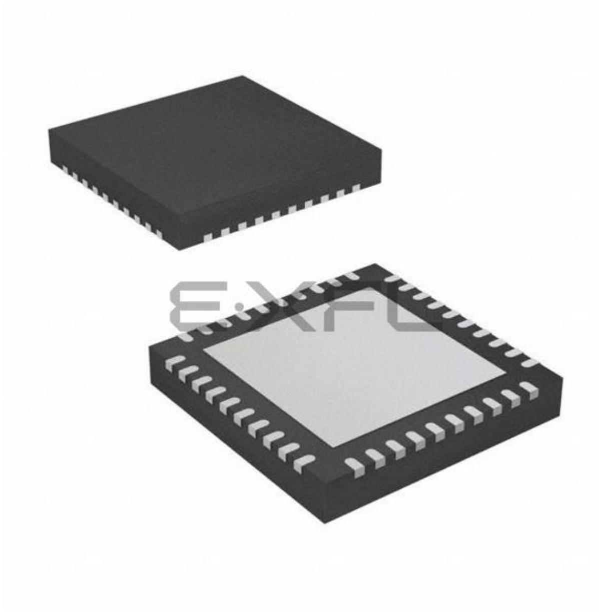
Microchip Technology - PIC18LF47K42-E/MV Datasheet



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	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	36
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)	1.8V ~ 3.6V
Data Converters	A/D 35x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UFQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf47k42-e-mv

TABLE 2: 40/44-PIN ALLOCATION TABLE FOR PIC18(L)F4XK42

I/O	40-Pin PDIP	44-Pin TQFP	40-Pin UQFN	44-Pin QFN	ADC	Voltage Reference	DAC	Comparators	Zero Cross Detect	I^{2}C	SPI	UART	DSM	Timers/SMT	CCP and PWM	CWG	CLC	NCO	Clock Reference (CLKR)	Interrupt-on-Change	Basic
RC3	18	37	33	37	ANC3	—	—	—	—	SCL1(3,4)	SCK1(1)	—	—	T2IN(1)	—	—	—	—	—	IOCC3	—
RC4	23	42	38	42	ANC4	—	—	—	—	SDA1(3,4)	SDI1(1)	—	—	—	—	—	—	—	—	IOCC4	—
RC5	24	43	39	43	ANC5	—	—	—	—	—	—	—	—	T4IN(1)	—	—	—	—	—	IOCC5	—
RC6	25	44	40	44	ANC6	—	—	—	—	—	—	CTS1(1)	—	—	—	—	—	—	—	IOCC6	—
RC7	26	1	1	1	ANC7	—	—	—	—	—	—	RX1(1)	—	—	—	—	—	—	—	IOCC7	—
RD0	19	38	34	38	AND0	—	—	—	—	—(4)	—	—	—	—	—	—	—	—	—	—	—
RD1	20	39	35	39	AND1	—	—	—	—	—(4)	—	—	—	—	—	—	—	—	—	—	—
RD2	21	40	36	40	AND2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD3	22	41	37	41	AND3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD4	27	2	2	2	AND4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD5	28	3	3	3	AND5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD6	29	4	4	4	AND6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD7	30	5	5	5	AND7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE0	8	25	23	25	ANE0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE1	9	26	24	26	ANE1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE2	10	27	25	27	ANE2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE3	1	18	16	18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCE3	$\overline{\text{MCLR}}$ V$_{PP}$
V$_{DD}$	11, 32	7, 28	7, 26	7, 28	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
V$_{SS}$	12, 31	6, 29	6, 27	6, 30	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.

2: All output signals shown in this row are PPS remappable.

3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

4: These pins can be configured for I^{2}C and SMB™ 3.0/2.0 logic levels; The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4/RD0/RD1 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the I^{2}C specific or SMBus input buffer thresholds.

Table of Contents

4.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 microcontroller devices:

- Program Flash Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate buses; this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Program Flash Memory and Data EEPROM Memory is provided in **Section 13.0 “Nonvolatile Memory (NVM) Control”**.

4.1 Program Flash Memory Organization

PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2 Mbyte program memory space. Accessing any unimplemented memory will return all ‘0’s (a NOP instruction).

These devices contain the following:

- PIC18(L)F45/55K42: 32 Kbytes of Flash memory, up to 16,384 single-word instructions
- PIC18(L)F26/46/56K42: 64 Kbytes of Flash memory, up to 32,768 single-word instructions
- PIC18(L)F27/47/57K42: 128 Kbytes of Flash memory, up to 65,536 single-word instructions

The Reset vector for the device is at address 000000h. PIC18(L)F26/27/45/46/47/55/56/57K42 devices feature a vectored interrupt controller with a dedicated interrupt vector table in the program memory, see **Section 9.0 “Interrupt Controller”**.

Note:	For memory information on this family of devices, see Table 4-1 and Table 4-3.

4.2 Memory Access Partition (MAP)

Program Flash memory is partitioned into:

- Application Block
- Boot Block, and
- Storage Area Flash (SAF) Block

4.2.1 APPLICATION BLOCK

Application block is where the user’s program resides by default. Default settings of the configuration bits ($\overline{\text{BBEN}}$ = 1 and $\overline{\text{SAFEN}}$ = 1) assign all memory in the program Flash memory area to the application block. The $\overline{\text{WRTAPP}}$ configuration bit is used to protect the application block.

4.2.2 BOOT BLOCK

Boot block is an area in program memory that is ideal for storing bootloader code. Code placed in this area can be executed by the CPU. The boot block can be write-protected, independent of the main application block. The Boot Block is enabled by the $\overline{\text{BBEN}}$ bit and size is based on the value of the BBSIZE bits of Configuration word (Register 5-7), see Table 5-1 for boot block sizes. The $\overline{\text{WRTB}}$ Configuration bit is used to write-protect the Boot Block.

4.2.3 STORAGE AREA FLASH

Storage Area Flash (SAF) is the area in program memory that can be used as data storage. SAF is enabled by the $\overline{\text{SAFEN}}$ bit of the Configuration word in Register 5-7. If enabled, the code placed in this area cannot be executed by the CPU. The SAF block is placed at the end of memory and spans 128 Words. The $\overline{\text{WRTSAF}}$ Configuration bit is used to write-protect the Storage Area Flash.

Note:	If write-protected locations are written to, memory is not changed and the WRERR bit defined in Register 13-1 is set.

TABLE 4-10: SPECIAL FUNCTION REGISTER MAP FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES BANK 57

39FFh	—	39DFh	OSCFRQ	39BFh	—	399Fh	—	397Fh	—	395Fh	WDTU	393Fh	—	391Fh	—
39FEh	—	39DEh	OSCTUNE	39BEh	—	399Eh	—	397Eh	—	395Eh	WDTH	393Eh	—	391Eh	—
39FDh	—	39DDh	OSCEN	39BDh	—	399Dh	—	397Dh	SCANTRIG	395Dh	WDTL	393Dh	—	391Dh	—
39FCh	—	39DCh	OSCSTAT	39BCh	—	399Ch	—	397Ch	SCANCON0	395Ch	WDTCON1	393Ch	—	391Ch	—
39FBh	—	39DBh	OSCCON3	39BBh	—	399Bh	—	397Bh	SCANHADRU	395Bh	WDTCON0	393Bh	—	391Bh	—
39FAh	—	39DAh	OSCCON2	39BAh	—	399Ah	PIE10	397Ah	SCANHADRH	395Ah	—	393Ah	—	391Ah	—
39F9h	—	39D9h	OSCCON1	39B9h	—	3999h	PIE9	3979h	SCANHADRL	3959h	—	3939h	—	3919h	—
39F8h	—	39D8h	CPUDOZE	39B8h	—	3998h	PIE8	3978h	SCANLADRU	3958h	—	3938h	—	3918h	—
39F7h	SCANPR	39D7h	—	39B7h	—	3997h	PIE7	3977h	SCANLADRH	3957h	—	3937h	—	3917h	—
39F6h	—	39D6h	—	39B6h	—	3996h	PIE6	3976h	SCANLADRL	3956h	—	3936h	—	3916h	—
39F5h	—	39D5h	—	39B5h	—	3995h	PIE5	3975h	—	3955h	—	3935h	—	3915h	—
39F4h	DMA2PR	39D4h	—	39B4h	—	3994h	PIE4	3974h	—	3954h	—	3934h	—	3914h	—
39F3h	DMA1PR	39D3h	—	39B3h	—	3993h	PIE3	3973h	—	3953h	—	3933h	—	3913h	—
39F2h	MAINPR	39D2h	—	39B2h	—	3992h	PIE2	3972h	—	3952h	—	3932h	—	3912h	—
39F1h	ISRPR	39D1h	VREGCON(1)	39B1h	—	3991h	PIE1	3971h	—	3951h	—	3931h	—	3911h	—
39F0h	—	39D0h	BORCON	39B0h	—	3990h	PIE0	3970h	—	3950h	—	3930h	—	3910h	—
39EFh	PRLOCK	39CFh	—	39AFh	—	398Fh	—	396Fh	—	394Fh	—	392Fh	—	390Fh	—
39EEh	—	39CEh	—	39AEh	—	398Eh	—	396Eh	—	394Eh	—	392Eh	—	390Eh	—
39EDh	—	39CDh	—	39ADh	—	398Dh	—	396Dh	—	394Dh	—	392Dh	—	390Dh	—
39ECh	—	39CCh	—	39ACh	—	398Ch	—	396Ch	—	394Ch	—	392Ch	—	390Ch	—
39EBh	—	39CBh	—	39ABh	—	398Bh	—	396Bh	—	394Bh	—	392Bh	—	390Bh	—
39EAh	—	39CAh	—	39AAh	PIR10	398Ah	IPR10	396Ah	—	394Ah	—	392Ah	—	390Ah	—
39E9h	—	39C9h	—	39A9h	PIR9	3989h	IPR9	3969h	CRCCON1	3949h	—	3929h	—	3909h	—
39E8h	—	39C8h	—	39A8h	PIR8	3988h	IPR8	3968h	CRCCON0	3948h	—	3928h	—	3908h	—
39E7h	—	39C7h	PMD7	39A7h	PIR7	3987h	IPR7	3967h	CRCXORH	3947h	—	3927h	—	3907h	—
39E6h	NVMCON2	39C6h	PMD6	39A6h	PIR6	3986h	IPR6	3966h	CRCXORL	3946h	—	3926h	—	3906h	—
39E5h	NVMCON1	39C5h	PMD5	39A5h	PIR5	3985h	IPR5	3965h	CRCSHIFTH	3945h	—	3925h	—	3905h	—
39E4h	—	39C4h	PMD4	39A4h	PIR4	3984h	IPR4	3964h	CRCSHIFTL	3944h	—	3924h	—	3904h	—
39E3h	NVMDAT	39C3h	PMD3	39A3h	PIR3	3983h	IPR3	3963h	CRCACCH	3943h	—	3923h	—	3903h	—
39E2h	—	39C2h	PMD2	39A2h	PIR2	3982h	IPR2	3962h	CRCACCL	3942h	—	3922h	—	3902h	—
39E1h	NVMADRH(4)	39C1h	PMD1	39A1h	PIR1	3981h	IPR1	3961h	CRCDATH	3941h	—	3921h	—	3901h	—
39E0h	NVMADRL	39C0h	PMD0	39A0h	PIR0	3980h	IPR0	3960h	CRCDATL	3940h	—	3920h	—	3900h	—

Legend: Unimplemented data memory locations and registers, read as ‘0’.

Note 1: Unimplemented in LF devices.

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

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

4: Unimplemented in PIC18(L)F45/55K42.

REGISTER 5-12: REVISION ID: REVISION ID REGISTER

R	R	R	R	R	R	R	R
`1`	`0`	`1`	`0`	MJRREV<5:2>			
bit 15							bit 8

R	R	R	R	R	R	R	R
MJRREV<1:0>		MNRREV<5:0>					
bit 7							bit 0

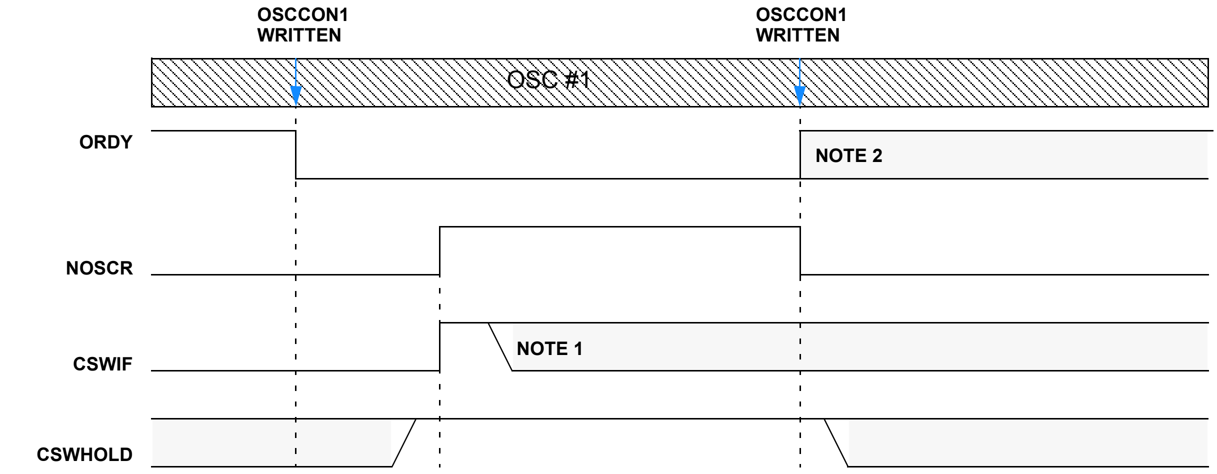
Legend:			
R = Readable bit	‘1’ = Bit is set	0’ = Bit is cleared	x = Bit is unknown

bit 15-12 **Read as ‘`1010`’**
These bits are fixed with value ‘`1010`’ for all devices in this family.

bit 11-6 **MJRREV<5:0>:** Major Revision ID bits
These bits are used to identify a major revision. A major revision is indicated by revision (A0, B0, C0, etc.)
Revision A = `0b00 0000`

bit 5-0 **MNRREV<5:0>:** Minor Revision ID bits
These bits are used to identify a minor revision.
Revision A0 = `0b00 0000`

FIGURE 7-8: CLOCK SWITCH ABANDONED



Note 1: CSWIF may be cleared before or after rewriting OSCCON1; CSWIF is not automatically cleared.
2: ORDY = 0 if OSCCON1 does not match OSCCON2; a new switch will begin.

9.3.2 NATURAL ORDER (HARDWARE) PRIORITY

When more than one interrupt with the same user specified priority level are requested, the priority conflict is resolved by using a method called “Natural Order Priority”. Natural order priority is a fixed priority scheme that is based on the Interrupt Vector Table. Table 9-2 shows the natural order priority and the interrupt vector number assigned for each source.

TABLE 9-2: INTERRUPT VECTOR PRIORITY TABLE

Vector Number	Interrupt Source
0	Software Interrupt
1	HLVD
2	OSF
3	CSW
4	NVM
5	SCAN
6	CRC
7	IOC
8	INT0
9	ZCD
10	AD
11	ADT
12	C1
13	SMT1
14	SMT1PRA
15	SMT1PWA
16	DMA1SCNT
17	DMA1DCNT
18	DMA1OR
19	DMA1A
20	SPI1RX
21	SPI1TX
22	SPI1
23	I2C1RX
24	I2C1TX
25	I2C1
26	I2C1E
27	U1RX
28	U1TX
29	U1E
30	U1
31	TMR0
32	TMR1
33	TMR1G
34	TMR2
35	CCP1
36	—
37	NCO
38	CWG1
39	CLC1
40	INT1
41	C2
42	DMA2SCNT
43	DMA2DCNT
44	DMA2OR
45	DMA2A
46	I2C2RX
47	I2C2TX
48	I2C2
49	I2C2E
50	U2RX
51	U2TX
52	U2E
53	U2
54	TMR3
55	TMR3G
56	TMR4
57	CCP2
58	—
59	CWG2
60	CLC2
61	INT2
62	—
63	—
64	—
65	—
66	—
67	—
68	—
69	—
70	TMR5
71	TMR5G
72	TMR6
73	CCP3
74	CWG3
75	CLC3
76	—
77	—
78	—
79	—
80	CCP4
81	CLC4

The natural order priority scheme has vector interrupt 0 as the highest priority and vector interrupt 81 as the lowest priority.

For example, when two concurrently occurring interrupt sources that are both designated high priority using the IPRx register will be resolved using the natural order priority (i.e., the interrupt with a lower corresponding vector number will preempt the interrupt with the higher vector number).

The ability for the user to assign every interrupt source to high or low priority levels means that the user program can give an interrupt with a low natural order priority a higher overall priority level.

9.4 Interrupt Operation

All pending interrupts are indicated by the flag bit being equal to a ‘1’ in the PIRx register. All pending interrupts are resolved using the priority scheme explained in **Section 9.3 “Interrupt Priority”**.

Once the interrupt source to be serviced is resolved, the program execution vectors to the resolved interrupt vector addresses, as explained in **Section 9.2 “Interrupt Vector Table (IVT)”**. The vector number is also stored in the WREG register. Most of the flag bits are required to be cleared by the application software, but in some cases, device hardware clears the interrupt automatically. Some flag bits are read-only in the PIRx registers, these flags are a summary of the source interrupts and the corresponding interrupt flags of the source must be cleared.

A valid interrupt can be either a high or low priority interrupt when in main routine or a high priority interrupt when in low priority Interrupt Service Routine. Depending on order of interrupt requests received and their relative timing, the CPU will be in the state of execution indicated by the STAT bits of the INTCON1 register (Register 9-2).

The State machine shown in Figure 9-1 and the subsequent sections detail the execution of interrupts when received in different orders.

Note:	The state of GIEH/L is not changed by the hardware when servicing an interrupt. The internal state machine is used to keep track of execution states. These bits can be manipulated in the user code resulting in transferring execution to the main routine and ignoring existing interrupts.

9.4.3 PREEMPTING LOW PRIORITY INTERRUPTS

Low-priority interrupts can be preempted by high priority interrupts. While in the low priority ISR, if a high-priority interrupt arrives, the high priority interrupt request is generated and the low priority ISR is suspended, while the high priority ISR is executed, see Figure 9-4.

After the high priority ISR is complete and if any other high priority interrupt requests are not active, the execution returns to the preempted low priority ISR.

> **Note 1:** The high priority interrupt flag must be cleared to avoid recursive interrupts.
>
> **2:** If a low-priority ISR was already serviced halfway before moving on to a high priority ISR, then the low priority ISR is completely serviced even if user code clears GIEL.

FIGURE 9-4: INTERRUPT EXECUTION: HIGH PRIORITY INTERRUPT PREEMPTING LOW PRIORITY INTERRUPTS

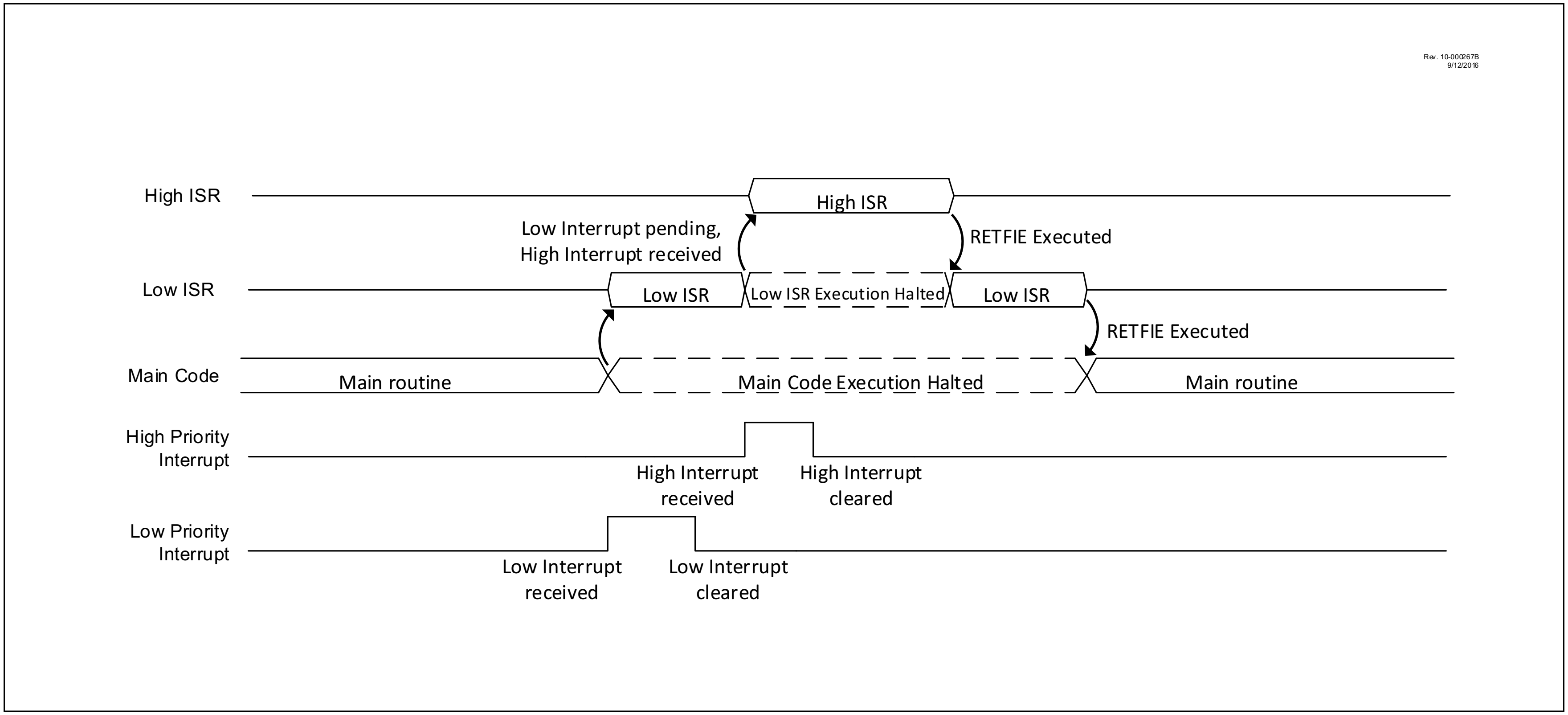
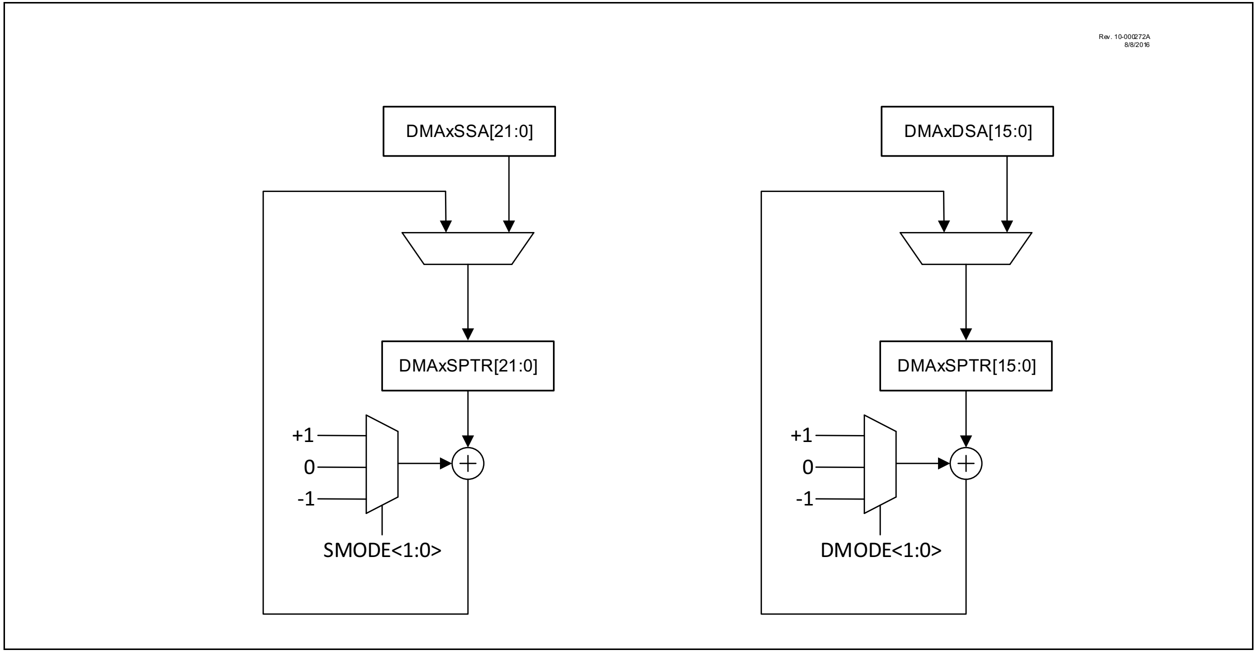


FIGURE 15-2: DMA POINTERS BLOCK DIAGRAM



The DMA can initiate data transfers from the PFM, Data EEPROM or SFR/GPR Space. The SMR<1:0> bits in the DMAxCON1 register are used to select the type of memory being pointed to by the Source Address Pointer. The SMR<1.0> bits are required because the PFM and SFR/GPR spaces have overlapping addresses that do not allow the specified address to uniquely define the memory location to be accessed.

Note 1: For proper memory read access to occur, the combination of address and space selection must be valid.

2: The destination does not have space selection bits because it can only write to the SFR/GPR space.

15.4.2 DMA MESSAGE SIZE/COUNTERS

A transaction is the transfer of one byte. A message consists of one or more transactions. A complete DMA process consists of one or more messages. The size registers determine how many transactions are in a message. The DMAxSSZ registers determine the source size and DMAxDSZ registers determine the destination size.

When a DMA transfer is initiated, the size registers are copied to corresponding counter registers that control the duration of the message. The DMAxSCNT registers count the source transactions and the DMAxDCNT registers count the destination transactions. Both are simultaneously decremented by one after each transaction.

A message is started by setting the DGO bit of the DMAxCON0 register and terminates when the smaller of the two counters reaches zero.

When either counter reaches zero the DGO bit is cleared and the counter and pointer registers are immediately reloaded with the corresponding size and address data. If the other counter did not reach zero then the next message will continue with the count and address corresponding to that register.

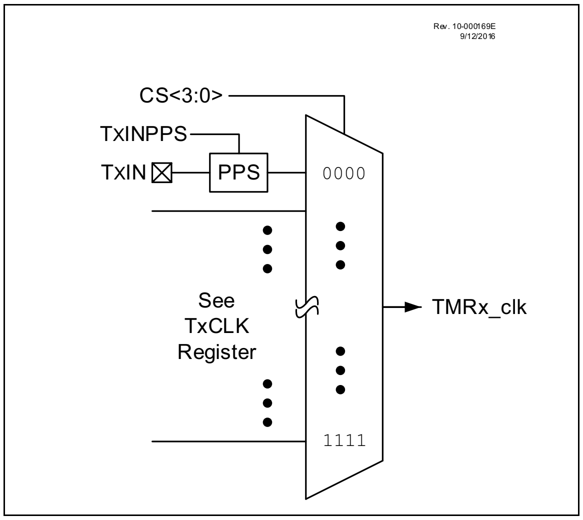
When the source and destination size registers are not equal, then the ratio of the largest to the smallest size determines how many messages are in the DMA process. For example, when the destination size is 6 and the source size is 2, then each message will consist of two transactions and the complete DMA process will consist of three messages. When the larger size is not an even integer of the smaller size, then the last message in the process will terminate early when the larger count reaches zero. In that case, the larger counter will reset and the smaller counter will have a remainder skewing any subsequent messages by that amount.

Note: Reading the DMAxSCNT or DMAxDCNT registers will never return zero. When either register is decremented from '1' it is immediately reloaded from the corresponding size register.

TABLE 17-1: PPS INPUT REGISTER DETAILS

Peripheral	PPS Input Register	Default Pin Selection at POR	Register Reset Value at POR	Input Available from Selected PORTx													
				PIC18(L)F26/27K42			PIC18(L)F45/46/47K42					PIC18(L)F55/56/57K42					
Interrupt 0	INT0PPS	RB0	0b0 1000	A	B	—	A	B	—	—	—	A	B	—	—	—	—
Interrupt 1	INT1PPS	RB1	0b0 1001	A	B	—	A	B	—	—	—	—	B	—	D	—	—
Interrupt 2	INT2PPS	RB2	0b0 1010	A	B	—	A	B	—	—	—	—	B	—	—	—	F
Timer0 Clock	T0CKIPPS	RA4	0b0 0100	A	B	—	A	B	—	—	—	A	—	—	—	—	F
Timer1 Clock	T1CKIPPS	RC0	0b1 0000	A	—	C	A	—	C	—	—	—	—	C	—	E	—
Timer1 Gate	T1GPPS	RB5	0b0 1101	—	B	C	—	B	C	—	—	—	B	C	—	—	—
Timer3 Clock	T3CKIPPS	RC0	0b1 0000	—	B	C	—	B	C	—	—	-	—	C	—	E	—
Timer3 Gate	T3GPPS	RC0	0b1 0000	A	—	C	A	—	C	—	—	A	—	C	—	—	—
Timer5 Clock	T5CKIPPS	RC2	0b1 0010	A	—	C	A	—	C	—	—	—	—	C	—	E	—
Timer5 Gate	T5GPPS	RB4	0b0 1100	—	B	C	—	B	—	D	—	—	B	—	D	—	—
Timer2 Clock	T2INPPS	RC3	0b1 0011	A	—	C	A	—	C	—	—	A	—	C	—	—	—
Timer4 Clock	T4INPPS	RC5	0b1 0101	—	B	C	—	B	C	—	—	—	B	C	—	—	—
Timer6 Clock	T6INPPS	RB7	0b0 1111	—	B	C	—	B	—	D	—	—	B	—	D	—	—
CCP1	CCP1PPS	RC2	0b1 0010	—	B	C	—	B	C	—	—	—	—	C	—	—	F
CCP2	CCP2PPS	RC1	0b1 0001	—	B	C	—	B	C	—	—	—	—	C	—	—	F
CCP3	CCP3PPS	RB5	0b0 1101	—	B	C	—	B	—	D	—	—	B	—	D	—	—
CCP4	CCP4PPS	RB0	0b0 1000	—	B	C	—	B	—	D	—	—	B	—	D	—	—
SMT1 Window	SMT1WINPPS	RC0	0b1 0000	—	B	C	—	B	C	—	—	—	—	C	—	—	F
SMT1 Signal	SMT1SIGPPS	RC1	0b1 0001	—	B	C	—	B	C	—	—	—	—	C	—	—	F
CWG1	CWG1PPS	RB0	0b0 1000	—	B	C	—	B	—	D	—	—	B	—	D	—	—
CWG2	CWG2PPS	RB1	0b0 1001	—	B	C	—	B	—	D	—	—	B	—	D	—	—
CWG3	CWG3PPS	RB2	0b0 1010	—	B	C	—	B	—	D	—	—	B	—	D	—	—
DSM1 Carrier Low	MD1CARLPPS	RA3	0b0 0011	A	—	C	A	—	—	D	—	A	—	—	D	—	—
DSM1 Carrier High	MD1CARHPPS	RA4	0b0 0100	A	—	C	A	—	—	D	—	A	—	—	D	—	—
DSM1 Source	MD1SRCPPS	RA5	0b0 0101	A	—	C	A	—	—	D	—	A	—	—	D	—	—
CLCx Input 1	CLCIN0PPS	RA0	0b0 0000	A	—	C	A	—	C	—	—	A	—	C	—	—	—
CLCx Input 2	CLCIN1PPS	RA1	0b0 0001	A	—	C	A	—	C	—	—	A	—	C	—	—	—
CLCx Input 3	CLCIN2PPS	RB6	0b0 1110	—	B	C	—	B	—	D	—	—	B	—	D	—	—
CLCx Input 4	CLCIN3PPS	RB7	0b0 1111	—	B	C	—	B	—	D	—	—	B	—	D	—	—

FIGURE 22-2: TIMER2 CLOCK SOURCE BLOCK DIAGRAM



22.1 Timer2 Operation

Timer2 operates in three major modes:

- Free Running Period
- One-Shot
- Monostable

Within each mode there are several options for starting, stopping, and reset. Table 22-1 lists the options.

In all modes the T2TMR count register is incremented on the rising edge of the clock signal from the programmable prescaler. When T2TMR equals T2PR then a high level is output to the postscaler counter. T2TMR is cleared on the next clock input.

An external signal from hardware can also be configured to gate the timer operation or force a T2TMR count Reset. In gate modes, the counter stops when the gate is disabled and resumes when the gate is enabled. In Reset modes the T2TMR count is reset on either the level or edge from the external source.

The T2TMR and T2PR registers are both directly readable and writable. The T2TMR register is cleared and the T2PR register initializes to FFh on any device Reset. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the T2TMR register
- a write to the TxCON register
- any device Reset
- External Reset Source event that resets the timer.

> **Note:** T2TMR is not cleared when TxCON is written.

22.1.1 FREE RUNNING PERIOD MODE

The value of T2TMR is compared to that of the Period register, T2PR, on each clock cycle. When the two values match, the comparator resets the value of T2TMR to 00h on the next cycle and increments the output postscaler counter. When the postscaler count equals the value in the OUTPS bits of the TxCON register, then a one clock period wide pulse occurs on the T2TMR_postscaled output, and the postscaler count is cleared.

22.1.2 ONE-SHOT MODE

The One-Shot mode is identical to the Free Running Period mode except that the ON bit is cleared and the timer is stopped when T2TMR matches T2PR and will not restart until the T2ON bit is cycled off and on. Postscaler OUTPS values other than 0 are meaningless in this mode because the timer is stopped at the first period event and the postscaler is reset when the timer is restarted.

22.1.3 MONOSTABLE MODE

Monostable modes are similar to One-Shot modes except that the ON bit is not cleared and the timer can be restarted by an external Reset event.

22.2 Timer2 Output

The Timer2 module's primary output is T2TMR_postscaled, which pulses for a single T2TMR_clk period when the postscaler counter matches the value in the OUTPS bits of the TxCON register. The T2PR postscaler is incremented each time the T2TMR value matches the T2PR value. This signal can be selected as an input to several other input modules.

Timer2 is also used by the CCP module for pulse generation in PWM mode. Both the actual T2TMR value as well as other internal signals are sent to the CCP module to properly clock both the period and pulse width of the PWM signal. See **Section 23.0 "Capture/Compare/PWM Module"** for more details on setting up Timer2 for use with the CCP, as well as the timing diagrams in **Section 22.5 "Operation Examples"** for examples of how the varying Timer2 modes affect CCP PWM output.

22.3 External Reset Sources

In addition to the clock source, the Timer2 also takes in an external Reset source. This external Reset source is selected for Timer2, Timer4, and Timer6 with the T2RST, T4RST, and T6RST registers, respectively. This source can control starting and stopping of the timer, as well as resetting the timer, depending on which mode the timer is in. The mode of the timer is controlled by the MODE bits of the T2HLT register. Edge Triggered modes require six Timer clock periods between external triggers. Level Triggered modes require the triggering level to be at least three Timer clock periods long. External triggers are ignored while in Debug Freeze mode.

REGISTER 23-3: CCPxCAP: CAPTURE INPUT SELECTION MULTIPLEXER REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/x	R/W-0/x	R/W-0/x
—	—	—	—	—	CTS<2:0>		
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 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **CTS<2:0>:** Capture Trigger Input Selection bits

CTS<1:0>	Connection			
	CCP1	CCP2	CCP3	CCP4
111	CLC4_out			
110	CLC3_out			
101	CLC2_out			
100	CLC1_out			
011	IOC_Interrupt			
010	CMP2_output			
001	CMP1_output			
000	Pin selected by CCP1PPS	Pin selected by CCP2PPS	Pin selected by CCP3PPS	Pin selected by CCP4PPS

REGISTER 23-4: CCPRxL: CCPx REGISTER LOW BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
RL<7:0>							
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 7-0 MODE = Capture Mode:
RL<7:0>: LSB of captured TMR1 value
MODE = Compare Mode:
RL<7:0>: LSB compared to TMR1 value
MODE = PWM Mode && FMT = 0:
RL<7:0>: CCPW<7:0> – Pulse-Width LS 8 bits
MODE = PWM Mode && FMT = 1:
RL<7:6>: CCPW<1:0> – Pulse-Width LS 2 bits
RL<5:0>: Not used

26.10 Auto-Shutdown

Auto-shutdown is a method to immediately override the CWG output levels with specific overrides that allow for safe shutdown of the circuit. The shutdown state can be either cleared automatically or held until cleared by software. The auto-shutdown circuit is illustrated in Figure 26-14.

26.10.1 SHUTDOWN

The shutdown state can be entered by either of the following two methods:

- Software generated
- External Input

26.10.1.1 Software Generated Shutdown

Setting the SHUTDOWN bit of the CWGxAS0 register will force the CWG into the shutdown state.

When the auto-restart is disabled, the shutdown state will persist as long as the SHUTDOWN bit is set.

When auto-restart is enabled, the SHUTDOWN bit will clear automatically and resume operation on the next rising edge event. The SHUTDOWN bit indicates when a shutdown condition exists. The bit may be set or cleared in software or by hardware.

26.10.1.2 External Input Source

External shutdown inputs provide the fastest way to safely suspend CWG operation in the event of a Fault condition. When any of the selected shutdown inputs goes active, the CWG outputs will immediately go to the specified override levels without software delay. The override levels are selected by the LSBD<1:0> and LSAC<1:0> bits of the CWGxAS0 register (Register 26-6). Several input sources can be selected to cause a shutdown condition. All input sources are active-low. The sources are:

- Pin selected by CWGxPPS
- Timer2 postscaled output
- Timer4 postscaled output
- Timer6 postscaled output
- Comparator 1 output
- Comparator 2 output
- CLC2 output

Shutdown input sources are individually enabled by the ASxE bits of the CWGxAS1 register (Register 26-7).

Note: Shutdown inputs are level sensitive, not edge sensitive. The shutdown state cannot be cleared, except by disabling auto-shutdown, as long as the shutdown input level persists.

26.10.1.3 Pin Override Levels

The levels driven to the CWG outputs during an auto-shutdown event are controlled by the LSBD<1:0> and LSAC<1:0> bits of the CWGxAS0 register (Register 26-6). The LSBD<1:0> bits control CWGxB/D output levels, while the LSAC<1:0> bits control the CWGxA/C output levels.

26.10.1.4 Auto-Shutdown Interrupts

When an auto-shutdown event occurs, either by software or hardware setting SHUTDOWN, the CWGxIF flag bit of the respective PIR register is set.

26.11 Auto-Shutdown Restart

After an auto-shutdown event has occurred, there are two ways to resume operation:

- Software controlled
- Auto-restart

In either case, the shutdown source must be cleared before the restart can take place. That is, either the shutdown condition must be removed, or the corresponding ASxE bit must be cleared.

26.11.1 SOFTWARE-CONTROLLED RESTART

If the REN bit of the CWGxAS0 register is clear (REN = 0), the CWG module must be restarted after an auto-shutdown event through software.

Once all auto-shutdown sources are removed, the software must clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.

Note: The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

26.11.2 AUTO-RESTART

If the REN bit of the CWGxAS0 register is set (REN = 1), the CWG module will restart from the shutdown state automatically.

Once all auto-shutdown conditions are removed, the hardware will automatically clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.

Note: The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

28.0 NUMERICALLY CONTROLLED OSCILLATOR (NCO) MODULE

The Numerically Controlled Oscillator (NCO) module is a timer that uses overflow from the addition of an increment value to divide the input frequency. The advantage of the addition method over simple counter driven timer is that the output frequency resolution does not vary with the divider value. The NCO is most useful for applications that require frequency accuracy and fine resolution at a fixed duty cycle.

Features of the NCO include:

- 20-bit Increment Function
- Fixed Duty Cycle mode (FDC) mode
- Pulse Frequency (PF) mode
- Output Pulse-Width Control
- Multiple Clock Input Sources
- Output Polarity Control
- Interrupt Capability

Figure 28-1 is a simplified block diagram of the NCO module.

TABLE 30-2: MD1SRC SELECTION MUX CONNECTIONS

MS<4:0>		Connection
1 1111 – 1 0111	31-23	Reserved
1 0110	22	SPI1 SDO
1 0101	21	Reserved
1 0100	20	UART2 TX
1 0011	19	UART1 TX
1 0010	18	CLC4 OUT
1 0001	17	CLC3 OUT
1 0000	16	CLC2 OUT
0 1111	15	CLC1 OUT
0 1110	14	CMP2 OUT
0 1101	13	CMP1 OUT
0 1100	12	NCO1 OUT
0 1011	11	Reserved
0 1010	10	Reserved
0 1001	9	PWM8 OUT
0 1000	8	PWM7 OUT
0 0111	7	PWM6 OUT
0 0110	6	PWM5 OUT
0 0101	5	CCP4 OUT
0 0100	4	CCP3 OUT
0 0011	3	CCP2 OUT
0 0010	2	CCP1 OUT
0 0001	1	DSM1 BIT
0 0000	0	Pin selected by MDSRCPPS

TABLE 30-3: SUMMARY OF REGISTERS ASSOCIATED WITH DATA SIGNAL MODULATOR MODE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
MD1CON0	EN	—	OUT	OPOL	—	—	—	BIT	469
MD1CON1	—	—	CHPOL	CHSYNC	—	—	CLPOL	CLSYNC	470
MD1CARH	—	—	—	—	—	CHS<2:0>			471
MD1CARL	—	—	—	—	—	CLS<2:0>			471
MDSRC	—	—	—	—	SRCS<3:0>				472

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used in the Data Signal Modulator mode.

TABLE 41-2: INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word	Status Affected	Notes
				MSb LSb		
BYTE-ORIENTED FILE REGISTER INSTRUCTIONS						
ADDWF	f, d ,a	Add WREG and f	1	`0010 01da ffff ffff`	C, DC, Z, OV, N	
ADDWFC	f, d, a	Add WREG and Carry bit to f	1	`0010 00da ffff ffff`	C, DC, Z, OV, N	
ANDWF	f, d, a	AND WREG with f	1	`0001 01da ffff ffff`	Z, N	
CLRF	f, a	Clear f	1	`0110 101a ffff ffff`	Z	
COMF	f, d, a	Complement f	1	`0001 11da ffff ffff`	Z, N	
DECF	f, d, a	Decrement f	1	`0000 01da ffff ffff`	C, DC, Z, OV, N	
INCF	f, d, a	Increment f	1	`0010 10da ffff ffff`	C, DC, Z, OV, N	
IORWF	f, d, a	Inclusive OR WREG with f	1	`0001 00da ffff ffff`	Z, N	
MOVF	f, d, a	Move f to WREG or f	1	`0101 00da ffff ffff`	Z, N	

MOVFF	f_s, f_d	Move f_s (source) to 1st word
		f_d (destination) 2nd word
			2	`1100 ffff ffff ffff`
				`1111 ffff ffff ffff`
					None	2, 3

MOVFFL	f_s, f_d	Move f_s (source) to
		g (full destination)
		f_d (full destination)3rd word
			3	`0000 0000 0110 ffff`
				`1111 ffff ffff ffgg`
				`1111 gggg gggg gggg`
					None	2

MOVWF	f, a	Move WREG to f	1	`0110 111a ffff ffff`	None	
MULWF	f, a	Multiply WREG with f	1	`0000 001a ffff ffff`	None	
NEGF	f, a	Negate f	1	`0110 110a ffff ffff`	C, DC, Z, OV, N	
RLCF	f, d, a	Rotate Left f through Carry	1	`0011 01da ffff ffff`	C, Z, N	
RLNCF	f, d, a	Rotate Left f (No Carry)	1	`0100 01da ffff ffff`	Z, N	
RRCF	f, d, a	Rotate Right f through Carry	1	`0011 00da ffff ffff`	C, Z, N	
RRNCF	f, d, a	Rotate Right f (No Carry)	1	`0100 00da ffff ffff`	Z, N	
SETF	f, a	Set f	1	`0110 100a ffff ffff`	None	
SUBFWB	f, d, a	Subtract f from WREG with borrow	1	`0101 01da ffff ffff`	C, DC, Z, OV, N	
SUBWF	f, d, a	Subtract WREG from f	1	`0101 11da ffff ffff`	C, DC, Z, OV, N	
SUBWFB	f, d, a	Subtract WREG from f with borrow	1	`0101 10da ffff ffff`	C, DC, Z, OV, N	
SWAPF	f, d, a	Swap nibbles in f	1	`0011 10da ffff ffff`	None	
XORWF	f, d, a	Exclusive OR WREG with f	1	`0001 10da ffff ffff`	Z, N	
BYTE-ORIENTED SKIP INSTRUCTIONS						
CPFSEQ	f, a	Compare f with WREG, skip =	1 (2 or 3)	`0110 001a ffff ffff`	None	1
CPFSGT	f, a	Compare f with WREG, skip >	1 (2 or 3)	`0110 010a ffff ffff`	None	1
CPFSLT	f, a	Compare f with WREG, skip <	1 (2 or 3)	`0110 000a ffff ffff`	None	1
DECFSZ	f, d, a	Decrement f, Skip if 0	1 (2 or 3)	`0010 11da ffff ffff`	None	1
DCFSNZ	f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	`0100 11da ffff ffff`	None	1
INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	`0011 11da ffff ffff`	None	1
INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	`0100 10da ffff ffff`	None	1
TSTFSZ	f, a	Test f, skip if 0	1 (2 or 3)	`0110 011a ffff ffff`	None	1
BIT-ORIENTED FILE REGISTER INSTRUCTIONS						
BCF	f, b, a	Bit Clear f	1	`1001 bbba ffff ffff`	None	
BSF	f, b, a	Bit Set f	1	`1000 bbba ffff ffff`	None	
BTG	f, d, a	Bit Toggle f	1	`0111 bbba ffff ffff`	None	
BIT-ORIENTED SKIP INSTRUCTIONS						
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	`1011 bbba ffff ffff`	None	1
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	`1010 bbba ffff ffff`	None	1

Note 1: If Program Counter (PC) is modified or a conditional test is true, the instruction requires an additional cycle. The extra cycle is executed as a `NOP`.

2: Some instructions are multi word instructions. The second/third words of these instructions will be decoded as a `NOP`, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.

3: f_s and f_d do not cover the full memory range. 2 MSBs of bank selection are forced to ‘b00 to limit the range of these instructions to lower 4k addressing space.

TSTFSZ — Test f, skip if 0

Syntax:	TSTFSZ f {,a}
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	skip if f = `0`
Status Affected:	None

Encoding:

0110	011a	ffff	ffff

Description: If ‘f’ = `0`, the next instruction fetched during the current instruction execution is discarded and a `NOP` is executed, making this a 2-cycle instruction.
If ‘a’ is ‘`0`’, the Access Bank is selected. If ‘a’ is ‘`1`’, the BSR is used to select the GPR bank.
If ‘a’ is ‘`0`’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 41.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    TSTFSZ  CNT, 1
NZERO   :
ZERO    :
```

Before Instruction
PC = Address `(HERE)`
After Instruction
If CNT = 00h,
PC = Address `(ZERO)`
If CNT ≠ 00h,
PC = Address `(NZERO)`

XORLW — Exclusive OR literal with W

Syntax:	XORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .XOR. k → W
Status Affected:	N, Z

Encoding:

0000	1010	kkkk	kkkk

Description: The contents of W are XORed with the 8-bit literal ‘k’. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to W

Example: `XORLW 0AFh`

Before Instruction
W = B5h
After Instruction
W = 1Ah

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3CE6h	CLKRCON	EN	—	—	DC		DIV			113
3CE5h	CLKRCLK	—	—	—	—	CLK				114
3CE4h - 3C7Fh	—	Unimplemented								
3C7Eh	CLCDATA0	—	—	—	—	CLC4OUT	CLC3OUT	CLC2OUT	CLC1OUT	447
3C7Dh	CLC1GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	446
3C7Ch	CLC1GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	445
3C7Bh	CLC1GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	444
3C7Ah	CLC1GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	443
3C79h	CLC1SEL3	D4S								442
3C78h	CLC1SEL2	D3S								442
3C77h	CLC1SEL1	D2S								442
3C76h	CLC1SEL0	D1S								442
3C75h	CLC1POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	441
3C74h	CLC1CON	EN	OE	OUT	INTP	INTN	MODE			440
3C73h	CLC2GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	446
3C72h	CLC2GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	445
3C71h	CLC2GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	444
3C70h	CLC2GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	443
3C6Fh	CLC2SEL3	D4S								442
3C6Eh	CLC2SEL2	D3S								442
3C6Dh	CLC2SEL1	D2S								442
3C6Ch	CLC2SEL0	D1S								442
3C6Bh	CLC2POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	441
3C6Ah	CLC2CON	EN	OE	OUT	INTP	INTN	MODE			440
3C69h	CLC3GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	446
3C68h	CLC3GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	445
3C67h	CLC3GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	444
3C66h	CLC3GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	443
3C65h	CLC3SEL3	D4S								442
3C64h	CLC3SEL2	D3S								442
3C63h	CLC3SEL1	D2S								442
3C62h	CLC3SEL0	D1S								443
3C61h	CLC3POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	441
3C60h	CLC3CON	EN	OE	OUT	INTP	INTN	MODE			440
3C5Fh	CLC4GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	446
3C5Eh	CLC4GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	445
3C5Dh	CLC4GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	444
3C5Ch	CLC4GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	443
3C5Bh	CLC4SEL3	D4S								442
3C5Ah	CLC4SEL2	D3S								442
3C59h	CLC4SEL1	D2S								442
3C58h	CLC4SEL0	D1S								443
3C57h	CLC4POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	441
3C56h	CLC4CON	EN	OE	OUT	INTP	INTN	MODE			440
3C55h - 3C00h	—	Unimplemented								
3BFFh	DMA1SIRQ	—	SIRQ							256

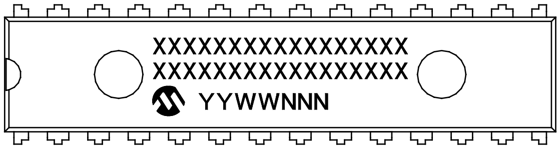
Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Unimplemented in LF devices.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.
4: Unimplemented in PIC18(L)F45/55K42.

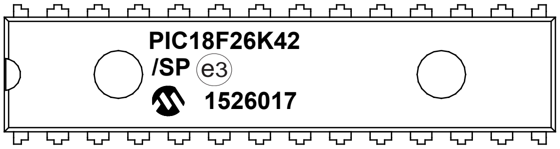
46.0 PACKAGING INFORMATION

Package Marking Information

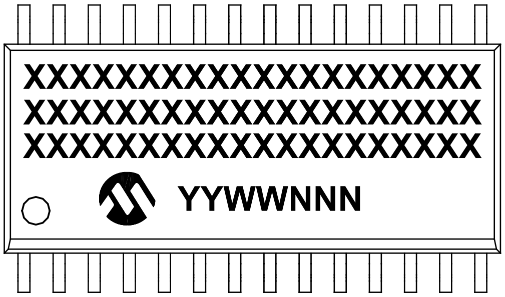
28-Lead SPDIP (.300")



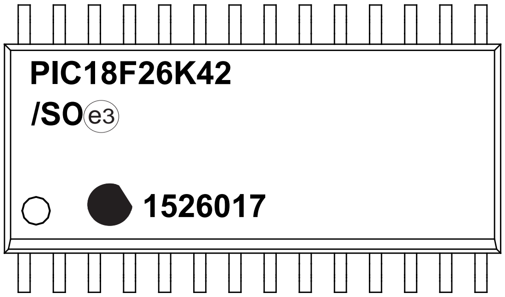
Example



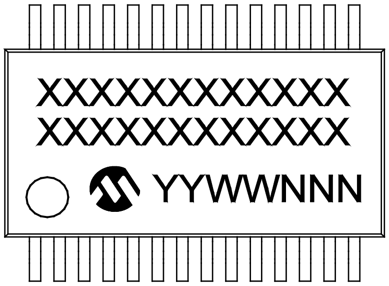
28-Lead SOIC (7.50 mm)



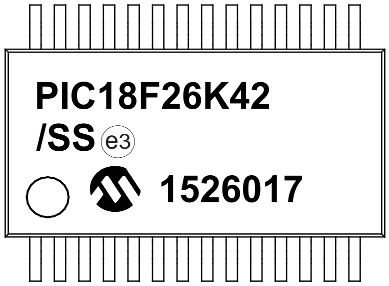
Example



28-Lead SSOP (5.30 mm)



Example

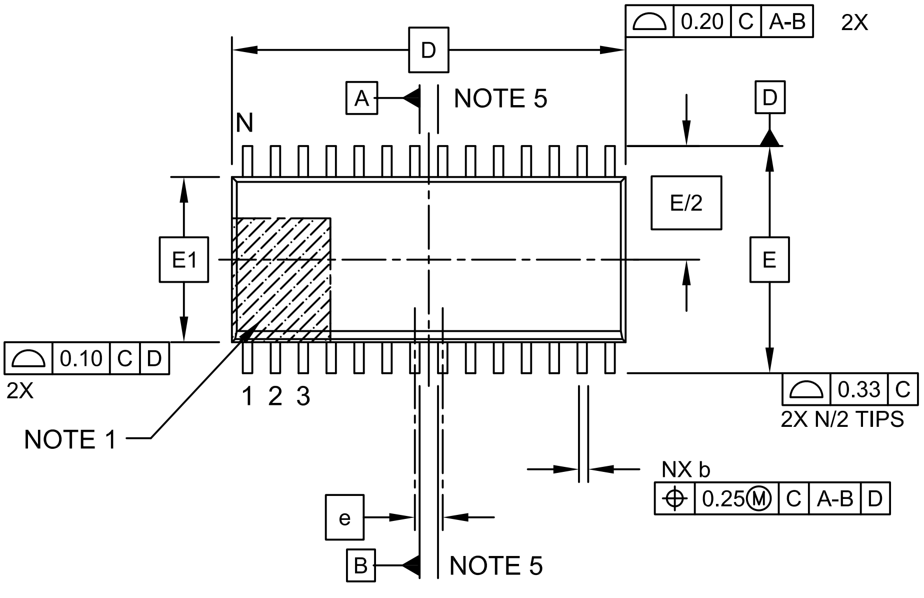


Legend:	XX...X	Customer-specific information or Microchip part number
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

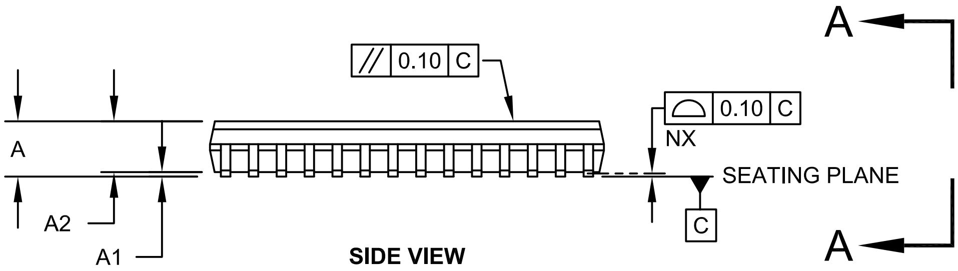
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

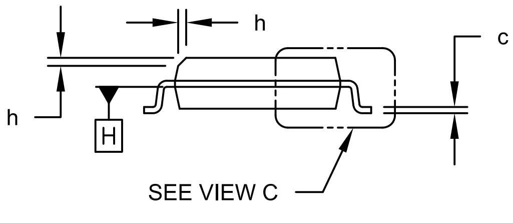
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW



VIEW A-A

Microchip Technology Drawing C04-052C Sheet 1 of 2