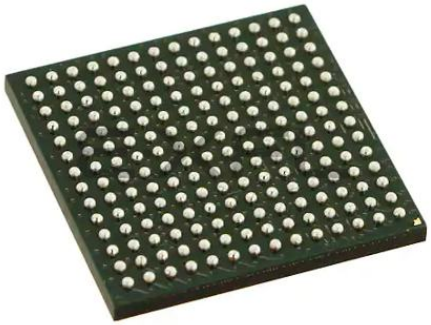

Welcome to E-XFL.COM

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	0°C ~ 70°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/mcf5272vm66r2

List of Figures (Continued)

Table of Contents (Continued)

Register Identifiers

Register identifiers in this user's manual were changed from names used in early versions of the manual released under non-disclosure agreement (NDA). Because a significant amount of collateral documentation, such as source code, was developed using the old register names, Table iii through Table xvi list the new register names and mnemonics as they should appear in the data book, along with the old ones. Identifiers that have changed are displayed in boldface type. Those that have not changed are marked "no change" in the 'New Mnemonic' column.

Table iii. On-Chip Peripherals and Configuration Registers Memory Map

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x0000	Module Base Address Register, Read Only	MBAR	No change
0x0004	System Configuration Register	SCR	No change
0x0006	System Protection Register	SPR	No change
0x0008	Power Management Register	PMR	No change
0x000E	Activate Low Power Register	ALPR	No change
0x0010	Device Identification Register	DIR	No change

Table iv. Interrupt Control Register Memory Map

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x0020	Interrupt Control Register 1	ICR1	No change
0x0024	Interrupt Control Register 2	ICR2	No change
0x0028	Interrupt Control Register 3	ICR3	No change
0x002C	Interrupt Control Register 4	ICR4	No change
0x0030	Interrupt Source Register	ISR	No change
0x0034	Programmable Interrupt Transition Register	PITR	No change
0x0038	Programmable Interrupt Wakeup Register	PIWR	No change
0x003F	Programmable Interrupt Vector Register	PIVR	No change

Table v. Chip Select Register Memory Map

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x0040	CS Base Register 0	CSBR0	No change
0x0044	CS Option Register 0	CSOR0	No change
0x0048	CS Base Register 1	CSBR1	No change
0x004C	CS Option Register 1	CSOR1	No change

Table 2-7. User-Mode Instruction Set Summary (continued)

Instruction	Operand Syntax	Operand Size	Operation
BCHG	Dy,<ea>x #<data>,<ea-1>x	.B,.L .B,.L	~(<bit number> of destination) → Z, Bit of destination
BCLR	Dy,<ea>x #<data>,<ea-1>x	.B,.L .B,.L	~(<bit number> of destination) → Z; 0 → bit of destination
BRA	<label>	.B,.W	PC + 2 + d_n → PC
BSET	Dy,<ea>x #<data>,<ea-1>x	.B,.L .B,.L	~(<bit number> of destination) → Z; 1→ bit of destination
BSR	<label>	.B,.W	SP – 4 → SP; next sequential PC→ (SP); PC + 2 + d_n → PC
BTST	Dy,<ea>x #<data>,<ea-1>x	.B,.L .B,.L	~(<bit number> of destination) → Z
CLR	<ea>y,Dx	.B,.W,.L	0 → destination
CMP	<ea>y,Ax	.L	Destination – source
CMPA	<ea>y,Dx	.L	Destination – source
CMPI	<ea>y,Dx	.L	Destination – immediate data
DIVS	<ea-1>y,Dx <ea>y,Dx	.W .L	Dx /<ea>y → Dx {16-bit remainder; 16-bit quotient} Dx /<ea>y → Dx {32-bit quotient} Signed operation
DIVU	<ea-1>y,Dx Dy,<ea>x	.W .L	Dx /<ea>y → Dx {16-bit remainder; 16-bit quotient} Dx /<ea>y → Dx {32-bit quotient} Unsigned operation
EOR	Dy,<ea>x	.L	Source ^ destination → destination
EORI	#<data>,Dx	.L	Immediate data ^ destination → destination
EXT	#<data>,Dx	.B →.W .W →.L	Sign-extended destination → destination
EXTB	Dx	.B →.L	Sign-extended destination → destination
HALT[1]	None	Unsized	Enter halted state
JMP	<ea-3>y	Unsized	Address of <ea> → PC
JSR	<ea-3>y	Unsized	SP – 4 → SP; next sequential PC → (SP); <ea> → PC
LEA	<ea-3>y,Ax	.L	<ea> → Ax
LINK	Ax,#<d16>	.W	SP – 4 → SP; Ax → (SP); SP → Ax; SP + d16 → SP
LSL	Dy,Dx #<data>,Dx	.L .L	X/C ← (Dx << Dy) ← 0 X/C ← (Dx << #<data>) ← 0
LSR	Dy,Dx #<data>,Dx	.L .L	0 → (Dx >> Dy) → X/C 0 → (Dx >> #<data>) → X/C
MAC	Ry,RxSF	.L + (.W × .W) → .L .L + (.L × .L) → .L	ACC + (Ry × Rx){<< 1 | >> 1} → ACC ACC + (Ry × Rx){<< 1 | >> 1} → ACC; (<ea>y{&MASK}) → Rw
MACL	Ry,RxSF,<ea-1>y,Rw	.L + (.W × .W) → .L, .L .L + (.L × .L) → .L, .L	ACC + (Ry × Rx){<< 1 | >> 1} → ACC ACC + (Ry × Rx){<< 1 | >> 1} → ACC; (<ea-1>y{&MASK}) → Rw
MOVE	<ea>y,<ea>x	.B,.W,.L	<ea>y → <ea>x

5.5.2 BDM Serial Interface

When the CPU is halted and PST reflects the halt status, the development system can send unrestricted commands to the debug module. The debug module implements a synchronous protocol using two inputs (DSCLK and DSI) and one output (DSO), where DSO is specified as a delay relative to the rising edge of the processor clock. See Table 5-1. The development system serves as the serial communication channel master and must generate DSCLK.

The serial channel operates at a frequency from DC to 1/5 of the PSTCLK frequency. The channel uses full-duplex mode, where data is sent and received simultaneously by both master and slave devices. The transmission consists of 17-bit packets composed of a status/control bit and a 16-bit data word. As shown in Figure 5-12, all state transitions are enabled on a rising edge of the PSTCLK clock when DSCLK is high; that is, DSI is sampled and DSO is driven.

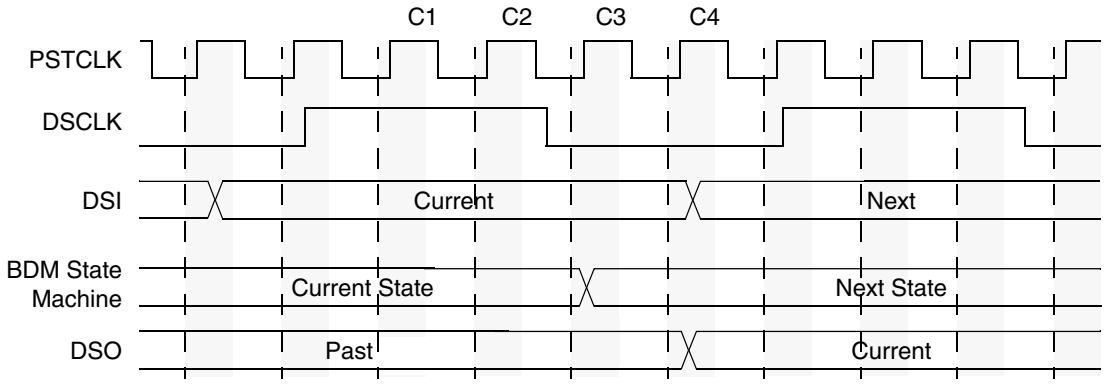

Figure 5-12. BDM Serial Interface Timing

DSCLK and DSI are synchronized inputs. DSCLK acts as a pseudo clock enable and is sampled on the rising edge of the processor CLK as well as the DSI. DSO is delayed from the DSCLK-enabled CLK rising edge (registered after a BDM state machine state change). All events in the debug module's serial state machine are based on the processor clock rising edge. DSCLK must also be sampled low (on a positive edge of CLK) between each bit exchange. The MSB is transferred first. Because DSO changes state based on an internally-recognized rising edge of DSCLK, DSDO cannot be used to indicate the start of a serial transfer. The development system must count clock cycles in a given transfer. C1–C4 are described as follows:

- C1—First synchronization cycle for DSI (DSCLK is high).
- C2—Second synchronization cycle for DSI (DSCLK is high).
- C3—BDM state machine changes state depending upon DSI and whether the entire input data transfer has been transmitted.
- C4—DSO changes to next value.

NOTE

A not-ready response can be ignored except during a memory-referencing cycle. Otherwise, the debug module can accept a new serial transfer after 32 processor clock periods.

5.5.3.3.12 Write Debug Module Register (WDMREG)

The operand (longword) data is written to the specified debug module register. All 32 bits of the register are altered by the write. DSCLK must be inactive while the debug module register writes from the CPU accesses are performed using the WDEBUG instruction.

Command Format:

Figure 5-39. WDMREG BDM Command Format

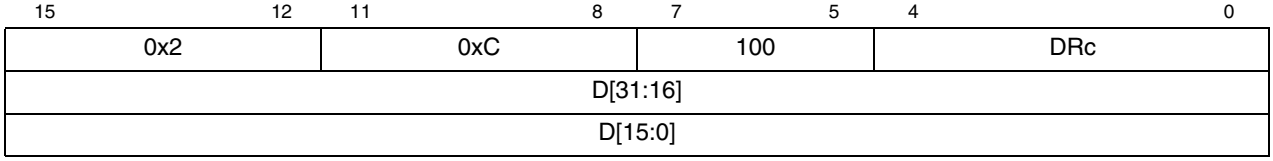

Table 5-3 shows the definition of the DRc write encoding.

Command Sequence:

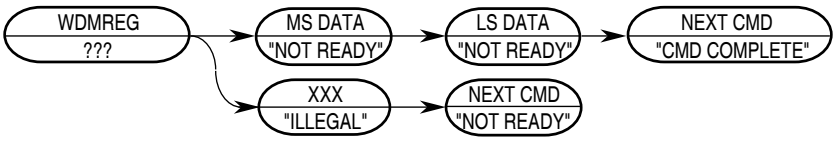

Figure 5-40. WDMREG Command Sequence

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

Result Data: Command complete status (0xFFFF) is returned when register write is complete.

5.6 Real-Time Debug Support

The ColdFire Family provides support debugging real-time applications. For these types of embedded systems, the processor must continue to operate during debug. The foundation of this area of debug support is that while the processor cannot be halted to allow debugging, the system can generally tolerate small intrusions into the real-time operation.

The debug module provides three types of breakpoints—PC with mask, operand address range, and data with mask. These breakpoints can be configured into one- or two-level triggers with the exact trigger response also programmable. The debug module programming model can be written from either the external development system using the debug serial interface or from the processor's supervisor programming model using the WDEBUG instruction. Only CSR is readable using the external development system.

Table 8-4. Chip Select Memory Address Decoding Priority

Priority	Chip Select
Highest	Chip select 0
	Chip select 1
	Chip select 2
	Chip select 3
	Chip select 4
	Chip select 5
	Chip select 6
Lowest	Chip select 7

8.2.2 Chip Select Option Registers (CSOR0–CSOR7)

CSOR0–CSOR7, Figure 8-2, are used to configure the address mask, additional setup/hold, extended burst capability, wait states, and read/write access.

	31 12	11	10	9	8	7	6 2	1	0
Field	BAM	ASET	WRAH	RDAH	EXTBURST	—	WS	RW	MRW
Reset	0xFFFF_F078								
R/W	R/W								
Addr	0x044 (CSOR0); 0x04C (CSOR1); 0x054 (CSOR2); 0x05C (CSOR3); 0x064 (CSOR4); 0x06C (CSOR5); 0x074 (CSOR6); 0x07C (CSOR7)								

Figure 8-2. Chip Select Option Registers (CSOR*n*)

Table 8-5 describes CSOR*n* fields.

Table 8-5. CSOR*n* Field Descriptions

Name	Name	Description
31–12	BAM	Address mask. Masks equivalent CSOR[BA] bits. The BAM setting chooses which BA bits to compare with the corresponding address bit to determine a match. 0 Mask address bit 1 Compare address bit
11	ASET	Address setup enable. Controls assertion of chip select with respect to assertion of a valid address that hits in the chip select address space. 0 Assert chip select on the rising edge of CLK that address is asserted. 1 Delay assertion of chip select for one CLK cycle after address is asserted. During write transfers, both chip select and R/$\overline{W}$ are delayed by 1 clock cycle. R/$\overline{W}$ asserts 1 clock cycle after assertion of the chip select.
10	WRAH	Controls the address, data, and attribute hold time after the termination, internal or external with $\overline{TA}$, of a write cycle that hits in the chip select address space. 0 Do not hold address, data, and attribute signals an extra cycle after chip select and R/$\overline{W}$ negate on writes. 1 Hold address, data, and attribute signals an extra cycle after $\overline{CS}x$ and R/$\overline{W}$ negate on writes.

To wake up the SDRAMs, SDCR[GSL] must be cleared. SDCR[SLEEP] remains set while the SDRAM is exiting sleep mode and is cleared when the SDRAM completes the correct sequence to exit sleep mode.

9.8 Performance

The maximum performance of the SDRAM controller is determined by the required number of cycles for page activation and precharge. The read access is influenced by the CAS latency. All SDRAM accesses are in page mode. The following table shows the number of required cycles including all dead cycles for each type of read/write SDRAM access. It assumes default timing configuration using an at least PC100-compliant SDRAM device at 66 MHz. Page miss latency includes the cycles to precharge the last open page and activate the new page before the read/write access. There are no precharge cycles when an address hits an open page.

In Table 9-9, the timing configuration is RTP = 61, RC = negligible, RCD = 0 (or 1), RP = 1 (or 0), and CLT = 1.

Table 9-9. SDRAM Controller Performance, 32-Bit Port, (RCD = 0, RP = 1) or (RCD = 1, RP = 0)

SDRAM Access		Number of System Clock Cycles	
		REG = 0, INV = 0	REG = 1, INV = 0
Single-beat read	Page miss	8	9
	Page hit	5	6
Single-beat write	Page miss	6	6
	Page hit	3	3
Burst read	Page miss	8-1-1-1 = 11	9-1-1-1 = 12
	Page hit	5-1-1-1 = 8	6-1-1-1 = 9
Burst write	Page miss	6-1-1-1 = 9	6-1-1-1 = 9
	Page hit	3-1-1-1 = 6	3-1-1-1 = 6

In Table 9-10, the timing configuration is RTP = 61, RC = negligible, RCD = 0, RP = 0, and CLT = 1.

Table 9-10. SDRAM Controller Performance, 32–Bit Port, (RCD = 0, RP = 0)

SDRAM Access		Number of System Clock Cycles	
		REG = 0, INV = 0	REG = 1, INV = 0
Single-beat read	Page miss	7	8
	Page hit	5	6
Single-beat write	Page miss	5	5
	Page hit	3	3
Burst read	Page miss	7-1-1-1 = 10	8-1-1-1 = 11
	Page hit	5-1-1-1 = 8	6-1-1-1 = 9
Burst write	Page miss	5-1-1-1 = 8	5-1-1-1 = 8
	Page hit	3-1-1-1 = 6	3-1-1-1 = 6

Table 11-3. Ethernet Address Recognition

Destination Address Type	FEC Address Processing
individual	The FEC compares the destination address field of the received frame with the 48-bit MAC address programmed into MALR and MAUR.
group	The FEC determines whether or not the group address is a broadcast address. If not broadcast, a hash table lookup is performed using the 64-entry hash table defined in HTLR and HTUR.
broadcast	The frame is accepted unconditionally.

In promiscuous mode (PROM = 1 in RCR), the FEC receives all of the incoming frames regardless of their address. In this mode, the destination address lookup is still performed and RxBD[MISS] is set accordingly. If address recognition did not achieve a match, the frame is received with RxBD[MISS] set. If address recognition achieves a match, the frame is received without setting RxBD[MISS].

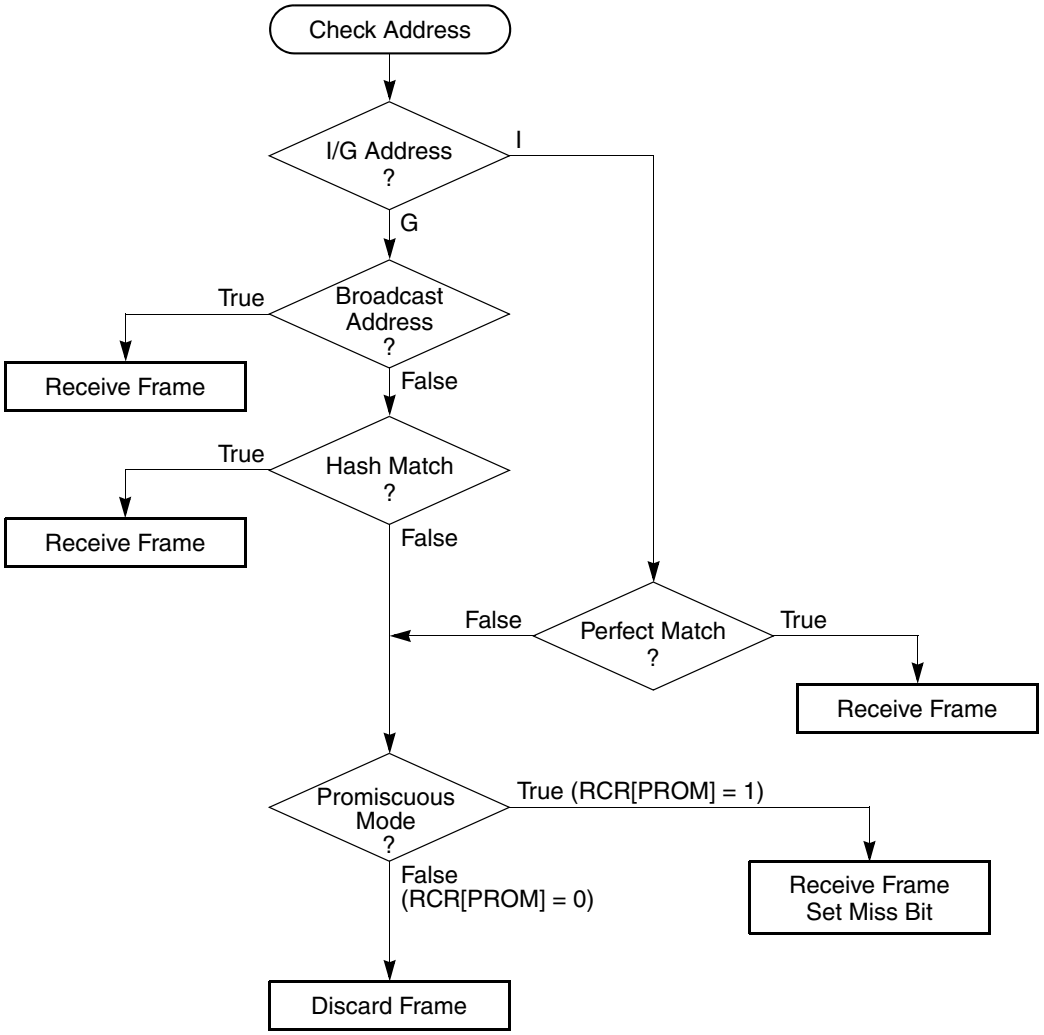

Figure 11-4. Ethernet Address Recognition Flowchart

Table 11-35. TxBD Field Descriptions (continued)

Bits	Name	Description
8	HB	Heartbeat error. Written by the FEC and is valid only if L = 1. The collision input was not asserted within the heartbeat window following the completion of transmission. Cannot be set unless the HBC bit is set in the TCR register.
7	LC	Late collision. Written by the FEC and is only valid if L = 1. A collision has occurred after 56 data bytes are transmitted. The FEC terminates the transmission.
6	RL	Retransmission limit. Written by the FEC and is only valid if L = 1. The transmitter has failed retry limit + 1 attempts to successfully transmit a message due to repeated collisions on the medium.
5–2	RC	Retry count. Written by the FEC and is valid only if L = 1. These four bits indicate the number of retries required before this frame is successfully transmitted. If RC = 0, then the frame was transmitted correctly the first time. If RC = 15, then the frame was transmitted successfully while the retry count was at its maximum value. If RL is set, then RC has no meaning.
1	UN	Underrun. Written by the FEC and is only valid if L = 1. The FEC encountered a transmit FIFO underrun while transmitting one or more of the data buffers associated with this frame. When transmit FIFO underrun occurs, transmission of the frame stops, and an incorrect CRC is appended. The remaining buffer(s) associated with this frame are sent as a DMA and dumped by the transmit logic.
0	CSL	Carrier sense lost. Written by the FEC and is valid only if L = 1. Carrier sense dropped out or never asserted during transmission of a frame without collision.
-	Data Length	Written by the user. Data length is the number of octets the FEC should transmit from this BD's data buffer. It is never modified by the FEC. Bits [10–0] are used by the DMA engine; bits [15–11] are ignored.
-	Tx Buffer Pointer	Written by the user. The transmit buffer pointer, which contains the address of the associated data buffer, may be even or odd. The buffer must reside in memory external to the FEC. This value is never modified by the FEC.

An underrun occurs when, during transmit, the transmit FIFO empties before the end of the frame. Then, a bad CRC is appended to the partially transmitted data. In addition, the UN bit is set in the last BD for the current frame. This situation can occur if the FEC cannot access an internal bus, or if the next BD in the frame is not available.

NOTE

Anytime the software driver sets an R bit in a transmit descriptor, the driver should immediately write to TDAR.

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.

12.3.2.13 USB Endpoint 0 Control Register (EP0CTL)

Figure 12-16 shows the USB endpoint 0 control register. provides both module level and endpoint 0 specific control (bits 0-9) functions.

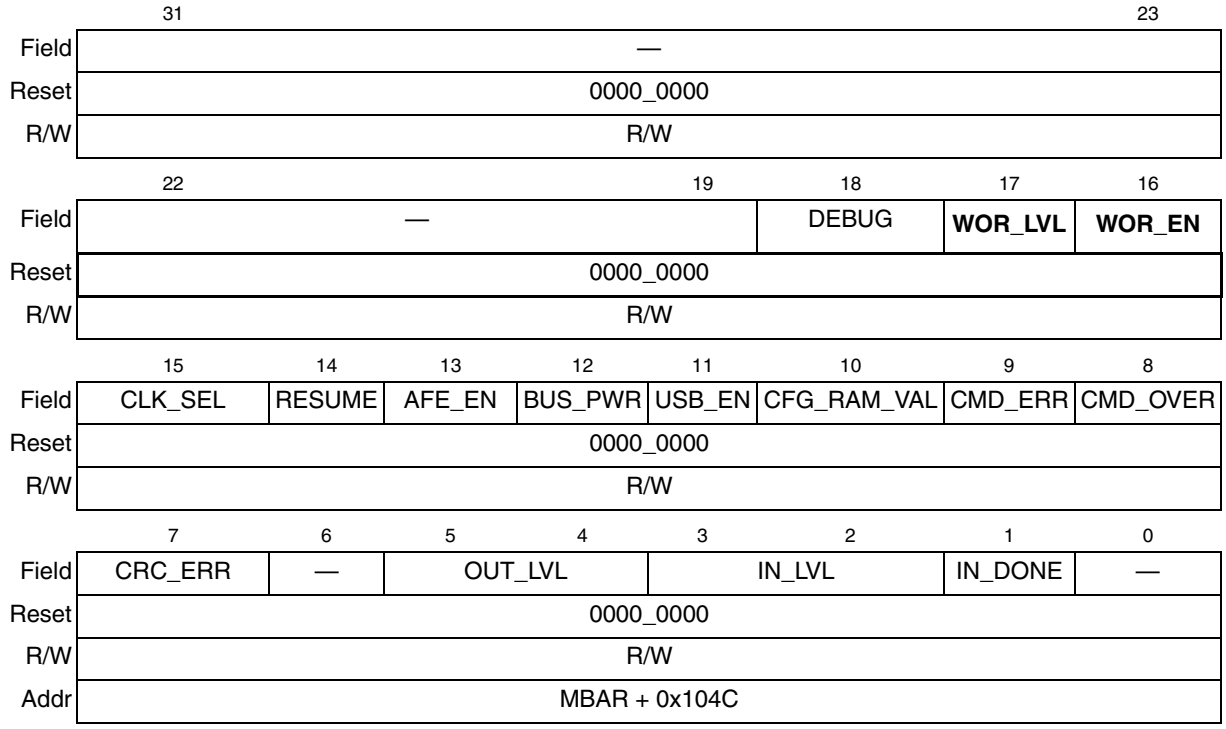

Figure 12-16. USB Endpoint 0 Control Register (EP0CTL)

Table 12-12 lists field descriptions for the USB endpoint 0 control register.

Table 12-12. EP0CTL Field Descriptions

Bits	Name	Description
31–19	—	Reserved, should be cleared.
18	DEBUG	Debug mode. Enters debug mode. Debug mode enables CRC error generation and notification of a change of address. 0 Normal operation 1 Enable debug mode functions
17	WOR_LVL	Wake-on-ring level select. Selects the active level of $\overline{\text{INT1}}$ for the wake-on-ring function. 0 Wake-on-ring function is invoked when $\overline{\text{INT1}}$ pin is 0. 1 Wake-on-ring function is invoked when $\overline{\text{INT1}}$ pin is 1.
16	WOR_EN	Wake-on-ring enable. Generates a RESUME when the active level is detected on $\overline{\text{INT1}}$ pin. 0 Wake-on-ring function disabled 1 Wake-on-ring function enabled Note: the wake-on-ring function generates a RESUME only if the USB module is enabled for remote wakeup by the host, for example, WAKE_ST = 1, and is suspended.
15	CLK_SEL	Clock source selection. Overrides the clock source for the USB module. If the USB_ExtCLK pin is selected after reset, setting this bit forces the USB module to use the internal system clock. 0 Clock is retrieved from the clock selected at reset. 1 Clock is retrieved from the internal system clock. Note: the selected clock must have a frequency of 48 MHz.

Table 12-14. EP0IMR and EP0ISR Field Descriptions (continued)

Bits	Name	Description
4	IN_EOT	End of transfer. This bit is set when the end of a transfer has been reached for an IN endpoint. An EOT interrupt is generated when a packet with a size less than the maximum packet size or the first zero-length packet following maximum size packets is sent. 0 No interrupt pending 1 Transfer completed
3	IN_EOP	End of packet. This bit is set when a packet has been sent successfully for endpoint 0 IN. 0 No interrupt pending 1 IN packet sent successfully
2	UNHALT	Unhalt. This bit is set when the endpoint 0 HALT_ST bit is cleared by a SETUP packet or USB reset. 0 No interrupt pending 1 Endpoint halt cleared
1	HALT	Halt. This bit is set when the endpoint 0 HALT_ST bit is set due to a STALL response to the host. 0 No interrupt pending 1 Endpoint halted
0	IN_LVL	IN FIFO threshold level. This bit indicates that the FIFO level has fallen below the level set in the EPCTL0 register. 0 No interrupt pending 1 IN FIFO threshold level reached

13.2.5.3 Interrupt Control

There are a number of control mechanisms for the periodic and aperiodic interrupts on the PLIC.

- Clearing the ON/OFF bit in the port configuration register, Section 13.5.7, “Port Configuration Registers (P0CR–P3CR),” turns the port off and masks all periodic and aperiodic interrupts for the affected port.
- Clearing the enable bits, ENB1 or ENB2, in the port configuration register masks the periodic transmit and receive interrupts associated with the respective B1 or B2 channel.
- Specific interrupt enables are provided in each port’s ICR. This includes a port interrupt enable, IE, which masks all periodic and aperiodic interrupts. In addition, there are interrupt enables for specific conditions. These are listed in Section 13.5.9, “Interrupt Configuration Registers (P0ICR–P3ICR).”

13.3 PLIC Timing Generator

13.3.1 Clock Synthesis

The PLIC clock generator employs a completely digital, synchronous design which can be used to synthesize a new clock by multiplying an incoming reference clock. This clock generator is not a PLL—it has no VCO or phase comparator.

The frequency multiplication factor is always an integral power of two between 2 and 256 inclusive. The amount of phase jitter exhibited by the synthesized clock increases as the synthesized clock frequency approaches CLKIN’s frequency. As a general guide, the maximum generated DCL should be no greater than one-twentieth of CLKIN’s frequency. Therefore, given a CLKIN of 66 MHz, the maximum frequency which can be synthesized with acceptable jitter is approximately 3.3 MHz.

The clock generator uses a 14-bit counter to divide CLKIN. This limits the reference clock’s minimum frequency to CLKIN divided by 16,384.

To summarize these two points:

- Synthesized clock x 20 < CLKIN(Recommended)
- Reference clock > CLKIN / 16,384(Required)

The control of the clock generator block is provided through the PCSR register detailed in Section 13.5.22, “Clock Select Register (PCSR).”

The process is illustrated by this example. Suppose the following:

- CPU clock = 66 MHz
- Reference clock = 64 KHz
- Synthesized clock = 1.024 MHz

The appropriate reference clock is selected by programming PLCLKSEL[CKI1, CKI0], Section 13.5.22, “Clock Select Register (PCSR).” The multiplication factor is 16 (1.024 MHz / 64 KHz) and is specified by PLCLKSEL[CMULT0-2]. The division ratio between the synthesized clock (GDCL), 1.024 MHz, and the synthesized frame sync (Gen_FSC) must be set. (A Gen_FSC of 8 KHz is assumed). This division ratio is selected by means of FDIV[2-0]. Finally, the clock generation block should be taken out of bypass by setting PCSR[NBP].

Table 13-1. PLIC Module Memory Map (continued)

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x0340	Port2 B2 Data Transmit (P2B2TR)			
0x0344	Port3 B2 Data Transmit (P3B2TR)			
0x0348	Port0 D Data Transmit (P0DTR)	Port1 D Data Transmit (P1DTR)	Port2 D Data Transmit (P2DTR)	Port3 D Data Transmit (P3DTR)
0x0350	Port0 GCI/IDL Configuration register (P0CR)		Port1 GCI/IDL Configuration register (P1CR)	
0x0354	Port2 GCI/IDL Configuration register (P2CR)		Port3 GCI/IDL Configuration register (P3CR)	
0x0358	Port0 Interrupt Configuration register (P0ICR)		Port1 Interrupt Configuration register (P1ICR)	
0x035C	Port2 Interrupt Configuration register (P2ICR)		Port3 Interrupt Configuration register (P3ICR)	
0x0360	Port0 GCI monitor Rx (P0GMR)		Port1 GCI monitor Rx (P1GMR)	
0x0364	Port2 GCI monitor Rx (P2GMR)		Port3 GCI monitor Rx (P3GMR)	
0x0368	Port0 GCI monitor Tx (P0GMT)		Port1 GCI monitor Tx (P1GMT)	
0x036C	Port2 GCI monitor Tx (P2GMT)		Port3 GCI monitor Tx (P3GMT)	
0x0370	Reserved	GCI monitor Tx status (PGMTS)	GCI monitor Tx abort (PGMTA)	Reserved
0x0374	Port0 GCI C/I Rx (P0GCIR)	Port1 GCI C/I Rx (P1GCIR)	Port2 GCI C/I Rx (P2GCIR)	Port3 GCI C/I Rx (P3GCIR)
0x0378	Port0 GCI C/I Tx (P0GCIT)	Port1 GCI C/I Tx (P1GCIT)	Port2 GCI C/I Tx (P2GCIT)	Port3 GCI C/I Tx (P3GCIT)
0x037C	Reserved			GCI C/I Tx Status (PGCITSR)
0x0383	Reserved			GCI C/I D-Channel Status (PDCSR)
0x0384	Port0 periodic status (P0PSR)		Port1 periodic status (P1PSR)	
0x0388	Port2 periodic status (P2PSR)		Port3 periodic status (P3PSR)	
0x038C	Aperiodic Interrupt status register (PASR)		Reserved	Loop back Control (PLCR)
0x0390	Reserved		D-Channel Request (PDRQR)	
0x0394	Port0 Sync Delay (P0SDR)		Port1 Sync Delay (P1SDR)	
0x0398	Port2 Sync Delay (P2SDR)		Port3 Sync Delay (P3SDR)	
0x039C	Reserved		Clock Select (PCSR)	

Chapter 14
Queued Serial Peripheral Interface (QSPI) Module

This chapter describes the queued serial peripheral interface (QSPI) module. Following a feature-set overview is a description of operation including details of the QSPI's internal RAM organization. The chapter concludes with the programming model and a timing diagram.

14.1 Overview

The queued serial peripheral interface module provides a serial peripheral interface with queued transfer capability. It allows users to enqueue up to 16 transfers at once, eliminating CPU intervention between transfers. Transfer RAMs in the QSPI are indirectly accessible using address and data registers.

Functionality is very similar, but not identical, to the QSPI portion of the QSM (queued serial module) implemented in the MC68332.

14.2 Features

- Programmable queue to support up to 16 transfers without user intervention
- Supports transfer sizes of 8 to 16 bits in 1-bit increments
- Four peripheral chip-select lines for control of up to 15 devices
- Baud rates from 129.4 Kbps to 16.5 Mbps at 66 MHz
- Programmable delays before and after transfers
- Programmable clock phase and polarity
- Supports wraparound mode for continuous transfers

14.3 Module Description

The QSPI module communicates with the integrated ColdFire CPU using internal memory mapped registers located starting at MBAR + 0xA0. See also Section 14.5, "Programming Model." A block diagram of the QSPI module is shown in Figure 14-1.

16.3.11 UART Divider Upper/Lower Registers (UDU*n*/UDL*n*)

The UDU*n* registers (formerly called UBG1*n*) hold the MSB, and the UDL*n* registers (formerly UBG2*n*) hold the LSB of the preload value. UDU*n* and UDL*n* concatenate to provide a divider to CLKIN for transmitter/receiver operation, as described in Section 16.5.1.2.1, “CLKIN Baud Rates.”

	7	0
Field	Divider MSB	
Reset	0000_0000	
R/W	Write only	
Address	MBAR + 0x118 (UDU0), 0x158 (UDU1)	

Figure 16-12. UART Divider Upper Registers (UDU*n*)

	7	0
Field	Divider LSB	
Reset	0000_0000	
R/W	Write only	
Address	MBAR + 0x11C (UDL0), 0x15C (UDL1)	

Figure 16-13. UART Divider Lower Registers (UDL*n*)

NOTE

The minimum value that can be loaded on the concatenation of UDU*n* with UDL*n* is 0x0002. Both UDU*n* and UDL*n* are write-only and cannot be read by the CPU.

16.3.12 UART Autobaud Registers (UABU*n*/UABL*n*)

The UABU*n* registers hold the MSB, and the UABL*n* registers hold the LSB of the calculated baud rate. If UCR*n*[ENAB] is set, the value in these registers is automatically loaded into UDU*n* and UDL*n*.

	7	0
Field	Autobaud MSB	
Reset	0000_0000	
R/W	Read only	
Address	MBAR + 0x120 (UABU0), 0x160 (UABU1)	

Figure 16-14. UART Autobaud Upper Registers (UABU*n*)

	7	0
Field	Autobaud LSB	
Reset	0000_0000	
R/W	Read only	
Address	MBAR + 0x124 (UABL0), 0x164 (UABL1)	

Figure 16-15. UART Autobaud Lower Registers (UABL*n*)

An internal interrupt request signal ($\overline{IRQ}$) is provided to notify the interrupt controller of an interrupt condition. The output is the logical NOR of unmasked UISR*n* bits. The interrupt levels of the UART modules are programmed in SIM register ICR2. See Section 7.2, “Interrupt Controller Registers.”

Table 16-15 briefly describes the UART module signals.

NOTE

The terms ‘assertion’ and ‘negation’ are used to avoid confusion between active-low and active-high signals. ‘Asserted’ indicates that a signal is active, independent of the voltage level; ‘negated’ indicates that a signal is inactive.

Table 16-15. UART Module Signals

Signal	Description
Transmitter Serial Data Output (URT_TxD)	URT_TxD is held high (mark condition) when the transmitter is disabled, idle, or operating in the local loop-back mode. Data is shifted out on URT_TxD on the falling edge of the clock source, with the least significant bit (lsb) sent first.
Receiver Serial Data Input (URT_RxD)	Data received on URT_RxD is sampled on the rising edge of the clock source, with the lsb received first.
Clear-to-Send ($\overline{URT\_CTS}$)	This input can generate an interrupt on a change of state.
Request-to-Send ($\overline{URT\_RTS}$)	This output can be programmed to be negated or asserted automatically by either the receiver or the transmitter. It can control serial data flow when connected to a transmitter’s $\overline{CTS}$.
Clock (URT_CLK)	The UART’s external clock source. It can be used in 1x or 16x mode. When both the transmitter and receiver use the timer as the clock source (UCR = 0xDD), the 16x clock is driven out on UARTCLK. If either the transmitter or receiver use an external clock (1x or 16x), URT_CLK is an input.

Figure 16-22 shows a signal configuration for a UART/RS-232 interface.

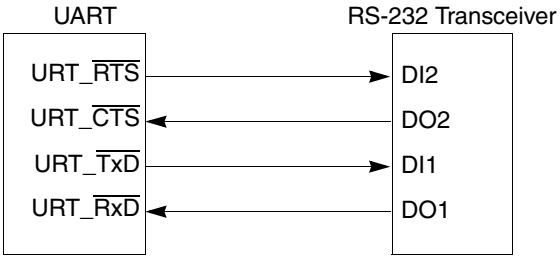

Figure 16-22. UART/RS-232 Interface

16.5 Operation

This section describes operation of the clock source generator, transmitter, and receiver.

16.5.1 Transmitter/Receiver Clock Source

CLKIN serves as the basic timing reference for the clock source generator logic, which consists of a clock generator and a programmable 16+4-bit divider (UDU, UDL, UFPD) dedicated to the UART. The clock generator cannot produce standard baud rates if CLKIN is used, so the 16-bit divider should be used.

Summary of the main features include:

- Double-buffered width register
- Variable-divide prescale
- Three independent PWM modules
- Byte-wide width register provides programmable duty cycle control

18.2 PWM Operation

The PWM is a simple free-running counter combined with a pulse-width register and a comparator such that the output is cleared whenever the counter value exceeds the width register value. When the counter overflows, or “wraps around,” its value becomes less than or equal to the value of the width register, and the output is set. With a suitable low-pass filter, the PWM can be used as a digital-to-analog converter.

The width register is double-buffered so that a new value can be loaded for the next cycle without affecting the current cycle. At the beginning of each period, the value of the width buffer register is loaded into the width register, which feeds the comparator. This value is used for comparison during the next cycle. The prescaler contains a variable divider that can reduce the incoming clock frequency by certain values between 1 and 32768.

18.3 PWM Programming Model

This section describes the registers and control bits in the PWM module. There are three independent PWM modules, each with its own control and width registers. The memory map for the PWM is shown in Table 18-1.

Table 18-1. PWM Module Memory Map

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x00C0	PWM Control Register 0 (PWCR0)	Reserved		
0x00C4	PWM Control Register 1 (PWCR1)	Reserved		
0x00C8	PWM Control Register 2 (PWCR2)	Reserved		
0x00D0	PWM Pulse-Width Register 0 (PWWD0)	Reserved		
0x00D4	PWM Pulse-Width Register 1 (PWWD1)	Reserved		
0x00D8	PWM Pulse-Width Register 2 (PWWD2)	Reserved		

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

Map BGA Pin	Pin Functions				Name	Description	I/O
	0 (Reset)	1	2	3			
B12	A1	SDA0	—	—	A1/SDA0	A1/SDRAM-16bit A0	O
B13	A5	SDA4	SDA3	—	A5/SDA4/SDA3	A5/SDRAM-16bit A4/SDRAM-32bit A3	O
B14	A6	SDA5	SDA4	—	A6/SDA5/SDA4	A6/SDRAM-16bit A5/SDRAM-32bit A4	O
C1	PST2	—	—	—	PST2	Internal processor status 2	O
C2	PST1	—	—	—	PST1	Internal processor status 1	O
C3	DDATA0	—	—	—	DDATA0	Debug data 0	O
C4	TCK	PSTCLK	—	—	TCK/PSTCLK	JTAG test clock in/ BDM PSTCLK output	I/O
C5	A21	—	—	—	A21	A21	O
C6	A18	—	—	—	A18	A18	O
C7	D19	D3	—	—	D19/D3	D19/D3	I/O
C8	BS1	—	—	—	$\overline{BS1}$	Byte strobe 1	O
C9	CAS0	—	—	—	CAS0	SDRAM column select strobe	O
C10	A14	SDA13	SDA12	—	A14/SDA13/SDA12	A14/SDRAM-16bit A13/SDRAM-32bit A12	O
C11	A11	—	SDA9	—	A11/SDA9	A11/SDRAM-32bit A9	O
C12	A7	SDA6	SDA5	—	A7/SDA6/SDA5	A7/SDRAM-16bit A6/SDRAM-32bit A5	O
C13	A8	SDA7	SDA6	—	A8/SDA7/SDA6	A8/SDRAM-16bit A7/SDRAM-32bit A6	O
C14	A9	SDA8	SDA7	—	A9/SDA8/SDA7	A9/SDRAM-16bit A8/SDRAM-32bit A7	O
D1	PA1	USB_RP	—	—	PA1/USB_RP	Port A bit 1/ USB Rx positive	I/O
D2	PA0	USB_TP	—	—	PA0/USB_TP	Port A bit 0/ USB Tx positive	I/O
D3	PST3	—	—	—	PST3	Internal processor status 3	O
D4	TRST	DSCLK	—	—	$\overline{TRST}$/DSCLK	JTAG reset/BDM clock	I
D5	TDO	DSO	—	—	TDO/DSO	JTAG test data out /BDM data out	O
D6	A19	—	—	—	A19	A19	O
D7	A17	—	—	—	A17	A17	O