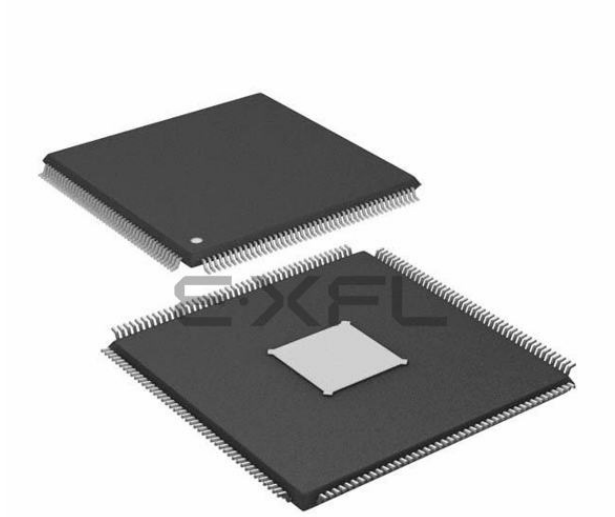
# Microchip Technology - PIC32MZ1025DAH176-I/2J Datasheet



**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 | MIPS32® microAptiv™ |
| Core Size | 32-Bit Single-Core |
| Speed | 200MHz |
| Connectivity | CANbus, EBI/EMI, Ethernet, I²C, IrDA, LINbus, PMP, SPI, SQI, UART/USART, USB OTG |
| Peripherals | Brown-out Detect/Reset, DMA, HLVD, I²S, POR, PWM, WDT |
| Number of I/O | 120 |
| Program Memory Size | 1MB (1M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.7V ~ 3.6V |
| Data Converters | A/D 45x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 176-LQFP Exposed Pad |
| Supplier Device Package | 176-LQFP (20x20) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic32mz1025dah176-i-2j |

**TABLE 1-4: OC1 THROUGH OC9 PINOUT I/O DESCRIPTIONS**

| Pin Name | Pin Number | | | Pin Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| | 169-pin LFBGA | 176-pin LQFP | 288-pin LFBGA | | | |
| **Output Compare** | | | | | | |
| OC1 | PPS | PPS | PPS | O | — | Output Compare Outputs 1-9 |
| OC2 | PPS | PPS | PPS | O | — | |
| OC3 | PPS | PPS | PPS | O | — | |
| OC4 | PPS | PPS | PPS | O | — | |
| OC5 | PPS | PPS | PPS | O | — | |
| OC6 | PPS | PPS | PPS | O | — | |
| OC7 | PPS | PPS | PPS | O | — | |
| OC8 | PPS | PPS | PPS | O | — | |
| OC9 | PPS | PPS | PPS | O | — | |
| OCFA | PPS | PPS | PPS | I | ST | Output Compare Fault A Input |
| OCFB | PPS | PPS | PPS | I | ST | Output Compare Fault B Input |

**Legend:** CMOS = CMOS-compatible input or output; Analog = Analog input; P = Power
ST = Schmitt Trigger input with CMOS levels; O = Output; I = Input
TTL = Transistor-transistor Logic input buffer; PPS = Peripheral Pin Select

**TABLE 1-5: EXTERNAL INTERRUPTS PINOUT I/O DESCRIPTIONS**

| Pin Name | Pin Number | | | Pin Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| | 169-pin LFBGA | 176-pin LQFP | 288-pin LFBGA | | | |
| **External Interrupts** | | | | | | |
| INT0 | C3 | 42 | A6 | I | ST | External Interrupt 0 |
| INT1 | PPS | PPS | PPS | I | ST | External Interrupt 1 |
| INT2 | PPS | PPS | PPS | I | ST | External Interrupt 2 |
| INT3 | PPS | PPS | PPS | I | ST | External Interrupt 3 |
| INT4 | PPS | PPS | PPS | I | ST | External Interrupt 4 |

**Legend:** CMOS = CMOS-compatible input or output; Analog = Analog input; P = Power
ST = Schmitt Trigger input with CMOS levels; O = Output; I = Input
TTL = Transistor-transistor Logic input buffer; PPS = Peripheral Pin Select

**TABLE 1-6: PORTA THROUGH PORTK PINOUT I/O DESCRIPTIONS**

| Pin Name | Pin Number | | | Pin Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| | 169-pin LFBGA | 176-pin LQFP | 288-pin LFBGA | | | |
| **PORTA** | | | | | | |
| RA0 | D2 | 53 | D4 | I/O | ST | PORTA is a bidirectional I/O port |
| RA1 | E11 | 160 | H16 | I/O | ST | |
| RA2 | L12 | 129 | U18 | I/O | ST | |
| RA3 | N12 | 128 | U17 | I/O | ST | |
| RA4 | L11 | 127 | V17 | I/O | ST | |
| RA5 | A11 | 174 | B18 | I/O | ST | |
| RA6 | E4 | 54 | E4 | I/O | ST | |
| RA7 | D1 | 55 | F4 | I/O | ST | |
| RA9 | B11 | 1 | A17 | I/O | ST | |
| RA10 | C10 | 2 | C15 | I/O | ST | |
| RA14 | M11 | 126 | T16 | I/O | ST | |
| RA15 | N11 | 125 | U16 | I/O | ST | |
| **PORTB** | | | | | | |
| RB0 | C12 | 169 | D18 | I/O | ST | PORTB is a bidirectional I/O port |
| RB1 | B9 | 11 | A14 | I/O | ST | |
| RB2 | A13 | 172 | C17 | I/O | ST | |
| RB3 | A10 | 8 | A16 | I/O | ST | |
| RB4 | A12 | 175 | B17 | I/O | ST | |
| RB5 | D11 | 167 | E17 | I/O | ST | |
| RB6 | D7 | 13 | B14 | I/O | ST | |
| RB7 | D12 | 170 | D17 | I/O | ST | |
| RB8 | A9 | 10 | A15 | I/O | ST | |
| RB9 | B12 | 173 | E16 | I/O | ST | |
| RB10 | C11 | 168 | E18 | I/O | ST | |
| RB11 | E7 | 12 | C14 | I/O | ST | |
| RB12 | B13 | 171 | C18 | I/O | ST | |
| RB13 | F7 | 9 | B15 | I/O | ST | |
| RB14 | E10 | 175 | J17 | I/O | ST | |
| RB15 | B10 | 7 | B16 | I/O | ST | |
| **PORTC** | | | | | | |
| RC1 | B7 | 17 | A12 | I/O | ST | PORTC is a bidirectional I/O port |
| RC2 | A8 | 14 | C13 | I/O | ST | |
| RC3 | A7 | 15 | B13 | I/O | ST | |
| RC4 | B8 | 16 | A13 | I/O | ST | |
| RC12 | E12 | 164 | G17 | I/O | ST | |
| RC13 | C13 | 162 | H17 | I | ST | |
| RC14 | D13 | 161 | H18 | I | ST | |
| RC15 | E13 | 163 | G18 | I/O | ST | |

**Legend:** CMOS = CMOS-compatible input or output; ST = Schmitt Trigger input with CMOS levels; TTL = Transistor-transistor Logic input buffer; Analog = Analog input; O = Output; PPS = Peripheral Pin Select; P = Power; I = Input

## 4.2 DDR2 SDRAM

Stacked DDR2 SDRAM memory devices support 32 MB of DDR2 SDRAM. Memory in these devices is organized as 4,194,304 x 4 banks x 16 bits. Refer to Figure 4-1 and Table 4-1 for the DDR2 SDRAM address ranges.

### 4.2.1 FEATURES

The DDR2 SDRAM includes the following features:

- Double Data Rate architecture: two data transfers per clock cycle
- CAS Latency: 3 and 4
- Burst Length: 8
- Bi-directional, differential data strobes (DDRUDQS, DDRLDQS and $\overline{\text{DDRUDQS}}$, $\overline{\text{DDRLDQS}}$) are transmitted / received with data
- Edge-aligned with Read data and center-aligned with Write data
- DLL aligns Data (DDRDQx) and Data Qualifier Strobe (DDRxDQS, $\overline{\text{DDRxDQS}}$) transitions with clock
- Differential clock inputs (DDRCK and /DDRCK)
- Data masks (DDRUDM, DDRLDM) for write data
- Commands entered on each positive DDRCK edge, data and data mask are referenced to both edges of DDRxDQS
- Posted CAS programmable additive latency supported to make command and data bus efficiency
- Read Latency = Additive Latency plus CAS Latency (RL = AL + CL)
- Off-Chip-Driver impedance adjustment (OCD) and On-Die-Termination (ODT) for better signal quality
- Auto-precharge operation for read and write bursts
- Auto Refresh and Self Refresh modes
- Precharged Power Down and Active Power Down
- Write Latency = Read Latency - 1 (WL = RL - 1)

Figure 4-3 provides a block diagram of the DDR2 SDRAM.

**FIGURE 4-3: DDR2 SDRAM BLOCK DIAGRAM**

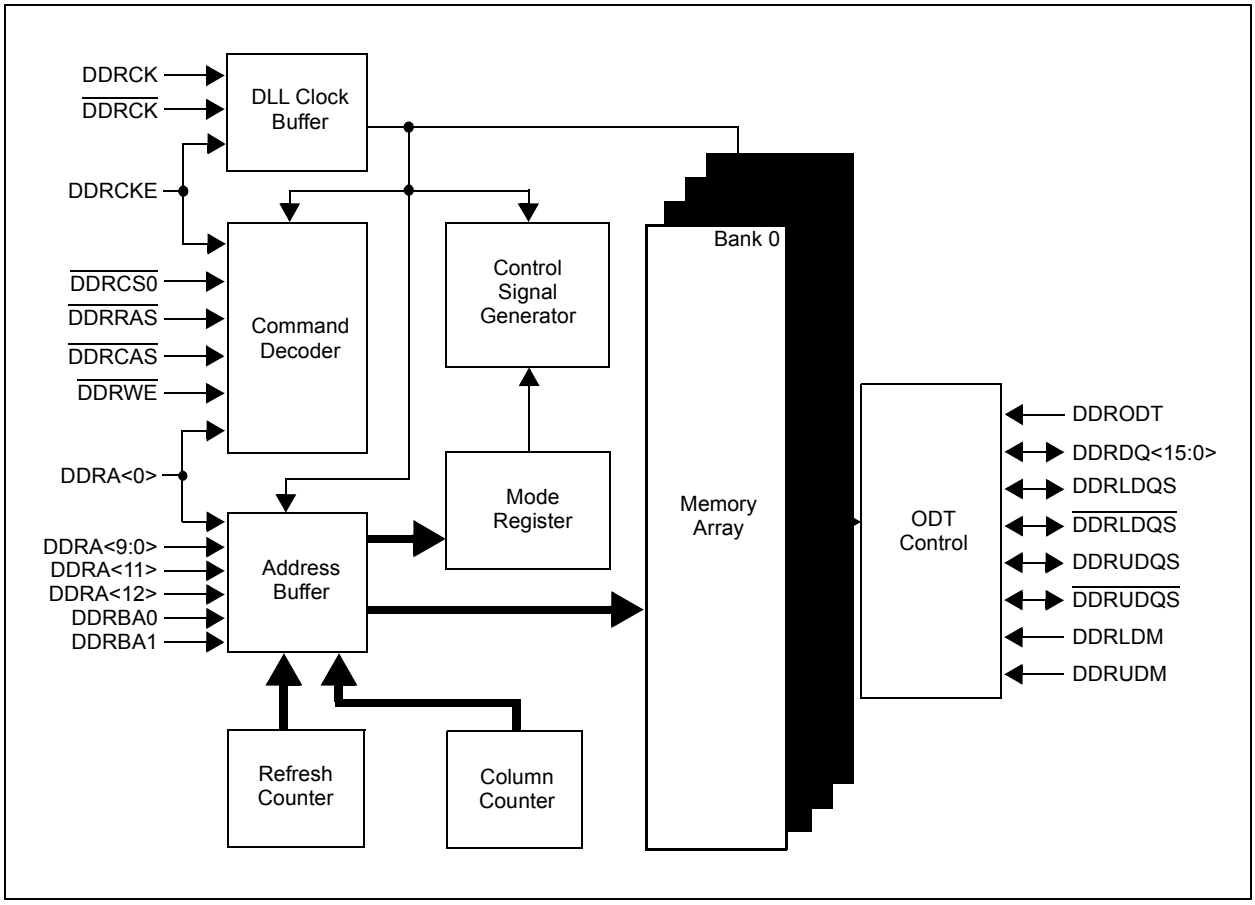


**TABLE 4-8: SYSTEM BUS TARGETS AND ASSOCIATED PROTECTION REGISTERS**

| Target Protection Number | Target Description (see Note 5) | SBTxREGy Register (see Note 7) | | | | | | | SBTxRDy Register | | SBTxWRy Register | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Name | Region Base (BASE<21:0>) (see Note 2) | Physical Start Address | Region Size (SIZE<4:0>) (see Note 3) | Region Size | Priority (PRI) | Priority Level | Name | Read Permission (GROUP3, GROUP2, GROUP1, GROUP0) | Name | Write Permission (GROUP3, GROUP2, GROUP1, GROUP0) |
| 0 | System Bus | SBT0REG0 | R | 0x1F8F0000 | R | 64 KB | — | 0 | SBT0RD0 | 0,1,1,1 | SBT0WR0 | 0,1,1,1 |
| | | SBT0REG1 | R | 0x1F8F8000 | R | 32 KB | — | 3 | SBT0RD1 | 0,0,0,1 | SBT0WR1 | 0,0,0,1 |
| 1 | Flash Memory(6): Program Flash Boot Flash Prefetch | SBT1REG0 | R | 0x1D000000 | R(4) | R(4) | — | 0 | SBT1RD0 | 0,0,0,0 | SBT1WR0 | 0,0,0,0 |
| | | SBT1REG2 | R | 0x1F8E0000 | R | 4 KB | 1 | 2 | SBT1RD2 | R/W(1) | SBT1WR2 | R/W(1) |
| | | SBT1REG3 | R/W | R/W | R/W | R/W | 1 | 2 | SBT1RD3 | 0,0,0,0 | SBT1WR3 | 0,0,0,0 |
| | | SBT1REG4 | R/W | R/W | R/W | R/W | 1 | 2 | SBT1RD4 | 0,0,0,0 | SBT1WR4 | 0,0,0,0 |
| | | SBT1REG5 | R/W | R/W | R/W | R/W | 1 | 2 | SBT1RD5 | 0,0,0,0 | SBT1WR5 | 0,0,0,0 |
| | | SBT1REG6 | R/W | R/W | R/W | R/W | 1 | 2 | SBT1RD6 | 0,0,0,0 | SBT1WR6 | 0,0,0,0 |
| | | SBT1REG7 | R/W | R/W | R/W | R/W | 0 | 1 | SBT1RD7 | 0,0,0,0 | SBT1WR7 | 0,0,0,0 |
| | | SBT1REG8 | R/W | R/W | R/W | R/W | 0 | 1 | SBT1RD8 | 0,0,0,0 | SBT1WR8 | 0,0,0,0 |
| 2 | RAM Bank 1 Memory | SBT2REG0 | R | 0 | R(4) | R(4) | — | 0 | SBT2RD0 | R/W(1) | SBT2WR0 | R/W(1) |
| | | SBT2REG1 | R/W | R/W | R/W | R/W | — | 3 | SBT2RD1 | R/W(1) | SBT2WR1 | R/W(1) |
| | | SBT2REG2 | R/W | R/W | R/W | R/W | 0 | 1 | SBT2RD2 | R/W(1) | SBT2WR2 | R/W(1) |
| 3 | RAM Bank 2 Memory | SBT3REG0 | R(4) | R(4) | R(4) | R(4) | — | 0 | SBT3RD0 | R/W(1) | SBT3WR0 | R/W(1) |
| | | SBT3REG1 | R/W | R/W | R/W | R/W | — | 3 | SBT3RD1 | R/W(1) | SBT3WR1 | R/W(1) |
| | | SBT3REG2 | R/W | R/W | R/W | R/W | 0 | 1 | SBT3RD2 | R/W(1) | SBT3WR2 | R/W(1) |
| 4 | External Memory via DDR2 and DDR2 Target 0 | SBT4REG0 | R | 0x08000000 | R | R(4) | — | 0 | SBT4RD0 | R/W(1) | SBT4WR0 | R/W(1) |
| | | SBT4REG1 | R/W | R/W | R/W | R/W | — | 3 | SBT4RD1 | R/W(1) | SBT4WR1 | R/W(1) |
| | | SBT4REG2 | R/W | R/W | R/W | R/W | 1 | 2 | SBT4RD2 | R/W(1) | SBT4WR2 | R/W(1) |
| | | SBT4REG3 | R/W | R/W | R/W | R/W | 1 | 2 | SBT4RD3 | R/W(1) | SBT4WR3 | R/W(1) |
| | | SBT4REG4 | R/W | R/W | R/W | R/W | 1 | 2 | SBT4RD4 | R/W(1) | SBT4WR4 | R/W(1) |
| 5 | External Memory via DDR2 and DDR2 Targets 1 and 2 | SBT5REG0 | R | 0x08000000 | R | R(4) | — | 0 | SBT5RD0 | R/W(1) | SBT5WR0 | R/W(1) |
| | | SBT5REG1 | R/W | R/W | R/W | R/W | — | 3 | SBT5RD1 | R/W(1) | SBT5WR1 | R/W(1) |
| | | SBT5REG2 | R/W | R/W | R/W | R/W | 1 | 2 | SBT5RD2 | R/W(1) | SBT5WR2 | R/W(1) |
| | | SBT5REG3 | R/W | R/W | R/W | R/W | 1 | 2 | SBT5RD3 | R/W(1) | SBT5WR3 | R/W(1) |
| | | SBT5REG4 | R/W | R/W | R/W | R/W | 1 | 2 | SBT5RD4 | R/W(1) | SBT5WR4 | R/W(1) |

**Legend:** R = Read; R/W = Read/Write; 'x' in a register name = 0-13; 'y' in a register name = 0-8.

**Note 1:** Reset values for these bits are '0', '1', '1', '1', respectively.

**2:** The BASE<21:0> bits must be set to the corresponding Physical Address and right shifted by 10 bits. For Read-only bits, this value is set by hardware on Reset.

**3:** The SIZE<4:0> bits must be set to the corresponding Region Size, based on the following formula: Region Size = $2^{(SIZE-1)}$ x 1024 bytes. For read-only bits, this value is set by hardware on Reset.

**4:** Refer to the Device Memory Map (Figure 4-1) for specific device memory sizes and start addresses.

**5:** See Table 4-2 for information on specific target memory size and start addresses.

**6:** The SBTxREG1 SFRs are reserved, and therefore, are not listed in this table for this target.

**7:** The 'x' in the SBTxREGy, SBTxRDy, and SBTxWRy registers represents the target protection number and not the actual target number (e.g., for SQI 'x' = 13 and not 11, whereas 11 is the actual target number).

**REGISTER 6-1: RCON: RESET CONTROL REGISTER**

bit 4 **WDTO:** Watchdog Timer Time-out Flag bit
1 = WDT Time-out has occurred
0 = WDT Time-out has not occurred

bit 3 **SLEEP:** Wake From Sleep Flag bit
1 = Device was in Sleep mode
0 = Device was not in Sleep mode

bit 2 **IDLE:** Wake From Idle Flag bit
1 = Device was in Idle mode
0 = Device was not in Idle mode

bit 1 **BOR:** Brown-out Reset Flag bit(1)
1 = Brown-out Reset has occurred
0 = Brown-out Reset has not occurred

bit 0 **POR:** Power-on Reset Flag bit(1)
1 = Power-on Reset has occurred
0 = Power-on Reset has not occurred

**Note 1:** User software must clear this bit to view the next detection.

## 7.2 Interrupts

The PIC32MZ DA family uses variable offsets for vector spacing. This allows the interrupt vector spacing to be configured according to application needs. A unique interrupt vector offset can be set for each vector using its associated OFFx register.

For details on the Variable Offset feature, refer to **8.5.2 "Variable Offset"** in **Section 8. "Interrupt Controller"** (DS60001108) of the *"PIC32 Family Reference Manual"*.

Table 7-2 provides the Interrupt IRQ, vector and bit location information.

**TABLE 7-2: INTERRUPT IRQ, VECTOR AND BIT LOCATION**

| Interrupt Source(1) | XC32 Vector Name | IRQ # | Vector # | Interrupt Bit Location | | | | Persistent Interrupt |
|---|---|---|---|---|---|---|---|---|
| | | | | Flag | Enable | Priority | Sub-priority | |
| Highest Natural Order Priority | | | | | | | | |
| Core Timer Interrupt | _CORE_TIMER_VECTOR | 0 | OFF000<17:1> | IFS0<0> | IEC0<0> | IPC0<4:2> | IPC0<1:0> | No |
| Core Software Interrupt 0 | _CORE_SOFTWARE_0_VECTOR | 1 | OFF001<17:1> | IFS0<1> | IEC0<1> | IPC0<12:10> | IPC0<9:8> | No |
| Core Software Interrupt 1 | _CORE_SOFTWARE_1_VECTOR | 2 | OFF002<17:1> | IFS0<2> | IEC0<2> | IPC0<20:18> | IPC0<17:16> | No |
| External Interrupt 0 | _EXTERNAL_0_VECTOR | 3 | OFF003<17:1> | IFS0<3> | IEC0<3> | IPC0<28:26> | IPC0<25:24> | No |
| Timer1 | _TIMER_1_VECTOR | 4 | OFF004<17:1> | IFS0<4> | IEC0<4> | IPC1<4:2> | IPC1<1:0> | No |
| Input Capture 1 Error | _INPUT_CAPTURE_1_ERROR_VECTOR | 5 | OFF005<17:1> | IFS0<5> | IEC0<5> | IPC1<12:10> | IPC1<9:8> | Yes |
| Input Capture 1 | _INPUT_CAPTURE_1_VECTOR | 6 | OFF006<17:1> | IFS0<6> | IEC0<6> | IPC1<20:18> | IPC1<17:16> | Yes |
| Output Compare 1 | _OUTPUT_COMPARE_1_VECTOR | 7 | OFF007<17:1> | IFS0<7> | IEC0<7> | IPC1<28:26> | IPC1<25:24> | No |
| External Interrupt 1 | _EXTERNAL_1_VECTOR | 8 | OFF008<17:1> | IFS0<8> | IEC0<8> | IPC2<4:2> | IPC2<1:0> | No |
| Timer2 | _TIMER_2_VECTOR | 9 | OFF009<17:1> | IFS0<9> | IEC0<9> | IPC2<12:10> | IPC2<9:8> | No |
| Input Capture 2 Error | _INPUT_CAPTURE_2_ERROR_VECTOR | 10 | OFF010<17:1> | IFS0<10> | IEC0<10> | IPC2<20:18> | IPC2<17:16> | Yes |
| Input Capture 2 | _INPUT_CAPTURE_2_VECTOR | 11 | OFF011<17:1> | IFS0<11> | IEC0<11> | IPC2<28:26> | IPC2<25:24> | Yes |
| Output Compare 2 | _OUTPUT_COMPARE_2_VECTOR | 12 | OFF012<17:1> | IFS0<12> | IEC0<12> | IPC3<4:2> | IPC3<1:0> | No |
| External Interrupt 2 | _EXTERNAL_2_VECTOR | 13 | OFF013<17:1> | IFS0<13> | IEC0<13> | IPC3<12:10> | IPC3<9:8> | No |
| Timer3 | _TIMER_3_VECTOR | 14 | OFF014<17:1> | IFS0<14> | IEC0<14> | IPC3<20:18> | IPC3<17:16> | No |
| Input Capture 3 Error | _INPUT_CAPTURE_3_ERROR_VECTOR | 15 | OFF015<17:1> | IFS0<15> | IEC0<15> | IPC3<28:26> | IPC3<25:24> | Yes |
| Input Capture 3 | _INPUT_CAPTURE_3_VECTOR | 16 | OFF016<17:1> | IFS0<16> | IEC0<16> | IPC4<4:2> | IPC4<1:0> | Yes |
| Output Compare 3 | _OUTPUT_COMPARE_3_VECTOR | 17 | OFF017<17:1> | IFS0<17> | IEC0<17> | IPC4<12:10> | IPC4<9:8> | No |
| External Interrupt 3 | _EXTERNAL_3_VECTOR | 18 | OFF018<17:1> | IFS0<18> | IEC0<18> | IPC4<20:18> | IPC4<17:16> | No |
| Timer4 | _TIMER_4_VECTOR | 19 | OFF019<17:1> | IFS0<19> | IEC0<19> | IPC4<28:26> | IPC4<25:24> | No |
| Input Capture 4 Error | _INPUT_CAPTURE_4_ERROR_VECTOR | 20 | OFF020<17:1> | IFS0<20> | IEC0<20> | IPC5<4:2> | IPC5<1:0> | Yes |
| Input Capture 4 | _INPUT_CAPTURE_4_VECTOR | 21 | OFF021<17:1> | IFS0<21> | IEC0<21> | IPC5<12:10> | IPC5<9:8> | Yes |
| Output Compare 4 | _OUTPUT_COMPARE_4_VECTOR | 22 | OFF022<17:1> | IFS0<22> | IEC0<22> | IPC5<20:18> | IPC5<17:16> | No |

**Note 1:** Not all interrupt sources are available on all devices. See the Family Features tables (Table 1 through Table 2) for the list of available peripherals.

**2:** Upon Reset, the GLCD interrupt (both HSYNC and VSYNC) are persistent. However, through the IRQCON bit (GLCDINT<31>), the type of interrupt can be changed to non-persistent.

**REGISTER 8-3: SPLLCON: SYSTEM PLL 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 | R/W-y | R/W-y | R/W-y |
| | — | — | — | — | — | PLLODIV<2:0> | | |
| 23:16 | U-0 | R/W-y | R/W-y | R/W-y | R/W-y | R/W-y | R/W-y | R/W-y |
| | — | PLLMULT<6:0> | | | | | | |
| 15:8 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-y | R/W-y | R/W-y |
| | — | | | | | PLLIDIV<2:0> | | |
| 7:0 | R/W-y | U-0 | U-0 | U-0 | U-0 | R/W-y | R/W-y | R/W-y |
| | PLLICLK | — | — | — | — | PLLRANGE<2:0> | | |

| Legend: | y = Value set from Configuration bits on POR | | |
|---|---|---|---|
| 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 31-27 **Unimplemented:** Read as '0'

bit 26-24 **PLLODIV<2:0>:** System PLL Output Clock Divider bits

111 = Reserved
110 = Reserved
101 = PLL Divide by 32
100 = PLL Divide by 16
011 = PLL Divide by 8
010 = PLL Divide by 4
001 = PLL Divide by 2
000 = Reserved

The default setting is specified by the FPLLODIV<2:0> Configuration bits in the DEVCFG2 register. Refer to Register 34-5 in **Section 34.0 "Special Features"** for information.

bit 23 **Unimplemented:** Read as '0'

bit 22-16 **PLLMULT<6:0>:** System PLL Multiplier bits

1111111 = Multiply by 128
1111110 = Multiply by 127
1111101 = Multiply by 126
1111100 = Multiply by 125
•
•
•
0000000 = Multiply by 1

The default setting is specified by the FPLLMULT<6:0> Configuration bits in the DEVCFG2 register. Refer to Register 34-5 in **Section 34.0 "Special Features"** for information.

bit 15-11 **Unimplemented:** Read as '0'

**Note 1:** Writes to this register require an unlock sequence. Refer to **Section 42. "Oscillators with Enhanced PLL"** (DS60001250) in the *"PIC32 Family Reference Manual"* for details.

**2:** Writes to this register are not allowed if the SPLL is selected as a clock source (COSC<2:0> = 001).

## 9.0 PREFETCH MODULE

**Note:** This data sheet summarizes the features of the PIC32MZ Graphics (DA) 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), which is available from the *Documentation > Reference Manual* section of the Microchip PIC32 web site (www.microchip.com/pic32).

The Prefetch module is a performance enhancing module that is included in PIC32MZ DA family 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.

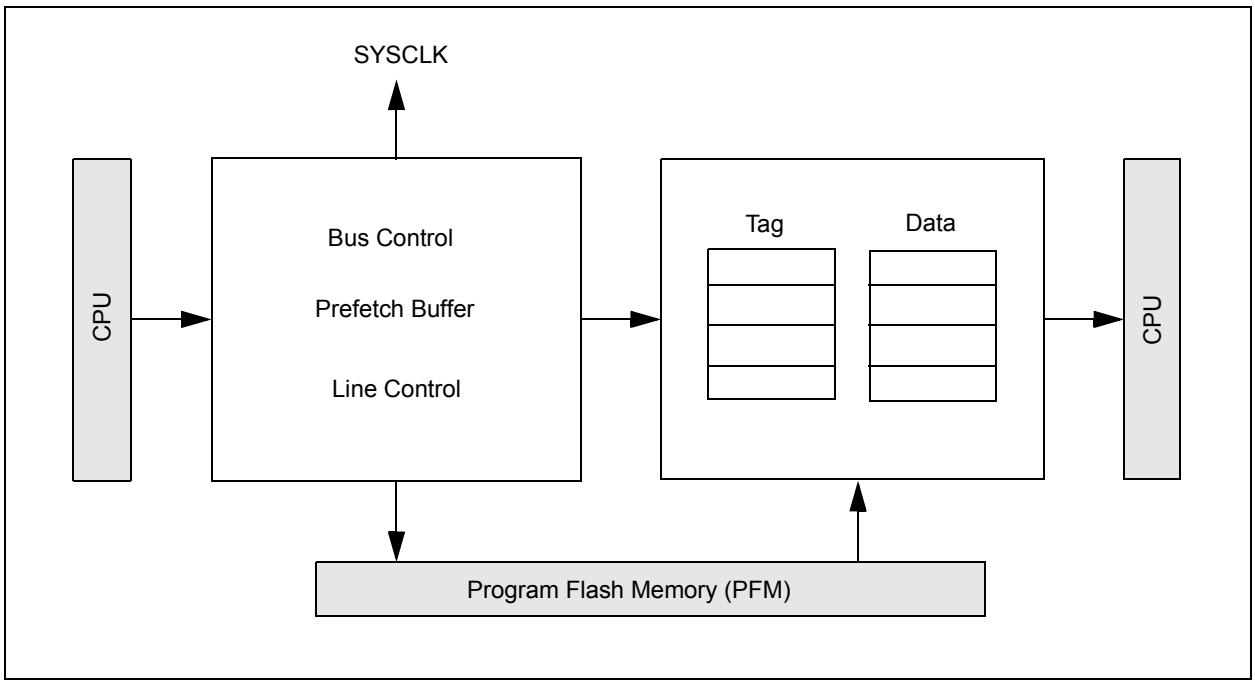
### 9.1 Features

The Prefetch module includes the following key features:

- 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**



**REGISTER 20-1: RTCCON: REAL-TIME CLOCK AND CALENDAR 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 | R/W-0 | R/W-0 |
| | — | — | — | — | — | — | CAL<9:8> | |
| 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 |
| | CAL<7:0> | | | | | | | |
| 15:8 | R/W-0 | U-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| | ON(1) | — | SIDL | — | — | RTCCLKSEL<1:0> | | RTC OUTSEL<1>(2) |
| 7:0 | R/W-0 | R-0 | U-0 | U-0 | R/W-0 | R-0 | R-0 | R/W-0 |
| | RTC OUTSEL<0>(2) | RTC CLKON | — | — | RTC WREN(3) | RTC SYNC | HALFSEC(4) | RTCOE |

| **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 31-26 **Unimplemented:** Read as '0'

bit 25-16 **CAL<9:0>:** Real-Time Clock Drift Calibration bits, which contain a signed 10-bit integer value

0111111111 = Maximum positive adjustment, adds 511 real-time clock pulses every one minute
•
•
•
0000000001 = Minimum positive adjustment, adds 1 real-time clock pulse every one minute
0000000000 = No adjustment
1111111111 = Minimum negative adjustment, subtracts 1 real-time clock pulse every one minute
•
•
•
1000000000 = Minimum negative adjustment, subtracts 512 real-time clock pulses every one minute

bit 15 **ON:** RTCC On bit(1)

1 = RTCC module is enabled
0 = RTCC module is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **SIDL:** Stop in Idle Mode bit

1 = Disables RTCC operation when CPU enters Idle mode
0 = Continue normal operation when CPU enters Idle mode

bit 12-11 **Unimplemented:** Read as '0'

**Note 1:** The ON bit is only writable when RTCWREN = 1.
**2:** Requires RTCOE = 1 (RTCCON<0>) for the output to be active.
**3:** The RTCWREN bit can be set only when the write sequence is enabled.
**4:** This bit is read-only. It is cleared to '0' on a write to the seconds bit fields (RTCTIME<14:8>).

**Note:** This register is reset only on a Power-on Reset (POR).

**REGISTER 27-5: CESTAT: CRYPTO ENGINE 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 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| | ERRMODE<2:0> | | | ERROP<2:0> | | | ERRPHASE<1:0> | |
| 23:16 | U-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| | — | — | BDSTATE | | | | START | ACTIVE |
| 15:8 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| | BDCTRL<15:8> | | | | | | | |
| 7:0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| | BDCTRL<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 31-29 **ERRMOD<2:0>:** Internal Error Mode Status bits
- 111 = Reserved
- 110 = Reserved
- 101 = Reserved
- 100 = Reserved
- 011 = CEK operation
- 010 = KEK operation
- 001 = Preboot authentication
- 000 = Normal operation

bit 28-26 **ERROP<2:0>:** Internal Error Operation Status bits
- 111 = Reserved
- 110 = Reserved
- 101 = Reserved
- 100 = Authentication
- 011 = Reserved
- 010 = Decryption
- 001 = Encryption
- 000 = Reserved

bit 25-24 **ERRPHASE<1:0>:** Internal Error Phase of DMA Status bits
- 11 = Destination data
- 10 = Source data
- 01 = Security Association (SA) access
- 00 = Buffer Descriptor (BD) access

bit 23-22 **Unimplemented:** Read as '0'

bit 21-18 **BDSTATE<3:0>:** Buffer Descriptor Processor State Status bits

These bits contain a number, which indicates the current state of the BDP:
- 1111 = Reserved
- •
- •
- •
- 0111 = Reserved
- 0110 = SA fetch
- 0101 = Fetch BDP is disabled
- 0100 = Descriptor is done
- 0011 = Data phase
- 0010 = BDP is loading
- 0001 = Descriptor fetch request is pending
- 0000 = BDP is idle

bit 17 **START:** DMA Start Status bit
- 1 = DMA start has occurred
- 0 = DMA start has not occurred

## 28.1 RNG Control Registers

**TABLE 28-1: 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.

**Step 4:** The user sets the ON bit to '1', which enables the ADC control clock. The following ADCx activation sequence is to be followed at all times:

**Step 5:** The user waits for the interrupt/polls the BGVRRDY bit (ADCCON2<31>) and the WKRDYx bit (ADCANCON<15,13:8>) = 1, which signals that the device analog environment (band gap and VREF) is ready.

**Step 6:** Set the DIGENx bit (ADCCON3<15,13:8>) to '1', which enables the digital circuitry to immediately begin processing incoming triggers to perform data conversions.

Standard non-interleaved dedicated Class_1 ADCx throughput rate formula is shown in Equation 29-1.

**EQUATION 29-1: THROUGHPUT RATE**

ADC Throughput Rate = 1/((Sample time + Conversion time)(TAD))
= 1 / ((SAMC+# bit resolution+1)(TAD))

Example:
SAMC = 3 TAD, 12-bit mode, TAD = 20 ns = 50 MHz:
Throughput rate:
= 1 / ((3+13)(20 ns))
= 1/(16 * 20 ns)
= 3.125 msps

**TABLE 29-1: PIC32MZXXDAXX INTERLEAVED ADC THROUGHPUT RATES**

| #No.of Interleaved ADC Possible | ADC TAD(min) = 20ns (50Mhz max) | | | |
|---|---|---|---|---|
| | 12-bit (max.) msps | 10-bit (max.) msps | 8-bit (max.) msps | 6-bit (max.) msps |
| 1 | 3.125 msps | 3.571 msps | 4.167 msps | 5.0 msps |
| 2 | 6.250 msps | 7.143 msps | 8.333 msps | 10.00 msps |
| 3 | 8.330 msps | 10.00 msps | 12.50 msps | 12.50 msps |
| 4 | 12.50 msps | 12.50 msps | 16.667 msps | 16.667 msps |

**Note:** Interleaved ADCs in this context means connecting the same analog source signal to multiple dedicated Class_1 ADCs (i.e., ADC0-ADC5), and using independent staggered trigger sources accordingly for each interleaved ADC.

**Note 1:** Prior to enabling the ADC module, the user application must copy the ADC calibration data (DEVADC0-DEVADC4, DEVADC7; see Register 41-8) from the Configuration memory into the ADC Configuration registers (ADC0CFG-ADC4CFG, ADC7CFG).

**2:** If $V_{DDIO}$ is greater than 2.5V, set the AICPMPEN bit (ADCCON1<12>) and the IOANCPEN bit (CFGCON<7>) to '0'. If $V_{DDIO}$ is less than 2.5V, set both bits to '1'.

## 29.2 ADC Control Registers

**TABLE 29-2: ADC REGISTER MAP**

| Virtual Address | 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 | |
| B000 | ADCCON1 | 31:16 | TRBEN | TRBERR | TRBMST<2:0> | | | TRBSLV<2:0> | | | FRACT | SELRES<1:0> | | STRGSRC<4:0> | | | | | 0060 |
| | | 15:0 | ON | — | SIDL | AICPMPEN | CVDEN | FSSCLKEN | FSPBCLKEN | — | — | IRQVS<2:0> | | | STRGLVL | — | — | — | 0000 |
| B004 | ADCCON2 | 31:16 | BGVRRDY | REFFLT | EOSRDY | CVDCPL<2:0> | | SAMC<9:0> | | | | | | | | | | | 0000 |
| | | 15:0 | BGVRIEN | REFFLTIEN | EOSIEN | ADCEIOVR | — | ADCEIS<2:0> | | | — | ADCDIV<6:0> | | | | | | | 0000 |
| B008 | ADCCON3 | 31:16 | ADCSEL<1:0> | | CONCLKDIV<5:0> | | | | | | DIGEN7 | — | — | DIGEN4 | DIGEN3 | DIGEN2 | DIGEN1 | DIGEN0 | 0000 |
| | | 15:0 | VREFSEL<2:0> | | | TRGSUSP | UPDIEN | UPDRDY | SAMP | RQCNVRT | GLSWTRG | GSWTRG | ADINSEL<5:0> | | | | | | 0000 |
| B00C | ADCTRGMODE | 31:16 | — | — | — | — | — | — | SH4ALT<1:0> | | SH3ALT<1:0> | | SH2ALT<1:0> | | SH1ALT<1:0> | | SH0ALT<1:0> | | 0000 |
| | | 15:0 | — | — | — | STRGEN4 | STRGEN3 | STRGEN2 | STRGEN1 | STRGEN0 | — | — | — | SSAMPEN4 | SSAMPEN3 | SSAMPEN2 | SSAMPEN1 | SSAMPEN0 | 0000 |
| B010 | ADCIMCON1 | 31:16 | DIFF15 | SIGN15 | DIFF14 | SIGN14 | DIFF13 | SIGN13 | DIFF12 | SIGN12 | DIFF11 | SIGN11 | DIFF10 | SIGN10 | DIFF9 | SIGN9 | DIFF8 | SIGN8 | 0000 |
| | | 15:0 | DIFF7 | SIGN7 | DIFF6 | SIGN6 | DIFF5 | SIGN5 | DIFF4 | SIGN4 | DIFF3 | SIGN3 | DIFF2 | SIGN2 | DIFF1 | SIGN1 | DIFF0 | SIGN0 | 0000 |
| B014 | ADCIMCON2 | 31:16 | DIFF31 | SIGN31 | DIFF30 | SIGN30 | DIFF29 | SIGN29 | DIFF28 | SIGN28 | DIFF27 | SIGN27 | DIFF26 | SIGN26 | DIFF25 | SIGN25 | DIFF24 | SIGN24 | 0000 |
| | | 15:0 | DIFF23 | SIGN23 | DIFF22 | SIGN22 | DIFF21 | SIGN21 | DIFF20 | SIGN20 | DIFF19 | SIGN19 | DIFF18 | SIGN18 | DIFF17 | SIGN17 | DIFF16 | SIGN16 | 0000 |
| B018 | ADCIMCON3 | 31:16 | — | — | — | — | — | — | — | — | DIFF43 | SIGN43 | DIFF42 | SIGN42 | DIFF41 | SIGN41 | DIFF40 | SIGN40 | 0000 |
| | | 15:0 | DIFF39 | SIGN39 | DIFF38 | SIGN38 | DIFF37 | SIGN37 | DIFF36 | SIGN36 | DIFF35 | SIGN35 | DIFF34 | SIGN34 | DIFF33 | SIGN33 | DIFF32 | SIGN32 | 0000 |
| B020 | ADCGIRQEN1 | 31:16 | AGIEN31 | AGIEN30 | AGIEN29 | AGIEN28 | AGIEN27 | AGIEN26 | AGIEN25 | AGIEN24 | AGIEN23 | AGIEN22 | AGIEN21 | AGIEN20 | AGIEN19 | AGIEN18 | AGIEN17 | AGIEN16 | 0000 |
| | | 15:0 | AGIEN15 | AGIEN14 | AGIEN13 | AGIEN12 | AGIEN11 | AGIEN10 | AGIEN9 | AGIEN8 | AGIEN7 | AGIEN6 | AGIEN5 | AGIEN4 | AGIEN3 | AGIEN2 | AGIEN1 | AGIEN0 | 0000 |
| B024 | ADCGIRQEN2 | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | — | — | — | — | AGIEN43 | AGIEN42 | AGIEN41 | AGIEN40 | AGIEN39 | AGIEN38 | AGIEN37 | AGIEN36 | AGIEN35 | AGIEN34 | AGIEN33 | AGIEN32 | 0000 |
| B028 | ADCCSS1 | 31:16 | CSS31 | CSS30 | CSS29 | CSS28 | CSS27 | CSS26 | CSS25 | CSS24 | CSS23 | CSS22 | CSS21 | CSS20 | CSS19 | CSS18 | CSS17 | CSS16 | 0000 |
| | | 15:0 | CSS15 | CSS14 | CSS13 | CSS12 | CSS11 | CSS10 | CSS9 | CSS8 | CSS7 | CSS6 | CSS5 | CSS4 | CSS3 | CSS2 | CSS1 | CSS0 | 0000 |
| B02C | ADCCSS2 | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | — | — | — | — | CSS43 | CSS42 | CSS41 | CSS40 | CSS39 | CSS38 | CSS37 | CSS36 | CSS35 | CSS34 | CSS33 | CSS32 | 0000 |
| B030 | ADCDSTAT1 | 31:16 | ARDY31 | ARDY30 | ARDY29 | ARDY28 | ARDY27 | ARDY26 | ARDY25 | ARDY24 | ARDY23 | ARDY22 | ARDY21 | ARDY20 | ARDY19 | ARDY18 | ARDY17 | ARDY16 | 0000 |
| | | 15:0 | ARDY15 | ARDY14 | ARDY13 | ARDY12 | ARDY11 | ARDY10 | ARDY9 | ARDY8 | ARDY7 | ARDY6 | ARDY5 | ARDY4 | ARDY3 | ARDY2 | ARDY1 | ARDY0 | 0000 |
| B034 | ADCDSTAT2 | 31:16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | 0000 |
| | | 15:0 | — | — | — | — | ARDY43 | ARDY42 | ARDY41 | ARDY40 | ARDY39 | ARDY38 | ARDY37 | ARDY36 | ARDY35 | ARDY34 | ARDY33 | ARDY32 | 0000 |
| B038 | ADCCMPEN1 | 31:16 | CMPE31 | CMPE30 | CMPE29 | CMPE28 | CMPE27 | CMPE26 | CMPE25 | CMPE24 | CMPE23 | CMPE22 | CMPE21 | CMPE20 | CMPE19 | CMPE18 | CMPE17 | CMPE16 | 0000 |
| | | 15:0 | CMPE15 | CMPE14 | CMPE13 | CMPE12 | CMPE11 | CMPE10 | CMPE9 | CMPE8 | CMPE7 | CMPE6 | CMPE5 | CMPE4 | CMPE3 | CMPE2 | CMPE1 | CMPE0 | 0000 |
| B03C | ADCCMP1 | 31:16 | DCMPHI<15:0> | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | DCMPLO<15:0> | | | | | | | | | | | | | | | | 0000 |
| B040 | ADCCMPEN2 | 31:16 | CMPE31 | CMPE30 | CMPE29 | CMPE28 | CMPE27 | CMPE26 | CMPE25 | CMPE24 | CMPE23 | CMPE22 | CMPE21 | CMPE20 | CMPE19 | CMPE18 | CMPE17 | CMPE16 | 0000 |
| | | 15:0 | CMPE15 | CMPE14 | CMPE13 | CMPE12 | CMPE11 | CMPE10 | CMPE9 | CMPE8 | CMPE7 | CMPE6 | CMPE5 | CMPE4 | CMPE3 | CMPE2 | CMPE1 | CMPE0 | 0000 |
| B044 | ADCCMP2 | 31:16 | DCMPHI<15:0> | | | | | | | | | | | | | | | | 0000 |
| | | 15:0 | DCMPLO<15:0> | | | | | | | | | | | | | | | | 0000 |
| B048 | ADCCMPEN3 | 31:16 | CMPE31 | CMPE30 | CMPE29 | CMPE28 | CMPE27 | CMPE26 | CMPE25 | CMPE24 | CMPE23 | CMPE22 | CMPE21 | CMPE20 | CMPE19 | CMPE18 | CMPE17 | CMPE16 | 0000 |
| | | 15:0 | CMPE15 | CMPE14 | CMPE13 | CMPE12 | CMPE11 | CMPE10 | CMPE9 | CMPE8 | CMPE7 | CMPE6 | CMPE5 | CMPE4 | CMPE3 | CMPE2 | CMPE1 | CMPE0 | 0000 |

**Note 1:** Before enabling the ADC, the user application must initialize the ADC calibration values by copying them from the factory-programmed DEVADCx Flash registers into the corresponding ADCxCFG registers.

**REGISTER 29-3: ADCCON3: ADC CONTROL REGISTER 3**

| 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 |
| | ADCSEL<1:0> | | CONCLKDIV<5:0> | | | | | |
| 23:16 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | DIGEN7 | — | — | DIGEN4 | DIGEN3 | DIGEN2 | DIGEN1 | DIGEN0 |
| 15:8 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0, HS, HC | R/W-0 | R-0, HS, HC |
| | VREFSEL<2:0> | | | TRGSUSP | UPDIEN | UPDRDY | SAMP(1,2,3,4) | RQCNVRT |
| 7:0 | R/W-0 | R/W, HC | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | GLSWTRG | GSWTRG | ADINSEL<5:0> | | | | | |

| Legend: | HC = Hardware Set | HS = Hardware Cleared | |
|---|---|---|---|
| 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 31-30 **ADCSEL<1:0>:** Analog-to-Digital Clock Source (TCLK) bits
11 = FRC
10 = REFCLK3
01 = System Clock (Tcy)
00 = PBCLK3

bit 29-24 **CONCLKDIV<5:0>:** Analog-to-Digital Control Clock (TQ) Divider bits
111111 = 64 * TCLK = TQ
•
•
•
000011 = 4 * TCLK = TQ
000010 = 3 * TCLK = TQ
000001 = 2 * TCLK = TQ
000000 = TCLK = TQ

bit 23 **DIGEN7:** Shared ADC (ADC7) Digital Enable bit
1 = ADC7 is digital enabled
0 = ADC7 is digital disabled

bit 22-21 **Unimplemented:** Read as '0'

bit 20 **DIGEN4:** ADC4 Digital Enable bit
1 = ADC4 is digital enabled
0 = ADC4 is digital disabled

bit 19 **DIGEN3:** ADC3 Digital Enable bit
1 = ADC3 is digital enabled
0 = ADC3 is digital disabled

**Note 1:** The SAMP bit has the highest priority and setting this bit will keep the S&H circuit in Sample mode until the bit is cleared. Also, usage of the SAMP bit will cause settings of SAMC<9:0> bits (ADCCON2<25:16>) to be ignored.

**2:** The SAMP bit only connects Class 2 and Class 3 analog inputs to the shared ADC, ADC7. All Class 1 analog inputs are not affected by the SAMP bit.

**3:** The SAMP bit is not a self-clearing bit and it is the responsibility of application software to first clear this bit and only after setting the RQCNVRT bit to start the analog-to-digital conversion.

**4:** Normally, when the SAMP and RQCNVRT bits are used by software routines, all TRGSRCx<4:0> bits and STRGSRC<4:0> bits should be set to '00000' to disable all external hardware triggers and prevent them from interfering with the software-controlled sampling command signal SAMP and with the software-controlled trigger RQCNVRT.

**REGISTER 29-5: ADCIMCON1: ADC INPUT MODE CONTROL REGISTER 1 (CONTINUED)**

bit 4 **SIGN2:** AN2 Signed Data Mode bit
1 = AN2 is using Signed Data mode
0 = AN2 is using Unsigned Data mode

bit 3 **DIFF1:** AN1 Mode bit
1 = AN1 is using Differential mode
0 = AN1 is using Single-ended mode

bit 2 **SIGN1:** AN1 Signed Data Mode bit
1 = AN1 is using Signed Data mode
0 = AN1 is using Unsigned Data mode

bit 1 **DIFF0:** AN0 Mode bit
1 = AN0 is using Differential mode
0 = AN0 is using Single-ended mode

bit 0 **SIGN0:** AN0 Signed Data Mode bit
1 = AN0 is using Signed Data mode
0 = AN0 is using Unsigned Data mode

**REGISTER 29-20: ADCCMPCON1: ADC DIGITAL COMPARATOR 1 CONTROL REGISTER**

bit 3 **IEHIHI:** High/High Digital Comparator 0 Event bit
1 = Generate a Digital Comparator 0 Event when DCMPHI<15:0> ≤ DATA<31:0>
0 = Do not generate an event

bit 2 **IEHILO:** High/Low Digital Comparator 0 Event bit
1 = Generate a Digital Comparator 0 Event when DATA<31:0> < DCMPHI<15:0>
0 = Do not generate an event

bit 1 **IELOHI:** Low/High Digital Comparator 0 Event bit
1 = Generate a Digital Comparator 0 Event when DCMPLO<15:0> ≤ DATA<31:0>
0 = Do not generate an event

bit 0 **IELOLO:** Low/Low Digital Comparator 0 Event bit
1 = Generate a Digital Comparator 0 Event when DATA<31:0> < DCMPLO<15:0>
0 = Do not generate an event

**REGISTER 30-17: CiFLTCON7: CAN FILTER CONTROL REGISTER 7**

| 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 |
| | FLTEN31 | MSEL31<1:0> | | FSEL31<4:0> | | | | |
| 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 |
| | FLTEN30 | MSEL30<1:0> | | FSEL30<4:0> | | | | |
| 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 |
| | FLTEN29 | MSEL29<1:0> | | FSEL29<4:0> | | | | |
| 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 |
| | FLTEN28 | MSEL28<1:0> | | FSEL28<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 **FLTEN31:** Filter 31 Enable bit
1 = Filter is enabled
0 = Filter is disabled

bit 30-29 **MSEL31<1:0>:** Filter 31 Mask Select bits
11 = Acceptance Mask 3 selected
10 = Acceptance Mask 2 selected
01 = Acceptance Mask 1 selected
00 = Acceptance Mask 0 selected

bit 28-24 **FSEL31<4:0>:** FIFO Selection bits
11111 = Message matching filter is stored in FIFO buffer 31
11110 = Message matching filter is stored in FIFO buffer 30
•
•
•
00001 = Message matching filter is stored in FIFO buffer 1
00000 = Message matching filter is stored in FIFO buffer 0

bit 23 **FLTEN30:** Filter 30Enable bit
1 = Filter is enabled
0 = Filter is disabled

bit 22-21 **MSEL30<1:0>:** Filter 30Mask Select bits
11 = Acceptance Mask 3 selected
10 = Acceptance Mask 2 selected
01 = Acceptance Mask 1 selected
00 = Acceptance Mask 0 selected

bit 20-16 **FSEL30<4:0>:** FIFO Selection bits
11111 = Message matching filter is stored in FIFO buffer 31
11110 = Message matching filter is stored in FIFO buffer 30
•
•
•
00001 = Message matching filter is stored in FIFO buffer 1
00000 = Message matching filter is stored in FIFO buffer 0

**Note:** The bits in this register can only be modified if the corresponding filter enable (FLTENn) bit is '0'.

**REGISTER 30-20: CiFIFOCONn: CAN FIFO CONTROL REGISTER (n = 0 THROUGH 31)**

bit 6 **TXABAT:** Message Aborted bit(2)
1 = Message was aborted
0 = Message completed successfully

bit 5 **TXLARB:** Message Lost Arbitration bit(3)
1 = Message lost arbitration while being sent
0 = Message did not loose arbitration while being sent

bit 4 **TXERR:** Error Detected During Transmission bit(3)
1 = A bus error occured while the message was being sent
0 = A bus error did not occur while the message was being sent

bit 3 **TXREQ:** Message Send Request
TXEN = 1: (FIFO configured as a Transmit FIFO)
Setting this bit to '1' requests sending a message.
The bit will automatically clear when all the messages queued in the FIFO are successfully sent
Clearing the bit to '0' while set ('1') will request a message abort.
TXEN = 0: (FIFO configured as a Receive FIFO)
This bit has no effect.

bit 2 **RTREN:** Auto RTR Enable bit
1 = When a remote transmit is received, TXREQ will be set
0 = When a remote transmit is received, TXREQ will be unaffected

bit 1-0 **TXPR<1:0>:** Message Transmit Priority bits
11 = Highest Message Priority
10 = High Intermediate Message Priority
01 = Low Intermediate Message Priority
00 = Lowest Message Priority

**Note 1:** These bits can only be modified when the CAN module is in Configuration mode (OPMOD<2:0> bits (CiCON<23:21>) = 100).
**2:** This bit is updated when a message completes (or aborts) or when the FIFO is reset.
**3:** This bit is reset on any read of this register or when the FIFO is reset.

**REGISTER 31-15: ETHSTAT: ETHERNET CONTROLLER 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 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | BUFCNT<7:0>(1) | | | | | | | |
| 15:8 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | — | — | — | — | — | — | — | — |
| 7:0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
| | ETHBUSY(5) | TXBUSY(2,6) | RXBUSY(3,6) | — | — | — | — | — |

| 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 31-24 **Unimplemented:** Read as '0'

bit 23-16 **BUFCNT<7:0>:** Packet Buffer Count bits(1)

Number of packet buffers received in memory. Once a packet has been successfully received, this register is incremented by hardware based on the number of descriptors used by the packet. Software decrements the counter (by writing to the BUFCDEC bit (ETHCON1<0>) for each descriptor used) after a packet has been read out of the buffer. The register does not roll over (0xFF to 0x00) when hardware tries to increment the register and the register is already at 0xFF. Conversely, the register does not roll under (0x00 to 0xFF) when software tries to decrement the register and the register is already at 0x0000. When software attempts to decrement the counter at the same time that the hardware attempts to increment the counter, the counter value will remain unchanged.

When this register value reaches 0xFF, the RX logic will halt (only if automatic Flow Control is enabled) awaiting software to write the BUFCDEC bit in order to decrement the register below 0xFF.

If automatic Flow Control is disabled, the RXDMA will continue processing and the BUFCNT will saturate at a value of 0xFF.

When this register is non-zero, the PKTPEND status bit will be set and an interrupt may be generated, depending on the value of the ETHIEN bit <PKTPENDIE> register.

When the ETHRXST register is written, the BUFCNT counter is automatically cleared to 0x00.

**Note:** BUFCNT will not be cleared when ON is set to '0'. This enables software to continue to utilize and decrement this count.

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **ETHBUSY:** Ethernet Module busy bit(4,5)

1 = Ethernet logic has been turned on (ON (ETHCON1<15>) = 1) or is completing a transaction
0 = Ethernet logic is idle

This bit indicates that the module has been turned on or is completing a transaction after being turned off.

**Note 1:** This bit is only used for RX operations.
**2:** This bit is only affected by TX operations.
**3:** This bit is only affected by RX operations.
**4:** This bit is affected by TX and RX operations.
**5:** This bit will be *set* when the ON bit (ETHCON1<15>) = 1.
**6:** This bit will be *cleared* when the ON bit (ETHCON1<15>) = 0.

## 37.0 2-D GRAPHICS PROCESSING UNIT (GPU)

**Note:** This data sheet summarizes the features of the PIC32MZ Graphics (DA) Family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the documents listed in the *Documentation > Reference Manual* section of the Microchip PIC32 web site (www.microchip.com/pic32).

The 2-D Graphics Processing Unit manipulates and alters the contents of the frame buffer in system RAM or DDR2 memory to accelerate the rendering of images for eventual pixel display. Hardware acceleration is brought to numerous 2-D applications, such as graphics user interfaces (menus, objects, and so on).

The 2-D GPU also provides accelerated on-the-fly rendering of vertical and horizontal lines, rectangles, copying of a rectangular area between different locations in memory. Once initiated, the hardware will perform the rendering through DMA, which makes the CPU available for other tasks.

A block diagram showing the interface for the 2-D Graphics Processing Unit is provided in Figure 37-1.

**Note:** For this peripheral, no hardware interface is documented. Use the Nano-2D Library, which is available in MPLAB Harmony, to manage this module.

The following are key features of the 2-D Graphics Processing Unit:

- 64-bit bus access to memory (higher throughput)
- Global clock gating (low power)
- Command buffers
- Fixed Functions:
  - Line draw
  - Rectangle fill
  - Rectangle clear
  - Bit blit (stretch/shrink/filter)
  - Programmable raster operation (ROP2), with full alpha blending and transparency
- Source data formats:
  - RGBA8888, RGB565, RGB5551, 8-bit Index
- Destination data formats:
  - RGBA8888, RGB565, RGB5551
- Dithering (18-bit)
- Orientation in 90-degree steps
- Clipping

**Note 1:** For RGB source formats, their related swizzle formats, such as ARGB and RGBA are supported.

**2:** The GPU is enabled and ready out of POR. However, the GPU can be soft Reset at run-time using the GPURESET bit (CFGCON2<0>). Make sure that the GPUMD bit is set to '0' and wait 10 µs before toggling the GPURESET bit to achieve proper soft Reset.

**FIGURE 37-1: 2-D GRAPHICS PROCESSING UNIT BLOCK DIAGRAM**