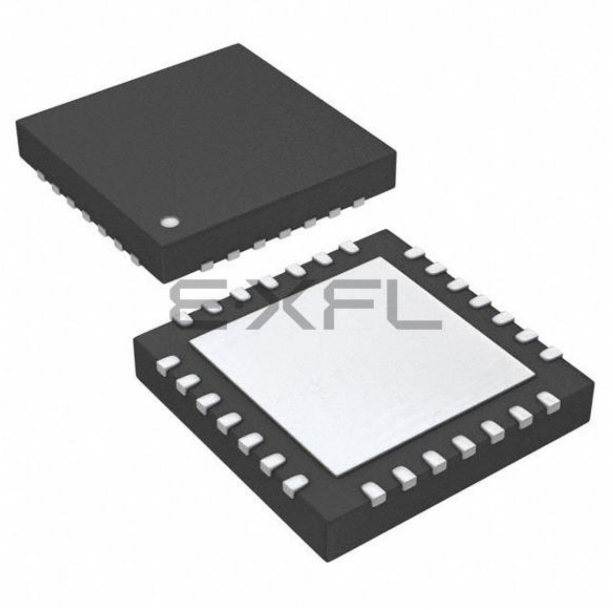
Microchip Technology - PIC18F24K42T-I/ML Datasheet



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	25
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K 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-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f24k42t-i-ml

TABLE 4-3: SPECIAL FUNCTION REGISTER MAP FOR PIC18(L)F24/25K42 DEVICES BANK 63

3FFFh	TOSU	3FDFh	INDF2	3FBFh	—	3F9Fh	T4PR	3F7Fh	CCP1CAP	3F5Fh	CCPTMRS1	3F3Fh	NCO1CLK	3F1Fh	SMT1CON1
3FFEh	TOSH	3FDEh	POSTINC2	3FBEh	—	3F9Eh	T4TMR	3F7Eh	CCP1CON	3F5Eh	CCPTMRS0	3F3Eh	NCO1CON	3F1Eh	SMT1CON0
3FFDh	TOSL	3FDDh	POSTDEC2	3FBDh	—	3F9Dh	T5CLK	3F7Dh	CCPR1H	3F5Dh	—	3F3Dh	NCO1INCU	3F1Dh	SMT1PRU
3FFCh	STKPTR	3FDCh	PRECIN2	3FBCh	LATC	3F9Ch	T5GATE	3F7Ch	CCPR1L	3F5Ch	—	3F3Ch	NCO1INCH	3F1Ch	SMT1PRH
3FFBh	PCLATU	3FDBh	PLUSW2	3FBBh	LATB	3F9Bh	T5GCON	3F7Bh	CCP2CAP	3F5Bh	—	3F3Bh	NCO1INCL	3F1Bh	SMT1PRL
3FFAh	PCLATH	3FDAh	FSR2H	3FBAh	LATA	3F9Ah	T5CON	3F7Ah	CCP2CON	3F5Ah	CWG1STR	3F3Ah	NCO1ACCU	3F1Ah	SMT1CPWU
3FF9h	PCL	3FD9h	FSR2L	3FB9h	T0CON1	3F99h	TMR5H	3F79h	CCPR2H	3F59h	CWG1AS1	3F39h	NCO1ACCH	3F19h	SMT1CPWH
3FF8h	TBLPTRU	3FD8h	STATUS	3FB8h	T0CON0	3F98h	TMR5L	3F78h	CCPR2L	3F58h	CWG1AS0	3F38h	NCO1ACCL	3F18h	SMT1CPWL
3FF7h	TBLPTRH	3FD7h	IVTBASEU	3FB7h	TMR0H	3F97h	T6RST	3F77h	CCP3CAP	3F57h	CWG1CON1	3F37h	—	3F17h	SMT1CPRU
3FF6h	TBLPTRL	3FD6h	IVTBASEH	3FB6h	TMR0L	3F96h	T6CLK	3F76h	CCP3CON	3F56h	CWG1CON0	3F36h	—	3F16h	SMT1CPRH
3FF5h	TABLAT	3FD5h	IVTBASEL	3FB5h	T1CLK	3F95h	T6HLT	3F75h	CCPR3H	3F55h	CWG1DBF	3F35h	—	3F15h	SMT1CPRL
3FF4h	PRODH	3FD4h	IVTLOCK	3FB4h	T1GATE	3F94h	T6CON	3F74h	CCPR3L	3F54h	CWG1DBR	3F34h	—	3F14h	SMT1TMRU
3FF3h	PRODL	3FD3h	INTCON1	3FB3h	T1GCON	3F93h	T6PR	3F73h	CCP4CAP	3F53h	CWG1ISM	3F33h	—	3F13h	SMT1TMRH
3FF2h	—	3FD2h	INTCON0	3FB2h	T1CON	3F92h	T6TMR	3F72h	CCP4CON	3F52h	CWG1CLK	3F32h	—	3F12h	SMT1TMRL
3FF1h	PCON1	3FD1h	—	3FB1h	TMR1H	3F91h	—	3F71h	CCPR4H	3F51h	CWG2STR	3F31h	—	3F11h	—
3FF0h	PCON0	3FD0h	—	3FB0h	TMR1L	3F90h	—	3F70h	CCPR4L	3F50h	CWG2AS1	3F30h	—	3F10h	—
3FEFh	INDF0	3FCFh	—	3FAFh	T2RST	3F8Fh	—	3F6Fh	—	3F4Fh	CWG2AS0	3F2Fh	—	3F0Fh	—
3FEEh	POSTINC0	3FCEh	PORTE	3FAEh	T2CLK	3F8Eh	—	3F6Eh	PWM5CON	3F4Eh	CWG2CON1	3F2Eh	—	3F0Eh	—
3FEDh	POSTDEC0	3FCDh	—	3FADh	T2HLT	3F8Dh	—	3F6Dh	PWM5DCH	3F4Dh	CWG2CON0	3F2Dh	—	3F0Dh	—
3FECh	PRECIN0	3FCCh	PORTC	3FACh	T2CON	3F8Ch	—	3F6Ch	PWM5DCL	3F4Ch	CWG2DBF	3F2Ch	—	3F0Ch	—
3FEBh	PLUSW0	3FCBh	PORTB	3FABh	T2PR	3F8Bh	—	3F6Bh	—	3F4Bh	CWG2DBR	3F2Bh	—	3F0Bh	—
3FEAh	FSR0H	3FCAh	—	3FAAh	T2TMR	3F8Ah	—	3F6Ah	PWM6CON	3F4Ah	CWG2ISM	3F2Ah	—	3F0Ah	—
3FE9h	FSR0L	3FC9h	—	3FA9h	T3CLK	3F89h	—	3F69h	PWM6DCH	3F49h	CWG2CLK	3F29h	—	3F09h	—
3FE8h	WREG	3FC8h	—	3FA8h	T3GATE	3F88h	—	3F68h	PWM6DCL	3F48h	CWG3STR	3F28h	—	3F08h	—
3FE7h	INDF1	3FC7h	—	3FA7h	T3GCON	3F87h	—	3F67h	—	3F47h	CWG3AS1	3F27h	—	3F07h	—
3FE6h	POSTINC1	3FC6h	—	3FA6h	T3CON	3F86h	—	3F66h	PWM7CON	3F46h	CWG3AS0	3F26h	—	3F06h	—
3FE5h	POSTDEC1	3FC5h	—	3FA5h	TMR3H	3F85h	—	3F65h	PWM7DCH	3F45h	CWG3CON1	3F25h	—	3F05h	—
3FE4h	PRECIN1	3FC4h	TRISC	3FA4h	TMR3L	3F84h	—	3F64h	PWM7DCL	3F44h	CWG3CON0	3F24h	—	3F04h	—
3FE3h	PLUSW1	3FC3h	TRISB	3FA3h	T4RST	3F83h	—	3F63h	—	3F43h	CWG3DBF	3F23h	SMT1WIN	3F03h	—
3FE2h	FSR1H	3FC2h	TRISA	3FA2h	T4CLK	3F82h	—	3F62h	PWM8CON	3F42h	CWG3DBR	3F22h	SMT1SIG	3F02h	—
3FE1h	FSR1L	3FC1h	—	3FA1h	T4HLT	3F81h	—	3F61h	PWM8DCH	3F41h	CWG3ISM	3F21h	SMT1CLK	3F01h	—
3FE0h	BSR	3FC0h	—	3FA0h	T4CON	3F80h	—	3F60h	PWM8DCL	3F40h	CWG3CLK	3F20h	SMT1STAT	3F00h	—

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

8.14 Register Definitions: Power Control

REGISTER 8-2: PCON0: POWER CONTROL REGISTER 0

R/W/HS-0/q	R/W/HS-0/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-0/u	R/W/HC-q/u
STKOVF	STKUNF	$\overline{\text{WDTWV}}$	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

Legend:

HC = Bit is cleared by hardware		HS = Bit is set by hardware
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-m/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 **STKOVF:** Stack Overflow Flag bit

1 = A Stack Overflow occurred (more `CALL`s than fit on the stack)
0 = A Stack Overflow has not occurred or set to ‘0’ by firmware

bit 6 **STKUNF:** Stack Underflow Flag bit

1 = A Stack Underflow occurred (more `RETURN`s than `CALL`s)
0 = A Stack Underflow has not occurred or set to ‘0’ by firmware

bit 5 **$\overline{\text{WDTWV}}$:** Watchdog Window Violation bit

1 = A WDT window violation has not occurred or set to ‘1’ by firmware
0 = A `CLRWDT` instruction was issued when the WDT Reset window was closed (set to ‘0’ in hardware when a WDT window violation Reset occurs)

bit 4 **$\overline{\text{RWDT}}$:** WDT Reset Flag bit

1 = A WDT overflow/time-out Reset has not occurred or set to ‘1’ by firmware
0 = A WDT overflow/time-out Reset has occurred (set to ‘0’ in hardware when a WDT Reset occurs)

bit 3 **$\overline{\text{RMCLR}}$:** $\overline{\text{MCLR}}$ Reset Flag bit

1 = A $\overline{\text{MCLR}}$ Reset has not occurred or set to ‘1’ by firmware
0 = A $\overline{\text{MCLR}}$ Reset has occurred (set to ‘0’ in hardware when a $\overline{\text{MCLR}}$ Reset occurs)

bit 2 **$\overline{\text{RI}}$:** `RESET` Instruction Flag bit

1 = A `RESET` instruction has not been executed or set to ‘1’ by firmware
0 = A `RESET` instruction has been executed (set to ‘0’ in hardware upon executing a `RESET` instruction)

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit

1 = No Power-on Reset occurred or set to ‘1’ by firmware
0 = A Power-on Reset occurred (set to ‘0’ in hardware when a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit

1 = No Brown-out Reset occurred or set to ‘1’ by firmware
0 = A Brown-out Reset occurred (set to ‘0’ in hardware when a Brown-out Reset occurs)

EXAMPLE 11-3: SETTING UP VECTORED INTERRUPTS USING MPASM

```
ISR_TMR0:  CODE        0x8C0                 ; ISR code at 0x08C0 in PFM
        BANKSEL        PIR0                  ; Select bank for PIR0
        BCF            PIR3, TMR0IF          ; Clear TMR0IF
        BTG            LATC, 0, ACCESS       ; Code to execute in ISR
        RETFIE         1                     ; Return from ISR

InterruptInit:
        BANKSEL        INTCON0               ; Select bank for INTCON0
        BSF            INTCON0, GIEH         ; Enable high priority interrupts
        BSF            INTCON0, GIEL         ; Enable low priority interrupts
        BSF            INTCON0, IPEN_INTCON0 ; Enable interrupt priority

        BANKSEL        PIE0                  ; Select bank for PIE0
        BSF            PIE3, TMR0IE          ; Enable TMR0 interrupt
        BSF            PIE4, TMR1IE          ; Enable TMR1 interrupt

        BCF            IPR3, TMR0IP          ; Make TMR0 interrupt low priority
        RETURN         1

VectorTableInit:
        ; Set IVTBASE (optional - default is 0x000008)
        MOVLW          0x00                  ; This is optional
        MOVWF          IVTBASEU, ACCESS      ; If not included, then the
        MOVLW          0x40                  ; hardware default value of
        MOVWF          IVTBASEH, ACCESS      ; 0x0008 will be taken.
        MOVLW          0x08
        MOVWF          IVTBASEL, ACCESS

        ; TMR0 vector at IVTBASE + 2*(TMR0 vector number i.e. 31) = 0x4046
        MOVLW          0x00                  ; Load TBLPTR with the
        MOVWF          TBLPTRU, ACCESS       ; PFM memory location to be
        MOVLW          0x40                  ; written to.
        MOVWF          TBLPTRH, ACCESS
        MOVLW          0x46
        MOVWF          TBLPTRL, ACCESS

        ; Write the contents of TMR0 vector location
        ; ISR_TMR0_ADDRESS >> 2 = 0x08C0 >> 2 = 0x0230
        MOVLW          0x30                  ; Low byte first
        MOVWF          TABLAT, ACCESS
        TBLWT*+                              ; Write to temp table latch

        MOVLW          0x02                  ; High byte next
        MOVWF          TABLAT, ACCESS
        TBLWT*+                              ; Write to temp table latch

        ; Write to PFM now using NVMCON
        BANKSEL        NVMCON1               ; Select bank for NVMCON1
        MOVLW          0x84                  ; Setting to write to PFM
        MOVWF          NVMCON1

        MOVLW          0x55                  ; Required unlock sequence
        MOVWF          NVMCON2
        MOVLW          0xAA
        MOVWF          NVMCON2
        BSF            NVMCON1, WR           ; Start writing to PFM

        BTFSC          NVMCON1, WR           ; Wait for write to complete
        GOTO           $-2

        RETURN         1
```

REGISTER 11-18: PIE4: PERIPHERAL INTERRUPT Enable REGISTER 4

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CLC1IE	CWG1IE	NCO1IE	—	CCP1IE	TMR2IE	TMR1GIE	TMR1IE
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 **CLC1IE:** CLC1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **CWG1IE:** CWG1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **NCO1IE:** NCO1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 4 **Unimplemented:** Read as ‘0’

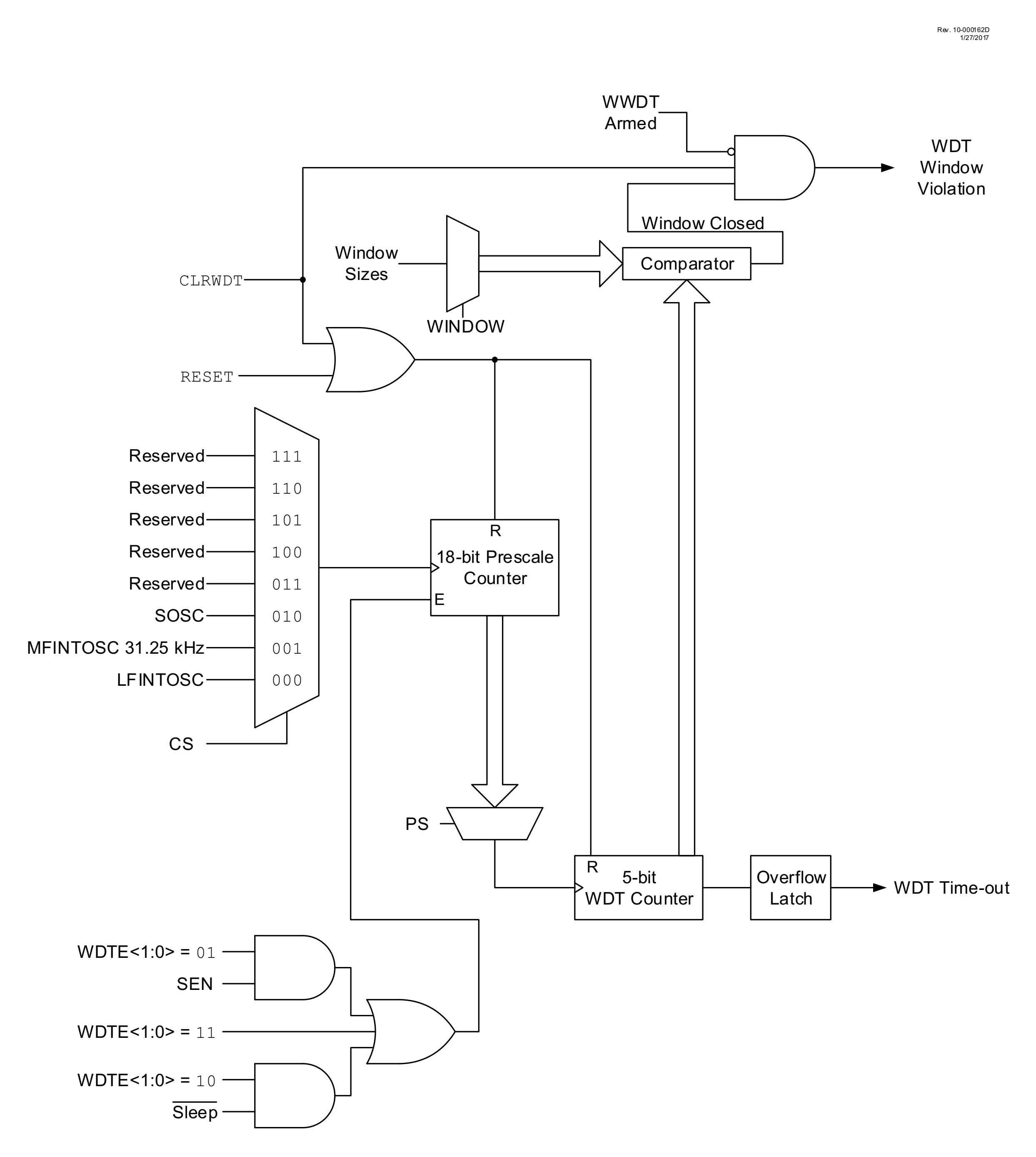
bit 3 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **TMR2IE:** TMR2 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **TMR1GIE:** TMR1 Gate Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **TMR1IE:** TMR1 Interrupt Enable bit
1 = Enabled
0 = Disabled

FIGURE 13-1: WINDOWED WATCHDOG TIMER BLOCK DIAGRAM



13.1 Independent Clock Source

The WWDT can derive its time base from either the 31 kHz LFINTOSC or 31.25 kHz MFINTOSC internal oscillators, depending on the value of WDTE<1:0> Configuration bits.

If WDTE = `2'b1x`, then the clock source will be enabled depending on the WDTCCS<2:0> Configuration bits.

If WDTE = `2'b01`, the SEN bit should be set by software to enable WWDT, and the clock source is enabled by the CS bits in the WDTCON1 register.

Time intervals in this chapter are based on a minimum nominal interval of 1 ms. See **Section 46.0 "Electrical Specifications"** for LFINTOSC and MFINTOSC tolerances.

13.2 WWDT Operating Modes

The Windowed Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Words. See Table 13-1.

13.2.1 WWDT IS ALWAYS ON

When the WDTE bits of Configuration Words are set to '`11`', the WWDT is always on.

WWDT protection is active during Sleep.

13.2.2 WWDT IS OFF IN SLEEP

When the WDTE bits of Configuration Words are set to '`10`', the WWDT is on, except in Sleep.

WWDT protection is not active during Sleep.

13.2.3 WWDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Words are set to '`01`', the WWDT is controlled by the SEN bit of the WDTCON0 register.

WWDT protection is unchanged by Sleep. See Table 13-1 for more details.

TABLE 13-1: WWDT OPERATING MODES

WDTE<1:0>	SEN	Device Mode	WWDT Mode
11	x	X	Active
10	x	Awake	Active
		Sleep	Disabled
01	1	X	Active
	0	X	Disabled
00	x	X	Disabled

13.3 Time-out Period

If the WDTCPS<4:0> Configuration bits default to `5'b11111`, then the PS bits of the WDTCON0 register set the time-out period from 1 ms to 256 seconds (nominal). If any value other than the default value is assigned to WDTCPS<4:0> Configuration bits, then the timer period will be based on the WDTCPS<4:0> bits in the CONFIG3L register. After a Reset, the default time-out period is 2s.

13.4 Watchdog Window

The Windowed Watchdog Timer has an optional Windowed mode that is controlled by the WDTCWS<2:0> Configuration bits and WINDOW<2:0> bits of the WDTCON1 register. In the Windowed mode, the `CLRWDT` instruction must occur within the allowed window of the WDT period. Any `CLRWDT` instruction that occurs outside of this window will trigger a window violation and will cause a WWDT Reset, similar to a WWDT time out. See Figure 13-2 for an example.

The window size is controlled by the WINDOW<2:0> Configuration bits, or the WINDOW<2:0> bits of WDTCON1, if WDTCWS<2:0> = `111`.

The five Most Significant bits of the WDTTMR register are used to determine whether the window is open, as defined by the WINDOW<2:0> bits of the WDTCON1 register.

In the event of a window violation, a Reset will be generated and the $\overline{\text{WDTWV}}$ bit of the PCON0 register will be cleared. This bit is set by a POR or can be set in firmware.

13.5 Clearing the WWDT

The WWDT is cleared when any of the following conditions occur:

- Any Reset
- Valid `CLRWDT` instruction is executed
- Device enters Sleep
- Exit Sleep by Interrupt
- WWDT is disabled
- Oscillator Start-up Timer (OST) is running
- Any write to the WDTCON0 or WDTCON1 registers

13.5.1 `CLRWDT` CONSIDERATIONS (WINDOWED MODE)

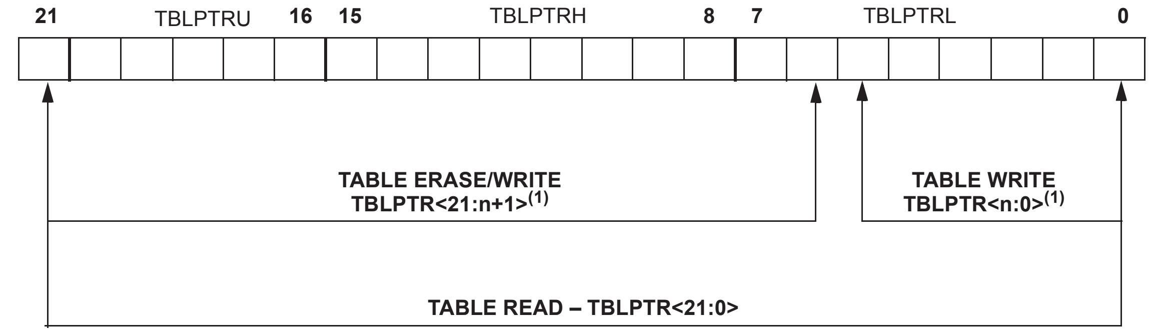
When in Windowed mode, the WWDT must be armed before a `CLRWDT` instruction will clear the timer. This is performed by reading the WDTCON0 register. Executing a `CLRWDT` instruction without performing such an arming action will trigger a window violation regardless of whether the window is open or not.

See Table 13-2 for more information.

TABLE 15-3: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS

Example	Operation on Table Pointer
`TBLRD*` `TBLWT*`	TBLPTR is not modified
`TBLRD*+` `TBLWT*+`	TBLPTR is incremented after the read/write
`TBLRD*-` `TBLWT*-`	TBLPTR is decremented after the read/write
`TBLRD+*` `TBLWT+*`	TBLPTR is incremented before the read/write

FIGURE 15-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



Note 1: Refer to Table 15-2 for the row size values.

Depending on the priority of the DMA with respect to CPU execution (Refer to **Section 3.2 "Memory Access Scheme"** for more information), the DMA Controller can move data through two methods:

- Stalling the CPU execution until it has completed its transfers (DMA has higher priority over the CPU in this mode of operation)
- Utilizing unused CPU cycles for DMA transfers (CPU has higher priority over the DMA in this mode of operation). Unused CPU cycles are referred to as bubbles which are instruction cycles available for use by the DMA to perform read and write operations. In this way, the effective bandwidth for handling data is increased; at the same time, DMA operations can proceed without causing a processor stall.

17.4 DMA Interface

The DMA module transfers data from the source to the destination one byte at a time, this smallest data movement is called a DMA data transaction. A DMA Message refers to one or more DMA data transactions.

Each DMA data transaction consists of two separate actions:

- Reading the Source Address Memory and storing the value in the DMA Buffer register
- Writing the contents of the DMA Buffer register to the Destination Address Memory

> **Note:** DMA data movement is a two-cycle operation.

The XIP bit (DMAxCON0 register) is a status bit to indicate whether or not the data in the DMAxBUF register has been written to the destination address. If the bit is set then data is waiting to be written to the destination. If clear it means that either data has been written to the destination or that no source read has occurred.

The DMA has read access to PFM, Data EEPROM, and SFR/GPR space, and write access to SFR/GPR space. Based on these memory access capabilities, the DMA can support the following memory transactions:

TABLE 17-1: DMA MEMORY ACCESS

Read Source	Write Destination
Program Flash Memory	GPR
Program Flash Memory	SFR
Data EE	GPR
Data EE	SFR
GPR	GPR
SFR	GPR
GPR	SFR
SFR	SFR

Even though the DMA module has access to all memory and peripherals that are also available to the CPU, it is recommended that the DMA does not access any register that is part of the System arbitration. The DMA, as a system arbitration client should not be read or written by itself or by another DMA instantiation.

The following sections discuss the various control interfaces required for DMA data transfers.

17.4.1 DMA ADDRESSING

The start addresses for the source read and destination write operations are set using the DMAxSSA <21:0> and DMAxDSA <15:0> registers, respectively.

When the DMA Message transfers are in progress, the DMAxSPTR <21:0> and DMAxDPTR <15:0> registers contain the current address pointers for each source read and destination write operation, these registers are modified after each transaction based on the Address mode selection bits.

The SMODE and DMODE bits in the DMAxCON1 control register determine the address modes of operation by controlling how the DMAxSPTR <21:0> and DMAxDPTR <15:0> bits are updated after every DMA data transaction combination (Figure 17-2).

Each address can be separately configured to:

- Remain unchanged
- Increment by 1
- Decrement by 1

REGISTER 17-6: DMAxSSAU – DMAx SOURCE START ADDRESS UPPER 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
—	—	SSA<21:16>					
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	1 = bit is set	0 = bit is cleared	x = bit is unknown
			u = bit is unchanged

bit 7-0 **SSA<21:16>:** Source Start Address bits

REGISTER 17-7: DMAxSPTRL – DMAx SOURCE POINTER LOW REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SPTR<7:0>							
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	1 = bit is set	0 = bit is cleared	x = bit is unknown
			u = bit is unchanged

bit 15-0 **SPTR<7:0>:** Current Source Address Pointer

REGISTER 17-8: DMAxSPTRH – DMAx SOURCE POINTER HIGH REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SPTR<15:8>							
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	1 = bit is set	0 = bit is cleared	x = bit is unknown
			u = bit is unchanged

bit 5-0 **SPTR<15:8>:** Current Source Address Pointer

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^{2}C2 (SDA)	—	B	C
6'b10 0011	I^{2}C2 (SCL)	—	B	C
6'b10 0010	I^{2}C1 (SDA)	—	B	C
6'b10 0001	I^{2}C1 (SCL)	—	B	C
6’b10 0000	SPI1 ($\overline{\text{SS}}$)	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 ($\overline{\text{RTS}}$)	—	B	C
6'b01 0111	UART2 (TXDE)	—	B	C
6'b01 0110	UART2 (TX)	—	B	C
6'b01 0101	UART1 ($\overline{\text{RTS}}$)	—	B	C
6'b01 0100	UART1 (TXDE)	—	B	C
6'b01 0011	UART1 (TX)	—	B	C
6’b01 0010 - 6’b01 0001	Reserved			

21.0 PERIPHERAL MODULE DISABLE (PMD)

Sleep, Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked, and thus, consume some amount of power. There may be cases where the application needs what these modes do not provide: the ability to allocate limited power resources to the CPU while eliminating power consumption from the peripherals.

The PIC18(L)F2x/4xK42 family addresses this requirement by allowing peripheral modules to be selectively enabled or disabled, placing them into the lowest possible power mode.

All modules are ON by default following any Reset.

21.1 Disabling a Module

Disabling a module has the following effects:

- All clock and control inputs to the module are suspended; there are no logic transitions, and the module will not function.
- The module is held in Reset.
- Any SFR becomes "unimplemented"
  - Writing is disabled
  - Reading returns 00h
- I/O functionality is prioritized as per **Section 18.2, I/O Priorities**
- All associated Input Selection registers are also disabled

21.2 Enabling a Module

When the PMD register bit is cleared, the module is re-enabled and will be in its Reset state (Power-on Reset). SFR data will reflect the POR Reset values.

Depending on the module, it may take up to one full instruction cycle for the module to become active. There should be no interaction with the module (e.g., writing to registers) for at least one instruction after it has been re-enabled.

21.3 Effects of a Reset

Following any Reset, each control bit is set to '0', enabling all modules.

21.4 System Clock Disable

Setting SYSCMD (PMD0, Register 21-1) disables the system clock (FOSC) distribution network to the peripherals. Not all peripherals make use of SYSCLK, so not all peripherals are affected. Refer to the specific peripheral description to see if it will be affected by this bit.

FIGURE 27-9: HIGH AND LOW MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAM

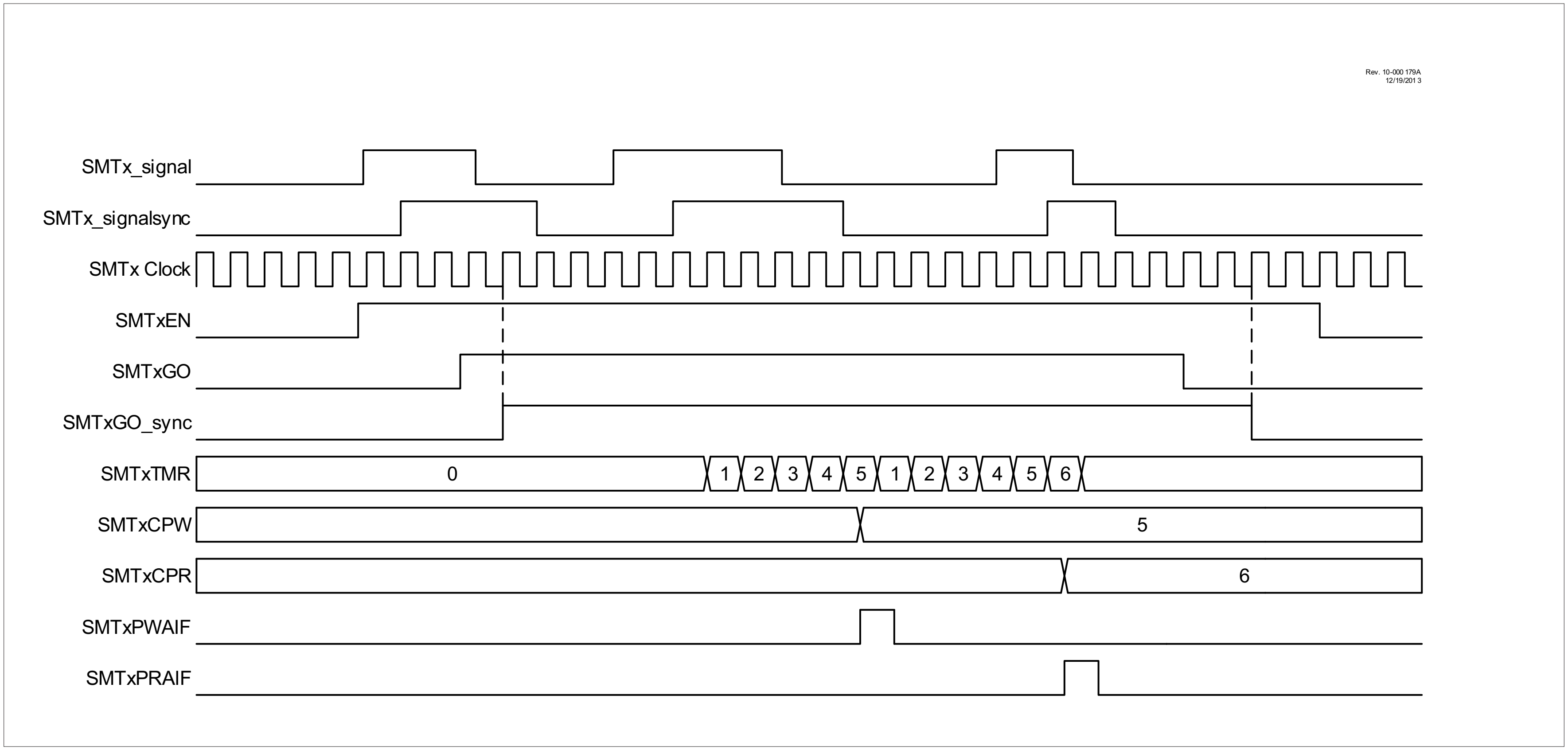
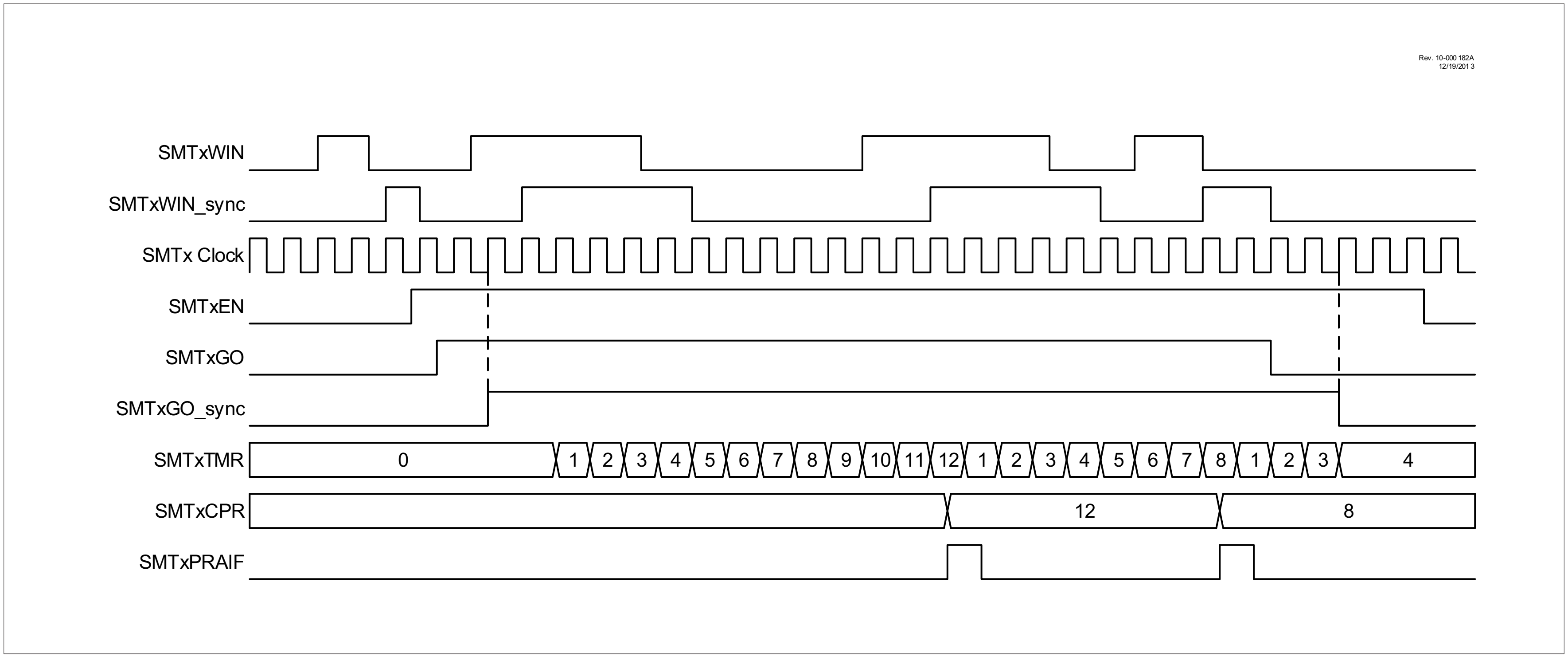


FIGURE 27-10: WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM



Preliminary

FIGURE 27-19: GATED COUNTER MODE REPEAT ACQUISITION TIMING DIAGRAM

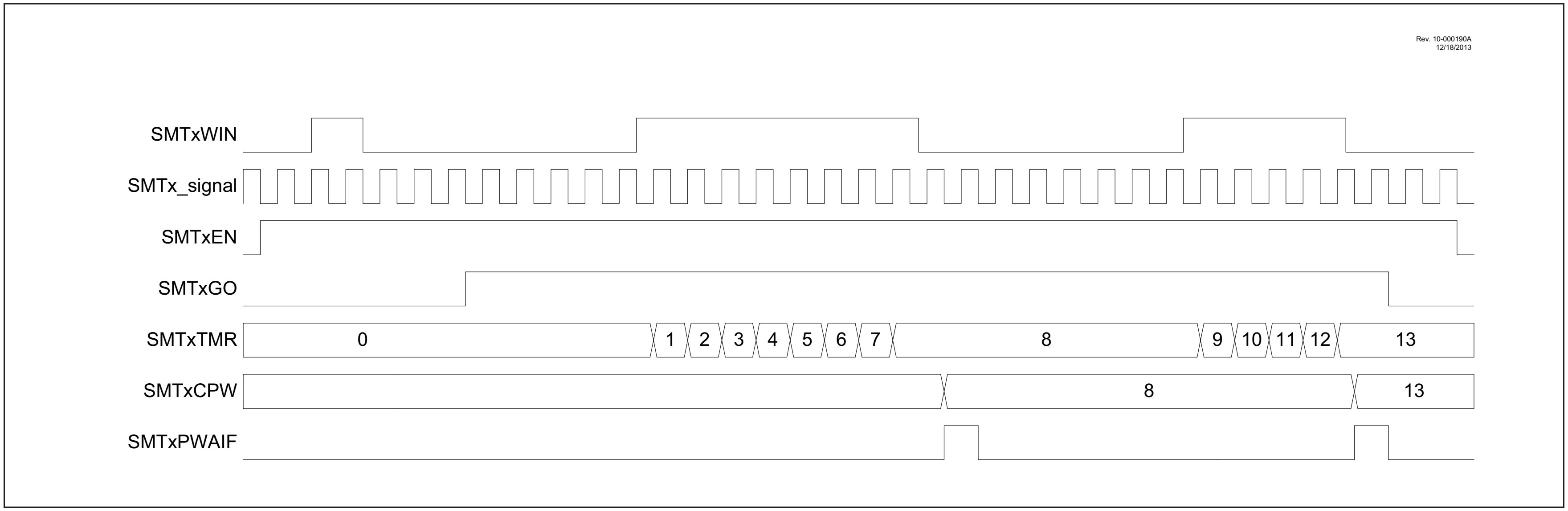
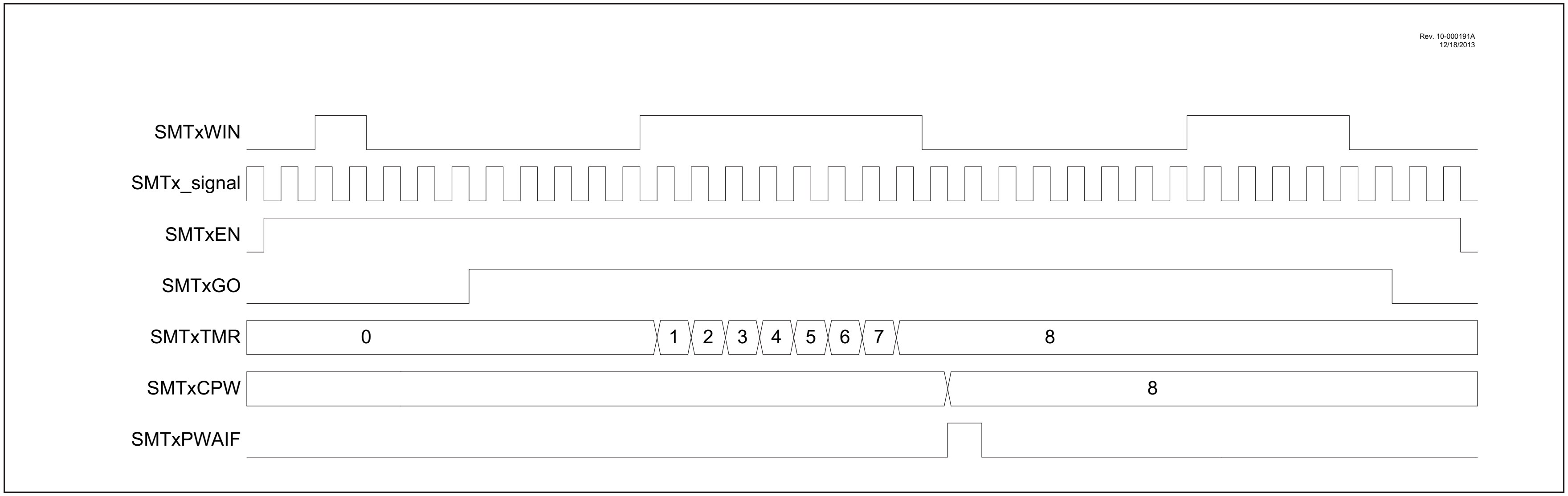


FIGURE 27-20: GATED COUNTER MODE SINGLE ACQUISITION TIMING DIAGRAM



29.0 CONFIGURABLE LOGIC CELL (CLC)

The Configurable Logic Cell (CLCx) module provides programmable logic that operates outside the speed limitations of software execution. The logic cell takes up to 32 input signals and, through the use of configurable gates, reduces the 32 inputs to four logic lines that drive one of eight selectable single-output logic functions.

Input sources are a combination of the following:

- I/O pins
- Internal clocks
- Peripherals
- Register bits

The output can be directed internally to peripherals and to an output pin.

There are four CLC modules available on this device - CLC1, CLC2, CLC3 and CLC4.

Note: The CLC1, CLC2, CLC3 and CLC4 are four separate module instances of the same CLC module design. Throughout this section, the lower case 'x' in register names is a generic reference to the CLC number (which should be substituted with 1, 2, 3, or 4 during code development). For example, the control register is generically described in this chapter as CLCxCON, but the actual device registers are CLC1CON, CLC2CON, CLC3CON and CLC4CON.

Refer to Figure 29-1 for a simplified diagram showing signal flow through the CLCx.

Possible configurations include:

- Combinatorial Logic
  - AND
  - NAND
  - AND-OR
  - AND-OR-INVERT
  - OR-XOR
  - OR-XNOR
- Latches
  - S-R
  - Clocked D with Set and Reset
  - Transparent D with Set and Reset

34.0 SERIAL PERIPHERAL INTERFACE (SPI) MODULE

34.1 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 (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 34-1 shows the block diagram of the SPI module.

REGISTER 38-6: ADCLK: ADC CLOCK 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
—	—	CS<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 **CS<5:0>**: ADC Conversion Clock Select bits
111111 = FOSC/128
111110 = FOSC/126
111101 = FOSC/124
•
•
•
000000 = FOSC/2

REGISTER 38-7: ADREF: ADC REFERENCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	NREF	—	—	PREF<1:0>	
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 7-5 **Unimplemented**: Read as '0'

bit 4 **NREF**: ADC Negative Voltage Reference Selection bit
1 = VREF- is connected to external VREF-
0 = VREF- is connected to VSS

bit 3-2 **Unimplemented**: Read as '0'

bit 1-0 **PREF**: ADC Positive Voltage Reference Selection bits
11 = VREF+ is connected to internal Fixed Voltage Reference (FVR) module
10 = VREF+ is connected to external VREF+
01 = Reserved
00 = VREF+ is connected to VDD

41.2 HLVD Setup

To set up the HLVD module:

1. Select the desired HLVD trip point by writing the value to the SEL<3:0> bits of the HLVDCON1 register.
2. Depending on the application to detect high-voltage peaks or low-voltage drops or both, set the INTH or INTL bit appropriately.
3. Enable the HLVD module by setting the EN bit.
4. Clear the HLVD interrupt flag (PIR2 register), which may have been set from a previous interrupt.
5. If interrupts are desired, enable the HLVD interrupt by setting the HLVDIE in the PIE2 register and GIE bits.

   An interrupt will not be generated until the RDY bit is set.

Note: Before changing any module settings (INTH, INTL, SEL<3:0>), first disable the module (EN = 0), make the changes and re-enable the module. This prevents the generation of false HLVD events.

41.3 Current Consumption

When the module is enabled, the HLVD comparator and voltage divider are enabled and consume static current. The total current consumption, when enabled, is specified in electrical specification Parameter **D206** (Table 46-3).

Depending on the application, the HLVD module does not need to operate constantly. To reduce current requirements, the HLVD circuitry may only need to be enabled for short periods where the voltage is checked. After such a check, the module could be disabled.

41.4 HLVD Start-up Time

The internal reference voltage of the HLVD module, specified in electrical specification (Table 46-17), may be used by other internal circuitry, such as the programmable Brown-out Reset. If the HLVD or other circuits using the voltage reference are disabled to lower the device’s current consumption, the reference voltage circuit will require time to become stable before a low or high-voltage condition can be reliably detected. This start-up time, TFVRST, is an interval that is independent of device clock speed. It is specified in electrical specification (Table 46-17).

The HLVD interrupt flag is not enabled until TFVRST has expired and a stable reference voltage is reached. For this reason, brief excursions beyond the set point may not be detected during this interval (see Figure 41-2 or Figure 41-3).

TABLE 43-1: PIC18(L)F24/25K42 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.

RLNCF Rotate Left f (No Carry)

Syntax: RLNCF f {,d {,a}}

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

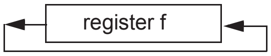
Operation: (f<n>) → dest<n + 1>,
(f<7>) → dest<0>

Status Affected: N, Z

Encoding:

0100	01da	ffff	ffff

Description: The contents of register ‘f’ are rotated one bit to the left. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’ (default).
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 43.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write to destination

Example: RLNCF REG, 1, 0

Before Instruction
REG = 1010 1011
After Instruction
REG = 0101 0111

RRCF Rotate Right f through Carry

Syntax: RRCF f {,d {,a}}

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

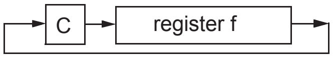
Operation: (f<n>) → dest<n – 1>,
(f<0>) → C,
(C) → dest<7>

Status Affected: C, N, Z

Encoding:

0011	00da	ffff	ffff

Description: The contents of register ‘f’ are rotated one bit to the right through the CARRY flag. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is placed back in register ‘f’ (default).
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 43.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write to destination

Example: RRCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 0111 0011
C = 0