RAM Size	1K x 32
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	196-LBGA
Supplier Device Package	196-LBGA (15x15)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcf5272cvm66

Table xi. UART1 Module Memory Map (continued)

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x016C	UART1 RxFIFO Control/Status Register	U2RxFCSR	**U1RxFCSR**
0x0174	UART1 CTS Unlatched Input	U2IP	**U1IP**
0x0178	UART1 RTS O/P Bit Set Command Register	U2OP1	**U1OP1**
0x017C	UART1 RTS O/P Bit Reset Command Register	U2OP0	**U1OP0**

Table xii. SDRAM Controller Memory Map

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x0182	SDRAM Configuration Register	SDCCR	**SDCR**
0x0186	SDRAM Timing Register	SDCTR	**SDTR**

Table xiii. Timer Module Memory Map

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x0200	Timer 0 Mode Register	TMR1	**TMR0**
0x0204	Timer 0 Reference Register	TRR1	**TRR0**
0x0208	Timer 0 Capture Register	TCR1	**TCAP0**
0x020C	Timer 0 Counter Register	TCN1	**TCN0**
0x0210	Timer 0 Event Register	TER1	**TER0**
0x0220	Timer 1 Mode Register	TMR2	**TMR1**
0x0224	Timer 1 Reference Register	TRR2	**TRR1**
0x0228	Timer 1 Capture Register	TCR2	**TCAP1**
0x022C	Timer 1 Counter Register	TCN2	**TCN1**
0x0230	Timer 1 Event Register	TER2	**TER1**
0x0240	Timer 2 Mode Register	TMR3	**TMR2**
0x0244	Timer 2 Reference Register	TRR3	**TRR2**
0x0248	Timer 2 Capture Register	TCR3	**TCAP2**
0x024C	Timer 2 Counter Register	TCN3	**TCN2**
0x0250	Timer 2 Event Register	TER3	**TER2**
0x0260	Timer 3 Mode Register	TMR4	**TMR3**
0x0264	Timer 3 Reference Register	TRR4	**TRR3**
0x0268	Timer 3 Capture Register	TCR4	**TCAP3**

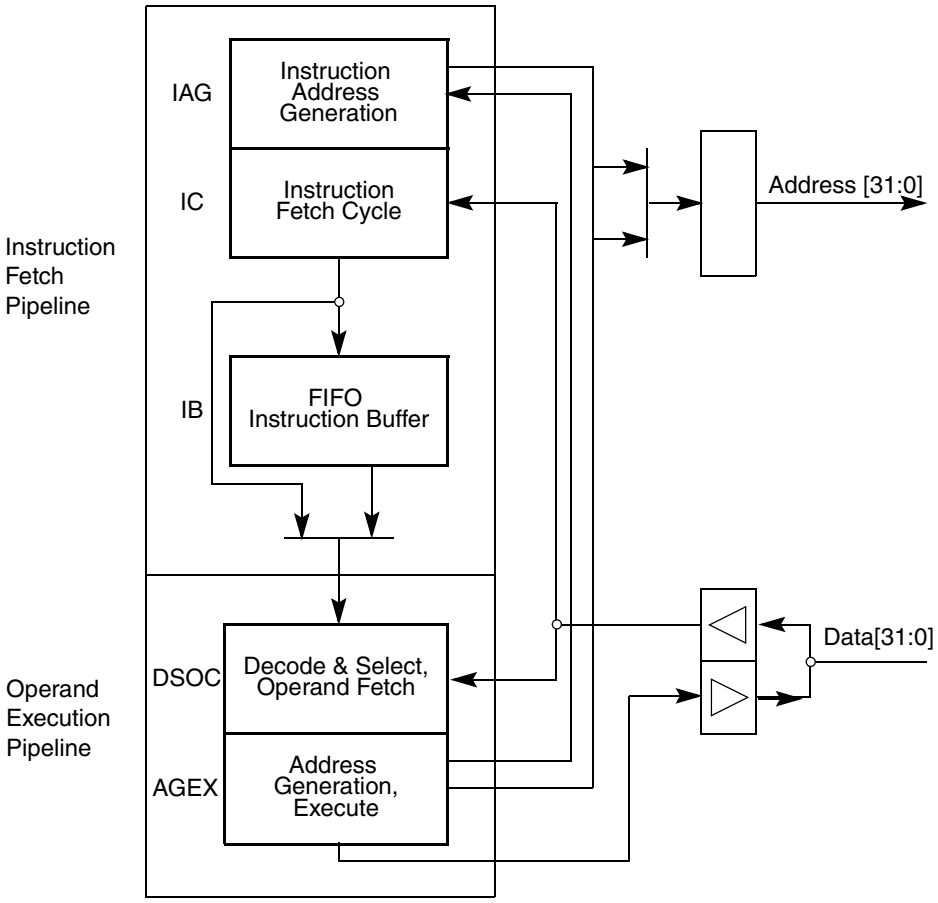

Figure 2-1. ColdFire Pipeline

2.1.1.1 Instruction Fetch Pipeline (IFP)

The IFP generates instruction addresses and fetches. Because the fetch and execution pipelines are decoupled by a three longword FIFO buffer, the IFP can prefetch instructions before the OEP needs them, minimizing stalls.

2.1.1.2 Operand Execution Pipeline (OEP)

The OEP is a two-stage pipeline featuring a traditional RISC datapath with a register file feeding an arithmetic/logic unit (ALU). For simple register-to-register instructions, the first stage of the OEP performs the instruction decode and fetching of the required register operands (OC), while the actual instruction execution is performed in the second stage (EX).

For memory-to-register instructions, the instruction is effectively staged through the OEP twice in the following way:

- The instruction is decoded and the components of the operand address are selected (DS).
- The operand address is generated using the execute engine (AG).
- The memory operand is fetched while any register operand is simultaneously fetched (OC).
- The instruction is executed (EX).

Table 2-14. Two-Operand Instruction Execution Times (continued)

Opcode	<ea>	Effective Address							
		Rn	(An)	(An)+	–(An)	(d16,An)	(d8,An,Xi*SF)	(xxx).wl	#<xxx>
sub.l	Dy,<ea>	—	3(1/1)	3(1/1)	3(1/1)	3(1/1)	4(1/1)	3(1/1)	—
subi.l	#imm,Dx	1(0/0)	—	—	—	—	—	—	—
subq.l	#imm,<ea>	1(0/0)	3(1/1)	3(1/1)	3(1/1)	3(1/1)	4(1/1)	3(1/1)	—
subx.l	Dy,Dx	1(0/0)	—	—	—	—	—	—	—

2.7.4 Miscellaneous Instruction Execution Times

Table 2-15 lists timings for miscellaneous instructions.

Table 2-15. Miscellaneous Instruction Execution Times

Opcode	<ea>	Effective Address							
		Rn	(An)	(An)+	–(An)	(d16,An)	(d8,An,Xi*SF)	(xxx).wl	#<xxx>
cpushl	(Ax)	—	11(0/1)	—	—	—	—	—	—
link.w	Ay,#imm	2(0/1)	—	—	—	—	—	—	—
move.w	CCR,Dx	1(0/0)	—	—	—	—	—	—	—
move.w	<ea>,CCR	1(0/0)	—	—	—	—	—	—	1(0/0)
move.w	SR,Dx	1(0/0)	—	—	—	—	—	—	—
move.w	<ea>,SR	7(0/0)	—	—	—	—	—	—	7(0/0)
movec	Ry,Rc	9(0/1)	—	—	—	—	—	—	—
movem.l [1]	<ea>,&list	—	1+n(n/0)	—	—	1+n(n/0)	—	—	—
movem.l	&list,<ea>	—	1+n(0/n)	—	—	1+n(0/n)	—	—	—
nop		3(0/0)	—	—	—	—	—	—	—
pea	<ea>	—	2(0/1)	—	—	2(0/1)[2]	3(0/1)[3]	2(0/1)	—
pulse		1(0/0)	—	—	—	—	—	—	—
stop	#imm	—	—	—	—	—	—	—	3(0/0)[4]
trap	#imm	—	—	—	—	—	—	—	15(1/2)
trapf		1(0/0)	—	—	—	—	—	—	—
trapf.w		1(0/0)	—	—	—	—	—	—	—
trapf.l		1(0/0)	—	—	—	—	—	—	—
unlk	Ax	2(1/0)	—	—	—	—	—	—	—
wddata.l	<ea>	—	3(1/0)	3(1/0)	3(1/0)	3(1/0)	4(1/0)	3(1/0)	—
wdebug.l	<ea>	—	5(2/0)	—	—	5(2/0)	—	—	—

1 *n* is the number of registers moved by the MOVEM opcode.

2 PEA execution times are the same for (d16,PC).

3 PEA execution times are the same for (d8,PC,Xi*SF).

4 The execution time for STOP is the time required until the processor begins sampling continuously for interrupts.

5.4.3 Address Breakpoint Registers (ABLR, ABHR)

The address breakpoint low and high registers (ABLR, ABHR), Figure 5-6, define regions in the processor's data address space that can be used as part of the trigger. These register values are compared with the address for each transfer on the processor's high-speed local bus. The trigger definition register (TDR) identifies the trigger as one of three cases:

1. identically the value in ABLR
2. inside the range bound by ABLR and ABHR inclusive
3. outside that same range

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

Figure 5-6. Address Breakpoint Registers (ABLR, ABHR)

Table 5-6 describes ABLR fields.

Table 5-6. ABLR Field Description

Bits	Name	Description
31–0	Address	Low address. Holds the 32-bit address marking the lower bound of the address breakpoint range. Breakpoints for specific addresses are programmed into ABLR.

Table 5-7 describes ABHR fields.

Table 5-7. ABHR Field Description

Bits	Name	Description
31–0	Address	High address. Holds the 32-bit address marking the upper bound of the address breakpoint range.

7.2.5 Programmable Interrupt Wakeup Register (PIWR)

The programmable interrupt wakeup register (PIWR), Figure 7-8, is used to specify which interrupt sources are capable of causing the CPU to wake up from low-power SLEEP or STOP modes when their source is active. All sources are disabled on reset. Note that only the external interrupt pins $\overline{\text{INT}}$[6:1] can wake up the CPU from STOP mode.

If more than one interrupt source has the same interrupt priority level (IPL) programmed in the ICRs, the interrupt controller daisy chains the interrupts with the priority order following the bit placement in the PIWR, with $\overline{\text{INT1}}$ having the highest priority and SWTO having the lowest priority as shown in Figure 7-8.

	31	30	29	28	27	26	25	24
Field	INT1	INT2	INT3	INT4	TMR0	TMR1	TMR2	TMR3
Reset	1111_1111							
R/W	R/W							

	23	22	21	20	19	18	17	16
Field	UART1	UART2	PLI_P	PLI_A	USB0	USB1	USB2	USB3
Reset	1111_1111							
R/W	R/W							

	15	14	13	12	11	10	9	8
Field	USB4	USB5	USB6	USB7	DMA	ERx	ETx	ENTC
Reset	1111_1111							
R/W	R/W							

	7	6	5	4	3	0
Field	QSPI	INT5	INT6	SWTO	—	
Reset	1111_0000					
R/W	R/W					
Address	MBAR+0x038					

Figure 7-8. Programmable Interrupt Wakeup Register (PIWR)

Table 7-6 describes PIWR fields.

Table 7-6. PIWR Field Descriptions

Bits	Field	Description
31–4	—	0 Interrupt cannot wake up the CPU when interrupt source is active. 1 Interrupt wakes up the CPU from low-power modes.
3–0	—	Reserved, should be cleared.

Table 9-7. SDCR Field Descriptions

Bits	Name	Description
15	—	Reserved, should be cleared.
14–13	MCAS	Maximum CAS address. Determines which device address output carries the column address msb. For example, if the SDRAM device has eight column addresses and the data bus is configured for 32 bits, the column address appears on A[9:2], so the maximum column address is A9. The lsb of the row address is therefore taken from internal address signal A10 and is used by the SDRAM controller to control address multiplexing. 00 A7 01 A8 10 A9 11 A10
12–11	—	Reserved, should be cleared.
10–8	BALOC	Bank address location. Determines the internal addresses that become SDRAM bank addresses. SDBA1SDBA0 000Reserved 001A21A20 010A22A21 011A23A22 100A24A23 101A25A24 110Reserved 111Reserved
7	GSL	Go to sleep. Setting GSL powers down the SDRAM and puts it into auto-refresh mode.
6–5	—	Reserved, should be cleared.
4	REG	Register read data for 66 MHz. Writing a 1 to REG enables pipeline mode for read data access. It forces the SDRAM controller to register the read data, adding one wait state to single-read accesses and to the first word read during a burst. REG must be 1 for clock frequencies above 48 MHz to meet input setup timing for data input (See electrical characteristics timing SD16). The description of INV shows how REG and INV interact.
3	INV	Invert clock. Inverts SDRAM clock output for timing refinement. If REG = 0 0 Do not add wait state for read accesses. 1 Shift SDCLK edge 180° If REG = 1 0 Add wait state for read accesses, all frequencies 1 Invalid, do not use.
2	SLEEP	SLEEP mode. This read-only status bit goes high when setting SDCR[GSL] has taken effect and the SDRAM is powered down. SLEEP is cleared when SDRAM is in auto-refresh mode.
1	ACT	Active. This read-only status bit goes high when the SDRAM controller completes its initialization. ACT is cleared by writing to SDCR.
0	INIT	Initialization enable. Setting INIT enables initialization of the SDRAM based on other SDCR bit values. Initialization starts after the first dummy write access to the SDRAM. CSOR7, CSBR7, and SDTR must be configured before setting INIT. CAUTION: CSOR7[WAITST] must equal 0x1F when $\overline{\text{CS7}}$/$\overline{\text{SDCS}}$ is configured for SDRAM.

11.6.1 FEC Buffer Descriptor Tables

The data for the fast Ethernet frames must reside in memory external to the FEC. The data for a frame is placed in one or more buffers. Each buffer has a pointer to it in a buffer descriptor (BD). In addition to pointing to the buffer, the BD contains the current state of the buffer. To permit maximum user flexibility, the BDs are also located in external memory.

Software defines buffers by allocating/initializing memory and initializing buffer descriptors. Setting the RxBD[E] or TxBD[R] produces the buffer. Software writing to either TDAR or RDAR tells the FEC that a buffer has been placed in external memory for the transmit or receive data traffic, respectively. The hardware reads the BDs and processes the buffers after they have been defined. After the data DMA is complete and the BDs have been written by the DMA engine, RxBD[E] or TxBD[R] are cleared by hardware to indicate that the buffer has been processed. Software may poll the BDs to detect when the buffers have been processed or may rely on the buffer/frame interrupts.

The ETHER_EN signal operates as a reset to the BD/DMA logic. When ETHER_EN is negated, the DMA engine BD pointers are reset to point to the starting transmit and receive BDs. The buffer descriptors are not initialized by hardware during reset. At least one transmit and receive BD must be initialized by software (write 0x0000_0000 to the most significant word of buffer descriptor) before the ETHER_EN bit is set.

The BDs are organized in two separate rings, one for receive buffers and one for transmit buffers. ERDSR defines the starting address of the receive BDs and ETDSR the same for the transmit BDs. The last buffer descriptor in each ring is defined by the wrap (W) bit. When set, W indicates that the next descriptor in the ring is at the location pointed to by ERDSR and ETDSR for the receive and transmit rings, respectively. Buffers descriptor rings must start on a double-word boundary.

The format of the transmit and receive buffer descriptors are given in Figure 11-27 and Figure 11-28.

11.6.1.1 Ethernet Receive Buffer Descriptor (RxBD)

In the RxBD, the user initializes the E and W bits in the first word and the pointer in the second word. When the buffer has been sent as a DMA, the FEC will modify the E, L, M, LG, NO, SH, CR, and OV bits and write the length of the used portion of the buffer in the first word. The M, LG, NO, SH, CR, and OV bits in the first word of the buffer descriptor are modified by the FEC only when the L bit is set.

	15	14	13	12	11	10 9	8	7	6	5	4	3	2	1	0
+0	E	RO1	W	RO2	L	—	M	BC	MC	LG	NO	SH	CR	OV	TR
+2	DATA LENGTH														
+4	Rx Data Buffer Pointer A[31–16]														
+6	Rx Data Buffer Pointer A[15–0]														

Figure 11-27. Receive Buffer Descriptor (RxBD)

The first word of the RxBD contains control and status bits. Its format is detailed below.

Table 12-1. USB Device Requests (continued)

Device Request	USB Request Processor Action
get_status	Returns the current status of the specified device, endpoint or interface. No user notification is provided, no user action required.
set_address	Loads the specified address into USBFAR. The control logic begins responding to the new address once the status stage of the request completes successfully. No user notification is provided unless debug mode is enabled.
set_configuration	Reads the configuration RAM. If the configuration is cleared, the USB module is placed into the unconfigured state. If a valid configuration is selected, the appropriate endpoint controllers are activated. An invalid configuration number or error in the configuration descriptor causes the USB module to return a STALL response to the host. The user is notified when this request completes successfully and must initialize the active endpoint controllers.
set_descriptor	Not supported. Returns request error.
set_feature	Sets the specified feature. Remote wakeup and endpoint halt are the only features defined in the USB Specification, Revision 1.0. If the specified feature is remote wakeup, the USB enables the remote wakeup functionality. If the specified feature is endpoint halt, the USB request processor halts the selected endpoint. A halted endpoint returns a STALL response to any requests.
set_interface	Allows the host to select an alternate setting for the specified interface. If a valid alternate setting is selected, the appropriate endpoint controllers are activated. The user is notified upon successful completion of this request and must reinitialize the affected endpoint controllers. NOTE: The user must read the descriptor structure to determine which endpoints correspond to a given interface.
sync_frame	Passed to the user as a vendor specific request.

14.5.7 Command RAM Registers (QCR0–QCR15)

The command RAM is accessed using the upper byte of QDR. The QSPI cannot modify information in command RAM.

There are 16 bytes in the command RAM. Each byte is divided into two fields. The chip select field enables external peripherals for transfer. The command field provides transfer operations.

NOTE

The command RAM is accessed only using the most significant byte of QDR and indirect addressing based on QAR[ADDR].

Figure 14-11 shows the command RAM register.

	15	14	13	12	11 … 8	7 … 0
Field	CONT	BITSE	DT	DSCK	QSPI_CS	–
Reset	Undefined					
R/W	Write Only					
Address	QAR[ADDR]					

Figure 14-11. Command RAM Registers (QCR0–QCR15)

Table 14-7 gives QCR field descriptions.

Table 14-7. QCR0–QCR15 Field Descriptions

Bits	Name	Description
15	CONT	Continuous. 0 Chip selects return to inactive level defined by QWR[CSIV] when transfer is complete. 1 Chip selects remain asserted between transfers for a transfer of up to 16 words of data.[1]
14	BITSE	Bits per transfer enable. 0 Eight bits 1 Number of bits set in QMR[BITS]
13	DT	Delay after transfer enable. 0 Default reset value. 1 The QSPI provides a variable delay at the end of serial transfer to facilitate interfacing with peripherals that have a latency requirement. The delay between transfers is determined by QDLYR[DTL].
12	DSCK	Chip select to QSPI_CLK delay enable. 0 Chip select valid to QSPI_CLK transition is one-half QSPI_CLK period. 1 QDLYR[QCD] specifies the delay from QSPI_CS valid to QSPI_CLK.
11–8	QSPI_CS	Peripheral chip selects. Used to select an external device for serial data transfer. More than one chip select may be active at once, and more than one device can be connected to each chip select. Bits 11–8 map directly to QSPI_CS[3:0], respectively. If it is desired to use those bits as a chip select value, then an external demultiplexor must be connected to the QSPI_CS[3:0] pins.
7–0	—	Reserved, should be cleared.

[1] In order to keep the chip setects asserted for all transfers, the QWR[CSIV] bit must be set to control the level that the chip selects return to after the first transfer.

16.3.5 UART Command Registers (UCR*n*)

The UART command registers (UCR*n*), Figure 16-6, supply commands to the UART. Only multiple commands that do not conflict can be specified in a single write to a UCR*n*. For example, RESET TRANSMITTER and ENABLE TRANSMITTER cannot be specified in one write.

	7	6	4	3	2	1	0
Field	ENAB	MISC		TC		RC	
Reset	0000_0000						
R/W	Write only						
Address	MBAR + 0x108, 0x148						

Figure 16-6. UART Command Registers (UCR*n*)

Table 16-6 describes UCR*n* fields and commands. Examples in Section 16.5.2, “Transmitter and Receiver Operating Modes,” show how these commands are used.

Table 16-6. UCR*n* Field Descriptions

Bits	Value	Command	Description
7	ENAB	—	Enable autobaud 0 Autobaud disabled. 1 Autobaud enabled. The transmission rate is calculated from the first received character. If the rate must be recalculated, ENAB must first be cleared and reset. UISR*n*[ABC] indicates a transmission rate has been calculated and loaded into the UART divider registers. UDU*n* and UDUL*n* must be initialized to 0x00 before enabling autobaud.
6–4	**MISC Field** (This field selects a single command.)		
	000	no command	—
	001	reset mode register pointer	Causes the mode register pointer to point to UMR1*n*.
	010	reset receiver	Immediately disables the receiver, clears USR*n*[FFULL,RxRDY], and reinitializes the receiver FIFO pointer. No other registers are altered. Because it places the receiver in a known state, use this command instead of RECEIVER DISABLE when reconfiguring the receiver.
	011	reset transmitter	Disables the transmitter and clears USR*n*[TxEMP,TxRDY]. No other registers are altered. Because it places the transmitter in a known state, use this command instead of TRANSMITTER DISABLE when reconfiguring the transmitter.
	100	reset error status	Clears USR*n*[RB,FE,PE,OE]. Also used in block mode to clear all error bits after a data block is received.
	101	reset break– change interrupt	Clears the delta break bit, UISR*n*[DB].
	110	start break	Forces TxD low. If the transmitter is empty, the break may be delayed up to one bit time. If the transmitter is active, the break starts when character transmission completes. The break is delayed until any character in the transmitter shift register is sent. Any character in the transmitter holding register is sent after the break. The transmitter must be enabled for the command to be accepted. This command ignores the state of $\overline{\text{CTS}}$.
	111	stop break	Causes TxD to go high (mark) within two bit times. Any characters in the transmitter buffer are sent.

16.3.14 UART Receiver FIFO Registers (URF*n*)

The URF*n* registers contain control and status bits for the receiver FIFO. Note that some bits are read only.

	7	6	5	4	0
Field	RXS		FULL	RXB	
Reset	0000_0000				
R/W	R/W		R	R	
Address	MBAR + 0x12C (URF0), 0x16C (URF1)				

Figure 16-17. UART Receiver FIFO Registers (URF*n*)

Table 16-11 describes URF*n* fields.

Table 16-11. URF*n* Field Descriptions

Bits	Name	Description
7–6	RXS	$\overline{\text{Receiver status}}$. When written to, these bits control the meaning of UISR*n*[RxFIFO]. 00 Inhibit receiver FIFO status indication in UISR*n*. 01 Receiver FIFO Š 25% full 10 Receiver FIFO Š 50% full 11 Receiver FIFO Š 75% full When read, these bits indicate the emptiness level of the FIFO. 00 Receiver FIFO < 25% full 01 Receiver FIFO Š 25% full 10 Receiver FIFO Š 50% full 11 Receiver FIFO Š 75% full
5	FULL	Receiver $\overline{\text{FIFO full}}$. 0 Receiver FIFO is not full and can be loaded with a character. 1 Receiver FIFO is full. Characters loaded from the receiver when the FIFO is full are lost. This bit is identical to USR*n*[FFULL].
4–0	RXB	Receiver buffer data level. Indicates the number of bytes, between 0 and 24, stored in the receiver FIFO.

16.5.1.1 Programmable Divider

As Figure 16-23 shows, the UART transmitter and receiver can use the following clock sources:

- An external clock signal on the URT_CLK pin that can be divided by 16. When not divided, URT_CLK provides a synchronous clock mode; when divided by 16, it is asynchronous.
- CLKIN supplies an asynchronous clock source that is prescaled by 32 and then divided by the 16-bit value programmed in UDU*n* and UDL*n*. See Section 16.3.11, “UART Divider Upper/Lower Registers (UDUn/UDLn).” The precision of this clock source can be tuned using the 4-bit value programmed in UFPD*n*.

Note that when autobaud mode is enabled, the UFPD*n*, UDU*n*, and UDL*n* are automatically loaded with the calculated baud rate. However, the calculated value can be overridden by a programmed value at any time.

The choice of URT_CLK or CLKIN is programmed in the UCSR.

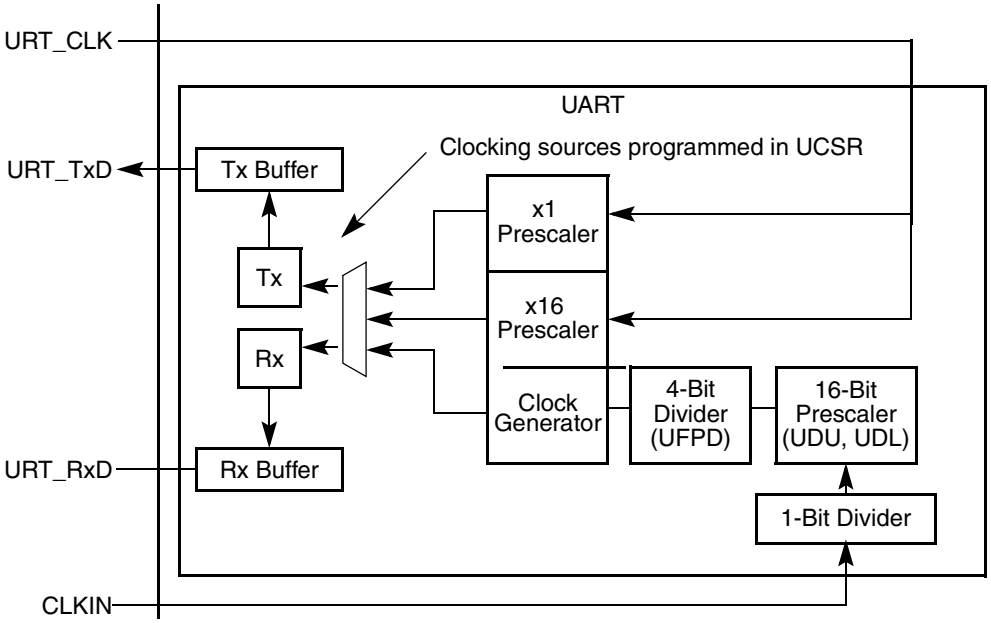

Figure 16-23. Clocking Source Diagram

16.5.1.2 Calculating Baud Rates

The following sections describe how to calculate baud rates.

16.5.1.2.1 CLKIN Baud Rates

When CLKIN is the UART clocking source, it goes through a divide-by-32 prescaler and then passes through the 16-bit divider of the concatenated UDU*n* and UDL*n* registers. The UFPD register can be used

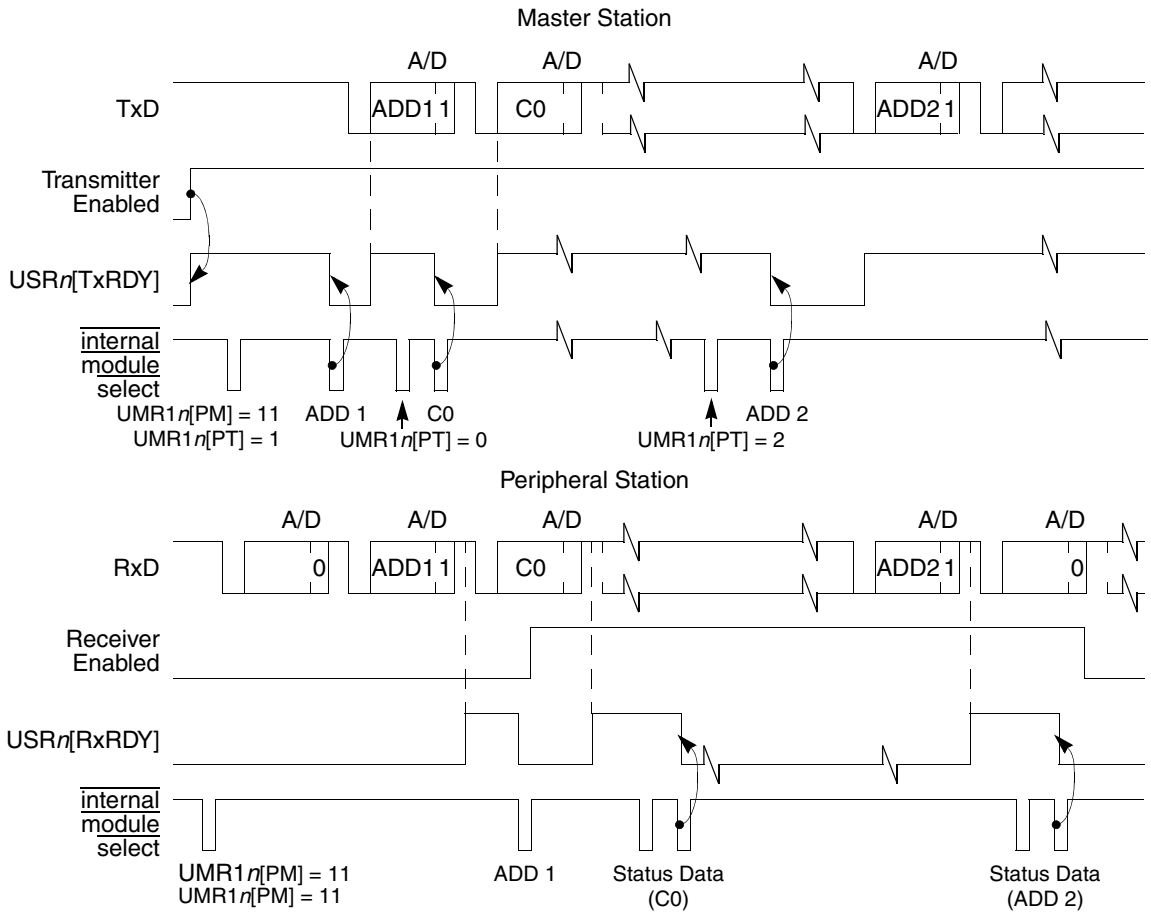

Figure 16-30. Multidrop Mode Timing Diagram

16.5.5 Bus Operation

This section describes bus operation during read, write, and interrupt acknowledge cycles to the UART module.

16.5.5.1 Read Cycles

The UART module responds to reads with byte data. Reserved registers return zeros.

16.5.5.2 Write Cycles

The UART module accepts write data as bytes. Write cycles to read-only or reserved registers complete normally without exception processing, but data is ignored.

16.5.5.3 Interrupt Acknowledge Cycles

An internal interrupt request signal notifies the interrupt controller of any unmasked interrupt conditions. The interrupt priority level is programmed in ICR2.

Table 19-2. Signal Name and Description by Pin Number (Sheet 4 of 8)

Map BGA Pin	Pin Functions				Name	Description	I/O
	0 (Reset)	1	2	3			
F6	VDD	+3.3V	—	—	VDD		
F7	GND	Ground	—	—	GND		
F8	GND	Ground	—	—	GND		
F9	VDD	+3.3V	—	—	VDD		
F10]	VDD	+3.3V	—	—	VDD		
F11	D12	PC12	—	—	D12/PC12	D12/port C bit 12	I/O
F12	D24	D8	—	—	D24/D8	D24/D8	I/O
F13	D25	D9	—	—	D25/D9	D25/D9	I/O
F14	D26	D10	—	—	D26/D10	D26/D10	I/O
G1	USB_VDD	—	—	—	USB_VDD	USB transceiver VDD	I
G2	USB_GND	—	—	—	USB_GND	USB transceiver GND	I
G3	PB4	URT0_CLK	—	—	PB4/URT0_CLK	Port B bit 4/UART0 baud clock	I/O
G4	PB6	—	—	—	PB6	Port B bit 6	I/O
G5	VDD	+3.3V	—	—	VDD		
G6	GND	Ground	—	—	GND		
G7	GND	Ground	—	—	GND		
G8	GND	Ground	—	—	GND		
G9	GND	Ground	—	—	GND		
G10	VDD	+3.3V	—	—	VDD		
G11	D11	PC11	—	—	D11/PC11	D11/port C bit 11	I/O
G12	D27	D11	—	—	D27/D11	D27/D11	I/O
G13	D28	D12	—	—	D28/D12	D28/D12	I/O
G14	D29	D13	—	—	D29/D13	D29/D13	I/O
H1	PB1	URT0_RxD	—	—	PB1/URT0_RxD	Port B bit 1/UART0 Rx data	I/O
H2	PB2	URT0_ CTS	—	—	PB2/$\overline{\text{URT0\_CTS}}$	Port B bit 2/UART0 CTS	I/O
H3	PB3	URT0_ RTS	—	—	PB3/$\overline{\text{URT0\_RTS}}$	Port B bit 3/UART0 RTS	I/O
H4	PB0	URT0_TxD	—	—	PB0/URT0_TxD	Port B bit 0/UART0 Tx data	I/O
H5	VDD	+3.3V	—	—	VDD		
H6	GND	Ground	—	—	GND		
H7	GND	Ground	—	—	GND		

Table 19-2. Signal Name and Description by Pin Number (Sheet 8 of 8)

Map BGA Pin	Pin Functions				Name	Description	I/O
	0 (Reset)	1	2	3			
N11	CS3	—	—	—	CS3	Chip select 3	O
N12	DRESETEN	—	—	—	DRESETEN	DRAM controller reset enable	I
N13	VDD	+3.3V	—	—			
N14	HIZ	—	—	—	HiZ	High impedance enable	I
P1	PA7	QSPI_CS3	DOUT3	—	PA7/QSPI_CS3/DOUT3	PA7/QSPI chip select 4/PLIC port 3 data output	I/O
P2	High Z	—	DIN3	INT4	DIN3/$\overline{\text{INT4}}$	Interrupt 4 input/PLIC port 3 data input	I
P3	INT2	—	—	—	INT2	Interrupt input 1	I
P4	QSPI_Din	—	—	—	QSPI_Din	QSPI data input	I
P5	High Z	PWM_ OUT1	TOUT1	—	PWM_OUT1/TOUT1	PWM output compare 1 /Timer 1 output compare	O
P6	E_COL	—	—	—	E_COL	Collision	I
P7	E_RxD0	—	—	—	E_RxD0	Ethernet Rx data	I
P8	E_TxEN	—	—	—	E_TxEN	Ethernet Tx enable	O
P9	PB11	E_RxD3	—	—	PB11/E_RxD3	Port B bit 11/Rx data bit 3 (100 Base-T Ethernet only)	I/O
P10	PB15	E_MDC	—	—	PB15/E_MDC	Port B bit 15/ Management Channel Clock (100 Base-T only)	I/O
P11	CS2	—	—	—	CS2	Chip select 2	O
P12	CS6	—	—	—	$\overline{\text{CS6}}$/AEN	Chip select 6	O
P13	$\overline{\text{OE}}$/RD	—	—	—	$\overline{\text{OE}}$/RD	Output enable/Read	O
P14	R/$\overline{\text{W}}$	—	—	—	R/$\overline{\text{W}}$	Read/Write	O

Table 19-7. Processor Status Encoding

PST[3:0]	Definition
0000	Continue execution
0001	Begin execution of an instruction
0010	Begin execution of PULSE instruction or WDDATA.
0011	Entry into user mode
0100	Begin execution of PULSE instruction
0101	Begin execution of taken branch
0110	Reserved
0111	Begin execution of RTE instruction
1000	Begin 1 byte transfer on DDATA
1001	Begin 2 byte transfer on DDATA
1010	Begin 3 byte transfer on DDATA
1011	Begin 4 byte transfer on DDATA
1100 [1]	Exception processing
1101 [1]	Emulator mode entry exception processing
1110 [1]	Processor is stopped, waiting for an interrupt
1111 [1]	Processor is halted

[1] These encodings are asserted for multiple cycles.

19.17.9 Debug Data (DDATA[3:0])

Debug data signals (DDATA[3:0]) display captured processor data and breakpoint status.

19.17.10 Device Test Enable ($\overline{\text{TEST}}$)

$\overline{\text{TEST}}$ is used to put the device into various manufacturing test modes. It should be tied to VDD for normal operation.

19.18 Operating Mode Configuration Pins

The MCF5272 has four mode-select signals, some of which are shared with output signals used during normal device operation. These signals are HI-Z, QSPI_Dout/WSEL, QSPI_CLK/BUSW1, and QSPI_CS0/BUSW0. BYPASS is a Freescale test mode signal and should never have a pull-down resistor. The remaining three mode-select signals must each have a 4.7-K¾ pull-up or pull-down resistor. These signals are sampled on the rising edge of Reset Output ($\overline{\text{RSTO}}$).

Table 20-3 through Table 20-5 describe data bus byte strobes. Note that most SDRAMs associate DQM3 with the MSB, thus $\overline{BS3}$ should be connected to the SDRAM's DQM3 input.

Table 20-3. Byte Strobe Operation for 32-Bit Data Bus

$\overline{BS3}$	$\overline{BS2}$	$\overline{BS1}$	$\overline{BS0}$	Access Type	Access Size	Data Located On
1	1	1	1	None	None	—
1	1	1	0	FLASH/SRAM	Byte	D[31:24]
1	1	0	1	FLASH/SRAM		D[23:16]
1	0	1	1	FLASH/SRAM		D[15:8]
0	1	1	1	FLASH/SRAM		D[7:0]
1	1	0	0	FLASH/SRAM	Word	D[31:16]
0	0	1	1	FLASH/SRAM		D[15:0]
0	0	0	0	FLASH/SRAM	Longword	D[31:0]
1	1	1	0	SDRAM	Byte	D[7:0]
1	1	0	1	SDRAM		D[15:8]
1	0	1	1	SDRAM		D[23:16]
0	1	1	1	SDRAM		D[31:24]
1	1	0	0	SDRAM	Word	D[15:0]
0	0	1	1	SDRAM		D[31:16]
0	0	0	0	SDRAM	Longword	D[31:0]

Table 20-4. Byte Strobe Operation for 16-Bit Data Bus—SRAM Cycles

BS1	BS0	Access Size	Data Located On
1	1	None	—
1	0	Byte	D[31:24]
0	1	Byte	D[23:16]
0	0	Word	D[31:16]

Table 20-5. Byte Strobe Operation for 16-Bit Data Bus—SDRAM Cycles

$\overline{BS3}$	$\overline{BS2}$	Access Type	Data Located On
1	1	None	—
1	0	Byte	D[23:16]
0	1	Byte	D[31:24]
0	0	Word	D[31:16]

Figure 20-20 shows a longword write access to a 32-bit port with a transfer error.

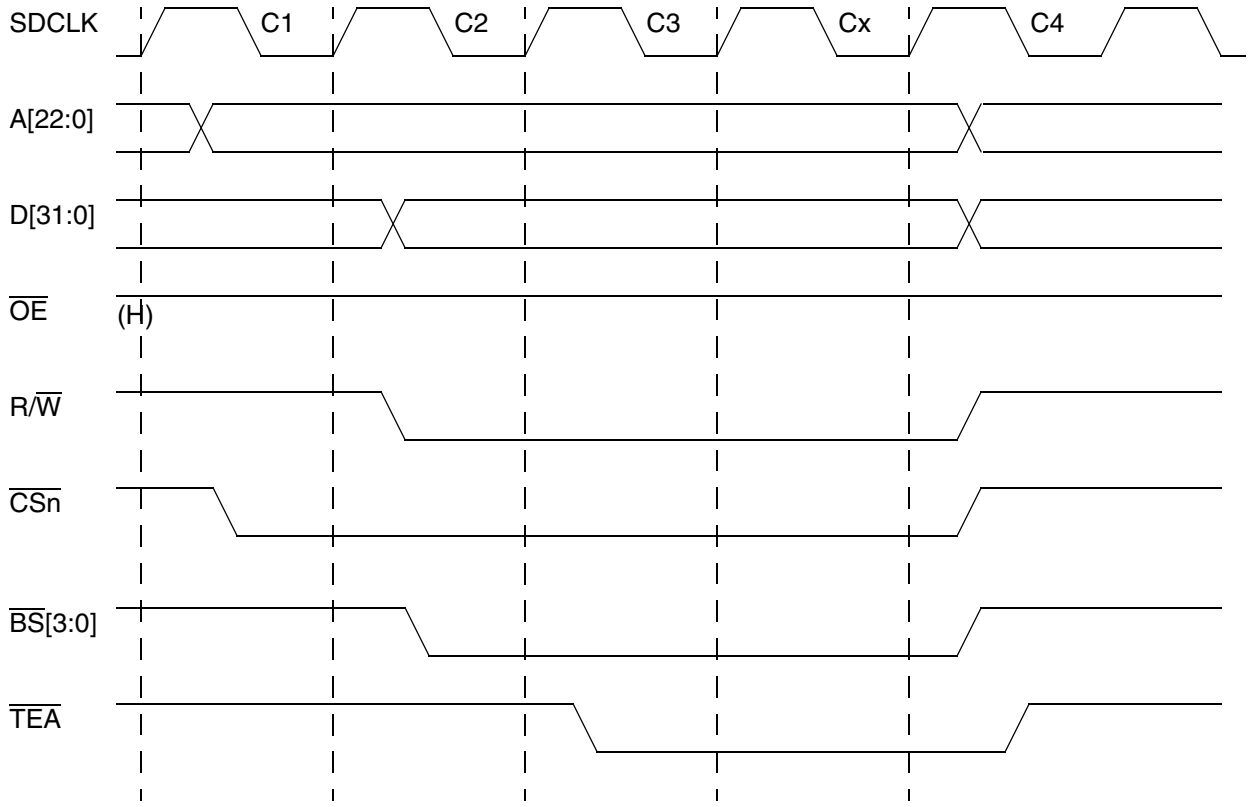

Figure 20-20. Longword Write Access To 32-Bit Port Terminated with TEA Timing

Clock 1 (C1)

The write cycle starts in C1. During C1, the MCF5272 places valid values on the address bus (A[22:0]) and the chip select signal.

Clock 2 (C2)

During C2, the MCF5272 drives the data bus, the byte strobes, and R/W.

Clock 3 (C3)

During C3, the selected device detects an error and asserts TEA. At the end of C3 or Cx, the MCF5272 samples the level of TEA. If it is asserted, the transfer of the longword is aborted and the transfer terminates.

NOTE

This example shows TEA being asserted during C3. TEA can be asserted earlier or later than C3.

NOTE

If TA is asserted when debug transfer error-acknowledge (TEA) is asserted, the transfer is terminated with a bus error.

23.6.4 MII Serial Management Channel Timing (MDIO and MDC)

Table 23-14 lists MII serial management channel timings. The FEC functions correctly with a maximum MDC frequency of 2.5 MHz.

Table 23-14. MII Serial Management Channel Timing

Num	Characteristic	Min	Max	Unit
M10	MDC falling edge to MDIO output invalid (minimum propagation delay)	0	—	nS
M11	MDC falling edge to MDIO output valid (max prop delay)	—	25	nS
M12	MDIO (input) to MDC rising edge setup	10	—	nS
M13	MDIO (input) to MDC rising edge hold	0	—	nS
M14	MDC pulse-width high	40%	60%	MDC period
M15	MDC pulse-width low	40%	60%	MDC period

Figure 23-14 shows MII serial management channel timings listed in Table 23-14.

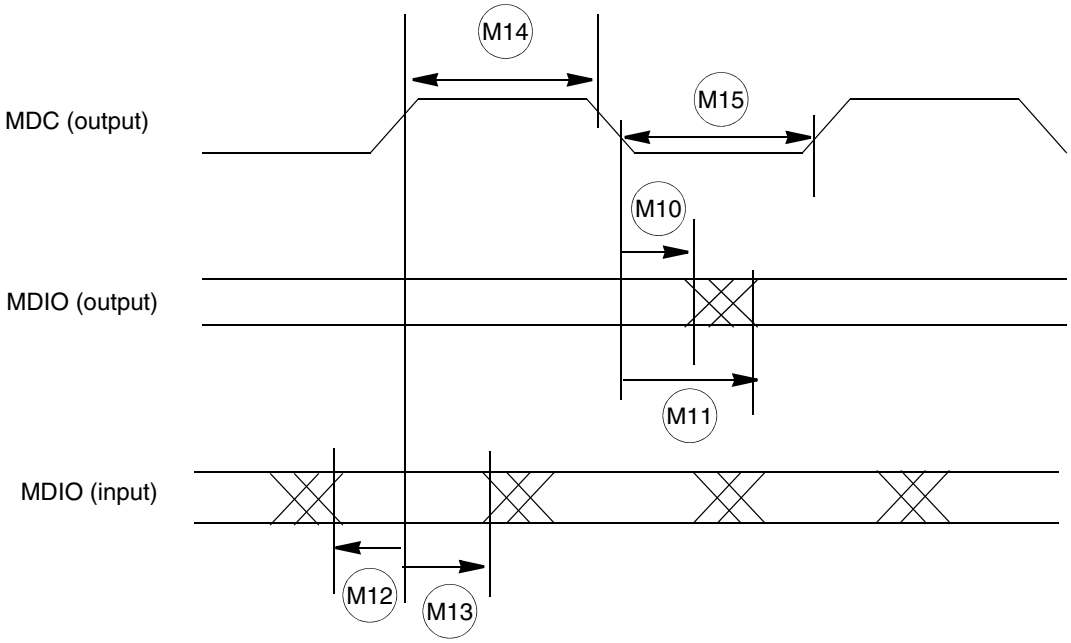

Figure 23-14. MII Serial Management Channel Timing Diagram