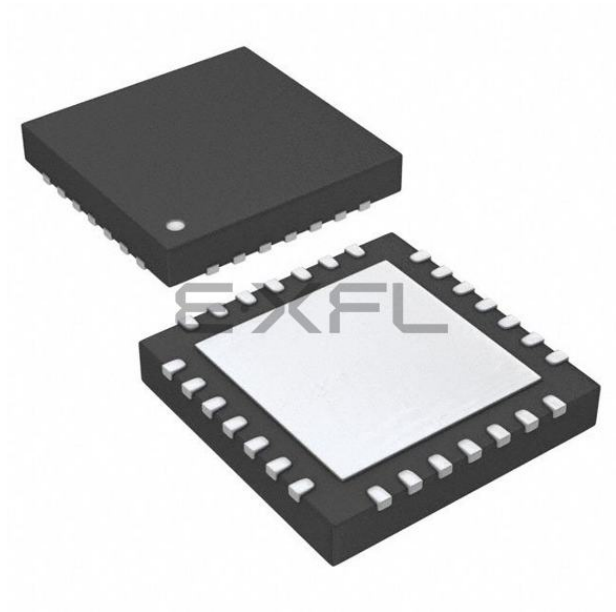


Welcome to E-XFL.COM

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	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
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-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf25k42t-i-mv

3.0 PIC18 CPU

This family of devices contains a PIC18 8-bit CPU core based on the modified Harvard architecture. The PIC18 CPU supports:

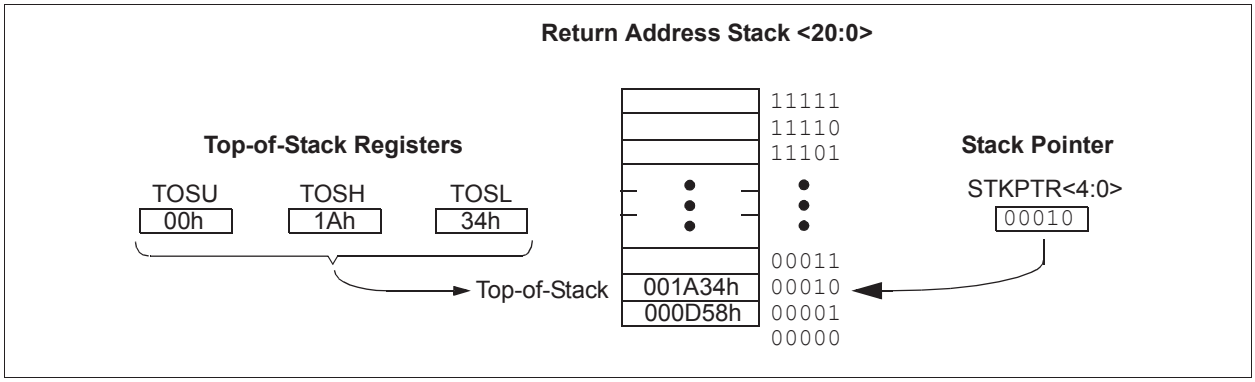
- System Arbitration which decides memory access allocation depending on user priorities
- Vectored Interrupt capability with automatic two level deep context saving
- 31-level deep hardware stack with overflow and underflow reset capabilities
- Support Direct, Indirect, and Relative Addressing modes
- 8x8 Hardware Multiplier

TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH CPU

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
ISRPR	—	—	—	—	—	ISRPR2	ISRPR1	ISRPR0	21
MAINPR	—	—	—	—	—	MAINPR2	MAINPR1	MAINPR0	21
DMA1PR	—	—	—	—	—	DMA1PR2	DMA1PR1	DMA1PR0	21
DMA2PR	—	—	—	—	—	DMA2PR2	DMA2PR1	DMA2PR0	22
SCANPR	—	—	—	—	—	SCANPR2	SCANPR1	SCANPR0	22
PRLOCK	—	—	—	—	—	—	—	PRLOCKED	22

Legend: — = Unimplemented location, read as '0'.

FIGURE 4-1: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



4.2.5.2 Return Stack Pointer (STKPTR)

The STKPTR register (Register 4-1) contains the Stack Pointer value. The STKOVF (Stack Overflow) Status bit and the STKUNF (Stack Underflow) Status bit can be accessed using the PCON0 register. The value of the Stack Pointer can be 0 through 31. On Reset, the Stack Pointer value will be zero. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System (RTOS) for stack maintenance. After the PC is pushed onto the stack 32 times (without popping any values off the stack), the STKOVF bit is set. The STKOVF bit is cleared by software or by a POR. The action that takes place when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 5.1 "Configuration Words"** for a description of the device Configuration bits.)

If STVREN is set (default), a Reset will be generated and a Stack Overflow will be indicated by the STKOVF bit when the 32nd push is initiated. This includes `CALL` and `CALLW` instructions, as well as stacking the return address during an interrupt response. The STKOVF bit will remain set and the Stack Pointer will be set to zero.

If STVREN is cleared, the STKOVF bit will be set on the 32nd push and the Stack Pointer will remain at 31 but no Reset will occur. Any additional pushes will overwrite the 31st push but the STKPTR will remain at 31.

Setting STKOVF = `1` in software will change the bit, but will not generate a Reset.

The STKUNF bit is set when a stack pop returns a value of zero. The STKUNF bit is cleared by software or by POR. The action that takes place when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 5.1 "Configuration Words"** for a description of the device Configuration bits).

If STVREN is set (default) and the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC, it will set the STKUNF bit and a Reset will be generated. This condition can be generated by the `RETURN`, `RETLW` and `RETFIE` instructions.

When STVREN = `0`, STKUNF will be set but no Reset will occur.

Note:	Returning a value of zero to the PC on an underflow has the effect of vectoring the program to the Reset vector, where the stack conditions can be verified and appropriate actions can be taken. This is not the same as a Reset, as the contents of the SFRs are not affected.

4.2.5.3 `PUSH` and `POP` Instructions

Since the Top-of-Stack is readable and writable, the ability to push values onto the stack and pull values off the stack without disturbing normal program execution is a desirable feature. The PIC18 instruction set includes two instructions, `PUSH` and `POP`, that permit the TOS to be manipulated under software control. TOSU, TOSH and TOSL can be modified to place data or a return address on the stack.

The `PUSH` instruction places the current PC value onto the stack. This increments the Stack Pointer and loads the current PC value onto the stack.

The `POP` instruction discards the current TOS by decrementing the Stack Pointer. The previous value pushed onto the stack then becomes the TOS value.

TABLE 4-11: REGISTER FILE SUMMARY FOR PIC18(L)F24/25K42 DEVICES (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
3BEDh	DMA1DPTRH	DPTR								00000000
3BECh	DMA1DPTRL	DPTR								00000000
3BEBh	DMA1DCNTH	—	—	—	—	DCNT				----0000
3BEAh	DMA1DCNTL	DCNT								00000000
3BE9h	DMA1BUF	BUF								xxxxxxxx
3BE8h - 3BE0h	—	Unimplemented								—
3BDFh	DMA2SIRQ	SIRQ								00000000
3BDEh	DMA2AIRQ	AIRQ								00000000
3BDDh	DMA2CON1	EN	SIRQEN	DGO	—	—	AIRQEN	—	XIP	000--0-0
3BDCh	DMA2CON0	DMODE		DSTP	SMR		SMODE		SSTP	00000000
3BDBh	DMA2SSAU	—	—	SSA						--000000
3BDAh	DMA2SSAH	SSA								00000000
3BD9h	DMA2SSAL	SSA								00000000
3BD8h	DMA2SSZH	—	—	—	—	SSZ				----0000
3BD7h	DMA2SSZL	SSZ								00000000
3BD6h	DMA2SPTRU	—	—	SPTR						--000000
3BD5h	DMA2SPTRH	SPTR								00000000
3BD4h	DMA2SPTRL	SPTR								00000000
3BD3h	DMA2SCNTH	—	—	—	—	SCNT				----0000
3BD2h	DMA2SCNTL	SCNT								00000000
3BD1h	DMA2DSAH	DSA								00000000
3BD0h	DMA2DSAL	SSA								00000000
3BCFh	DMA2DSZH	—	—	—	—	DSZ				----0000
3BCEh	DMA2DSZL	DSZ								xxxxxxxx
3BCDh	DMA2DPTRH	DPTR								00000000
3BCCh	DMA2DPTRL	DPTR								00000000
3BCBh	DMA2DCNTH	—	—	—	—	DCNT				----0000
3BCAh	DMA2DCNTL	DCNT								00000000
3BC9h	DMA2BUF	BUF								xxxxxxxx
3BC8h - 3AEBh	—	Unimplemented								—
3AEAh	U2CTSPPS	—	—	—	U2CTSPPS					---01101
3AE8h	U2RXPPS	—	—	—	U2RXPPS					---01111
3AE7h	U1CTSPPS	—	—	—	U1CTSPPS					---10101
3AE5h	U1RXPPS	—	—	—	U1RXPPS					---10111
3AE4h	I2C2SDAPPS	—	—	—	I2C2SDAPPS					---00010
3AE3h	I2C2SCLPPS	—	—	—	I2C2SCLPPS					---01001
3AE2h	I2C1SDAPPS	—	—	—	I2C1SDAPPS					---10100
3AE1h	I2C1SCLPPS	—	—	—	I2C1SCLPPS					---10011
3AE0h	SPI1SSPPS	—	—	—	SPI1SSPPS					---00101
3ADFh	SPI1SDIPPS	—	—	—	SPI1SDIPPS					---10100
3ADEh	SPI1SCKPPS	—	—	—	SPI1SCKPPS					---10011
3ADDh	ADACTPPS	—	—	—	ADACTPPS					---01100
3ADCh	CLCIN3PPS	—	—	—	CLCIN3PPS					---01111
3ADBh	CLCIN2PPS	—	—	—	CLCIN2PPS					---01110
3ADAh	CLCIN1PPS	—	—	—	CLCIN1PPS					---00001
3AD9h	CLCIN0PPS	—	—	—	CLCIN0PPS					---00000
3AD8h	MD1SRCPPS	—	—	—	MD1SRCPPS					---00101

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition
Note 1: Not present in LF devices.

REGISTER 11-6: PIR3: PERIPHERAL INTERRUPT REGISTER 3

R/W/HS-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
TMR0IF	U1IF	U1EIF	U1TXIF	U1RXIF	I2C1EIF	I2C1IF	I2C1TXIF
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 = Bit is set in hardware

bit 7 **TMR0IF:** TMR0 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 6 **U1IF:** UART1 Interrupt Flag bit
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 5 **U1EIF:** UART1 Framing Error Interrupt Flag bit
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 4 **U1TXIF:** UART1 Transmit Interrupt Flag bit
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 3 **U1RXIF:** UART1 Receive Interrupt Flag bit
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 2 **I2C1EIF:** I^2C1 Error Interrupt Flag bit
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 1 **I2C1IF:** I^2C1 Interrupt Flag bit
1 = Interrupt has occurred
0 = Interrupt event has not occurred

bit 0 **I2C1TXIF:** I^2C1 Transmit Interrupt Flag bit
1 = Interrupt has occurred
0 = Interrupt event has not occurred

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 11-11: PIR8: PERIPHERAL INTERRUPT REGISTER 8

R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	U-0	U-0	U-0
TMR5GIF	TMR5IF	—	—	—	—	—	—
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 = Bit is set in hardware

bit 7 **TMR5GIF:** TMR5 Gate Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 6 **TMR5IF:** TMR5 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 5-0 **Unimplemented:** Read as '0'

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 11-12: PIR9: PERIPHERAL INTERRUPT REGISTER 9

U-0	U-0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	—	—	CLC3IF	CWG3IF	CCP3IF	TMR6IF
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 7-4 **Unimplemented:** Read as '0'

bit 3 **CLC3IF:** CLC3 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 2 **CWG3IF:** CWG3 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 1 **CCP3IF:** CCP3 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

bit 0 **TMR6IF:** TMR6 Interrupt Flag bit
1 = Interrupt has occurred (must be cleared by software)
0 = Interrupt event has not occurred

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit, or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

15.3.5 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

EXAMPLE 15-5: DATA EEPROM READ

```
; Data Memory Address to read
            CLRF      NVMCON1          ; Setup Data EEPROM Access
            MOVF      EE_ADDRL, W      ;
            MOVWF     NVMADRL          ; Setup Address
            BSF       NVMCON1, RD      ; Issue EE Read
            MOVF      NVMDAT, W        ; W = EE_DATA
```

EXAMPLE 15-6: DATA EEPROM WRITE

```
; Data Memory Address to write
            CLRF      NVMCON1          ; Setup Data EEPROM Access
            MOVF      EE_ADDRL, W      ;
            MOVWF     NVMADRL          ; Setup Address
; Data Memory Value to write
            MOVF      EE_DATA, W       ;
            MOVWF     NVMDAT           ;
; Enable writes
            BSF       NVMCON1, WREN    ;
; Disable interrupts
            BCF       INTCON0, GIE     ;
; Required unlock sequence
            MOVLW     55h              ;
            MOVWF     NVMCON2          ;
            MOVLW     AAh              ;
            MOVWF     NVMCON2          ;
; Set WR bit to begin write
            BSF       NVMCON1, WR      ;
; Enable INT
            BSF       INTCON0, GIE     ;
; Wait for interrupt, write done
            SLEEP                      ;
; Disable writes
            BCF       NVMCON1, WREN    ;
```

15.3.6 OPERATION DURING CODE-PROTECT

Data EEPROM Memory has its own code-protect bits in Configuration Words. External read and write operations are disabled if code protection is enabled.

If the Data EEPROM is write-protected or if NVMADR points an invalid address location, the WR bit is cleared without any effect. WRERR is signaled in this scenario.

15.3.7 PROTECTION AGAINST SPURIOUS WRITE

There are conditions when the user may not want to write to the Data EEPROM Memory. To protect against spurious EEPROM writes, various mechanisms have been implemented. On power-up, the WREN bit is cleared. In addition, writes to the EEPROM are blocked during the Power-up Timer period (TPWRT).

The unlock sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch or software malfunction.

16.8 Scanner Module Overview

The Scanner allows segments of the Program Flash Memory or Data EEPROM, to be read out (scanned) to the CRC Peripheral. The Scanner module interacts with the CRC module and supplies it data one word at a time. Data is fetched from the address range defined by SCANLADR registers up to the SCANHADR registers.

The Scanner begins operation when the SGO bit is set (SCANCON0 Register) and ends when either SGO is cleared by the user or when SCANLADR increments past SCANHADR. The SGO bit is also cleared by clearing the EN bit (CRCCON0 register).

16.9 Configuring the Scanner

The scanner module may be used in conjunction with the CRC module to perform a CRC calculation over a range of program memory or Data EEPROM addresses. In order to set up the scanner to work with the CRC, perform the following steps:

1. Set up the CRC module (See **Section 16.7 "Configuring the CRC"**) and enable the Scanner module by setting the EN bit in the SCANCON0 register.
2. Choose which memory region the Scanner module should operate on and set the MREG bit of the SCANCON0 register appropriately.
3. If trigger is used for scanner operation, set the TRIGEN bit of the SCANCON0 register and select the trigger source using SCANTRIG register. Select the trigger source using SCANTRIG register and then set the TRIGEN bit of the SCANCON0 register. See Table 16-2 for Scanner Operation.
4. If Burst mode of operation is desired, set the BURSTMD bit (SCANCON0 register). See Table 16-2 for Scanner Operation.
5. Set the SCANLADRL/H/U and SCANHADRL/H/U registers with the beginning and ending locations in memory that are to be scanned.
6. Select the priority level for the Scanner module (See **Section 3.1 "System Arbitration"**) and lock the priorities (See **Section 3.1.1 "Priority Lock"**).
7. Both CRCEN and CRCGO bits must be enabled to use the scanner. Setting the SGO bit will start the scanner operation.

16.10 Scanner Interrupt

The scanner will trigger an interrupt when the SCANLADR increments past SCANHADR. The SCANIF bit can only be cleared in software.

16.11 Scanning Modes

The interaction of the scanner with the system operation is controlled by the priority selection in the System Arbiter (see **Section 3.2 "Memory Access Scheme"**). Additionally, BURSTMD and TRIGEN also determine the operation of the Scanner.

16.11.1 TRIGEN = 0, BURSTMD = 0

In this case, the memory access request is granted to the scanner if no other higher priority source is requesting access.

All sources with lower priority than the scanner will get the memory access cycles that are not utilized by the scanner.

16.11.2 TRIGEN = 1, BURSTMD = 0

In this case, the memory access request is generated when the CRC module is ready to accept.

The memory access request is granted to the scanner if no other higher priority source is requesting access. All sources with lower priority than the scanner will get the memory access cycles that are not utilized by the scanner.
The memory access request is granted to the scanner if no other higher priority source is requesting access. All sources with lower priority than the scanner will get the memory access cycles that are not utilized by the scanner.

16.11.3 TRIGEN = x, BURSTMD = 1

In this case, the memory access is always requested by the scanner.

The memory access request is granted to the scanner if no other higher priority source is requesting access. The memory access cycles will not be granted to lower priority sources than the scanner until it completes operation i.e. SGO = 0 (SCANCON0 register)

Note:	If TRIGEN = 1 and BURSTMD = 1, the user should ensure that the trigger source is active for the Scanner operation to complete.

REGISTER 17-1: DMAxCON0 – DMAx CONTROL REGISTER 0

R/W-0/0	R/W/HC-0/0	R/W/HS/HC-0/0	U-0	U-0	R/W/HC-0/0	U-0	R/HS/HC-0/0
EN	SIRQEN	DGO	—	—	AIRQEN	—	XIP
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n/n = Value at POR and BOR/Value at all other Resets		0 = bit is cleared	x = bit is unknown u = bit is unchanged

bit 7 **EN:** DMA Module Enable bit
- 1 = Enables module
- 0 = Disables module

bit 6 **SIRQEN:** Start of Transfer Interrupt Request Enable bits
- 1 = Hardware triggers are allowed to start DMA transfers
- 0 = Hardware triggers are not allowed to start DMA transfers

bit 5 **DGO:** DMA transaction bit
- 1 = DMA transaction is in progress
- 0 = DMA transaction is not in progress

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **AIRQEN:** Abort of Transfer Interrupt Request Enable bits
- 1 = Hardware triggers are allowed to abort DMA transfers
- 0 = Hardware triggers are not allowed to abort DMA transfers

bit 1 **Unimplemented:** Read as '0'

bit 0 **XIP:** Transfer in Progress Status bit
- 1 = The DMAxBUF register currently holds contents from a read operation and has not transferred data to the destination.
- 0 = The DMAxBUF register is empty or has successfully transferred data to the destination address

REGISTER 19-2: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER

U-0	U-0	R/W-0/u
—	—	RxyPPS<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 **RxyPPS<5:0>:** Pin Rxy Output Source Selection bits

RxyPPS<5:0>	Pin Rxy Output Source	Device Configuration		
		PIC18(L)F24/25K42		
6'b11 1111 – 6'b11 0011	Reserved			
6'b11 0010	ADGRDB	A	—	C
6'b11 0001	ADGRDA	A	—	C
6'b11 0000	CWG3D	A	—	C
6'b10 1111	CWG3C	A	—	C
6'b10 1110	CWG3B	A	—	C
6'b10 1101	CWG3A	—	B	C
6'b10 1100	CWG2D	—	B	C
6'b10 1011	CWG2C	—	B	C
6'b10 1010	CWG2B	—	B	C
6'b10 1001	CWG2A	—	B	C
6'b10 1000	DSM1	A	—	C
6'b10 0111	CLKR	—	B	C
6'b10 0110	NCO1	A	—	C
6'b10 0101	TMR0	—	B	C
6'b10 0100	I^2C2 (SDA)	—	B	C
6'b10 0011	I^2C2 (SCL)	—	B	C
6'b10 0010	I^2C1 (SDA)	—	B	C
6'b10 0001	I^2C1 (SCL)	—	B	C
6'b10 0000	SPI1 ($\overline{SS}$)	A	—	C
6'b01 1111	SPI1 (SDO)	—	B	C
6'b01 1110	SPI1 (SCK)	—	B	C
6'b01 1101	C2OUT	A	—	C
6'b01 1100	C1OUT	A	—	C
6'b01 1011 – 6'b01 1001	Reserved			
6'b01 1000	UART2 ($\overline{RTS}$)	—	B	C
6'b01 0111	UART2 (TXDE)	—	B	C
6'b01 0110	UART2 (TX)	—	B	C
6'b01 0101	UART1 ($\overline{RTS}$)	—	B	C
6'b01 0100	UART1 (TXDE)	—	B	C
6'b01 0011	UART1 (TX)	—	B	C
6'b01 0010 – 6'b01 0001	Reserved			

REGISTER 21-5: PMD4: PMD CONTROL REGISTER 4

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0
CWG3MD	CWG2MD	CWG1MD	—	—	—	—	—
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	q = Value depends on condition

bit 7 **CWG3MD:** Disable CWG3 Module bit
1 = CWG3 module disabled
0 = CWG3 module enabled

bit 6 **CWG2MD:** Disable CWG2 Module bit
1 = CWG2 module disabled
0 = CWG2 module enabled

bit 5 **CWG1MD:** Disable CWG1 Module bit
1 = CWG1 module disabled
0 = CWG1 module enabled

bit 4-0 **Unimplemented:** Read as '0'

26.1 PWMx Pin Configuration

All PWM outputs are multiplexed with the PORT data latch. The user must configure the pins as outputs by clearing the associated TRIS bits.

26.1.1 FUNDAMENTAL OPERATION

The PWM module produces a 10-bit resolution output. The PWM timer can be selected using the PxTSEL bits in the CCPTMRS1 register. The default selection for PWMx is T2TMR. Please note that the PWM module operation in the following sections is described with respect to T2TMR. Timer2 and T2PR set the period of the PWM. The PWMxDCL and PWMxDCH registers configure the duty cycle. The period is common to all PWM modules, whereas the duty cycle is independently controlled.

Note: The Timer2 postscaler is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

All PWM outputs associated with Timer2 are set when T2TMR is cleared. Each PWMx is cleared when T2TMR is equal to the value specified in the corresponding PWMxDCH (8 MSb) and PWMxDCL<7:6> (2 LSb) registers. When the value is greater than or equal to T2PR, the PWM output is never cleared (100% duty cycle).

Note: The PWMxDCH and PWMxDCL registers are double buffered. The buffers are updated when Timer2 matches T2PR. Care should be taken to update both registers before the timer match occurs.

26.1.2 PWM OUTPUT POLARITY

The output polarity is inverted by setting the PWMxPOL bit of the PWMxCON register.

26.1.3 PWM PERIOD

The PWM period is specified by the T2PR register of Timer2. The PWM period can be calculated using the formula of Equation 26-1. It is required to have FOSC/4 as clock input to Timer2/4/6 for correct PWM operation.

EQUATION 26-1: PWM PERIOD

$$PWM\ Period = [(T2PR) + 1] \bullet 4 \bullet TOSC \bullet (TMR2\ Prescale\ Value)$$

Note: TOSC = 1/FOSC

When T2TMR is equal to T2PR, the following three events occur on the next increment cycle:

- T2TMR is cleared
- The PWM output is active. (Exception: When the PWM duty cycle = 0%, the PWM output will remain inactive.)
- The PWMxDCH and PWMxDCL register values are latched into the buffers.

Note: The Timer2 postscaler has no effect on the PWM operation.

26.1.4 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the PWMxDCH and PWMxDCL register pair. The PWMxDCH register contains the eight MSbs and the PWMxDCL<7:6>, the two LSbs. The PWMxDCH and PWMxDCL registers can be written to at any time.

Equation 26-2 is used to calculate the PWM pulse width.

Equation 26-3 is used to calculate the PWM duty cycle ratio.

EQUATION 26-2: PULSE WIDTH

$$Pulse\ Width = (PWMxDCH:PWMxDCL<7:6>) \bullet TOSC \bullet (TMR2\ Prescale\ Value)$$

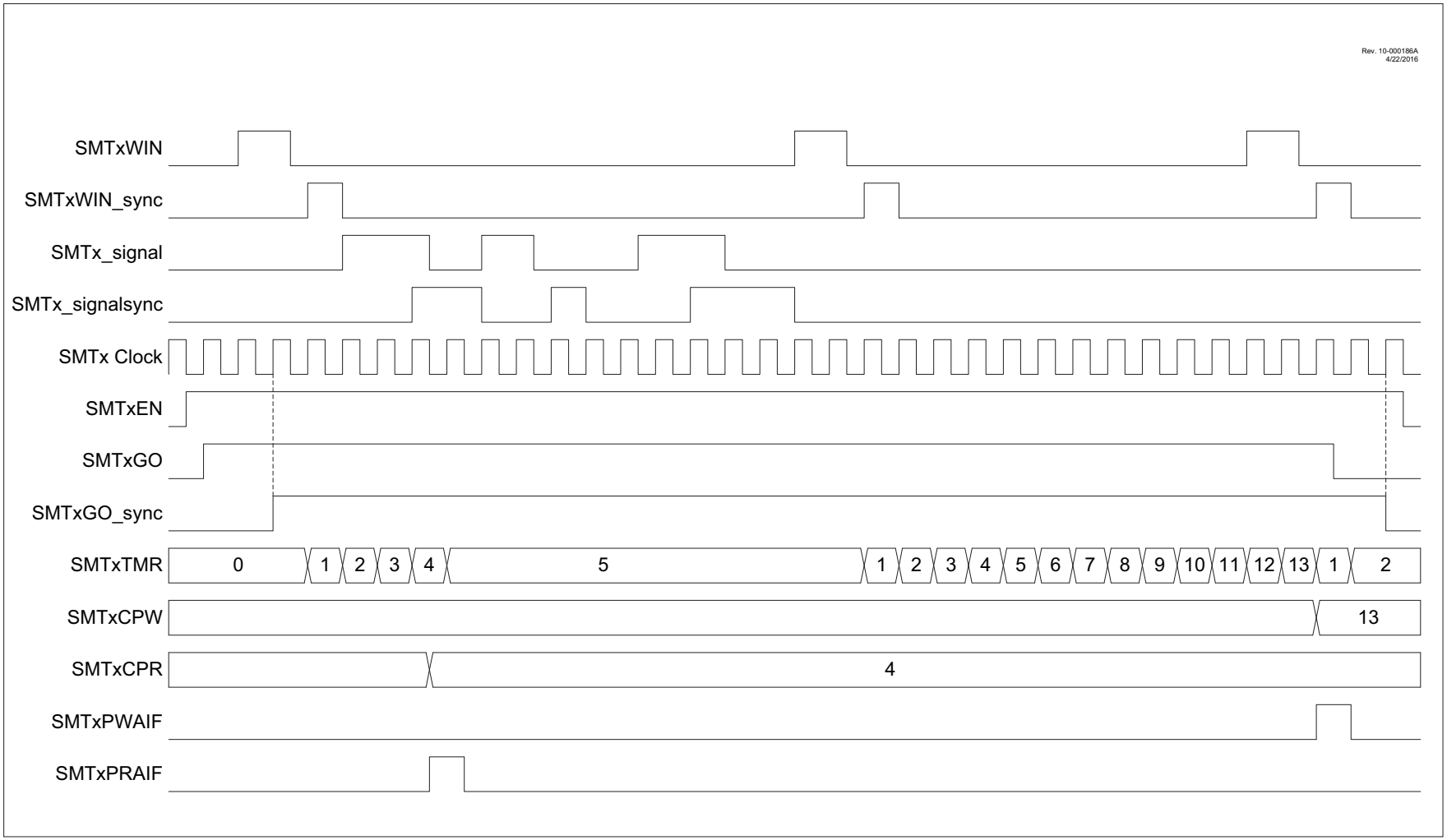
Note: TOSC = 1/FOSC

EQUATION 26-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(PWMxDCH:PWMxDCL<7:6>)}{4(T2PR + 1)}$$

The 8-bit timer T2TMR register is concatenated with the two Least Significant bits of 1/FOSC, adjusted by the Timer2 prescaler to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

FIGURE 27-14: TIME OF FLIGHT MODE REPEAT ACQUISITION TIMING DIAGRAM



27.6.11 WINDOWED COUNTER MODE

This mode counts pulses on the SMT1_signal input, within a window dictated by the SMT1WIN input. It begins counting upon seeing a rising edge of the SMT1WIN input, updates the SMT1CPW register on a falling edge of the SMT1WIN input, and updates the SMT1CPR register on each rising edge of the SMT1WIN input beyond the first. See Figure 27-21 and Figure 27-22.

REGISTER 27-13: SMT1CPWL: SMT CAPTURED PULSE WIDTH REGISTER – LOW BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMT1CPW<7:0>							
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 7-0 **SMT1CPW<7:0>**: Significant bits of the SMT PW Latch – Low Byte

REGISTER 27-14: SMT1CPWH: SMT CAPTURED PULSE WIDTH REGISTER – HIGH BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMT1CPW<15:8>							
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 7-0 **SMT1CPW<15:8>**: Significant bits of the SMT PW Latch – High Byte

REGISTER 27-15: SMT1CPWU: SMT CAPTURED PULSE WIDTH REGISTER – UPPER BYTE

R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x	R-x/x
SMT1CPW<23:16>							
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 7-0 **SMT1CPW<23:16>**: Significant bits of the SMT PW Latch – Upper Byte

33.2.1.1 Enabling the Transmitter

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

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

All other UART control bits are assumed to be in their default state.

Setting the TXEN bit in the UxCON0 register enables the transmitter circuitry of the UART. The MODE<3:0> bits in the UxCON0 register select the desired mode. Setting the ON bit in the UxCON1 register enables the UART. When TXEN is set and the transmitter is not idle, the TX pin is automatically configured as an output. When the transmitter is idle, the TX pin drive is relinquished to the port TRIS control. If the TX pin is shared with an analog peripheral, the analog I/O function should be disabled by clearing the corresponding ANSEL bit.

Note:	The UxTXIF Transmitter Interrupt flag is set when the TXEN enable bit is set and the UxTXB register can accept data.

33.2.1.2 Transmitting Data

A transmission is initiated by writing a character to the UxTXB register. If this is the first character, or the previous character has been completely transmitted from the TSR, the data in the UxTXB is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the UxTXB until the previous character transmission is complete. The pending character in the UxTXB is then transferred to the TSR at the beginning of the previous character Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the completion of all of the previous character's Stop bits.

33.2.1.3 Transmit Data Polarity

The polarity of the transmit data is controlled with the TXPOL bit in the UxCON2 register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the TXPOL bit to '1' will invert the transmit data, resulting in low true idle and data bits. The TXPOL bit controls transmit data polarity in all modes.

33.2.1.4 Transmit Interrupt Flag

The UxTXIF interrupt flag bit in the PIR register is set whenever the UART transmitter is enabled and no character is being held for transmission in the UxTXB. In other words, the UxTXIF bit is clear only when the TSR is busy with a character and a new character has been queued for transmission in the UxTXB.

The UxTXIF interrupt can be enabled by setting the UxTXIE interrupt enable bit in the PIE register. However, the UxTXIF flag bit will be set whenever the UxTXB is empty, regardless of the state of UxTXIE enable bit.The UxTXIF bit is read-only and cannot be set or cleared by software.

To use interrupts when transmitting data, set the UxTXIE bit only when there is more data to send. Clear the UxTXIE interrupt enable bit upon writing UxTXB with the last character of the transmission.

33.2.1.5 TSR Status

The TXMTIF bit in the UxERRIR register indicates the status of the TSR. This is a read-only bit. The TXMTIF bit is set when the TSR is empty and idle. The TXMTIF bit is cleared when a character is transferred to the TSR from the UxTXB. The TXMTIF bit remains clear until all bits, including the Stop bits, have been shifted out of the TSR and a byte is not waiting in the UxTXB register.

The TXMTIF will generate an interrupt when the TXMTIE bit in the UxERRIE register is set.

Note:	The TSR is not mapped in data memory, so it is not available to the user.

33.2.1.6 Transmitter 7-bit Mode

7-Bit mode is selected when the MODE<3:0> bits are set to '0001'. In 7-bit mode, only the seven Least Significant bits of the data written to UxTXB are transmitted. The Most Significant bit is ignored.

33.2.1.7 Transmitter Parity Modes

When the Odd or even Parity mode is selected, all data is sent as nine bits. The first eight bits are data and the 9th bit is parity. Even and odd parity is selected when the MODE<3:0> bits are set to '0011' and '0010', respectively. Parity is automatically determined by the module and inserted in the serial data stream.

34.3 SPI MODE OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SPIxCON0<2:0>, SPIxCON1<7:4>, SPIxCON1<2:0>, and SPIxCON2<2:0>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Input, Output, and Slave Select Polarity
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on first/second edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Master or Slave mode)
- MSB-First or LSB-First
- Receive/Transmit Modes
  - Full duplex
  - Receive-without-transmit
  - Transmit-without-receive
- Transfer Counter Mode (Transmit-without-receive mode)

34.3.1 ENABLING AND DISABLING THE SPI MODULE

To enable the serial peripheral, the SPI enable bit (EN in SPIxCON0) must be set. To reset or reconfigure SPI mode, clear the EN bit, re-initialize the SSPxCONx registers and then set the EN bit. Setting the EN bit enables the SPI inputs and outputs: SDI, SDO, SCK(out), SCK(in), SS(out), and SS(in). All of these inputs and outputs are steered by PPS, and thus must have their functions properly mapped to device pins to function (see **Section 19.0 "Peripheral Pin Select (PPS) Module"**). In addition, SS(out) and SCK(out) must have the pins they are steered to set as outputs (TRIS bits must be 0) in order to properly output. Clearing the TRIS bit of the SDO pin will cause the SPI module to always control that pin, but is not necessary for SDO functionality. (see **Section 34.3.5 "Input and Output Polarity Bits"**). Configurations selected by the following registers should not be changed while the EN bit is set:

- SPIxBAUD
- SPIxCON1
- SPIxCON0 (except to clear the EN bit)

Clearing the EN bit aborts any transmissions in progress, disables the setting of interrupt flags by hardware, and resets the FIFO occupancy (see **Section 34.3.3 "Transmit and Receive FIFOs"** for more FIFO details).

34.3.2 BUSY BIT

While a data transfer is in progress, the SPI module sets the BUSY bit of SPIxCON2. This bit can be polled by the user to determine the current status of the SPI module, and to know when a communication is complete. The following registers/bits should not be written by software while the BUSY bit is set:

- SPIxTCNTH/L
- SPIxTWIDTH
- SPIxCON2
- The CLRBF bit of SPIxSTATUS

Note:	It is also not recommended to read SPIx-TCNTH/L while the BUSY bit is set, as the value in the registers may not be a reliable indicator of the Transfer Counter. Use the Transfer Count Zero Interrupt Flag (the TCZIF bit of SPIxINTF) to accurately determine that the Transfer Counter has reached zero.

TABLE 44-1: REGISTER FILE SUMMARY FOR PIC18(L)F24/25K42 DEVICES (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3A58h	CCDPB	CCDPB7	CCDPB6	CCDPB5	CCDPB4	CCDPB3	CCDPB2	CCDPB1	CCDPB0	277
3A57h	IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	290
3A56h	IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	290
3A55h	IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	290
3A54h	INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	INLVLB3	INLVLB2	INLVLB1	INLVLB0	274
3A53h	SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	SLRB3	SLRB2	SLRB1	SLRB0	273
3A52h	ODCONB	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	272
3A51h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	271
3A50h	ANSELB	ANSELB7	ANSELB6	ANSELB5	ANSELB4	ANSELB3	ANSELB2	ANSELB1	ANSELB0	270
3A4Fh - 3A4Ah	—	Unimplemented								
3A49h	CCDNA	CCDNA7	CCDNA6	CCDNA5	CCDNA4	CCDNA3	CCDNA2	CCDNA1	CCDNA0	278
3A48h	CCDPA	CCDNP7	CCDNP6	CCDNP5	CCDNP4	CCDNP3	CCDNP2	CCDNP1	CCDNP0	277
3A47h	IOCAF	IOCAF7	IOCAF6	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	290
3A46h	IOCAN	IOCAN7	IOCAN6	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	290
3A45h	IOCAP	IOCAP7	IOCAP6	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	290
3A44h	INLVLA	INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	274
3A43h	SLRCONA	SLRA7	SLRA6	SLRA5	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0	273
3A42h	ODCONA	ODCA7	ODCA6	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	272
3A41h	WPUA	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	271
3A40h	ANSELA	ANSELA7	ANSELA6	ANSELA5	ANSELA4	ANSELA3	ANSELA2	ANSELA1	ANSELA0	270
3A3Fh - 3A18h	—	Unimplemented								
3A17h	RC7PPS	—	—	—	RC7PPS4	RC7PPS3	RC7PPS2	RC7PPS1	RC7PPS0	284
3A16h	RC6PPS	—	—	—	RC6PPS4	RC6PPS3	RC6PPS2	RC6PPS1	RC6PPS0	284
3A15h	RC5PPS	—	—	—	RC5PPS4	RC5PPS3	RC5PPS2	RC5PPS1	RC5PPS0	284
3A14h	RC4PPS	—	—	—	RC4PPS4	RC4PPS3	RC4PPS2	RC4PPS1	RC4PPS0	284
3A13h	RC3PPS	—	—	—	RC3PPS4	RC3PPS3	RC3PPS2	RC3PPS1	RC3PPS0	284
3A12h	RC2PPS	—	—	—	RC2PPS4	RC2PPS3	RC2PPS2	RC2PPS1	RC2PPS0	284
3A11h	RC1PPS	—	—	—	RC1PPS4	RC1PPS3	RC1PPS2	RC1PPS1	RC1PPS0	284
3A10h	RC0PPS	—	—	—	RC0PPS4	RC0PPS3	RC0PPS2	RC0PPS1	RC0PPS0	284
3A0Fh	RB7PPS	—	—	—	RB7PPS4	RB7PPS3	RB7PPS2	RB7PPS1	RB7PPS0	284
3A0Eh	RB6PPS	—	—	—	RB6PPS4	RB6PPS3	RB6PPS2	RB6PPS1	RB6PPS0	284
3A0Dh	RB5PPS	—	—	—	RB5PPS4	RB5PPS3	RB5PPS2	RB5PPS1	RB5PPS0	284
3A0Ch	RB4PPS	—	—	—	RB4PPS4	RB4PPS3	RB4PPS2	RB4PPS1	RB4PPS0	284
3A0Bh	RB3PPS	—	—	—	RB3PPS4	RB3PPS3	RB3PPS2	RB3PPS1	RB3PPS0	284
3A0Ah	RB2PPS	—	—	—	RB2PPS4	RB2PPS3	RB2PPS2	RB2PPS1	RB2PPS0	284
3A09h	RB1PPS	—	—	—	RB1PPS4	RB1PPS3	RB1PPS2	RB1PPS1	RB1PPS0	284
3A08h	RB0PPS	—	—	—	RB0PPS4	RB0PPS3	RB0PPS2	RB0PPS1	RB0PPS0	284
3A07h	RA7PPS	—	—	—	RA7PPS4	RA7PPS3	RA7PPS2	RA7PPS1	RA7PPS0	284
3A06h	RA6PPS	—	—	—	RA6PPS4	RA6PPS3	RA6PPS2	RA6PPS1	RA6PPS0	284
3A05h	RA5PPS	—	—	—	RA5PPS4	RA5PPS3	RA5PPS2	RA5PPS1	RA5PPS0	284
3A04h	RA4PPS	—	—	—	RA4PPS4	RA4PPS3	RA4PPS2	RA4PPS1	RA4PPS0	284
3A03h	RA3PPS	—	—	—	RA3PPS4	RA3PPS3	RA3PPS2	RA3PPS1	RA3PPS0	284
3A02h	RA2PPS	—	—	—	RA2PPS4	RA2PPS3	RA2PPS2	RA2PPS1	RA2PPS0	284
3A01h	RA1PPS	—	—	—	RA1PPS4	RA1PPS3	RA1PPS2	RA1PPS1	RA1PPS0	284
3A00h	RA0PPS	—	—	—	RA0PPS4	RA0PPS3	RA0PPS2	RA0PPS1	RA0PPS0	284
39FFh - 39F8h	—	Unimplemented								
39F7h	SCANPR	—	—	—	—	—	PR			22

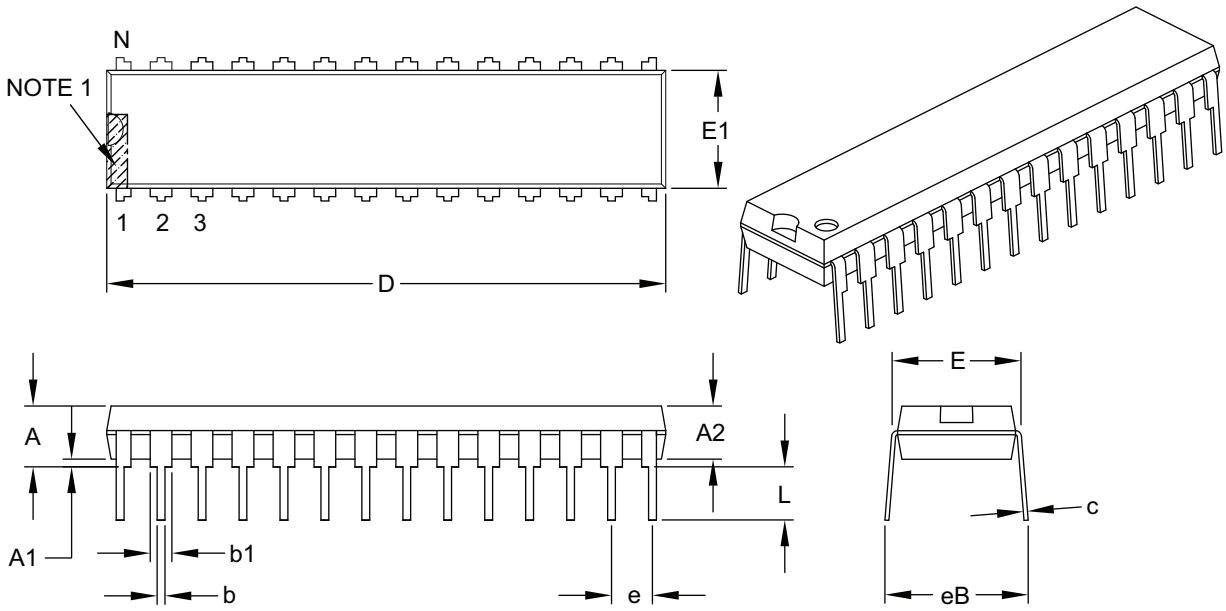
Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition
Note 1: Not present in LF devices.

48.1 Package Details

The following sections give the technical details of the packages.

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-070B

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X](2)	-	X	/XX	XXX
Device	Tape and Reel Option		Temperature Range	Package	Pattern

Device:	PIC18F24K42, PIC18LF24K42, PIC18F25K42, PIC18LF25K42
Tape and Reel Option:	Blank = standard packaging (tube or tray) T = Tape and Reel(1), (2)
Temperature Range:	E = -40°C to +125°C (Extended) I = -40°C to +85°C (Industrial)
Package:	ML = 28-lead QFN 6x6mm MV = 28-lead UQFN 4x4x0.5mm SO = 28-lead SOIC SP = 28-lead Skinny Plastic DIP SS = 28-lead SSOP
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

a) PIC18F24K42-E/P 301 = Extended temp., PDIP package, QTP pattern #301.

b) PIC18F25K42-E/SO = Extended temp., SOIC package.

c) PIC18F24K42T-I/ML = Tape and reel, Industrial temp., QFN package.

Note 1: Tape and Reel option is available for ML, MV, PT, SO and SS packages with industrial Temperature Range only.

2: Tape and Reel identifier only appears in catalog part number description. This identifier is used for ordering purposes and is not printed on the device package.