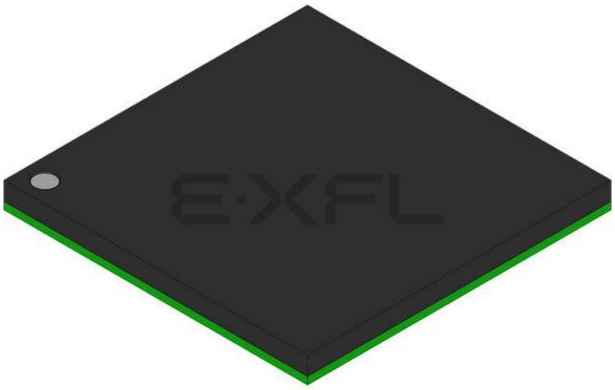
Freescale Semiconductor - MCF5272VM66 Datasheet



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	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/pro/item?MUrl=&PartUrl=mcf5272vm66

Table of Contents (Continued)

Paragraph Number — Title — Page Number

Chapter 12
Universal Serial Bus (USB)

Table vii. QSPI Module Memory Map (continued)

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x00AC	QSPI Interrupt Register	SPINT	**QIR**
0x00B0	QSPI Address Register	SPADDR	**QAR**
0x00B4	QSPI Data Register	SPDATA	**QDR**

Table viii. PWM Module Memory Map

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x00C0	**PWM Control Register 0**	PWMCR1	**PWCR0**
0x00C4	**PWM Control Register 1**	PWMCR2	**PWCR1**
0x00C8	**PWM Control Register 2**	PWMCR3	**PWCR2**
0x00D0	**PWM Pulse-Width Register 0**	PWMWD1	**PWWD0**
0x00D4	**PWM Pulse-Width Register 1**	PWMWD2	**PWWD1**
0x00D8	**PWM Pulse-Width Register 2**	PWMWD3	**PWWD2**

Table ix. DMA Module Memory Map

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x00E0	DMA Mode Register	DCMR	No change
0x00E6	DMA Interrupt Register	DCIR	No change
0x00E8	DMA Byte Count Register	DBCR	No change
0x00EC	DMA Source Address Register	DSAR	No change
0x00F0	DMA Destination Address Register	DDAR	No change

Table x. UART0 Module Memory Map

MBAR Offset	Register Name	Old Mnemonic	New Mnemonic
0x0100	UART0 Mode Register 1/2	U1MR1/U1MR2	**U0MR1/U0MR2**
0x0104	UART0 Status	U1SR	**U0SR**
0x0104	UART0 Clock Select Register	U1CSR	**U0CSR**
0x0108	UART0 Command Register	U1CR	**U0CR**
0x010C	UART0 Receive Buffer	U1RxB	**U0RxB**
0x010C	UART0 Transmit Buffer	U1TxB	**U0TxB**

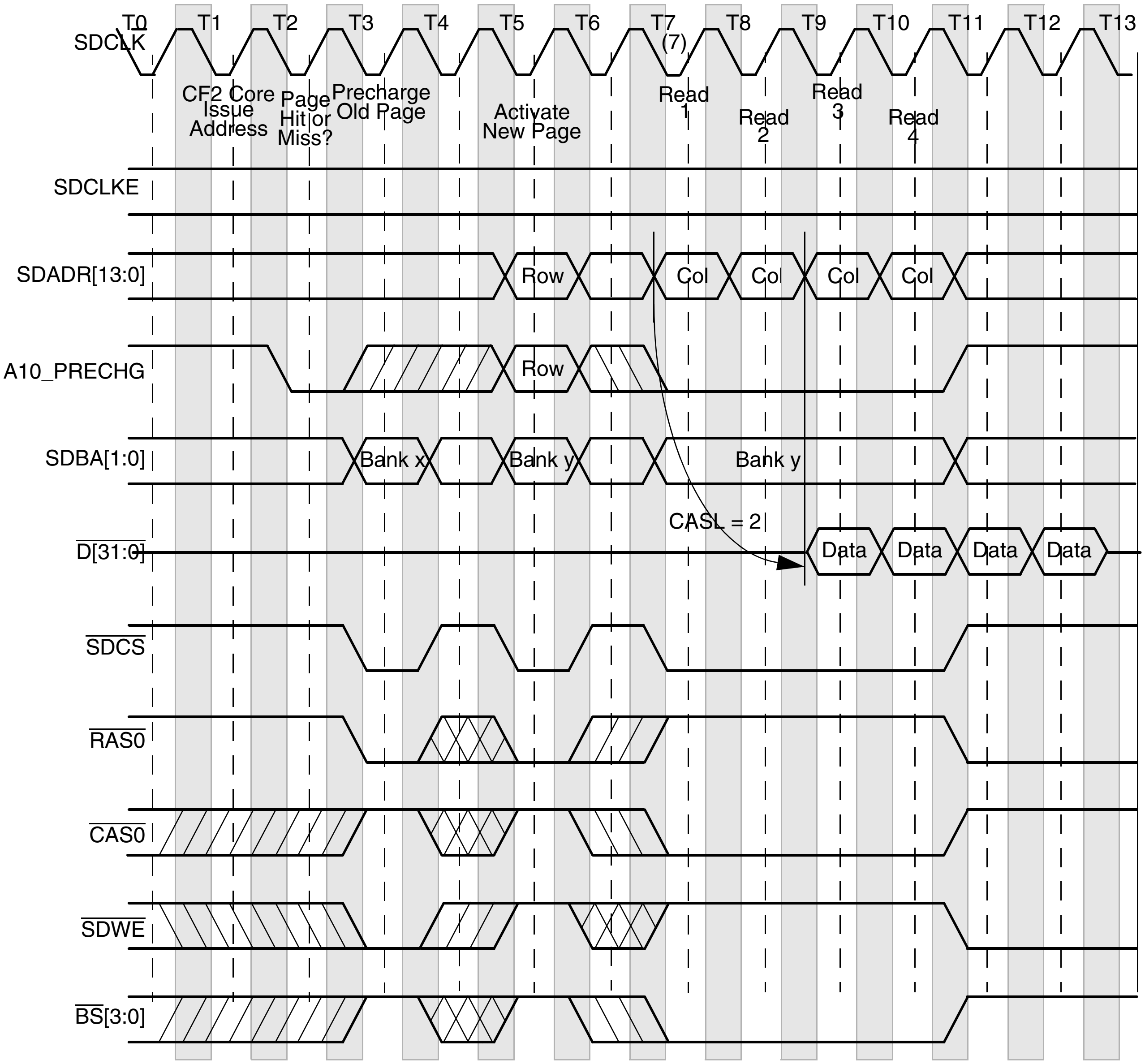


Figure 9-9. SDRAM Burst Read, 32-Bit Port, Page Miss, Access = 9-1-1-1

Table 11-32. User Initialization Process (before ETHER_EN) (continued)

Step	Description
7	Set HTUR and HTLR
8	Set EMRBR
9	Set ERDSR
10	Set ETDSR
11	Set RCR
12	Set TCR
13	Set MSCR (optional)
14	Initialize (Empty) TxBD
15	Initialize (Empty) RxBD

11.5.24 FEC Initialization

In the FEC, the descriptor control machine initializes a few registers whenever the ETHER_EN control is asserted. The transmit and receive FIFO pointers are reset, the transmit backoff random number is initialized, and the transmit and receive blocks are activated. After the FEC initialization sequence is complete, the hardware is ready for operation, waiting for RDAR and TDAR to be asserted by the user.

11.5.24.1 User Initialization (after setting ETHER_EN)

The user initializes portions of the FEC after setting ETHER_EN. The exact values depend on the particular application. The sequence probably resembles the steps shown in Table 11-33, though these could also be done before asserting ETHER_EN.

Table 11-33. User Initialization (after ETHER_EN)

Step	Description
1	Fill Receive Descriptor Ring with Empty Buffers
2	Set RDAR

11.6 Buffer Descriptors

Data associated with the FEC controller is stored in buffers, which are referenced by buffer descriptors (BDs) organized as tables in the dual-port RAM. These tables have the same basic configuration as those used by the USB.

The BD table allows users to define separate buffers for transmission and reception. Each table forms a circular queue, or ring. The FEC uses status and control fields in the BDs to inform the core that the buffers have been serviced, to confirm reception and transmission events, or to indicate error conditions.

12.3.2.18 USB Endpoint 0–7 Data Registers (EP*n*DR)

Figure 12-21 shows the USB endpoint 0-7 data registers.

	31	16
Field	DATA	
Reset	0000_0000_0000_0000	
R/W	R/W	

	15	0
Field	DATA	
Reset	0000_0000_0000_0000	
R/W	R/W	
Addr	MBAR + 0x10AC, 0x10B0, 0x10B4, 0x10B8, 0x10BC, 0x10C0, 0x10C4, 0x10C8	

Figure 12-21. USB Endpoint 0-7 Data Registers (EP*n*DR)

Table 12-17 lists field descriptions for the USB endpoint 0–7 data registers.

Table 12-17. EP*n*DR Field Descriptions

Bits	Name	Description
31–0	DATA	The EP*n*DR registers allow data to be written to/from each endpoint's FIFO. For IN endpoints, the registers are write-only and writing to a full FIFO is ignored. For OUT endpoints, the registers are read-only, and reading from an empty FIFO returns undefined data. These registers can be accessed using 8-, 16-, or 32-bit accesses in order to read/write 1, 2, or 4 bytes from/to the FIFO at one time.

The above settings can be made by a single write of the 16-bit value 0x802B to PCSR.

The following restrictions should be observed when using the clock generator module:

- The smallest multiplication factor is 2.
- CLKIN should be significantly greater than (> 20 times) the synthesized clock.

Figure 13-11 and Figure 13-12 show the connectivity and relationship of the timing signals within the PLIC block.

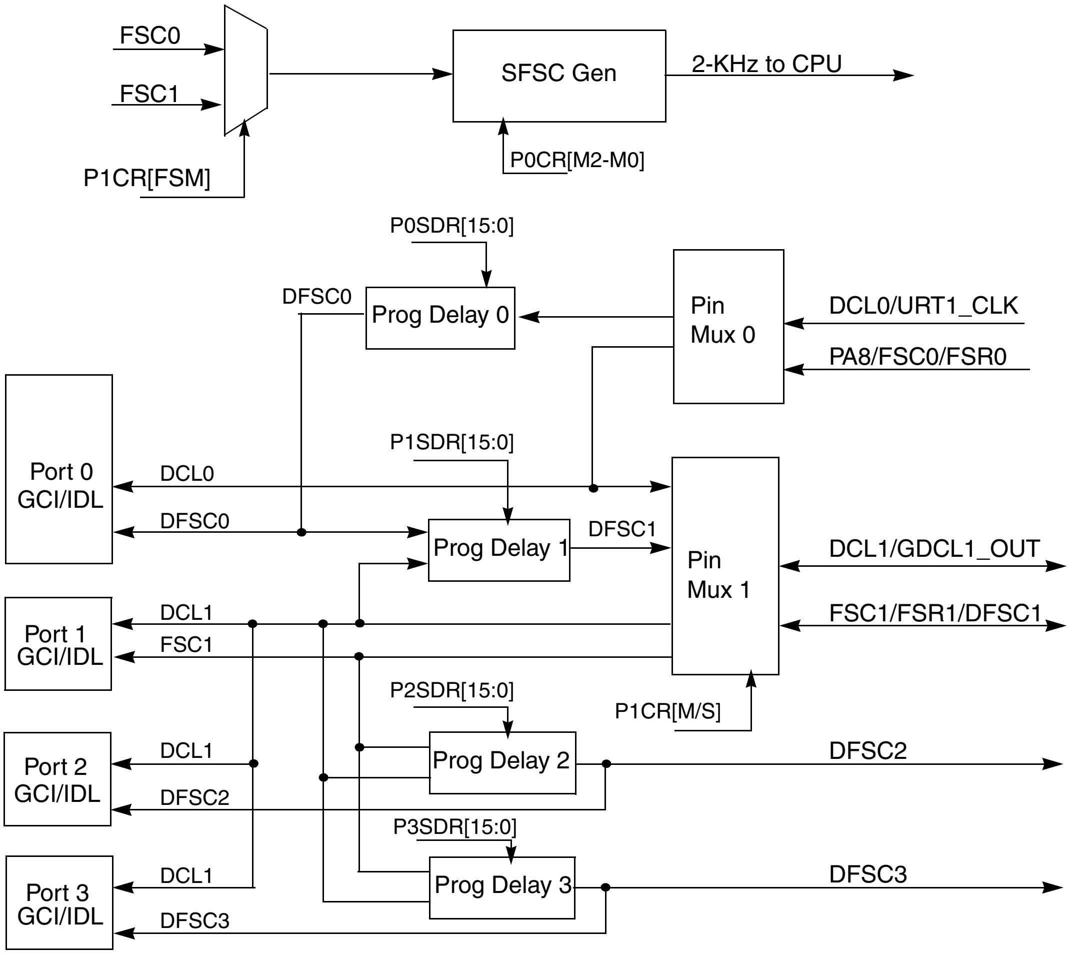


Figure 13-11. PLIC Internal Timing Signal Routing

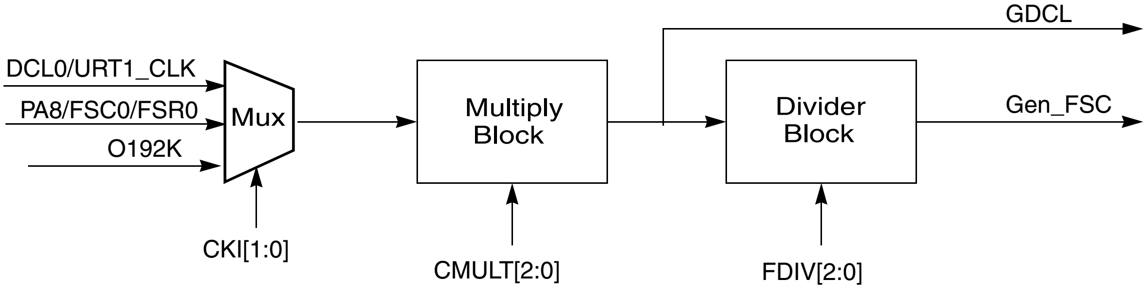


Figure 13-12. PLIC Clock Generator

13.3.2 Super Frame Sync Generation

Figure 13-11 shows the generation of the 2-KHz super frame sync. The choice of either FSC0 or FSC1 is possible using P1CR[FSM]. This allows either the port 0 or port 1 timing to be used to generate the 2-KHz super frame sync interrupt. The SFSC block then divides this accordingly. When P1CR[FSM] is set, FSC1 is the source of the super frame sync. In case P1CR[MS] is 0 (that is, port 1 is in slave mode), the interrupt is ultimately driven by an external source. In case the M/S bit is 1 (that is, port 1 is in master mode), FSC1 ultimately comes from port 0.

13.3.3 Frame Sync Synthesis

Figure 13-11 illustrates the relationships between the various frame sync clocks. DFSC1 is generated through programmable delay 1 referenced to DFSC0. DFSC2 and DFSC3 are generated through programmable delays 2 and 3 referenced to DFSC1. Note well the following:

- P0SDR settings affect DFSC[0–3]
- P1SDR settings affect DFSC[1–3]
- P2SDR settings affect only DFSC2
- P3SDR settings affect only DFSC3

13.4 PLIC Register Memory Map

Table 13-1 lists the PLIC registers with their offset address from MBAR and their default value on reset.

Table 13-1. PLIC Module Memory Map

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x0300	Port0 B1 Data Receive (P0B1RR)			
0x0304	Port1 B1 Data Receive (P1B1RR)			
0x0308	Port2 B1 Data Receive (P2B1RR)			
0x030C	Port3 B1 Data Receive (P3B1RR)			
0x0310	Port0 B2 Data Receive (P0B2RR)			
0x0314	Port1 B2 Data Receive (P1B2RR)			
0x0318	Port2 B2 Data Receive (P2B2RR)			
0x031C	Port3 B2 Data Receive (P3B2RR)			
0x0320	Port0 D Data Receive (P0DRR)	Port1 D Data Receive (P1DRR)	Port2 D Data Receive (P2DRR)	Port3 D Data Receive (P3DRR)
0x0328	Port0 B1 Data Transmit (P0B1TR)			
0x032C	Port1 B1 Data Transmit (P1B1TR)			
0x0330	Port2 B1 Data Transmit (P2B1TR)			
0x0334	Port3 B1 Data Transmit (P3B1TR)			
0x0338	Port0 B2 Data Transmit (P0B2TR)			
0x033C	Port1 B2 Data Transmit (P1B2TR)			

13.5.20 D-Channel Request Register (PDRQR)

All bits in this read/write register are cleared on hardware or software reset.

The PDRQR register contains D-channel control bits for all four ports on the MCF5272.

	15 12	11	10	9	8	7 2	1 0
Field	—	SHDD(1)	DCNTI(1)	SHDD(0)	DCNTI(0)	—	DRQ
Reset	0000_0000_0000_0000						
R/W	Read/Write						
Addr	MBAR + 0x392						

Figure 13-32. D-Channel Request Registers (PDRQR)

Table 13-15. PDRQR Field Descriptions

Bits	Name	Description
15–12	—	Reserved, should be cleared.
11, 9	SHDD	D-channel shift direction. 0 D-channel data is msb first. The first bit received is assumed to be the most significant bit and is loaded into the msb position of the D-channel receive register for the respective port. SHDD(1) configures the shift direction for ports 1, 2 and 3, SHDD(0) configures the shift direction for port 0. 1 D-channel data is lsb first for the D channel. The first bit received is assumed to be the least significant bit and is loaded into the lsb position of the D-channel receive register for the respective port.
10, 8	DCNTI	D-channel control ignore. Allows the D-Channel contention function to be ignored. 00 contention active on both ports 01 ignore contention on port 0 10 ignore contention on port 1 11 ignore contention on both ports
7–2	—	Reserved, should be cleared.
1–0	DRQ	The value written to these bits is driven onto the DREQ pins associated with port 0 and port 1. When set, a logic high, 1, is driven on to the corresponding pin.

Two of Freescale’s MC145574 S/T transceivers are shown connected to ports 0 and 1. The frame sync control signal FSC0 is connected to S/T transceiver one, while FSC1 is connected to transceiver two.

Figure 13-42 shows an example of the IDL bus timing relationship of the S/T transceivers when in standard IDL2 8-bit mode with a common frame sync.

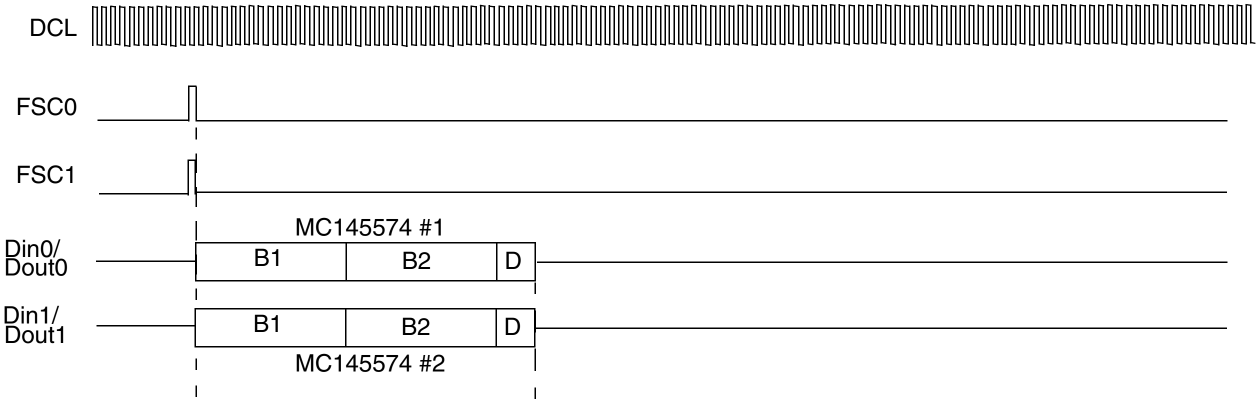


Figure 13-42. Standard IDL2 8-Bit Mode

15.3.5 Timer Event Registers (TER0–TER3)

TERs are used to report events recognized by the timer. On recognition of an event, the timer sets the appropriate TER*n* bit, regardless of the corresponding interrupt enable bits (ORI and CE) in the TMR*n*. Writing a 1 to a bit clears it; writing 0 has no effect. Both bits must be cleared before the timer can negate the request to the interrupt controller. Both bits may be cleared simultaneously.

	15	2	1	0
Field	—		REF	CAP
Reset	0000_0000_0000_0000			
R/W	Read/Write			
Addr	MBAR + 0x210 (TER0); 0x230 (TER1); 0x250 (TER2); 0x270 (TER3)			

Figure 15-6. Timer Event Registers (TER0–TER3)

Table 15-2 describes TER*n* fields.

Table 15-2. TER*n* Field Descriptions

Bits	Name	Description
15–2	—	Reserved, should be cleared.
1	REF	Output reference event. 0 The counter has not reached the TRR value 1 The counter reached the TRR value. TMR[ORI] is used to enable the interrupt request caused by this event. Write a 1 to this bit to clear the event condition.
0	CAP	Capture event. 0 The counter value has not been latched into the TCAP. 1 The counter value is latched in the TCAP. TMR[CE] is used to enable capture and the interrupt request caused by this event. Write a 1 to this bit to clear the event condition.

16.3.15 UART Fractional Precision Divider Control Registers (UFPD*n*)

The UFPD*n* registers allow greater accuracy when deriving a transmitter/receiver clock source from CLKIN. The use of the UFPD*n* registers is optional; if the contents are left in the reset state, code written for other ColdFire devices containing UART modules will not be affected by the addition of these registers. The contents of these registers allow the frequency to be divided by a factor of up to 16. When autobaud is used, these registers are updated automatically to reflect the clock rate being used. Host software can write to these registers to make fine adjustments to the clock rate. See Section 16.5.1.2, “Calculating Baud Rates,” for an example of UFPD*n* programming.

	7	4	3	0
Field	—		FD	
Reset	0000_0000			
R/W	R/W			
Address	MBAR + 0x130 (UFPD0), 0x170 (UFPD1)			

Figure 16-18. UART Fractional Precision Divider Control Registers (UFPD*n*)

Table 16-12 describes UFPD*n* fields.

Table 16-12. UFPD*n* Field Descriptions

Bits	Name	Description
7–4	—	Reserved, should be cleared.
3–0	FD	Fractional divider. The value of these bits, from 0 to 15, determine the scale factor by which the clocking source for the transmitter and/or receiver is scaled.

16.3.16 UART Input Port Registers (UIP*n*)

The UIP registers, Figure 16-19, show the current state of the $\overline{\text{CTS}}$ input.

	7	1	0
Field	—		CTS
Reset	1111_1111		
R/W	Read only		
Address	MBAR + 0x134 (UIP0), 0x174 (UIP1)		

Figure 16-19. UART Input Port Registers (UIP*n*)

Table 16-13 describes UIP*n* fields.

Table 16-13. UIP*n* Field Descriptions

Bits	Name	Description
7–1	—	Reserved, should be cleared.
0	CTS	$\overline{\text{CTS}}$ state.The $\overline{\text{CTS}}$ value is latched and reflects the state of the input pin when UIP*n* is read. Note: This bit has the same function and value as UIPCR*n*[RTS]. 0 The current state of the $\overline{\text{CTS}}$ input is logic 0. 1 The current state of the $\overline{\text{CTS}}$ input is logic 1.

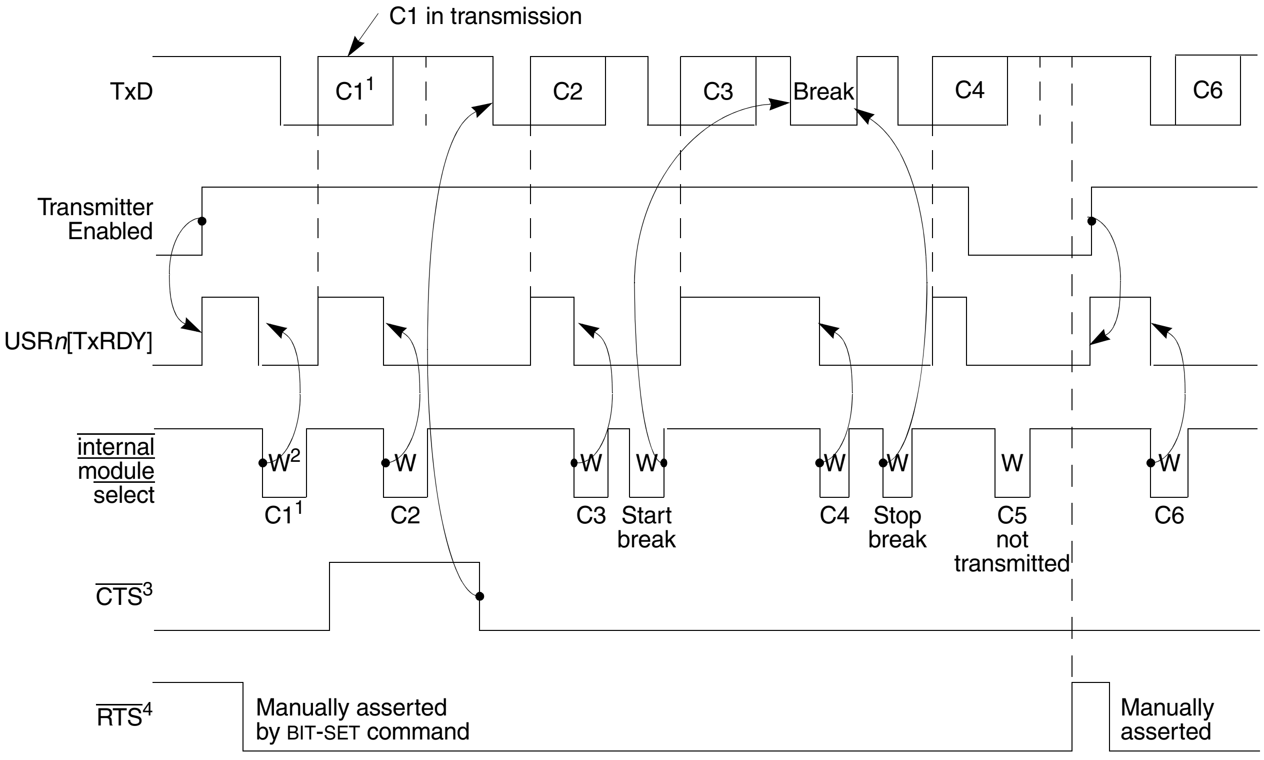
If the transmitter is reset through a software command, operation stops immediately (see Section 16.3.5, “UART Command Registers (UCRn)”). The transmitter is reenabled through the UCR*n* to resume operation after a disable or software reset.

If the clear-to-send operation is enabled, $\overline{\text{CTS}}$ must be asserted for the character to be transmitted. If $\overline{\text{CTS}}$ is negated in the middle of a transmission, the character in the shift register is sent and TxD remains in mark state until $\overline{\text{CTS}}$ is reasserted. If the transmitter is forced to send a continuous low condition by issuing a START BREAK command, the transmitter ignores the state of $\overline{\text{CTS}}$.

If the transmitter is programmed to automatically negate $\overline{\text{RTS}}$ when a message transmission completes, $\overline{\text{RTS}}$ must be asserted manually before a message is sent. In applications in which the transmitter is disabled after transmission is complete and $\overline{\text{RTS}}$ is appropriately programmed, $\overline{\text{RTS}}$ is negated one bit time after the character in the shift register is completely transmitted. The transmitter must be manually reenabled by reasserting $\overline{\text{RTS}}$ before the next message is to be sent.

The transmitter must be enabled prior to accepting a START BREAK command. If the transmitter is disabled while the BREAK is active, the BREAK is not terminated. The BREAK can only be terminated by using the STOP BREAK command.

Figure 16-25 shows the functional timing information for the transmitter.



[1] C*n* = transmit characters
[2] W = write
[3] UMR2*n*[TxCTS] = 1
[4] UMR2*n*[TxRTS] = 1

Figure 16-25. Transmitter Timing

Figure 16-31. UART Mode Programming Flowchart (Sheet 5 of 5)

Table 17-4. Port A Control Register Function Bits

Pin Number	PACNT[*xx*] = 00 (Function 0b00)	PACNT[*xx*] = 01 (Function 0b01)	PACNT[*xx*] = 10 (Function 0b10)	PACNT[*xx*] = 11 (Function 0b11)
D2	PA0	USB_TP	—	—
D1	PA1	USB_RP	—	—
E5	PA2	USB_RN	—	—
E4	PA3	USB_TN	—	—
E3	PA4	USB_Susp	—	—
E2	PA5	USB_TxEN	—	—
E1	PA6	USB_RxD	—	—
P1	PA7	QSPI_CS3	DOUT3	—
J2	PA8	FSC0/FSR0	—	—
J3	PA9	DGNT0	—	—
K5	PA10	DREQ0	—	—
L1	PA11	Reserved	QSPI_CS1	—
L2	PA12	DFSC2	—	—
L3	PA13	DFSC3	—	—
M2	PA14	DREQ1	—	—
M3	PA15	DGNT1[1]	—	—

[1] If this pin is programmed to function as $\overline{\text{INT6}}$, it is not available as a GPIO.

17.2.2 Port B Control Register (PBCNT)

PBCNT, shown in Figure 17-2, is used to configure the pins assigned to signals that are multiplexed with GPIO port B.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	PBCNT15		PBCNT14		PBCNT13		PBCNT12		PBCNT11		PBCNT10		PBCNT9		PBCNT8	
Reset	0000_0000_0000_0000															
R/W	Read/Write															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	PBCNT7		PBCNT6		PBCNT5		PBCNT4		PBCNT3		PBCNT2		PBCNT1		PBCNT0	
Reset	0000_0000_0000_0000															
R/W	Read/Write															
Addr	MBAR + 0x0088															

Figure 17-2. Port B Control Register (PBCNT)

Table 17-5 describes PBCNT fields. Table 17-6 provides the same information organized by function.

19.11.4 USB Transmit Data Negative (USB_TN/PA3)

USB mode: USB_TN is the inverted data transmit output.

Port A mode: This pin can also be configured as the PA3 I/O.

19.11.5 USB Suspend Driver (USB_SUSP/PA4)

USB mode: USB_SUSP is used to put the USB driver in suspend operation.

Port A mode: This pin can also be configured as the PA4 I/O.

19.11.6 USB Transmitter Output Enable ($\overline{\text{USB\_TxEN}}$/PA5)

USB mode: $\overline{\text{USB\_TxEN}}$ enables the transceiver to transmit data on the bus. It requires a 4.7-K¾ pullup resistor to ensure that the external USB Tx driver is off between the MCF5272 coming out of reset and initializing the port A pin configuration register.

Port A mode: This pin can also be configured as the PA5 I/O.

19.11.7 USB Rx Data Output (USB_RxD/PA6)

USB mode: USB_RxD is the receive data output from the differential receiver inputs USB_RN and USB_RP.

Port A mode: This pin can also be configured as the PA6 I/O.

19.11.8 USB_D+ and USB_D-

USB_D+ and USB_D- are the on-chip USB interface transceiver signals. When these signals are enabled, the USB module uses them to communicate to an external USB bus. When not used, each signal should be pulled to VDD using a 4.7-K¾ resistor.

19.11.9 USB_CLK

USB_CLK is used to connect an external 48-MHz oscillator to the USB module. When this pin is tied to GND or VDD, the USB module automatically uses the internal CPU clock. In this case the CLKIN must be 48 MHz if the system is to use the USB function.

19.11.10 $\overline{\text{INT1}}$/USB Wake-on-Ring (USB_WOR)

The USB module allows for $\overline{\text{INT1}}$ to generate the USB wake-on-ring signal to the USB host controller. This function is enabled by a control bit in the USB module. WOR is provided to allow the CPU and the USB interface to be woken up when in power down mode. This occurs when the USB controller detects a resume state at the USB inputs.

The interrupt output of an ISDN transceiver, such as the MC145574, can be connected to $\overline{\text{INT1}}$/USB_WOR. Before putting the device into sleep mode, the USB module’s wake on ring function

Table 21-1. JTAG Signals (continued)

Signal	Description
TDI/DSI	Test and debug data in. Input provided for loading serial data port shift registers (boundary-scan, bypass, and instruction registers). Shifting in of data depends on the state of the JTAG controller state machine and the instruction currently in the instruction register. Data is shifted in on the rising edge of TCK.
$\overline{\text{TRST}}$/ DSCLK	JTAG test reset. $\overline{\text{TRST}}$ asynchronously resets the JTAG TAP logic when low.
MTMOD	Freescale test mode select. Negating MTMOD enables JTAG mode; asserting it enables BDM mode.

21.3 TAP Controller

The TAP controller is a synchronous state machine that controls JTAG logic and interprets the sequence of logical values on TMS. The value adjacent to each arrow in the state machine in Figure 21-2 reflects the value of TMS sampled on the rising edge of TCK. For a description of the TAP controller states, refer to the IEEE 1149.1 document.

Figure 21-2. TAP Controller State Machine

23.3 AC Electrical Specifications

NOTE

AC timing specifications may be subject to change during ongoing qualification.

AC timing specifications assume maximum output load capacitance on all output pins including SDCLK. If this value is different, the input and output timing specifications would need to be adjusted to match the clock load.

AC timing specifications referenced to SDCLK assume SDRAM control register bit 3 is 0. After reset this bit is set.

23.3.1 Clock Input and Output Timing Specifications

Table 23-6 lists clock input and output timings.

Table 23-6. Clock Input and Output Timing Specifications

Name	Characteristic	0–66 MHz		Unit
		Min	Max	
	Frequency of operation	0	66.00	MHz
C1	CLKIN period (T) [1]	15	—	nS
C2 [2]	CLKIN fall time (from V_h = 2.4 V to V_l = 0.5 V)	—	2	nS
C3 [2]	CLKIN rise time (from V_l = 0.5 V to V_h = 2.4 V)	—	2	nS
C4	CLKIN duty cycle (measured at 1.5 V)	45	55	%
C4a [3]	CLKIN pulse-width high (measured at 1.5 V)	6.75	8.25	nS
C4b [3]	CLKIN pulse-width low (measured at 1.5 V)	6.75	8.25	nS

[1] The clock period is referred to as T in the electrical specifications. The time for T is always in nS. Timing specifications can be given in terms of T. For example, 2T+5 nS

[2] Specification values are not tested.

[3] Specification values listed are for maximum frequency of operation.

Clock input and output timings listed in Table 23-6 are shown in Figure 23-1.

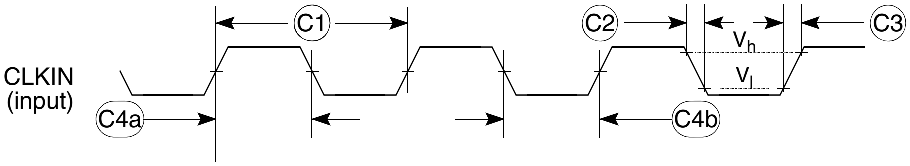


Figure 23-1. Clock Input Timing Diagram

Table A-2. CPU Space Registers Memory Map

CPU SPACE ADDRESS	NAME	Size	SYSTEM CONFIGURATION REGISTERS	Program Access	Debug Access
0x0002	(CACR)	32	Cache Control Register	MOVEC	RCREG, WCREG
0x0004	(ACR0)	32	Cache Access Control Register 0	MOVEC	RCREG, WCREG
0x0005	(ACR1)	32	Cache Access Control Register 1	MOVEC	RCREG, WCREG
0x008x	A7:A0	32	Address registers A7:A0	MOVE	RAREG, WAREG
0x008x	D7:D0	32	Data registers D7:D0	MOVE	RDREG, WDREG
0x0801	(VBR)	32	Vector Base Register	MOVEC	RCREG, WCREG
0x080E	CCR	8	Condition Code Register (Debug only)	MOVE to/from CCR	RCREG, WCREG
0x080F	PC	32	Program Counter (Debug only)		RCREG, WCREG
0x0C00	ROMBAR	32	ROM Base Address Register	MOVEC	RCREG, WCREG
0x0C04	RAMBAR	32	SRAM Base Address Register	MOVEC	RCREG, WCREG
0x0C0F	MBAR	32	Module Base Address Register	MOVEC	

NOTE

MBAR must only be written using the MOVEC instruction. Writing to address MBAR+0x0000 causes unpredictable device operation.

Table A-3. On-Chip Peripherals and Configuration Registers Memory Map

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x0000	Module Base Address Register, Read Only (MBAR)			
0x0004	System Configuration Register (SCR)		Reserved	
0x0006	Reserved		System Protection Register (SPR)	
0x0008	Power Management Register (PMR)			
0x000E	Reserved		Activate Low Power Register (ALPR)	
0x0010	Device Identification Register (DIR)			

Table A-4. Interrupt Control Register Memory Map

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x0020	Interrupt Control Register 1 (ICR1)			
0x0024	Interrupt Control Register 2 (ICR2)			
0x0028	Interrupt Control Register 3 (ICR3)			
0x002C	Interrupt Control Register 4 (ICR4)			
0x0030	Interrupt Source Register (ISR)			
0x0034	Programmable Interrupt Transition Register (PITR)			
0x0038	Programmable Interrupt Wakeup Register (PIWR)			
0x003F	Reserved			Programmable Interrupt Vector Register (PIVR)