# Microchip Technology - PIC32MZ1024EFK124-I/TL 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 | Obsolete |
| Core Processor | MIPS32® M-Class |
| Core Size | 32-Bit Single-Core |
| Speed | 200MHz |
| Connectivity | CANbus, EBI/EMI, Ethernet, I²C, PMP, SPI, SQI, UART/USART, USB OTG |
| Peripherals | Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT |
| Number of I/O | 97 |
| Program Memory Size | 1MB (1M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.1V ~ 3.6V |
| Data Converters | A/D 48x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 124-VFTLA Dual Rows, Exposed Pad |
| Supplier Device Package | 124-VTLA (9x9) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic32mz1024efk124-i-tl |

The System Bus arbitration scheme implements a non-programmable, Least Recently Serviced (LRS) priority, which provides Quality Of Service (QOS) for most initiators. However, some initiators can use Fixed High Priority (HIGH) arbitration to guarantee their access to data.

The arbitration scheme for the available initiators is shown in Table 4-5.

TABLE 4-5: INITIATOR ID AND QOS

| Name | ID | QOS |
|---|---|---|
| CPU | 1 | LRS(1) |
| CPU | 2 | HIGH(1,2) |
| DMA Read | 3 | LRS(1) |
| DMA Read | 4 | HIGH(1,2) |
| DMA Write | 5 | LRS(1) |
| DMA Write | 6 | HIGH(1,2) |
| USB | 7 | LRS |
| Ethernet Read | 8 | LRS |
| Ethernet Write | 9 | LRS |
| CAN1 | 10 | LRS |
| CAN2 | 11 | LRS |
| SQI1 | 12 | LRS |
| Flash Controller | 13 | HIGH(2) |
| Crypto | 14 | LRS |

Note 1: When accessing SRAM, the DMAPRI bit (CFGCON<25>) and the CPUPRI bit (CFGCON<24>) provide arbitration control for the DMA and CPU (when servicing an interrupt (i.e., EXL = 1)), respectively, by selecting the use of LRS or HIGH When using HIGH, the DMA and CPU get arbitration preference over all initiators using LRS.

2: Using HIGH arbitration can have serious negative effects on other initiators. Therefore, it is recommended to not enable this type of arbitration for an initiator that uses significant system bandwidth. HIGH arbitration is intended to be used for low bandwidth applications that require low latency, such as LCC graphics applications.

## 4.3 Permission Access and System Bus Registers

The System Bus on PIC32MZ EF family of microcontrollers provides access control capabilities for the transaction initiators on the System Bus.

The System Bus divides the entire memory space into fourteen target regions and permits access to each target by initiators via permission groups. Four Permission Groups (0 through 3) can be assigned to each initiator. Each permission group is independent of the others and can have exclusive or shared access to a region.

Using the CFGPG register (see Register 34-10 in Section 34.0 “Special Features” ), Boot firmware can assign a permission group to each initiator, which can make requests on the System Bus.

The available targets and their regions, as well as the associated control registers to assign protection, are described and listed in Table 4-6.

Register 4-2 through Register 4-10 are used for setting and controlling access permission groups and regions.

To change these registers, they must be unlocked in hardware. The register lock is controlled by the PGLOCK Configuration bit (CFGCON<11>). Setting PGLOCK prevents writes to the control registers; clearing PGLOCK allows writes.

To set or clear the PGLOCK bit, an unlock sequence must be executed. Refer to Section 42. “Oscillators with Enhanced PLL” in the *“PIC32 Family Reference Manual”* for details.

## 6.0 RESETS

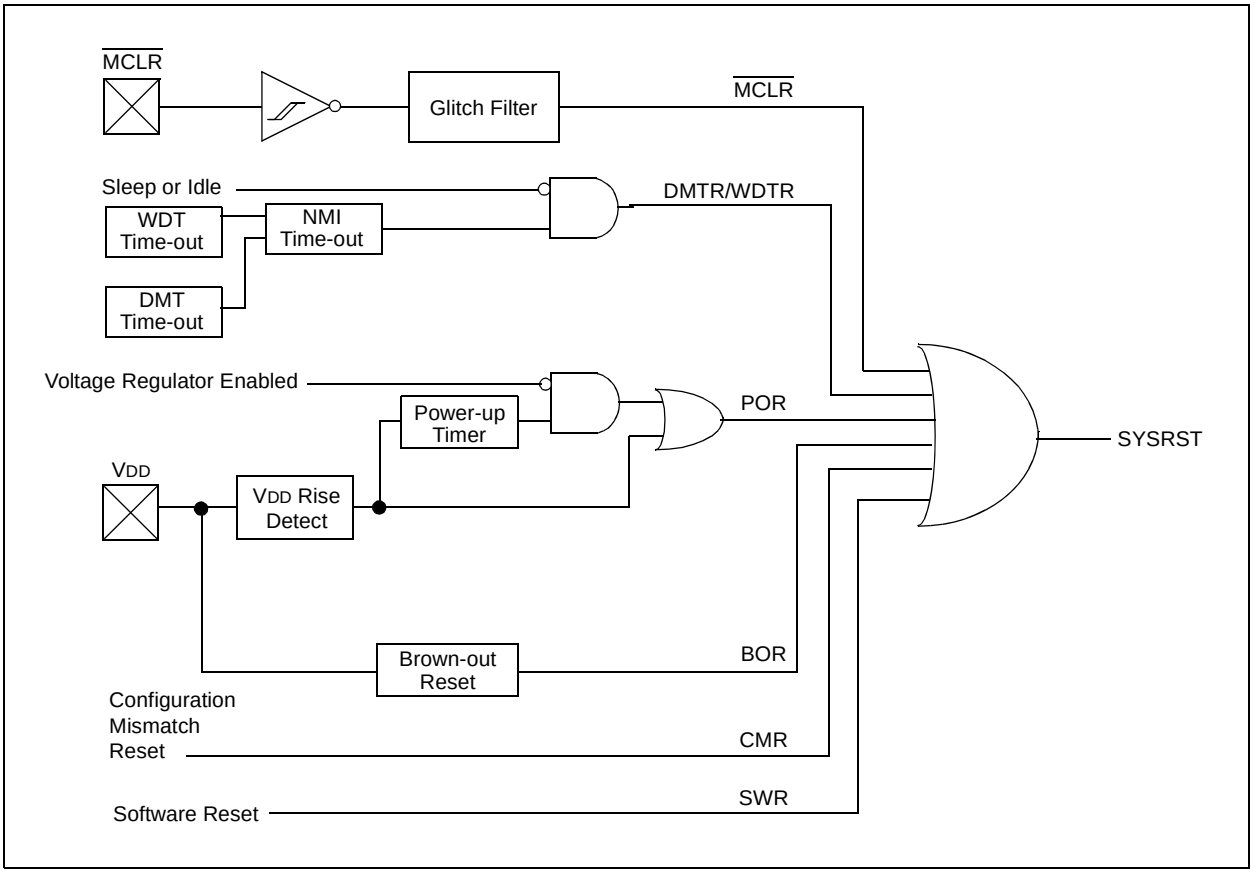
| Note: | This data sheet summarizes the features of the PIC32MZ EF family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to Section 7. "Resets" (DS60001118) in the *"PIC32 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com/PIC32). |
|---|---|

The Reset module combines all Reset sources and controls the device Master Reset signal, SYSRST. The device Reset sources are as follows:

- Power-on Reset (POR)
- Master Clear Reset pin ($\overline{MCLR}$)
- Software Reset (SWR)
- Watchdog Timer Reset (WDTR)
- Brown-out Reset (BOR)
- Configuration Mismatch Reset (CMR)
- Deadman Timer Reset (DMTR)

A simplified block diagram of the Reset module is illustrated in Figure 6-1.

FIGURE 6-1: SYSTEM RESET BLOCK DIAGRAM



REGISTER 7-3: INTSTAT: INTERRUPT STATUS REGISTER

| Bit Range | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 |
|---|---|---|---|---|---|---|---|---|
| 31:24 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 23:16 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 15:8 | U-0 | U-0 | U-0 | U-0 | U-0 | R-0 | R-0 | R-0 |
| | — | — | — | — | — | SRIPL<2:0>(1) | | |
| 7:0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| | SIRQ<7:0> | | | | | | | |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 31-11 Unimplemented: Read as '0'

bit 10-8 SRIPL<2:0>: Requested Priority Level bits for Single Vector Mode bits(1)
111-000 = The priority level of the latest interrupt presented to the CPU

bit 7-6 Unimplemented: Read as '0'

bit 7-0 SIRQ<7:0>: Last Interrupt Request Serviced Status bits
11111111-00000000 = The last interrupt request number serviced by the CPU

Note 1: This value should only be used when the interrupt controller is configured for Single Vector mode.

REGISTER 7-4: IPTMR: INTERRUPT PROXIMITY TIMER REGISTER

| Bit Range | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 |
|---|---|---|---|---|---|---|---|---|
| 31:24 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | IPTMR<31:24> | | | | | | | |
| 23:16 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | IPTMR<23:16> | | | | | | | |
| 15:8 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | IPTMR<15:8> | | | | | | | |
| 7:0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | IPTMR<7:0> | | | | | | | |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 31-0 IPTMR<31:0>: Interrupt Proximity Timer Reload bits
Used by the Interrupt Proximity Timer as a reload value when the Interrupt Proximity timer is triggered by an interrupt event.

## 9.0 PREFETCH MODULE

| Note: | This data sheet summarizes the features of the PIC32MZ EF family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to Section 41. "Prefetch Module for Devices with L1 CPU Cache" (DS60001183) in the *"PIC32 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com/PIC32). |
|---|---|

The Prefetch module is a performance enhancing module that is included in the PIC32MZ EF family of devices. When running at high-clock rates, Wait states must be inserted into Program Flash Memory (PFM) read transactions to meet the access time of the PFM. Wait states can be hidden to the core by prefetching and storing instructions in a temporary holding area that the CPU can access quickly. Although the data path to the CPU is 32 bits wide, the data path to the PFM is 128 bits wide. This wide data path provides the same bandwidth to the CPU as a 32-bit path running at four times the frequency.

The Prefetch module holds a subset of PFM in temporary holding spaces known as lines. Each line contains a tag and data field. Normally, the lines hold a copy of what is currently in memory to make instructions or data available to the CPU without Flash Wait states.

The following are key features of the Prefetch module:

- 4x16 byte fully-associative lines
- One line for CPU instructions
- One line for CPU data
- Two lines for peripheral data
- 16-byte parallel memory fetch
- Configurable predictive prefetch
- Error detection and correction

A simplified block diagram of the Prefetch module is shown in Figure 9-1.

FIGURE 9-1: PREFETCH MODULE BLOCK DIAGRAM

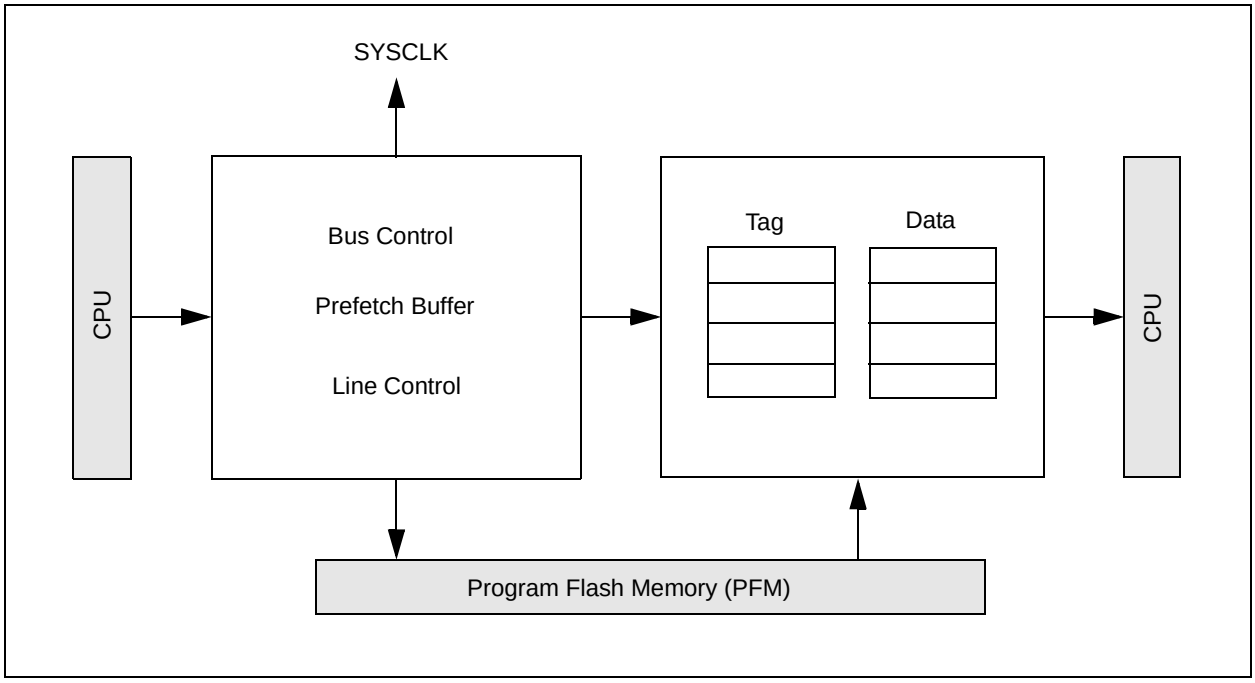


TABLE 11-1: USB REGISTER MAP 1 (CONTINUED)

| Virtual Address (BF8E_#) | Register Name | Bit Range | Bits | | | | | | | | | | | | | | | | All Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 31/15 | 30/14 | 29/13 | 28/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 | |
| 3128 | USB E2CSR2 | 31:16 | Indexed by the same bits in USBIE2CSR2 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 312C | USB E2CSR3 | 31:16 | Indexed by the same bits in USBIE2CSR3 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3130 | USB E3CSR0 | 31:16 | Indexed by the same bits in USBIE3CSR0 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3134 | USB E3CSR1 | 31:16 | Indexed by the same bits in USBIE3CSR1 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3138 | USB E3CSR2 | 31:16 | Indexed by the same bits in USBIE3CSR2 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 313C | USB E3CSR3 | 31:16 | Indexed by the same bits in USBIE3CSR3 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3140 | USB E4CSR0 | 31:16 | Indexed by the same bits in USBIE4CSR0 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3144 | USB E4CSR1 | 31:16 | Indexed by the same bits in USBIE4CSR1 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3148 | USB E4CSR2 | 31:16 | Indexed by the same bits in USBIE4CSR2 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 314C | USB E4CSR3 | 31:16 | Indexed by the same bits in USBIE4CSR3 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3150 | USB E5CSR0 | 31:16 | Indexed by the same bits in USBIE5CSR0 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3154 | USB E5CSR1 | 31:16 | Indexed by the same bits in USBIE5CSR1 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3158 | USB E5CSR2 | 31:16 | Indexed by the same bits in USBIE5CSR2 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 315C | USB E5CSR3 | 31:16 | Indexed by the same bits in USBIE5CSR3 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3160 | USB E6CSR0 | 31:16 | Indexed by the same bits in USBIE6CSR0 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3164 | USB E6CSR1 | 31:16 | Indexed by the same bits in USBIE6CSR1 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 3168 | USB E6CSR2 | 31:16 | Indexed by the same bits in USBIE6CSR2 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 316C | USB E6CSR3 | 31:16 | Indexed by the same bits in USBIE6CSR3 | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: Device mode.
2: Host mode.
3: Definition for Endpoint 0 (ENDPOINT<3:0> (USBCSR<19:16>) = 0).
4: Definition for Endpoints 1-7 (ENDPOINT<3:0> (USBCSR<19:16>) = 1 through 7).

REGISTER 11-28: USBLPMR1: USB LINK POWER MANAGEMENT CONTROL REGISTER 1 (CONTINUED)

bit 16 LPMXMT: LPM Transition to the L1 State bit

When in *Device mode*:

1 = USB module will transition to the L1 state upon the receipt of the next LPM transaction. LPMEN must be set to '0b11 . Both LPMXMT and LPMEN must be set in the same cycle.

0 = Maintain current state

When LPMXMT and LPMEN are set, the USB module can respond in the following ways:

- If no data is pending (all TX FIFOs are empty), the USB module will respond with an ACK. The bit will self clear and a software interrupt will be generated.
- If data is pending (data resides in at least one TX FIFO), the USB module will respond with a NYET. In this case, the bit will not self clear however a software interrupt will be generated.

When in *Host mode*:

1 = USB module will transmit an LPM transaction. This bit is self clearing, and will be immediately cleared upon receipt of any Token or three time-outs have occurred.

0 = Maintain current state

bit 15-12 ENDPOINT<3:0>: LPM Token Packet Endpoint bits

This is the endpoint in the token packet of the LPM transaction.

bit 11-9 Unimplemented: Read as '0'

bit 8 RMTWAK: Remote Wake-up Enable bit

This bit is applied on a temporary basis only and is only applied to the current suspend state.

1 = Remote wake-up is enabled

0 = Remote wake-up is disabled

bit 7-4 HIRD<3:0>: Host Initiated Resume Duration bits

The minimum time the host will drive resume on the bus. The value in this register corresponds to an actual resume time of:

Resume Time = 50 µs + HIRD * 75 µs. The resulting range is 50 µs to 1200 µs.

bit 3-0 LNKSTATE<3:0>: Link State bits

This value is provided by the host to the peripheral to indicate what state the peripheral must transition to after the receipt and acceptance of a LPM transaction. The only valid value for this register is '1' for Sleep State (L1). All other values are reserved.

TABLE 12-15: PORTG REGISTER MAP FOR 100- PIN, 124-PIN, AND 144-PIN DEVICES ONLY

| Virtual Address (BF86_#) | Register Name(1) | Bit Range | Bits | | | | | | | | | | | | | | | | All Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 31/15 | 30/14 | 29/13 | 28/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 | |
| 0600 | ANSELG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | ANSG15 | — | — | — | — | — | ANSG9 | ANSG8 | ANSG7 | ANSG6 | — | — | — | — | — | — | 83C0 |
| 0610 | TRISG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | TRISG15 | TRISG14 | TRISG13 | TRISG12 | — | — | TRISG9 | TRISG8 | TRISG7 | TRISG6 | — | — | — | — | TRISG1 | TRISG0 | F3C3 |
| 0620 | PORTG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | RG15 | RG14 | RG13 | RG12 | — | — | RG9 | RG8 | RG7 | RG6 | — | — | — | — | RG1 | RG0 | xxxx |
| 0630 | LATG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | LATG15 | LATG14 | LATG13 | LATG12 | — | — | LATG9 | LATG8 | LATG7 | LATG6 | — | — | — | — | LATG1 | LATG0 | xxxx |
| 0640 | ODCG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | ODCG15 | ODCG14 | ODCG13 | ODCG12 | — | — | ODCG9 | ODCG8 | ODCG7 | ODCG6 | — | — | — | — | ODCG1 | ODCG0 | 0000 |
| 0650 | CNPUG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | CNPUG15 | CNPUG14 | CNPUG13 | CNPUG12 | — | — | CNPUG9 | CNPUG8 | CNPUG7 | CNPUG6 | — | — | — | — | CNPUG1 | CNPUG0 | 0000 |
| 0660 | CNPDG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | CNPDG15 | CNPDG14 | CNPDG13 | CNPDG12 | — | — | CNPDG9 | CNPDG8 | CNPDG7 | CNPDG6 | — | — | — | — | CNPDG1 | CNPDG0 | 0000 |
| 0670 | CNCONG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | ON | — | — | — | EDGE DETECT | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| 0680 | CNENG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | CNENG15 | CNENG14 | CNENG13 | CNENG12 | — | — | CNENG9 | CNENG8 | CNENG7 | CNENG6 | — | — | — | — | CNENG1 | CNENG0 | 0000 |
| 0690 | CNSTATG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | CN STATG15 | CN STATG14 | CN STATG13 | CN STATG12 | — | — | CN STATG9 | CN STATG8 | CN STATG7 | CN STATG6 | — | — | — | — | CN STATG1 | CN STATG0 | 0000 |
| 06A0 | CNNEG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | CNNEG15 | CNNEG14 | CNNEG13 | CNNEG12 | — | — | CNNEG9 | CNNEG8 | CNNEG7 | CNNEG6 | — | — | — | — | CNNEG1 | CNNEG0 | 0000 |
| 06B0 | CNFG | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | CNFG15 | CNFG14 | CNFG13 | CNFG12 | — | — | CNFG9 | CNFG8 | CNFG7 | CNFG6 | — | — | — | — | CNFG1 | CNFG0 | 0000 |
| 06C0 | SRCON0G | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | — | SR0G14 | SR0G13 | SR0G12 | — | — | SR0G9 | — | — | SR0G6 | — | — | — | — | — | — | 0000 |
| 06D0 | SRCON1G | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | — | SR1G14 | SR1G13 | SR1G12 | — | — | SR1G9 | — | — | SR1G6 | — | — | — | — | — | — | 0000 |

Legend: x = Unknown value on Reset; — = Unimplemented, read as '0'; Reset values are shown in hexadecimal.

Note 1: All registers in this table have corresponding CLR, SET and INV registers at its virtual address, plus an offset of 0x4, 0x8, and 0xC, respectively. See Section 12.3 "CLR, SET, and INV Registers" for more information.

# 19.0 SERIAL PERIPHERAL INTERFACE (SPI) AND INTER-IC SOUND ($I^2S$)

| Note: | This data sheet summarizes the features of the PIC32MZ EF family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to Section 23. "Serial Peripheral Interface (SPI)" (DS60001106) in the *"PIC32 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com/PIC32). |
|---|---|

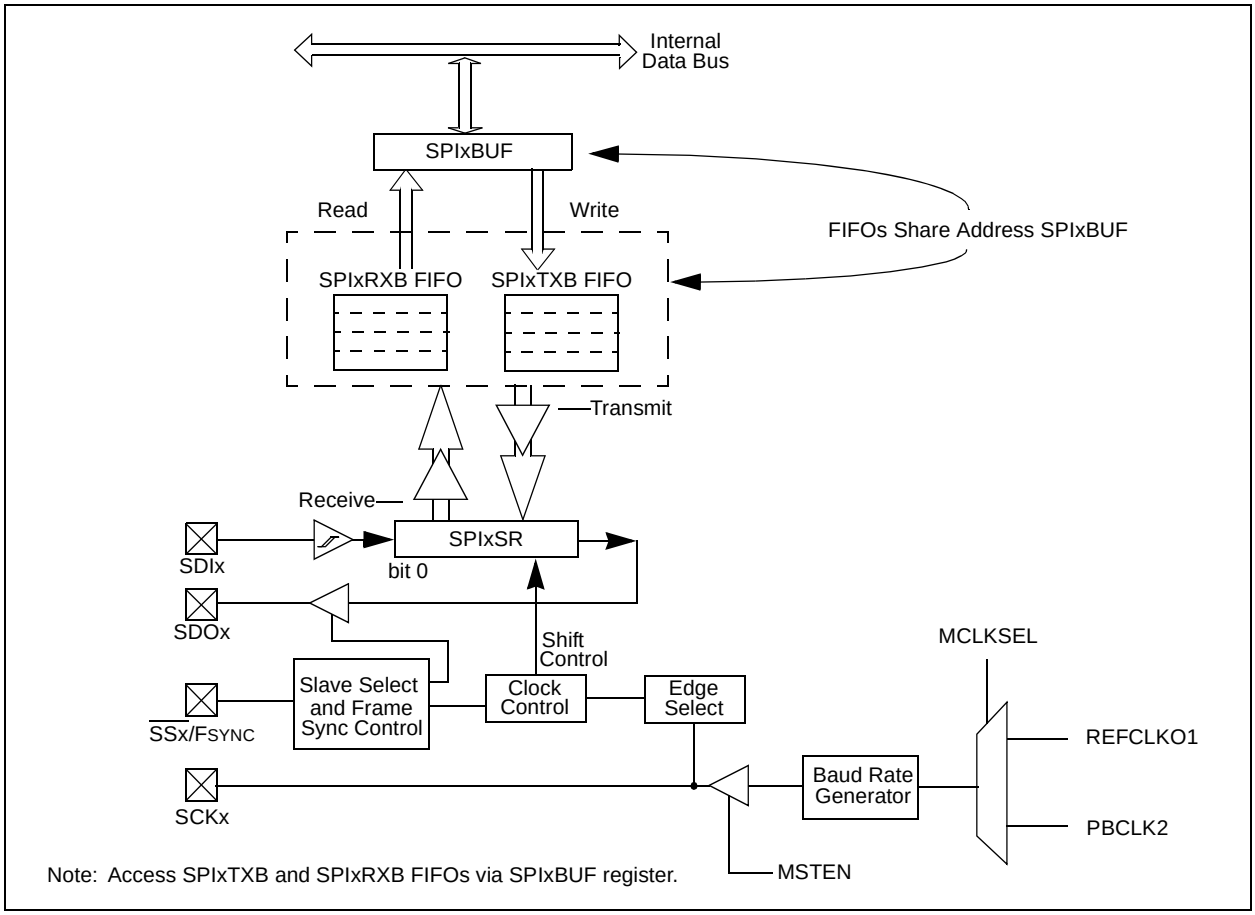
The SPI/$I^2S$ module is a synchronous serial interface that is useful for communicating with external peripherals and other microcontroller devices, as well as digital audio devices. These peripheral devices may be Serial EEPROMs, Shift registers, display drivers, Analog-to-Digital Converters, and so on.

The SPI/$I^2S$ module is compatible with Motorola® SPI and SIOP interfaces.

The following are key features of the SPI module:

- Master and Slave modes support
- Four different clock formats
- Enhanced Framed SPI protocol support
- User-configurable 8-bit, 16-bit and 32-bit data width
- Separate SPI FIFO buffers for receive and transmit
  - FIFO buffers act as 4/8/16-level deep FIFOs based on 32/16/8-bit data width
- Programmable interrupt event on every 8-bit, 16-bit and 32-bit data transfer
- Operation during Sleep and Idle modes
- Audio Codec Support:
  - $I^2S$ protocol
  - Left-justified
  - Right-justified
  - PCM

FIGURE 19-1: SPI/$I^2S$ MODULE BLOCK DIAGRAM



## 23.1 PMP Control Registers

TABLE 23-1: PARALLEL MASTER PORT REGISTER MAP

| Virtual Address (BF82_#) | Register Name(1) | Bit Range | Bits | | | | | | | | | | | | | | | | All Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 31/15 | 30/14 | 29/13 | 28/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 | |
| E000 | PMCON | 31:16 | — | — | — | — | — | — | — | — | RDSTART | — | — | — | — | — | DUALBUF | — | 0000 |
| | | 15:0 | ON | — | SIDL | ADRMUX<1:0> | | PMPTTL | PTWREN | PTRDEN | CSF<1:0> | | ALP | CS2P | CS1P | — | WRSP | RDSP | 0000 |
| E010 | PMMODE | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | BUSY | IRQM<1:0> | | INCM<1:0> | | MODE16 | MODE<1:0> | | WAITB<1:0> | | WAITM<3:0> | | | | WAITE<1:0> | | 0000 |
| E020 | PMADDR | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | CS2 | CS1 | ADDR<13:0> | | | | | | | | | | | | | | 0000 |
| | | | ADDR15 | ADDR14 | | | | | | | | | | | | | | | 0000 |
| E030 | PMDOUT | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | DATAOUT<15:0> | | | | | | | | | | | | | | | | 0000 |
| E040 | PMDIN | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | DATAIN<15:0> | | | | | | | | | | | | | | | | 0000 |
| E050 | PMAEN | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | PTEN<15:0> | | | | | | | | | | | | | | | | 0000 |
| E060 | PMSTAT | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | IBF | IBOV | — | — | IB3F | IB2F | IB1F | IB0F | OBE | OBUF | — | — | OB3E | OB2E | OB1E | OB0E | 008F |
| E070 | PMWADDR | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | WCS2 | WCS1 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | | WADDR15 | WADDR14 | WADDR<13:0> | | | | | | | | | | | | | | 0000 |
| E080 | PMRADDR | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | RCS2 | RCS1 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | | RADDR15 | RADDR14 | RADDR<13:0> | | | | | | | | | | | | | | 0000 |
| E090 | PMRDIN | 31:16 | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | 15:0 | RDATAIN<15:0> | | | | | | | | | | | | | | | 0000 |

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: All registers in this table have corresponding CLR, SET and INV registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See Section 12.3 "CLR, SET, and INV Registers" for more information.

REGISTER 23-9: PMRADDR: PARALLEL PORT READ ADDRESS REGISTER

| Bit Range | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 |
|---|---|---|---|---|---|---|---|---|
| 31:24 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 23:16 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 15:8 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | RCS2(1) | RCS1(3) | RADDR<13:8> | | | | | |
| | RADDR15(2) | RADDR14(4) | | | | | | |
| 7:0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | RADDR<7:0> | | | | | | | |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 31-16 Unimplemented: Read as '0'

bit 15 RCS2: Chip Select 2 bit(1)
1 = Chip Select 2 is active
0 = Chip Select 2 is inactive (RADDR15 function is selected)

bit 15 RADDR<15>: Target Address bit 15(2)

bit 14 RCS1: Chip Select 1 bit(3)
1 = Chip Select 1 is active
0 = Chip Select 1 is inactive (RADDR14 function is selected)

bit 14 RADDR<14>: Target Address bit 14(4)

bit 13-0 RADDR<13:0>: Address bits

Note 1: When the CSF<1:0> bits (PMCON<7:6>) = 10 or 01 .
2: When the CSF<1:0> bits (PMCON<7:6>) = 00.
3: When the CSF<1:0> bits (PMCON<7:6>) = 10.
4: When the CSF<1:0> bits (PMCON<7:6>) = 00 or 01 .

Note: This register is only used when the DUALBUF bit (PMCON<17>) is set to '1'.

REGISTER 26-8: CEPOLLCON: CRYPTO ENGINE POLL CONTROL REGISTER

| Bit Range | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 |
|---|---|---|---|---|---|---|---|---|
| 31:24 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 23:16 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 15:8 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | BDPPLCON<15:8> | | | | | | | |
| 7:0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | BDPPLCON<7:0> | | | | | | | |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 31-16 Unimplemented: Read as '0'

bit 15 BDPPLCON<15:0>: Buffer Descriptor Processor Poll Control bits

These bits determine the number of SYSCLK cycles that the Crypto DMA would wait before refetching the descriptor control word if the Buffer Descriptor fetched was disabled.

## 26.3 Security Association Structure

Table 26-4 shows the Security Association Structure. The Crypto Engine uses the Security Association to determine the settings for processing a Buffer Descriptor Processor. The Security Association contains:

- Which algorithm to use
- Whether to use engines in parallel (for both authentication and encryption/decryption)
- The size of the key
- Authentication key
- Encryption/decryption key
- Authentication Initialization Vector (IV)
- Encryption IV

TABLE 26-4: CRYPTO ENGINE SECURITY ASSOCIATION STRUCTURE

| Name | | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 |
|---|---|---|---|---|---|---|---|---|---|
| SA_CTRL | 31:24 | — | — | VERIFY | — | NO_RX | OR_EN | ICVONLY | IRFLAG |
| | 23:16 | LNC | LOADIV | FB | FLAGS | — | — | — | ALGO<6> |
| | 15:8 | ALGO<5:0> | | | | | | ENCTYPE | KEYSIZE<1> |
| | 7:0 | KEYSIZE<0> | MULTITASK<2:0> | | | CRYPTOALGO<3:0> | | | |
| SA_AUTHKEY1 | 31:24 | AUTHKEY<31:24> | | | | | | | |
| | 23:16 | AUTHKEY<23:16> | | | | | | | |
| | 15:8 | AUTHKEY<15:8> | | | | | | | |
| | 7:0 | AUTHKEY<7:0> | | | | | | | |
| SA_AUTHKEY2 | 31:24 | AUTHKEY<31:24> | | | | | | | |
| | 23:16 | AUTHKEY<23:16> | | | | | | | |
| | 15:8 | AUTHKEY<15:8> | | | | | | | |
| | 7:0 | AUTHKEY<7:0> | | | | | | | |
| SA_AUTHKEY3 | 31:24 | AUTHKEY<31:24> | | | | | | | |
| | 23:16 | AUTHKEY<23:16> | | | | | | | |
| | 15:8 | AUTHKEY<15:8> | | | | | | | |
| | 7:0 | AUTHKEY<7:0> | | | | | | | |
| SA_AUTHKEY4 | 31:24 | AUTHKEY<31:24> | | | | | | | |
| | 23:16 | AUTHKEY<23:16> | | | | | | | |
| | 15:8 | AUTHKEY<15:8> | | | | | | | |
| | 7:0 | AUTHKEY<7:0> | | | | | | | |
| SA_AUTHKEY5 | 31:24 | AUTHKEY<31:24> | | | | | | | |
| | 23:16 | AUTHKEY<23:16> | | | | | | | |
| | 15:8 | AUTHKEY<15:8> | | | | | | | |
| | 7:0 | AUTHKEY<7:0> | | | | | | | |
| SA_AUTHKEY6 | 31:24 | AUTHKEY<31:24> | | | | | | | |
| | 23:16 | AUTHKEY<23:16> | | | | | | | |
| | 15:8 | AUTHKEY<15:8> | | | | | | | |
| | 7:0 | AUTHKEY<7:0> | | | | | | | |
| SA_AUTHKEY7 | 31:24 | AUTHKEY<31:24> | | | | | | | |
| | 23:16 | AUTHKEY<23:16> | | | | | | | |
| | 15:8 | AUTHKEY<15:8> | | | | | | | |
| | 7:0 | AUTHKEY<7:0> | | | | | | | |
| SA_AUTHKEY8 | 31:24 | AUTHKEY<31:24> | | | | | | | |
| | 23:16 | AUTHKEY<23:16> | | | | | | | |
| | 15:8 | AUTHKEY<15:8> | | | | | | | |
| | 7:0 | AUTHKEY<7:0> | | | | | | | |
| SA_ENCKEY1 | 31:24 | ENCKEY<31:24> | | | | | | | |
| | 23:16 | ENCKEY<23:16> | | | | | | | |
| | 15:8 | ENCKEY<15:8> | | | | | | | |
| | 7:0 | ENCKEY<7:0> | | | | | | | |
| SA_ENCKEY2 | 31:24 | ENCKEY<31:24> | | | | | | | |
| | 23:16 | ENCKEY<23:16> | | | | | | | |

## 27.1 RNG Control Registers

TABLE 27-2: RANDOM NUMBER GENERATOR (RNG) REGISTER MAP

| Virtual Address (BF8E_#) | Register Name | Bit Range | Bits | | | | | | | | | | | | | | | | All Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 31/15 | 30/14 | 29/13 | 28/12 | 27/11 | 26/10 | 25/9 | 24/8 | 23/7 | 22/6 | 21/5 | 20/4 | 19/3 | 18/2 | 17/1 | 16/0 | |
| 6000 | RNGVER | 31:16 | ID<15:0> | | | | | | | | | | | | | | | | xxxx |
| | | 15:0 | VERSION<7:0> | | | | | | | | REVISION<7:0> | | | | | | | | xxxx |
| 6004 | RNGCON | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | — | — | — | LOAD | TRNGMODE | CONT | PRNGEN | TRNGEN | PLEN<7:0> | | | | | | | | 0064 |
| 6008 | RNGPOLY1 | 31:16 | POLY<31:0> | | | | | | | | | | | | | | | | FFFF |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 600C | RNGPOLY2 | 31:16 | POLY<31:0> | | | | | | | | | | | | | | | | FFFF |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 6010 | RNGNUMGEN1 | 31:16 | RNG<31:0> | | | | | | | | | | | | | | | | FFFF |
| | | 15:0 | | | | | | | | | | | | | | | | | FFFF |
| 6014 | RNGNUMGEN2 | 31:16 | RNG<31:0> | | | | | | | | | | | | | | | | FFFF |
| | | 15:0 | | | | | | | | | | | | | | | | | FFFF |
| 6018 | RNGSEED1 | 31:16 | SEED<31:0> | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 601C | RNGSEED2 | 31:16 | SEED<31:0> | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | | | | | | | | | | | | | | | | | 0000 |
| 6020 | RNGCNT | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | — | — | — | — | — | — | — | — | — | RCNT<6:0> | | | | | | | 0000 |

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

REGISTER 28-18: ADCTRG2: AD C TRIGGER SOURCE 2 REGISTER

| Bit Range | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 |
|---|---|---|---|---|---|---|---|---|
| 31:24 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | — | — | — | TRGSRC7<4:0> | | | | |
| 23:16 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | — | — | — | TRGSRC6<4:0> | | | | |
| 15:8 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | — | — | — | TRGSRC5<4:0> | | | | |
| 7:0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | — | — | — | TRGSRC4<4:0> | | | | |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at POR | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

bit 31-29 Unimplemented: Read as ‘0’

bit 28-24 TRGSRC7<4:0>: Trigger Source for Conversion of Analog Input AN7 Select bits
11111 = Reserved
•
•
•
01101 = Reserved
01100 = Comparator 2 (COUT)
01011 = Comparator 1 (COUT)
01010 = OCMP5
01001 = OCMP3
01000 = OCMP1
00111 = TMR5 match
00110 = TMR3 match
00101 = TMR1 match
00100 = INT0 External interrupt
00011 = STRIG
00010 = Global level software trigger (GLSWTRG)
00001 = Global software edge Trigger (GSWTRG)
00000 = No Trigger

For STRIG, in addition to setting the trigger, it also requires programming of the STRGSRC<4:0> bits (ADCCON1<20:16>) to select the trigger source, and requires the appropriate CSS bits to be set in the ADCCSS*x* registers.

bit 23-21 Unimplemented: Read as ‘0’

bit 20-16 TRGSRC6<4:0>: Trigger Source for Conversion of Analog Input AN6 Select bits
See bits 28-24 for bit value definitions.

bit 15-13 Unimplemented: Read as ‘0’

bit 12-8 TRGSRC5<4:0>: Trigger Source for Conversion of Analog Input AN5 Select bits
See bits 28-24 for bit value definitions.

bit 7-5 Unimplemented: Read as ‘0’

bit 4-0 TRGSRC4<4:0>: Trigger Source for Conversion of Analog Input AN4 Select bits
See bits 28-24 for bit value definitions.

REGISTER 30-24: EMAC1CFG2: ETHERNET CONTROLLER MAC CONFIGURATION 2 REGISTER

| Bit Range | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 25/16/8/0 |
|---|---|---|---|---|---|---|---|---|
| 31:24 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 23:16 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 15:8 | U-0 | R/W-1 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 |
| | — | EXCESSDFR | BPNOBKOFF | NOBKOFF | — | — | LONGPRE | PUREPRE |
| 7:0 | R/W-1 | R/W-0 | R/W-1 | R/W-1 | R/W-0 | R/W-0 | R/W-1 | R/W-0 |
| | AUTOPAD(1,2) | VLANPAD(1,2) | PADENABLE(1,3) | CRCENABLE | DELAYCRC | HUGEFRM | LENGTHCK | FULLDPLX |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 31-15 Unimplemented: Read as '0'

bit 14 EXCESSDER: Excess Defer bit
- 1 = The MAC will defer to carrier indefinitely as per the Standard
- 0 = The MAC will abort when the excessive deferral limit is reached

bit 13 BPNOBKOFF: Backpressure/No Backoff bit
- 1 = The MAC after incidentally causing a collision during backpressure will immediately retransmit without backoff reducing the chance of further collisions and ensuring transmit packets get sent
- 0 = The MAC will not remove the backoff

bit 12 NOBKOFF: No Backoff bit
- 1 = Following a collision, the MAC will immediately retransmit rather than using the Binary Exponential Back-off algorithm as specified in the Standard
- 0 = Following a collision, the MAC will use the Binary Exponential Backoff algorithm

bit 11-10 Unimplemented: Read as '0'

bit 9 LONGPRE: Long Preamble Enforcement bit
- 1 = The MAC only allows receive packets which contain preamble fields less than 12 bytes in length
- 0 = The MAC allows any length preamble as per the Standard

bit 8 PUREPRE: Pure Preamble Enforcement bit
- 1 = The MAC will verify the content of the preamble to ensure it contains 0x55 and is error-free. A packet with errors in its preamble is discarded
- 0 = The MAC does not perform any preamble checking

bit 7 AUTOPAD: Automatic Detect Pad Enable bit(1,2)
- 1 = The MAC will automatically detect the type of frame, either tagged or untagged, by comparing the two octets following the source address with 0x8100 (VLAN Protocol ID) and pad accordingly
- 0 = The MAC does not perform automatic detection

Note 1: Table 30-6 provides a description of the pad function based on the configuration of this register.
2: This bit is ignored if the PADENABLE bit is cleared.
3: This bit is used in conjunction with the AUTOPAD and VLANPAD bits.

Note: Both 16-bit and 32-bit accesses are allowed to these registers (including the SET, CLR and INV registers). 8-bit accesses are not allowed and are ignored by the hardware

REGISTER 30-28: EMAC1MAXF: ETHERNET CONTROLLER MAC MAXIMUM FRAME LENGTH REGISTER

| Bit Range | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 |
|---|---|---|---|---|---|---|---|---|
| 31:24 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 23:16 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 15:8 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/W-1 |
| | MACMAXF<15:8>(1) | | | | | | | |
| 7:0 | R/W-1 | R/W-1 | R/W-1 | R/W-0 | R/W-1 | R/W-1 | R/W-1 | R/W-0 |
| | MACMAXF<7:0>(1) | | | | | | | |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 31-16 Unimplemented: Read as '0'

bit 15-0 MACMAXF<15:0>: Maximum Frame Length bits(1)

These bits reset to 0x05EE, which represents a maximum receive frame of 1518 octets. An untagged maximum size Ethernet frame is 1518 octets. A tagged frame adds four octets for a total of 1522 octets. If a shorter/longer maximum length restriction is desired, program this 16-bit field.

Note 1: If a proprietary header is allowed, this bit should be adjusted accordingly. For example, if 4-byte headers are prepended to frames, MACMAXF could be set to 1527 octets. This would allow the maximum VLAN tagged frame plus the 4-byte header.

| Note: | Both 16-bit and 32-bit accesses are allowed to these registers (including the SET, CLR and INV registers). 8-bit accesses are not allowed and are ignored by the hardware. |
|---|---|

REGISTER 34-4: DEVCFG1/ADEVCFG1: DEVICE CONFIGURATION WORD 1

| Bit Range | Bit 31/23/15/7 | Bit 30/22/14/6 | Bit 29/21/13/5 | Bit 28/20/12/4 | Bit 27/19/11/3 | Bit 26/18/10/2 | Bit 25/17/9/1 | Bit 24/16/8/0 |
|---|---|---|---|---|---|---|---|---|
| 31:24 | R/P | R/P | R/P | R/P | R/P | R/P | R/P | R/P |
| | FDMTEN | DMTCNT<4:0> | | | | | FWDTWINSZ<1:0> | |
| 23:16 | R/P | R/P | R/P | R/P | R/P | R/P | R/P | R/P |
| | FWDTEN | WINDIS | WDTSPGM | WDTPS<4:0> | | | | |
| 15:8 | R/P | R/P | r-1 | r-1 | r-1 | R/P | R/P | R/P |
| | FCKSM<1:0> | | — | — | — | OSCIOFNC | POSCMOD<1:0> | |
| 7:0 | R/P | R/P | R/P | R/P | R/P | R/P | R/P | R/P |
| | IESO | FSOSCEN | DMTINTV<2:0> | | | FNOSC<2:0> | | |

| Legend: | r = Reserved bit | P = Programmable bit | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 31 FDMTEN: Deadman Timer enable bit
1 = Deadman Timer is enabled and *cannot* be disabled by software
0 = Deadman Timer is disabled and *can* be enabled by software

bit 30-26 DMTCNT<4:0>: Deadman Timer Count Select bits
11111 = Reserved
•
•
•
11000 = Reserved
10111 = $2^{31}$ (2147483648)
10110 = $2^{30}$ (1073741824)
10101 = $2^{29}$ (536870912)
10100 = $2^{28}$ (268435456)
•
•
•
00001 = $2^{9}$ (512)
00000 = $2^{8}$ (256)

bit 25-24 FWDTWINSZ<1:0>: Watchdog Timer Window Size bits
11 = Window size is 25%
10 = Window size is 37.5%
01 = Window size is 50%
00 = Window size is 75%

bit 23 FWDTEN: Watchdog Timer Enable bit
1 = Watchdog Timer is enabled and cannot be disabled by software
0 = Watchdog Timer is not enabled; it can be enabled in software

bit 22 WINDIS: Watchdog Timer Window Enable bit
1 = Watchdog Timer is in non-Window mode
0 = Watchdog Timer is in Window mode

bit 21 WDTSPGM: Watchdog Timer Stop During Flash Programming bit
1 = Watchdog Timer stops during Flash programming
0 = Watchdog Timer runs during Flash programming (for read/execute while programming Flash applications)

TABLE 37-31: SPIx MODULE MASTER MODE (CKE = **1**) TIMING REQUIREMENTS

| AC CHARACTERISTICS | | | Standard Operating Conditions: 2.1V to 3.6V (unless otherwise stated)<br>Operating temperature -40°C dTA d+85°C for Industrial<br>-40°C dTA d+125°C for Extended | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics (1) | Min. | Typ. (2) | Max. | Units | Conditions |
| SP10 | TscL | SCKx Output Low Time (Note 3) | Tsck/2 | — | — | ns | — |
| SP11 | TscH | SCKx Output High Time (Note 3) | Tsck/2 | — | — | ns | — |
| SP15 | TscK | SPI Clock Speed (Note 5) | —<br>—<br>—<br>—<br>— | —<br>—<br>—<br>—<br>— | 25<br>50<br>25<br>50<br>25 | MHz<br>MHz<br>MHz<br>MHz<br>MHz | SPI1, SPI4 through SPI6<br>SPI2 on RPB3, RPB5<br>SPI2 on other I/O<br>SPI3 on RPB10, RPB9, RPF0<br>SPI3 on other I/O |
| SP20 | TscF | SCKx Output Fall Time (Note 4) | — | — | — | ns | See parameter DO32 |
| SP21 | TscR | SCKx Output Rise Time (Note 4) | — | — | — | ns | See parameter DO31 |
| SP30 | TdoF | SDOx Data Output Fall Time (Note 4) | — | — | — | ns | See parameter DO32 |
| SP31 | TdoR | SDOx Data Output Rise Time (Note 4) | — | — | — | ns | See parameter DO31 |
| SP35 | TscH2doV, TscL2doV | SDOx Data Output Valid after SCKx Edge | — | — | 7 | ns | VDD > 2.7V |
| | | | — | — | 10 | ns | VDD < 2.7V |
| SP36 | TdoV2sc, TdoV2scL | SDOx Data Output Setup to First SCKx Edge | — | — | 7 | ns | — |
| SP40 | TdiV2scH, TdiV2scL | Setup Time of SDIx Data Input to SCKx Edge | 7 | — | — | ns | VDD > 2.7V |
| | | | 10 | — | — | ns | VDD < 2.7V |
| SP41 | TscH2diL, TscL2diL | Hold Time of SDIx Data Input to SCKx Edge | 7 | — | — | ns | VDD > 2.7V |
| | | | 10 | — | — | ns | VDD < 2.7V |

Note 1: These parameters are characterized, but not tested in manufacturing.

2: Data in “Typical” column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

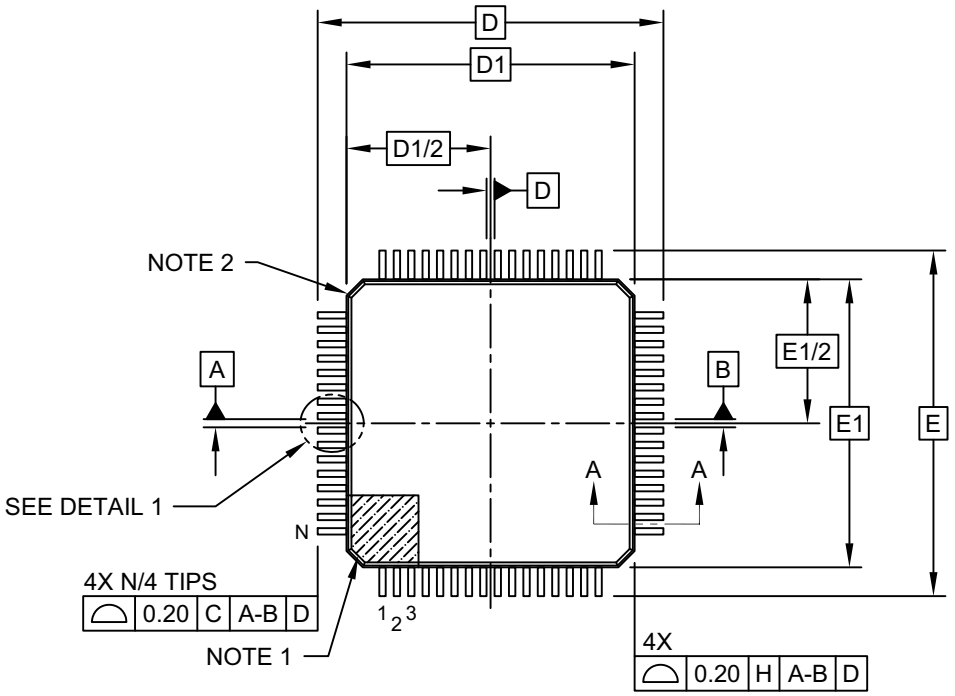
3: The minimum clock period for SCKx is 20 ns. Therefore, the clock generated in Master mode must not violate this specification.

4: Assumes 30 pF load on all SPIx pins.

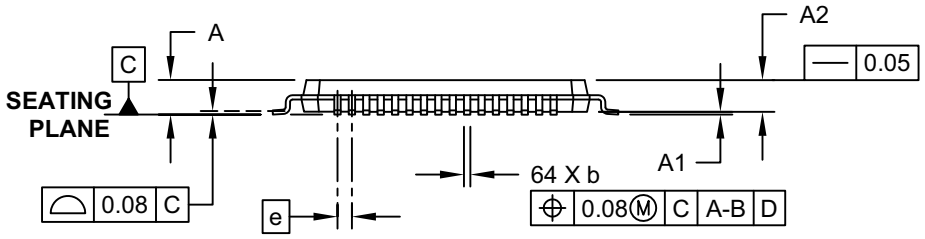
5: To achieve maximum data rate, VDD must be t 3.3V, the SMP bit (SPIxCON<9>) must be equal to ‘1’, and the operating temperature must be within the range of -40°C to +105°C.

**64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|




TOP VIEW




SIDE VIEW

Microchip Technology Drawing C04-085C Sheet 1 of 2