



Welcome to [E-XFL.COM](https://www.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	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f27k42-i-ss

9.5 Context Saving

The Interrupt controller supports a two-level deep context saving (Main routine context and Low ISR context). Refer to state machine shown in [Figure 9-6](#) for details.

The Program Counter (PC) is saved on the dedicated device PC stack. CPU registers saved include STATUS, WREG, BSR, FSR0/1/2, PRODL/H and PCLATH/U.

After WREG has been saved to the context registers, the resolved vector number of the interrupt source to be serviced is copied into WREG. Context save and restore operation is completed by the interrupt controller based on current state of the interrupts and the order in which they were sent to the CPU.

Context save/restore works the same way in both states of MVECEN. When IPEN = 0, there is only one level interrupt active. Hence, only the main context is saved when an interrupt is received.

9.5.1 ACCESSING SHADOW REGISTERS

The Interrupt controller automatically saves the context information in the shadow registers available in Bank 56. Both the saved context values (i.e., main routine and low ISR) can be accessed using the same set of shadow registers. By clearing the SHADLO bit in the SHADCON register ([Register 9-43](#)), the CPU register values saved for main routine context can accessed, and by setting the SHADLO bit of the CPU register, values saved for low ISR context can accessed. Low ISR context is automatically restored to the CPU registers upon exiting the high ISR. Similarly, the main context is automatically restored to the CPU registers upon exiting the low ISR.

The Shadow registers in Bank 56 are readable and writable, so if the user desires to modify the context, then the corresponding shadow register should be modified and the value will be restored when exiting the ISR. Depending on the user's application, other registers may also need to be saved.

9.6 Returning from Interrupt Service Routine (ISR)

The “Return from Interrupt” instruction (`RETFIE`) is used to mark the end of an ISR.

When `RETFIE 1` instruction is executed, the PC is loaded with the saved PC value from the top of the PC stack. Saved context is also restored with the execution of this instruction. Thus, execution returns to the previous state of operation that existed before the interrupt occurred.

When `RETFIE 0` instruction is executed, the saved context is not restored back to the registers.

9.7 Interrupt Latency

By assigning each interrupt with a vector address/number (`MVECEN = 1`), scanning of all interrupts is not necessary to determine the source of the interrupt.

When `MVECEN = 1`, Vectored interrupt controller requires three clock cycles to vector to the ISR from main routine, thereby removing dependency of interrupt timing on compiled code.

There is a fixed latency of three instruction cycles between the completion of the instruction active when the interrupt occurred and the first instruction of the Interrupt Service Routine. [Figure 9-7](#), [Figure 9-8](#) and [Figure 9-9](#) illustrate the sequence of events when a peripheral interrupt is asserted when the last executed instruction is one-cycle, two-cycle and three-cycle respectively, when `MVECEN = 1`.

After the Interrupt Flag Status bit is set, the current instruction completes executing. In the first latency cycle, the contents of the PC, STATUS, WREG, BSR, FSR0/1/2, PRODL/H and PCLATH/U registers are context saved and the IVTBASE+ Vector number is calculated. In the second latency cycle, the PC is loaded with the calculated vector table address for the interrupt source and the starting address of the ISR is fetched. In the third latency cycle, the PC is loaded with the ISR address. All the latency cycles are executed as a `FNOP` instruction.

When `MVECEN = 0`, Vectored interrupt controller requires two clock cycles to vector to the ISR from main routine. There is a latency of two instruction cycles plus the software latency between the completion of the instruction active when the interrupt occurred and the first instruction of the Interrupt Service Routine.

10.2.1 WAKE-UP FROM SLEEP

The device can wake up from Sleep through one of the following events:

1. External Reset input on $\overline{\text{MCLR}}$ pin, if enabled
2. BOR Reset, if enabled
3. Low-Power Brown-Out Reset (LPBOR), if enabled
4. POR Reset
5. Windowed Watchdog Timer, if enabled
6. All interrupt sources except clock switch interrupt can wake up the part.

The first five events will cause a device Reset. The last one event is considered a continuation of program execution. To determine whether a device Reset or wake-up event occurred, refer to [Section 6.13 “Power Control \(PCON0/PCON1\) Register”](#).

When the `SLEEP` instruction is being executed, the next instruction (`PC + 2`) is prefetched. For the device to wake-up through an interrupt event, the corresponding Interrupt Enable bit must be enabled. Wake-up will occur regardless of the state of the GIE bit. If the GIE bit is disabled, the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is enabled, the device executes the instruction after the `SLEEP` instruction, the device will then call the Interrupt Service Routine. In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

The WDT is cleared when the device wakes-up from Sleep, regardless of the source of wake-up.

Upon a wake from a Sleep event, the core will wait for a combination of three conditions before beginning execution. The conditions are:

- PFM Ready
- COSC-Selected Oscillator Ready
- BOR Ready (unless BOR is disabled)

10.2.2 WAKE-UP USING INTERRUPTS

When any interrupt source, with the exception of the clock switch interrupt, has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a `SLEEP` instruction
 - `SLEEP` instruction will execute as a `NOP`
 - WDT and WDT prescaler will not be cleared
 - $\overline{\text{TO}}$ bit of the STATUS register will not be set
 - $\overline{\text{PD}}$ bit of the STATUS register will not be cleared
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction
 - `SLEEP` instruction will be completely executed
 - Device will immediately wake-up from Sleep
 - WDT and WDT prescaler will be cleared
 - $\overline{\text{TO}}$ bit of the STATUS register will be set
 - $\overline{\text{PD}}$ bit of the STATUS register will be cleared

Even if the flag bits were checked before executing a `SLEEP` instruction, it may be possible for flag bits to become set before the `SLEEP` instruction completes. To determine whether a `SLEEP` instruction executed, test the $\overline{\text{PD}}$ bit. If the $\overline{\text{PD}}$ bit is set, the `SLEEP` instruction was executed as a `NOP`.

PIC18(L)F26/27/45/46/47/55/56/57K42

EXAMPLE 13-4: WRITING TO PROGRAM FLASH MEMORY (CONTINUED)

```
WRITE_BYTE_TO_HREGS
    MOVF     POSTINC0, W           ; get low byte of buffer data
    MOVWF    TABLAT               ; present data to table latch
    TBLWT+*                       ; write data, perform a short write
                                ; to internal TBLWT holding register.
                                ; loop until holding registers are full

    DECFSZ   COUNTER
    BRA      WRITE_WORD_TO_HREGS

PROGRAM_MEMORY
    BCF      NVMCON1, REG0        ; point to Program Flash Memory
    BSF      NVMCON1, REG1        ; point to Program Flash Memory
    BSF      NVMCON1, WREN        ; enable write to memory
    BCF      NVMCON1, FREE        ; enable write to memory
    BCF      INTCON0, GIE        ; disable interrupts
    MOVLW    55h
    Required MOVWF    NVMCON2        ; write 55h
Sequence    MOVLW    0AAh
    MOVWF    NVMCON2            ; write 0AAh
    BSF      NVMCON1, WR         ; start program (CPU stall)
    DCFSZ    COUNTER2           ; repeat for remaining write blocks
    BRA      WRITE_BYTE_TO_HREGS
    BSF      INTCON0, GIE        ; re-enable interrupts
    BCF      NVMCON1, WREN        ; disable write to memory
```

15.10 Reset

The DMA registers are set to the default state on any Reset. The registers are also reset to the default state when the enable bit is cleared (DMA1CON1bits.EN=0).

15.11 Power Saving Mode Operation

The DMA utilizes system clocks and it is treated as a peripheral when it comes to power saving operations. Like other peripherals, the DMA also uses Peripheral Module Disable bits to further tailor its operation in low-power states.

15.11.1 SLEEP MODE

When the device enters Sleep mode, the system clock to the module is shut down, therefore no DMA operation is supported in Sleep. Once the system clock is disabled, the requisite read and write clocks are also disabled, without which the DMA cannot perform any of its tasks.

Any transfers that may be in progress are resumed on exiting from Sleep mode. Register contents are not affected by the device entering or leaving Sleep mode. It is recommended that DMA transactions be allowed to finish before entering Sleep mode.

15.11.2 IDLE MODE

In IDLE mode, all of the system clocks (including the read and write clocks) are still operating but the CPU is not using them to save power.

Therefore, every instruction cycle is available to the system arbiter and if the bubble is granted to the DMA, it may be utilized to move data.

15.11.3 DOZE MODE

Similar to the Idle mode, the CPU does not utilize all of the available instruction cycles slots that are available to it in order to save power. It only executes instructions based on its settings from the Doze settings.

Therefore, every instruction not used by the CPU is available for system arbitration and may be utilized by the DMA if granted by the arbiter.

15.11.4 PERIPHERAL MODULE DISABLE

The Peripheral Module Disable (PMD) registers provide a method to disable DMA by gating all clock sources supplied to it. The respective DMAxMD bit needs to be set in order to disable the DMA.

15.12 DMA Register Interfaces

The DMA can transfer data to any GPR or SFR location. For better user accessibility, some of the more commonly used SFR spaces have their Mirror registers placed in Bank 64 (0x4000-0x40FF). These Mirror registers can be only accessed through the DMA Source and Destination Address registers. Refer to [Table 4-3](#) for details about Bank 64 Registers.

PIC18(L)F26/27/45/46/47/55/56/57K42

REGISTER 16-7: SLRCONx: SLEW RATE CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SLRx7	SLRx6	SLRx5	SLRx4	SLRx3	SLRx2	SLRx1	SLRx0
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

bit 7-0

SLRx<7:0>: Slew Rate Control on Pins Rx<7:0>, respectively

1 = Port pin slew rate is limited

0 = Port pin slews at maximum rate

TABLE 16-8: SLEW RATE CONTROL REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SLRCONA	SLRA7	SLRA6	SLRA5	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0
SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	SLRB3	SLRB2	SLRB1	SLRB0
SLRCONC	SLRC7	SLRC6	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0
SLRCOND ⁽¹⁾	SLRD7	SLRD6	SLRD5	SLRD4	SLRD3	SLRD2	SLRD1	SLRD0
SLRCONE ⁽¹⁾	—	—	—	—	—	SLRE2	SLRE1	SLRE0
SLRCONF ⁽²⁾	SLRF7	SLRF6	SLRF5	SLRF4	SLRF3	SLRF2	SLRF1	SLRF0

Note 1: Unimplemented in PIC18(L)F26/27K42.

Note 2: Unimplemented in PIC18(L)F26/27/45/46/47K42.

PIC18(L)F26/27/45/46/47/55/56/57K42

23.5 Register Definitions: CCP Control

Long bit name prefixes for the CCP peripherals are shown below. Refer to [Section 1.3.2.2 “Long Bit Names”](#) for more information.

Peripheral	Bit Name Prefix
CCP1	CCP1
CCP2	CCP2
CCP3	CCP3
CCP4	CCP4

REGISTER 23-1: CCPxCON: CCPx CONTROL REGISTER

R/W-0/0	U-0	R-x	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	OUT	FMT	MODE<3:0>			
bit 7				bit 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 7 **EN:** CCP Module Enable bit
 1 = CCP is enabled
 0 = CCP is disabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **OUT:** CCPx Output Data bit (read-only)

bit 4 **FMT:** CCPW (pulse-width) Alignment bit
 MODE = Capture mode:
 Unused
 MODE = Compare mode:
 Unused
 MODE = PWM mode:
 1 = Left-aligned format
 0 = Right-aligned format

bit 3-0 **MODE<3:0>:** CCPx Mode Select bits

MODE	Operating Mode	Operation	Set CCPxIF
11xx	PWM	PWM operation	Yes
1011	Compare	Pulse output; clear TMR1 ⁽²⁾	Yes
1010		Pulse output	Yes
1001		Clear output ⁽¹⁾	Yes
1000		Set output ⁽¹⁾	Yes
0111	Capture	Every 16th rising edge of CCPx input	Yes
0110		Every 4th rising edge of CCPx input	Yes
0101		Every rising edge of CCPx input	Yes
0100		Every falling edge of CCPx input	Yes
0011		Every edge of CCPx input	Yes
0010	Compare	Toggle output	Yes
0001		Toggle output; clear TMR1 ⁽²⁾	Yes
0000	Disabled		—

Note 1: The set and clear operations of the Compare mode are reset by setting MODE = 4'b0000 or EN = 0.
Note 2: When MODE = 0001 or 1011, then the timer associated with the CCP module is cleared. TMR1 is the default selection for the CCP module, so it is used for indication purpose only.

26.7 Rising Edge and Reverse Dead Band

In Half-Bridge mode, the rising edge dead band delays the turn-on of the CWGxA output after the rising edge of the CWG data input. In Full-Bridge mode, the reverse dead-band delay is only inserted when changing directions from Forward mode to Reverse mode, and only the modulated output CWGxB is affected.

The CWGxDBR register determines the duration of the dead-band interval on the rising edge of the input source signal. This duration is from 0 to 64 periods of the CWG clock.

Dead band is always initiated on the edge of the input source signal. A count of zero indicates that no dead band is present.

If the input source signal reverses polarity before the dead-band count is completed, then no signal will be seen on the respective output.

The CWGxDBR register value is double-buffered. When EN = 0 ([Register 26-1](#)), the buffer is loaded when CWGxDBR is written. If EN = 1, then the buffer will be loaded at the rising edge following the first falling edge of the data input, after the LD bit ([Register 26-1](#)) is set. Refer to [Figure 26-12](#) for an example.

26.8 Falling Edge and Forward Dead Band

In Half-Bridge mode, the falling edge dead band delays the turn-on of the CWGxB output at the falling edge of the CWG data input. In Full-Bridge mode, the forward dead-band delay is only inserted when changing directions from Reverse mode to Forward mode, and only the modulated output CWGxD is affected.

The CWGxDBF register determines the duration of the dead-band interval on the falling edge of the input source signal. This duration is from zero to 64 periods of CWG clock.

Dead-band delay is always initiated on the edge of the input source signal. A count of zero indicates that no dead band is present.

If the input source signal reverses polarity before the dead-band count is completed, then no signal will be seen on the respective output.

The CWGxDBF register value is double-buffered. When EN = 0 ([Register 26-1](#)), the buffer is loaded when CWGxDBF is written. If EN = 1, then the buffer will be loaded at the rising edge following the first falling edge of the data input after the LD ([Register 26-1](#)) is set. Refer to [Figure 26-13](#) for an example.

PIC18(L)F26/27/45/46/47/55/56/57K42

TABLE 27-1: CLCx DATA INPUT SELECTION

DyS<5:0> Value	CLCx Input Source
111111 [63]	Reserved
.	
.	
.	
110100 [52]	Reserved
110011 [51]	CWG3B_out
110010 [50]	CWG3A_out
110001 [49]	CWG2B_out
110000 [48]	CWG2A_out
101111 [47]	CWG1B_out
101110 [46]	CWG1A_out
101101 [45]	SS1
101100 [44]	SCK1
101011 [43]	SDO1
101010 [42]	Reserved
101001 [41]	UART2_tx_out
101000 [40]	UART1_tx_out
100111 [39]	CLC4_out
100110 [38]	CLC3_out
100101 [37]	CLC2_out
100100 [36]	CLC1_out
100011 [35]	DSM1_out
100010 [34]	IOC_flag
100001 [33]	ZCD_out
100000 [32]	CMP2_out
011111 [31]	CMP1_out
011110 [30]	NCO1_out
011101 [29]	Reserved
011100 [28]	Reserved
011011 [27]	PWM8_out
011010 [26]	PWM7_out
011001 [25]	PWM6_out
011000 [24]	PWM5_out
010111 [23]	CCP4_out
010110 [22]	CCP3_out
010101 [21]	CCP2_out
010100 [20]	CCP1_out
010011 [19]	SMT1_out
010010 [18]	TMR6_out
010001 [17]	TMR5_overflow
010000 [16]	TMR4_out
001111 [15]	TMR3_overflow

TABLE 27-1: CLCx DATA INPUT SELECTION (CONTINUED)

DyS<5:0> Value	CLCx Input Source
001110 [14]	TMR2_out
001101 [13]	TMR1_overflow
001100 [12]	TMR0_overflow
001011 [11]	CLKR_out
001010 [10]	ADCRC
001001 [9]	SOSC
001000 [8]	MFINTOSC (32 kHz)
000111 [7]	MFINTOSC (500 kHz)
000110 [6]	LFINTOSC
000101 [5]	HFINTOSC
000100 [4]	Fosc
000011 [3]	CLCIN3PPS
000010 [2]	CLCIN2PPS
000001 [1]	CLCIN1PPS
000000 [0]	CLCIN0PPS

Measuring VCPINV can be difficult, especially when the waveform is relative to VDD. However, by combining Equations 29-2 and 29-3, the resistor value can be determined from the time difference between the ZCD_output high and low intervals. Note that the time difference, ΔT , is $4 \cdot T_{OFFSET}$. The equation for determining the pull-up and pull-down resistor values from the high and low ZCD_output periods is shown in Equation 29-4.

EQUATION 29-4: PULL-UP/DOWN RESISTOR VALUES

$$R = R_{SERIES} \left(\frac{V_{BIAS}}{V_{PEAK} \left(\sin \left(\pi Freq \frac{\Delta T}{2} \right) \right)} - 1 \right)$$

R is pull-up or pull-down resistor.

VBIAS is VPULLUP when R is pull-up or VDD when R is pull-down.

ΔT is the ZCDOUT high and low period difference.

29.6 Handling VPEAK Variations

If the peak amplitude of the external voltage is expected to vary, the series resistor must be selected to keep the ZCD current source and sink below the design maximum range of $\pm 600 \mu A$ and above a reasonable minimum range. A general rule of thumb is that the maximum peak voltage can be no more than six times the minimum peak voltage. To ensure that the maximum current does not exceed $\pm 600 \mu A$ and the minimum is at least $\pm 100 \mu A$, compute the series resistance as shown in Equation 29-5. The compensating pull-up for this series resistance can be determined with Equation 29-3 because the pull-up value is not dependent to the peak voltage.

EQUATION 29-5: SERIES R FOR V RANGE

$$R_{SERIES} = \frac{V_{MAXPEAK} + V_{MINPEAK}}{7 \times 10^{-4}}$$

29.7 Operation During Sleep

The ZCD current sources and interrupts are unaffected by Sleep.

29.8 Effects of a Reset

The ZCD circuit can be configured to default to the active or inactive state on Power-on-Reset (POR). When the \overline{ZCD} Configuration bit is cleared, the ZCD circuit will be active at POR. When the \overline{ZCD} Configuration bit is set, the SEN bit of the ZCDCON register must be set to enable the ZCD module.

29.9 Disabling the ZCD Module

The ZCD module can be disabled in two ways:

1. Configuration Word 2H has the \overline{ZCD} bit which disables the ZCD module when set, but it can be enabled using the SEN bit of the ZCDCON register (Register 29-1). If the \overline{ZCD} bit is clear, the ZCD is always enabled.
2. The ZCD can also be disabled using the ZCDMD bit of the respective PMD2 register (Register 19-3). This is subject to the status of the \overline{ZCD} bit.

31.3 Asynchronous Address Mode

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems.

When Asynchronous Address mode is enabled, all data is transmitted and received as 9-bit characters. The 9th bit determines whether the character is an address or data. When the 9th bit is set, the eight Least Significant bits are the address. When the 9th bit is clear, the Least Significant bits are data. In either case, the 9th bit is stored in PERIF when the byte is written to the receive FIFO. When PERIE is also set, the RXIF will be suppressed, thereby suspending DMA transfers allowing software to process the received address.

An address character will enable all receivers that match the address and disable all other receivers. Once a receiver is enabled, all non-address characters will be received until an address character is received that does not match.

31.3.1 ADDRESS MODE TRANSMIT

The UART transmitter is enabled for asynchronous address operation by configuring the following control bits:

- TXEN = 1
- MODE<3:0> = 0100
- UxBRGH:L = desired baud rate
- RxyPPS = code for desired output pin
- ON = 1

Addresses are sent by writing to the UxP1L register. This transmits the written byte with the 9th bit set, which indicates that the byte is an address.

Data is sent by writing to the UxTXB register. This transmits the written byte with the 9th bit cleared, which indicates that the byte is data.

To send data to a particular device on the transmission bus, first transmit the address of the intended device. All subsequent data will be accepted only by that device until an address of another device is transmitted.

Writes to UxP1L take precedence over writes to UxTXB. When both the UxP1L and UxTXB registers are written while the TSR is busy, the next byte to be transmitted will be from UxP1L.

To ensure that all data intended for one device is sent before the address is changed, wait until the TXMTIF bit is high before writing UxP1L with the new address.

31.3.2 ADDRESS MODE RECEIVE

The UART receiver is enabled for asynchronous address operation by configuring the following control bits:

- RXEN = 1
- MODE<3:0> = 0100
- UxBRGH:L = desired baud rate
- RXPPS = code for desired input pin
- Input pin ANSEL bit = 0
- UxP2L = receiver address
- UxP3L = address mask
- ON = 1

In Address mode, no data will be transferred to the input FIFO until a valid address is received. This is the default state. Any of the following conditions will cause the UART to revert to the default state:

- ON = 0
- RXEN = 0
- Received address does not match

When a character with the 9th bit set is received, the Least Significant eight bits of that character will be qualified by the values in the UxP2L and UxP3L registers.

The byte is XOR'd with UxP2L then AND'd with UxP3L. A match occurs when the result is 0h, in which case, the unaltered received character is stored in the receive FIFO, thereby setting the UxRXIF interrupt bit. The 9th bit is stored in the corresponding PERIF bit, identifying this byte as an address.

An address match also enables the receiver for all data such that all subsequent characters without the 9th bit set will be stored in the receive FIFO.

When the 9th bit is set and a match does not occur, the character is not stored in the receive FIFO and all subsequent data is ignored.

The UxP3L register mask allows a range of addresses to be accepted. Software can then determine the sub-address of the range by processing the received address character.

32.0 SERIAL PERIPHERAL INTERFACE (SPI) MODULE

32.1 SPI Module Overview

The SPI (Serial Peripheral Interface) module is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a Chip Select known as Slave Select. Example slave devices include serial EEPROMs, shift registers, display drivers, A/D converters, or another PIC® device.

The SPI bus specifies four signal connections:

- Serial Clock (SCK)
- Serial Data Out (SDO)
- Serial Data IN (SDI)
- Slave Select (\overline{SS})

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Polarity and Edge Select
- SDI, SDO, and SS Polarity Control
- Separate Transmit and Receive Enables
- Slave Select Synchronization
- Daisy-chain connection of slave devices
- Separate Transmit and Receive Buffers with 2-byte FIFO and DMA capabilities

[Figure 32-1](#) shows the block diagram of the SPI module.

PIC18(L)F26/27/45/46/47/55/56/57K42

REGISTER 33-15: I2CXADR3: I²C ADDRESS 3 REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0
ADR14	ADR13	ADR12	ADR11	ADR10	ADR9	ADR8	—
bit 15							bit 8
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	—
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

HS = Hardware set

HC = Hardware clear

bit 7-0

ADR<7-0>: Address 3 bits

MODE<2:0> = 000 | 110 - 7-bit Slave/Multi-Master Modes

ADR<7:1>: 7-bit Slave Address

ADR<0>: Unused in this mode; bit state is a don't care

MODE<2:0> = 001 | 111 - 7-bit Slave/Multi-Master Mode with Masking

MSK1<7:1>: 7-bit Slave Address

MSK1<0>: Unused in this mode; bit state is a don't care

MODE<2:0> = 010 - 10-Bit Slave Mode

ADR<14-10>: Bit pattern sent by master is fixed by I²C specification and must be equal to '11110'. However, these bit values are compared by hardware to the received data to determine a match. It is up to the user to set these bits as '11110'

ADR<9-8>: Two Most Significant bits of 10-bit address

MODE<2:0> = 011 - 10-Bit Slave Mode with Masking

MSK0<14-8>: The received address byte, bit *n*, is compared to I2CxADR0 to detect I²C address match

PIC18(L)F26/27/45/46/47/55/56/57K42

36.7 Register Definitions: ADC Control

REGISTER 36-1: ADCON0: ADC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0	R/W-0/0	U-0	R/W/HC-0
ON	CONT	—	CS	—	FM	—	GO
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Bit is cleared by hardware

- bit 7 **ON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled
- bit 6 **CONT:** ADC Continuous Operation Enable bit
1 = GO is retriggered upon completion of each conversion trigger until ADTIF is set (if ADSOI is set) or until GO is cleared (regardless of the value of ADSOI)
0 = ADC is cleared upon completion of each conversion trigger
- bit 5 **Unimplemented:** Read as '0'
- bit 4 **CS:** ADC Clock Selection bit
1 = Clock supplied from FRC dedicated oscillator
0 = Clock supplied by FOSC, divided according to ADCLK register
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **FM:** ADC results Format/alignment Selection
1 = ADRES and PREV data are right-justified
0 = ADRES and PREV data are left-justified, zero-filled
- bit 1 **Unimplemented:** Read as '0'
- bit 0 **GO:** ADC Conversion Status bit⁽¹⁾
1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle. The bit is cleared by hardware as determined by the CONT bit
0 = ADC conversion completed/not in progress

Note 1: This bit requires ON bit to be set.

- 2:** If cleared by software while a conversion is in progress, the results of the conversion up to this point will be transferred to ADRES and the state machine will be reset, but the ADIF interrupt flag bit will not be set; filter and threshold operations will not be performed.

PIC18(L)F26/27/45/46/47/55/56/57K42

REGISTER 36-8: ADPCH: ADC POSITIVE CHANNEL SELECTION REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	ADPCH<5:0>					
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **ADPCH<5:0>:** ADC Positive Input Channel Selection bits

111111 = FVR buffer 2 ⁽²⁾	010111 = ANC7
111110 = FVR buffer 1 ⁽²⁾	010110 = ANC6
111101 = DAC1 output ⁽¹⁾	010101 = ANC5
111100 = Temperature indicator ⁽³⁾	010100 = ANC4
111011 = Vss (Analog Ground)	010011 = ANC3
111010 = Reserved. No channel connected.	010010 = ANC2
•	010001 = ANC1
•	010000 = ANC0
•	001111 = ANB7
110000 = Reserved. No channel connected.	001110 = ANB6
101111 = ANF7 ⁽⁴⁾	001101 = ANB5
101110 = ANF6 ⁽⁴⁾	001100 = ANB4
101101 = ANF5 ⁽⁴⁾	001011 = ANB3
101100 = ANF4 ⁽⁴⁾	001010 = ANB2
101011 = ANF3 ⁽⁴⁾	001001 = ANB1
101010 = ANF2 ⁽⁴⁾	001000 = ANB0
101001 = ANF1 ⁽⁴⁾	000111 = ANA7
101000 = ANF0 ⁽⁴⁾	000110 = ANA6
100111 = Reserved. No channel connected.	000101 = ANA5
•	000100 = ANA4
•	000011 = ANA3
100011 = Reserved. No channel connected.	000010 = ANA2
100010 = ANE2 ⁽⁵⁾	000001 = ANA1
100001 = ANE1 ⁽⁵⁾	000000 = ANA0
100000 = ANE0 ⁽⁵⁾	
011111 = AND7 ⁽⁵⁾	
011110 = AND6 ⁽⁵⁾	
011101 = AND5 ⁽⁵⁾	
011100 = AND4 ⁽⁵⁾	
011011 = AND3 ⁽⁵⁾	
011010 = AND2 ⁽⁵⁾	
011001 = AND1 ⁽⁵⁾	
011000 = AND0 ⁽⁵⁾	

- Note**
- 1: See [Section 37.0 “5-Bit Digital-to-Analog Converter \(DAC\) Module”](#) for more information.
 - 2: See [Section 34.0 “Fixed Voltage Reference \(FVR\)”](#) for more information.
 - 3: See [Section 35.0 “Temperature Indicator Module”](#) for more information.
 - 4: Reserved on PIC18(L)F26/27/45/46/47K42 parts.
 - 5: Reserved on PIC18(L)F26/27K42 parts.

37.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The positive input source ($V_{SOURCE+}$) of the DAC can be connected to:

- FVR Buffer
- External V_{REF+} pin
- V_{DD} supply voltage

The negative input source ($V_{SOURCE-}$) of the DAC can be connected to:

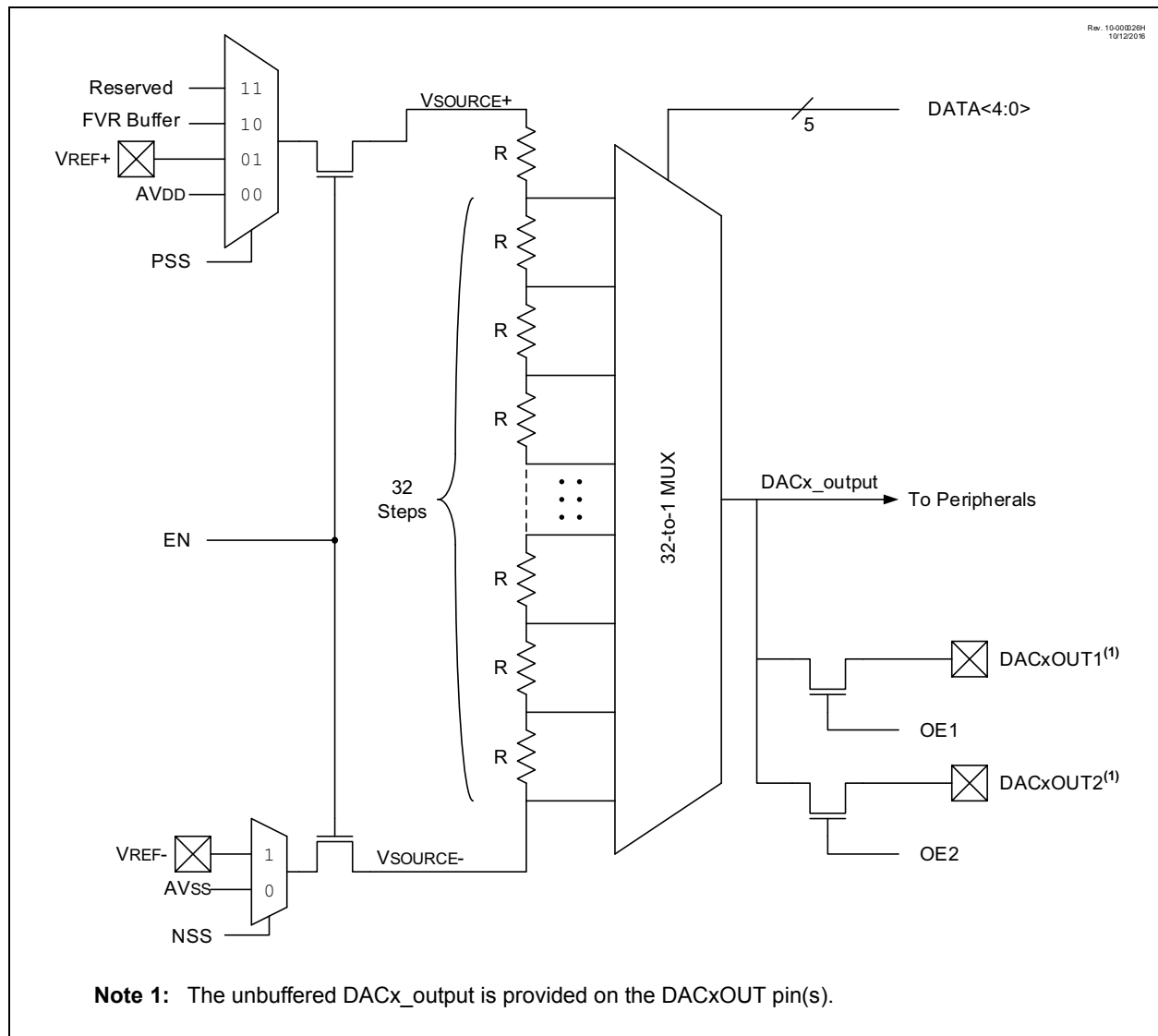
- External V_{REF-} pin
- V_{SS}

The output of the DAC ($DAC1_output$) can be selected as a reference voltage to the following:

- Comparator positive input
- ADC input channel
- $DAC1OUT1$ pin
- $DAC1OUT2$ pin

The Digital-to-Analog Converter (DAC) can be enabled by setting the EN bit of the $DAC1CON0$ register.

FIGURE 37-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM



PIC18(L)F26/27/45/46/47/55/56/57K42

37.6 Register Definitions: DAC Control

Long bit name prefixes for the DAC peripheral is shown below. Refer to [Section 1.3.2.2 “Long Bit Names”](#) for more information.

Peripheral	Bit Name Prefix
DAC1	DAC1

REGISTER 37-1: DAC1CON0: DAC CONTROL REGISTER

R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
EN	—	OE1	OE2	PSS<1:0>	—	NSS	
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

- bit 7 **EN:** DAC Enable bit
1 = DAC is enabled
0 = DAC is disabled⁽¹⁾
- bit 6 **Unimplemented:** Read as '0'
- bit 5 **OE1:** DAC Voltage Output Enable bit
1 = DAC voltage level is output on the DAC1OUT1 pin
0 = DAC voltage level is disconnected from the DAC1OUT1 pin
- bit 4 **OE2:** DAC Voltage Output Enable bit
1 = DAC voltage level is output on the DAC1OUT2 pin
0 = DAC voltage level is disconnected from the DAC1OUT2 pin
- bit 3-2 **PSS<1:0>:** DAC Positive Source Select bit
11 = Reserved
10 = FVR buffer 2
01 = VREF+
00 = VDD
- bit 1 **Unimplemented:** Read as '0'
- bit 0 **NSS:** DAC Negative Source Select bit
1 = VREF-
0 = VSS

Note 1: DAC1OUTx output pins are still active.

43.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

43.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

43.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/librarian features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

43.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

45.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified V_{DD} range). This is for **information only** and devices are ensured to operate properly only within the specified range.

Unless otherwise noted, all graphs apply to both the L and LF devices.

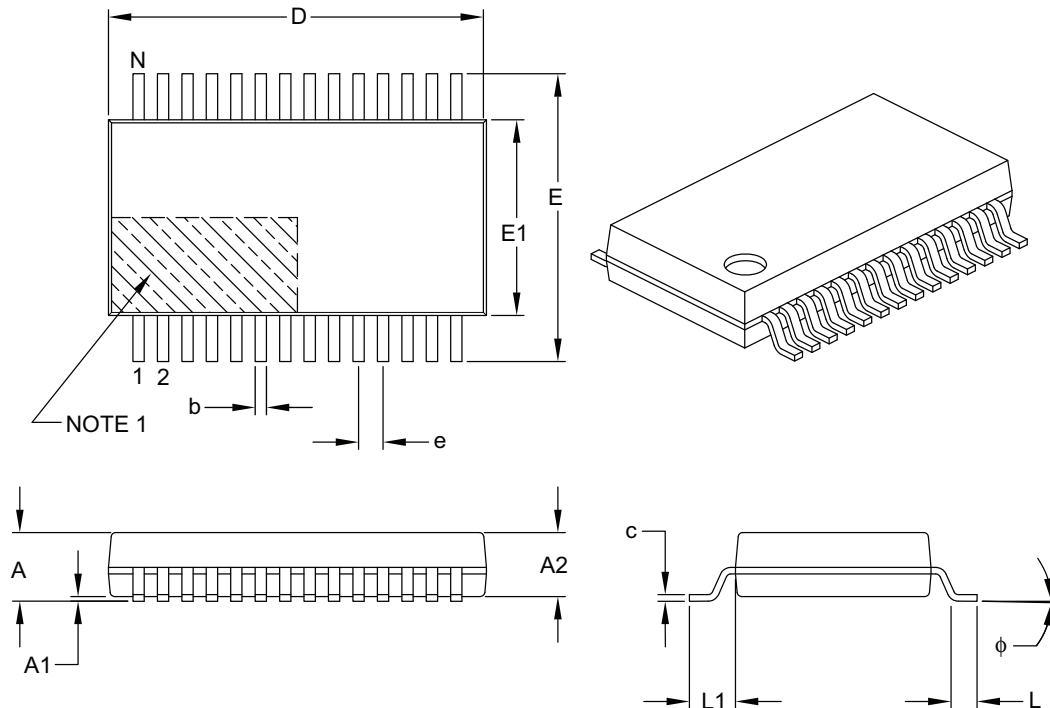
Note:	The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.
--------------	--

“Typical” represents the mean of the distribution at 25°C. “Maximum”, “Max.”, “Minimum” or “Min.” represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over each temperature range.

PIC18(L)F26/27/45/46/47/55/56/57K42

28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	–	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B