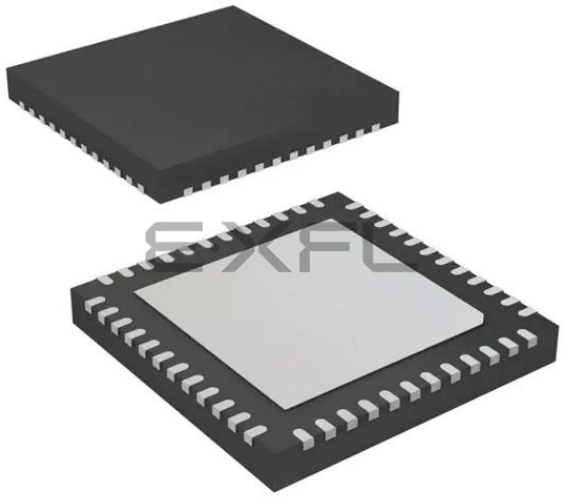


Welcome to E-XFL.COM

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, LVD, POR, PWM, WDT
Number of I/O	44
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 43x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-UFQFN Exposed Pad
Supplier Device Package	48-UQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf55k42t-i-mv

Pin Allocation Tables

TABLE 1: 28-PIN ALLOCATION TABLE (PIC18(L)F2XK42)

I/O	28-Pin SPDIP/SOIC/SSOP	28-Pin (U)QFN	ADC	Voltage Reference	DAC	Comparators	Zero Cross Detect	I^2C	SPI	UART	DSM	Timers/SMT	CCP and PWM	CWG	CLC	NCO	Clock Reference (CLKR)	Interrupt-on-Change	Basic
RA0	2	27	ANA0	—	—	C1IN0- C2IN0-	—	—	—	—	—	—	—	—	CLCIN0(1)	—	—	IOCA0	—
RA1	3	28	ANA1	—	—	C1IN1- C2IN1-	—	—	—	—	—	—	—	—	CLCIN1(1)	—	—	IOCA1	—
RA2	4	1	ANA2	VREF-	DAC1OUT1	C1IN0+ C2IN0+	—	—	—	—	—	—	—	—	—	—	—	IOCA2	—
RA3	5	2	ANA3	VREF+	—	C1IN1+	—	—	—	—	MDCARL(1)	—	—	—	—	—	—	IOCA3	—
RA4	6	3	ANA4	—	—	—	—	—	—	—	MDCARH(1)	T0CKI(1)	—	—	—	—	—	IOCA4	—
RA5	7	4	ANA5	—	—	—	—	—	SS1(1)	—	MDSRC(1)	—	—	—	—	—	—	IOCA5	—
RA6	10	7	ANA6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCA6	OSC2 CLKOUT
RA7	9	6	ANA7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCA7	OSC1 CLKIN
RB0	21	18	ANB0	—	—	C2IN1+	ZCD	—	—	—	—	—	CCP4(1)	CWG1IN(1)	—	—	—	INT0(1) IOCB0	—
RB1	22	19	ANB1	—	—	C1IN3- C2IN3-	—	SCL2(3,4)	—	—	—	—	—	CWG2IN(1)	—	—	—	INT1(1) IOCB1	—
RB2	23	20	ANB2	—	—	—	—	SDA2(3,4)	—	—	—	—	—	CWG3IN(1)	—	—	—	INT2(1) IOCB2	—
RB3	24	21	ANB3	—	—	C1IN2- C2IN2-	—	—	—	—	—	—	—	—	—	—	—	IOCB3	—
RB4	25	22	ANB4 ADCACT(1)	—	—	—	—	—	—	—	—	T5G(1)	—	—	—	—	—	IOCB4	—
RB5	26	23	ANB5	—	—	—	—	—	—	—	—	T1G(1)	CCP3(1)	—	—	—	—	IOCB5	—
RB6	27	24	ANB6	—	—	—	—	—	—	CTS2(1)	—	—	—	—	CLCIN2(1)	—	—	IOCB6	ICSPCLK
RB7	28	25	ANB7	—	DAC1OUT2	—	—	—	—	RX2(1)	—	T6IN(1)	—	—	CLCIN3(1)	—	—	IOCB7	ICSPDAT

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^2C and SMB™ 3.0/2.0 logic levels; The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 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^2C specific or SMBus input buffer thresholds.

TABLE 4-7: SPECIAL FUNCTION REGISTER MAP FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES BANK 60

3CFFh	—	3CDFh	—	3CBFh	—	3C9Fh	—	3C7Fh	—	3C5Fh	CLC4GLS3	3C3Fh	—	3C1Fh	—
3CFEh	MD1CARH	3CDEh	—	3CBEh	—	3C9Eh	—	3C7Eh	CLCDATA0	3C5Eh	CLC4GLS2	3C3Eh	—	3C1Eh	—
3CFDh	MD1CARL	3CDDh	—	3CBDh	—	3C9Dh	—	3C7Dh	CLC1GLS3	3C5Dh	CLC4GLS1	3C3Dh	—	3C1Dh	—
3CFCh	MD1SRC	3CDCh	—	3CBCh	—	3C9Ch	—	3C7Ch	CLC1GLS2	3C5Ch	CLC4GLS0	3C3Ch	—	3C1Ch	—
3CFBh	MD1CON1	3CDBh	—	3CBBh	—	3C9Bh	—	3C7Bh	CLC1GLS1	3C5Bh	CLC4SEL3	3C3Bh	—	3C1Bh	—
3CFAh	MD1CON0	3CDAh	—	3CBAh	—	3C9Ah	—	3C7Ah	CLC1GLS0	3C5Ah	CLC4SEL2	3C3Ah	—	3C1Ah	—
3CF9h	—	3CD9h	—	3CB9h	—	3C99h	—	3C79h	CLC1SEL3	3C59h	CLC4SEL1	3C39h	—	3C19h	—
3CF8h	—	3CD8h	—	3CB8h	—	3C98h	—	3C78h	CLC1SEL2	3C58h	CLC4SEL0	3C38h	—	3C18h	—
3CF7h	—	3CD7h	—	3CB7h	—	3C97h	—	3C77h	CLC1SEL1	3C57h	CLC4POL	3C37h	—	3C17h	—
3CF6h	—	3CD6h	—	3CB6h	—	3C96h	—	3C76h	CLC1SEL0	3C56h	CLC4CON	3C36h	—	3C16h	—
3CF5h	—	3CD5h	—	3CB5h	—	3C95h	—	3C75h	CLC1POL	3C55h	—	3C35h	—	3C15h	—
3CF4h	—	3CD4h	—	3CB4h	—	3C94h	—	3C74h	CLC1CON	3C54h	—	3C34h	—	3C14h	—
3CF3h	—	3CD3h	—	3CB3h	—	3C93h	—	3C73h	CLC2GLS3	3C53h	—	3C33h	—	3C13h	—
3CF2h	—	3CD2h	—	3CB2h	—	3C92h	—	3C72h	CLC2GLS2	3C52h	—	3C32h	—	3C12h	—
3CF1h	—	3CD1h	—	3CB1h	—	3C91h	—	3C71h	CLC2GLS1	3C51h	—	3C31h	—	3C11h	—
3CF0h	—	3CD0h	—	3CB0h	—	3C90h	—	3C70h	CLC2GLS0	3C50h	—	3C30h	—	3C10h	—
3CEFh	—	3CCFh	—	3CAFh	—	3C8Fh	—	3C6Fh	CLC2SEL3	3C4Fh	—	3C2Fh	—	3C0Fh	—
3CEEh	—	3CCEh	—	3CAEh	—	3C8Eh	—	3C6Eh	CLC2SEL2	3C4Eh	—	3C2Eh	—	3C0Eh	—
3CEDh	—	3CCDh	—	3CADh	—	3C8Dh	—	3C6Dh	CLC2SEL1	3C4Dh	—	3C2Dh	—	3C0Dh	—
3CECh	—	3CCCh	—	3CACh	—	3C8Ch	—	3C6Ch	CLC2SEL0	3C4Ch	—	3C2Ch	—	3C0Ch	—
3CEBh	—	3CCBh	—	3CABh	—	3C8Bh	—	3C6Bh	CLC2POL	3C4Bh	—	3C2Bh	—	3C0Bh	—
3CEAh	—	3CCAh	—	3CAAh	—	3C8Ah	—	3C6Ah	CLC2CON	3C4Ah	—	3C2Ah	—	3C0Ah	—
3CE9h	—	3CC9h	—	3CA9h	—	3C89h	—	3C69h	CLC3GLS3	3C49h	—	3C29h	—	3C09h	—
3CE8h	—	3CC8h	—	3CA8h	—	3C88h	—	3C68h	CLC3GLS2	3C48h	—	3C28h	—	3C08h	—
3CE7h	—	3CC7h	—	3CA7h	—	3C87h	—	3C67h	CLC3GLS1	3C47h	—	3C27h	—	3C07h	—
3CE6h	CLKRCLK	3CC6h	—	3CA6h	—	3C86h	—	3C66h	CLC3GLS0	3C46h	—	3C26h	—	3C06h	—
3CE5h	CLKRCON	3CC5h	—	3CA5h	—	3C85h	—	3C65h	CLC3SEL3	3C45h	—	3C25h	—	3C05h	—
3CE4h	—	3CC4h	—	3CA4h	—	3C84h	—	3C64h	CLC3SEL2	3C44h	—	3C24h	—	3C04h	—
3CE3h	—	3CC3h	—	3CA3h	—	3C83h	—	3C63h	CLC3SEL1	3C43h	—	3C23h	—	3C03h	—
3CE2h	—	3CC2h	—	3CA2h	—	3C82h	—	3C62h	CLC3SEL0	3C42h	—	3C22h	—	3C02h	—
3CE1h	—	3CC1h	—	3CA1h	—	3C81h	—	3C61h	CLC3POL	3C41h	—	3C21h	—	3C01h	—
3CE0h	—	3CC0h	—	3CA0h	—	3C80h	—	3C60h	CLC3CON	3C40h	—	3C20h	—	3C00h	—

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.

REGISTER 5-3: CONFIGURATION WORD 2L (30 0002h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
BOREN<1:0>		$\overline{\text{LPBOREN}}$	IVT1WAY	MVECEN	PWRTS<1:0>		MCLRE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '1'	
-n = Value for blank device	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **BOREN<1:0>:** Brown-out Reset Enable bits

When enabled, Brown-out Reset Voltage (V_{BOR}) is set by the BORV bit.
11 = Brown-out Reset is enabled, SBOREN bit is ignored
10 = Brown-out Reset is enabled while running, disabled in Sleep; SBOREN is ignored
01 = Brown-out Reset is enabled according to SBOREN
00 = Brown-out Reset is disabled

bit 5 **$\overline{\text{LPBOREN}}$:** Low-Power BOR Enable bit

1 = Low-Power BOR is disabled
0 = Low-Power BOR is enabled

bit 4 **IVT1WAY:** IVTLOCK bit One-Way Set Enable bit

1 = IVTLOCKED bit can be cleared and set only once; IVT registers remain locked after one clear/set cycle
0 = IVTLOCK ED bit can be set and cleared multiple times (subject to the unlock sequence)

bit 3 **MVECEN:** Multi-vector Enable bit

1 = Multi-vector enabled; Vector table used for interrupts
0 = Legacy interrupt behavior

bit 2-1 **PWRTS<1:0>:** Power-up Timer Selection bits

11 = PWRT is disabled
10 = PWRT set at 64 ms (2048 LFINTOSC Cycles)
01 = PWRT set at 16 ms (512 LFINTOSC Cycles)
00 = PWRT set at 1 ms (32 LFINTOSC Cycles)

bit 0 **MCLRE:** Master Clear ($\overline{\text{MCLR}}$) Enable bit

If LVP = 1:
RE3 pin function is $\overline{\text{MCLR}}$
If LVP = 0:
1 = $\overline{\text{MCLR}}$ pin is $\overline{\text{MCLR}}$
0 = $\overline{\text{MCLR}}$ pin function is a port defined function

REGISTER 5-4: CONFIGURATION WORD 2H (30 0003h)

R/W-1	U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{XINST}}$	—	$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	$\overline{\text{ZCD}}$	BORV<1:0>(1)	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '1'	
-n = Value for blank device	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{XINST}}$:** Extended Instruction Set Enable bit
1 = Extended instruction set and Indexed Addressing mode are disabled (Legacy mode)
0 = Extended instruction set and Indexed Addressing mode are enabled

bit 6 **Unimplemented:** Read as '1'

bit 5 **$\overline{\text{DEBUG}}$:** Debugger Enable bit
1 = Background debugger is disabled
0 = Background debugger is enabled

bit 4 **STVREN:** Stack Overflow/Underflow Reset Enable bit
1 = Stack Overflow or Underflow will cause a Reset
0 = Stack Overflow or Underflow will not cause a Reset

bit 3 **PPS1WAY:** PPSLOCKED One-Way Set Enable bit
1 = PPSLOCKED bit can be cleared and set only once; PPS registers remain locked after one clear/set cycle
0 = PPSLOCKED bit can be set and cleared multiple times (subject to the unlock sequence)

bit 2 **$\overline{\text{ZCD}}$:** Zero-Cross Detect Enable bit
1 = ZCD is disabled; ZCD can be enabled by setting the bit SEN of the ZCDCON register
0 = ZCD is always enabled

bit 1-0 **BORV<1:0>:** Brown-out Reset Voltage Selection bits(1)
PIC18FXXK42 Devices:
11 = Brown-out Reset Voltage (V_{BOR}) is set to 2.45V
10 = Brown-out Reset Voltage (V_{BOR}) is set to 2.45V
01 = Brown-out Reset Voltage (V_{BOR}) is set to 2.7V
00 = Brown-out Reset Voltage (V_{BOR}) is set to 2.85V
PIC18LFXXK42 Device:
11 = Brown-out Reset Voltage (V_{BOR}) is set to 1.90V
10 = Brown-out Reset Voltage (V_{BOR}) is set to 2.45V
01 = Brown-out Reset Voltage (V_{BOR}) is set to 2.7V
00 = Brown-out Reset Voltage (V_{BOR}) is set to 2.85V

Note 1: The higher voltage setting is recommended for operation at or above 16 MHz.

REGISTER 5-5: CONFIGURATION WORD 3L (30 0004h)

U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	WDTE<1:0>		WDTCPS<4:0>				
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '1'	
-n = Value for blank device	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '1'

bit 6-5 **WDTE<1:0>:** WDT Operating Mode bits
00 = WDT is disabled, SWDTEN is ignored
01 = WDT is enabled/disabled by the SWDTEN bit in WDTCON0
10 = WDT is enabled while Sleep = 0, suspended when Sleep = 1; SWDTEN is ignored
11 = WDT is enabled regardless of Sleep; SWDTEN is ignored

bit 4-0 **WDTCPS<4:0>:** WDT Period Select bits

WDTCPS<4:0>	WDTPS at POR				Software Control of WDTPS?
	Value	Divider Ratio		Typical Time-out (FIN = 31 kHz)	
00000	00000	1:32	2^5	1 ms	No
00001	00001	1:64	2^6	2 ms	
00010	00010	1:128	2^7	4 ms	
00011	00011	1:256	2^8	8 ms	
00100	00100	1:512	2^9	16 ms	
00101	00101	1:1024	2^{10}	32 ms	
00110	00110	1:2048	2^{11}	64 ms	
00111	00111	1:4096	2^{12}	128 ms	
01000	01000	1:8192	2^{13}	256 ms	
01001	01001	1:16384	2^{14}	512 ms	
01010	01010	1:32768	2^{15}	1s	
01011	01011	1:65536	2^{16}	2s	
01100	01100	1:131072	2^{17}	4s	
01101	01101	1:262144	2^{18}	8s	
01110	01110	1:524299	2^{19}	16s	
01111	01111	1:1048576	2^{20}	32s	
10000	10000	1:2097152	2^{21}	64s	
10001	10001	1:4194304	2^{22}	128s	
10010	10010	1:8388608	2^{23}	256s	
10011 ... 11110	10011 ... 11110	1:32	2^5	1 ms	No
11111	01011	1:65536	2^{16}	2s	Yes

REGISTER 5-9: CONFIGURATION WORD 5L (30 0008h)

U-1	U-1	U-1	U-1	U-1	U-1	U-1	R/W-1
—	—	—	—	—	—	—	$\overline{CP}$
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '1'	
-n = Value for blank device	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-1 **Unimplemented:** Read as '1'

bit 0 **$\overline{CP}$:** User Program Flash Memory and Data EEPROM Code Protection bit
1 = User Program Flash Memory and Data EEPROM code protection is disabled
0 = User Program Flash Memory and Data EEPROM code protection is enabled

REGISTER 5-10: CONFIGURATION WORD 5H (30 0009h)

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '1'	
-n = Value for blank device	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **Unimplemented:** Read as '1'

TABLE 5-2: SUMMARY OF CONFIGURATION WORDS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
30 0000h	CONFIG1L	—	RSTOSC<2:0>			—	FEXTOSC<2:0>			1111 1111
30 0001h	CONFIG1H	—	—	FCMEN	—	CSWEN	—	PR1WAY	$\overline{CLKOUTEN}$	1111 1111
30 0002h	CONFIG2L	BOREN<1:0>		$\overline{LPBOREN}$	IVT1WAY	MVECEN	PWRTS<1:0>		MCLRE	1111 1111
30 0003h	CONFIG2H	$\overline{XINST}$	—	$\overline{DEBUG}$	STVREN	PPS1WAY	$\overline{ZCD}$	BORV<1:0>		1111 1111
30 0004h	CONFIG3L	—	WDTE<1:0>		WDTCPS<4:0>					1111 1111
30 0005h	CONFIG3H	—	—	WDTCCS<2:0>			WDTCWS<2:0>			1111 1111
30 0006h	CONFIG4L	$\overline{WRTAPP}$	—	—	$\overline{SAFEN}$	$\overline{BBEN}$	BBSIZE<2:0>			1111 1111
30 0007h	CONFIG4H	—	—	LVP	—	$\overline{WRTSAF}$	$\overline{WRTD}$	$\overline{WRTC}$	$\overline{WRTB}$	1111 1111
30 0008h	CONFIG5L	—	—	—	—	—	—	—	$\overline{CP}$	1111 1111
30 0009h	CONFIG5H	—	—	—	—	—	—	—	—	1111 1111

TABLE 15-6: EXAMPLE DMA USE CASE TABLE

Source Module	Source Register(s)	Destination Module	Destination Register(s)	DCHxSIRQ	Comment
Signal Measurement Timer (SMT)	SMTxCPW[U:H:L]	GPR	GPR[x,y,z]	SMTxPWAIF	Store Captured Pulse-width values
	SMTxCPR[U:H:L]			SMTxPRAIF	Store Captured Period values
GPR/SFR/Program Flash/Data EEPROM	MEMORY[x,y]	TMR0	TMR0[H:L]	TMR0IF	Use as a Timer0 reload for custom 16-bit value
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x]	TMR0	PR0	ANY	Update TMR0 frequency based on a specific trigger
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x,y]	TMR1	TMR1[H:L]	TMR1IF	Use as a Timer1 reload for custom 16-bit value
TMR1	TMR1[H:L]	GPR	GPR[x,y]	TMR1GIF	Use TMR1 Gate interrupt flag to read data out of TMR1 register
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x]	TMR2	PR2	TMR2IF	
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x,y,z]	TMR2 CCP or PWM	PR2 CCPR[H:L] or PWMDC[H:L]	ANY	Frequency generator with 50% duty cycle look up table
CCP	CCPR[H:L]	GPR	GPR[x,y]	CCPxIF	Move data from CCP 16b Capture
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x,y]	CCP	CCPR[H:L]	ANY	Load Compare value or PWM values into the CCP
GPR/SFR/ Program Flash/Data EEPROM	MEMORY [x,y,z,u,v,w]	CCPx CCPy CCPz	CCPxR[H:L] CCPyR[H:L] CCPzR[H:L]	ANY	Update multiple PWM values at the same time e.g. 3-phase motor control
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x,y,z]	NCO	NCOxINC[U:H:L]	ANY	Frequency Generator look-up table
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x]	DAC	DACxCON0	ANY	Update DAC values
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x]	OSCTUNE	OSCTUNE	ANY	Automated Frequency dithering

REGISTER 15-1: 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 17-2: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER

U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	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
See Table 17-2 for the list of available ports.

TABLE 21-3: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1/3/5 AS A TIMER/COUNTER

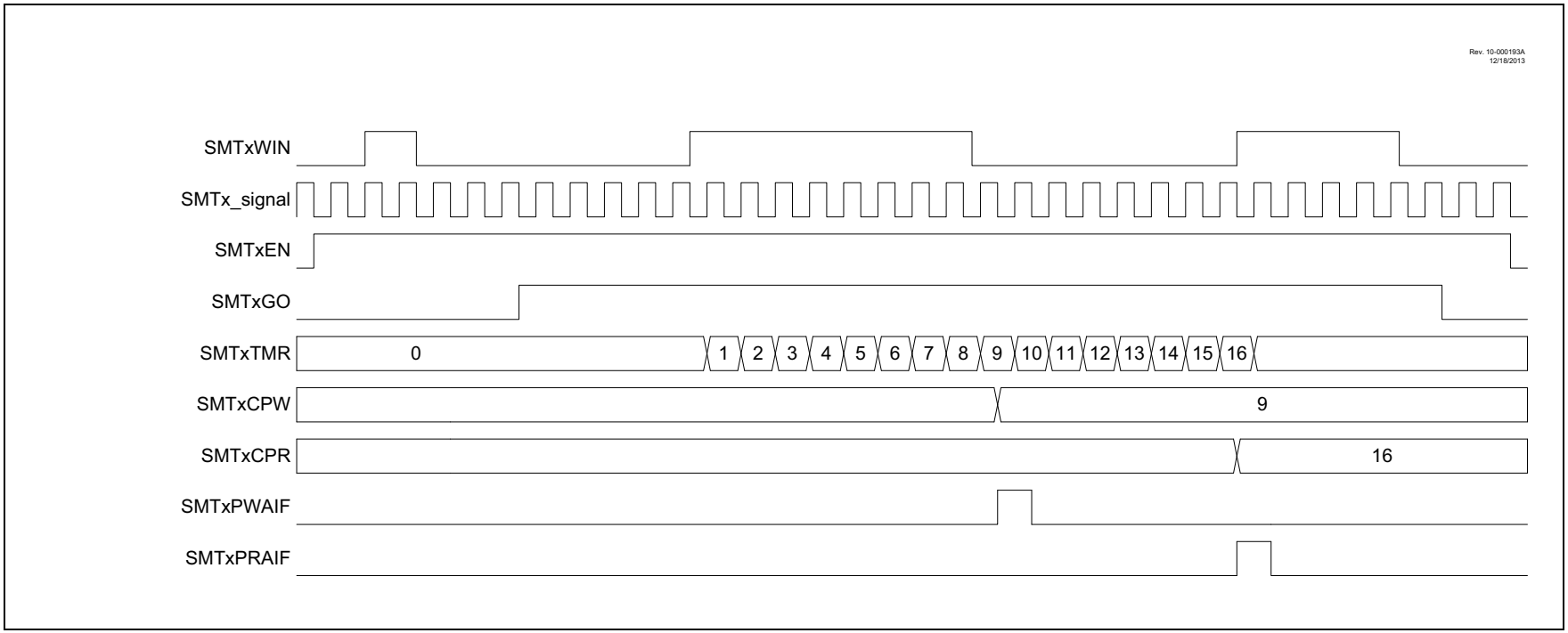
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
TxCON	—	—	CKPS<1:0>		—	$\overline{\text{SYNC}}$	RD16	ON	313
TxGCON	GE	GPOL	GTM	GSPM	GO/$\overline{\text{DONE}}$	GVAL	—	—	314
TxCLK	—	—	—	CS<4:0>					315
TxGATE	—	—	—	GSS<4:0>					316
TMRxL	Least Significant Byte of the 16-bit TMR3 Register								317
TMRxH	Holding Register for the Most Significant Byte of the 16-bit TMR3 Register								317

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by TIMER1/3/5.

25.6.11 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 25-21 and Figure 25-22.

FIGURE 25-22: WINDOWED COUNTER MODE SINGLE ACQUISITION TIMING DIAGRAM



REGISTER 25-2: SMT1CON1: SMT CONTROL REGISTER 1

R/W/HC-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
GO	REPEAT	—	—	MODE<3:0>			
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	-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 **GO:** GO Data Acquisition bit
1 = Incrementing, acquiring data is enabled
0 = Incrementing, acquiring data is disabled

bit 6 **REPEAT:** SMT Repeat Acquisition Enable bit
1 = Repeat Data Acquisition mode is enabled
0 = Single Acquisition mode is enabled

bit 5-4 **Unimplemented**: Read as '0'

bit 3-0 **MODE<3:0>** SMT Operation Mode Select bits
1111 = Reserved
•
•
•
1011 = Reserved
1010 = Windowed counter
1001 = Gated counter
1000 = Counter
0111 = Capture
0110 = Time of flight
0101 = Gated windowed measure
0100 = Windowed measure
0011 = High and low time measurement
0010 = Period and Duty-Cycle Acquisition
0001 = Gated Timer
0000 = Timer

31.21 Register Definitions: UART Control

Long bit name prefixes for the UART peripherals are shown below. Refer to **Section 1.3 "Register and Bit naming conventions"**for more information.

Peripheral	Bit Name Prefix
UART 1	U1
UART 2	U2

REGISTER 31-1: UxCON0: UART CONTROL REGISTER 0

R/W-0/0	R/W/HS/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
BRGS	ABDEN	TXEN	RXEN	MODE<3:0>			
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	HC = Hardware clear

bit 7 **BRGS:** Baud rate Generator Speed Select bit
- 1 = Baud rate generator is high speed with 4 baud clocks per bit
- 0 = Baud rate generator is normal speed with 16 baud clocks per bit

bit 6 **ABDEN:** Auto-baud Detect Enable bit(3)
- 1 = Auto-baud is enabled. Receiver is waiting for Sync character (0x55)
- 0 = Auto-baud is not enabled or auto-baud is complete

bit 5 **TXEN:** Transmit Enable Control bit(2)
- 1 = Transmit is enabled. TX output pin drive is forced on when transmission is active, and controlled by PORT TRIS control when transmission is idle.
- 0 = Transmit is disabled. TX output pin drive is controlled by PORT TRIS control

bit 4 **RXEN:** Receive Enable Control bit(2)
- 1 = Receiver is enabled
- 0 = Receiver is disabled

bit 3-0 **MODE<3:0>:** UART Mode Select bits(1)
- 1111 = Reserved
- 1110 = Reserved
- 1101 = Reserved
- 1100 = LIN Master/Slave mode(4)
- 1011 = LIN Slave-Only mode(4)
- 1010 = DMX mode(4)
- 1001 = DALI Control Gear mode(4)
- 1000 = DALI Control Device mode(4)
- 0111 = Reserved
- 0110 = Reserved
- 0101 = Reserved
- 0100 = Asynchronous 9-bit UART Address mode. 9th bit: 1 = address, 0 = data
- 0011 = Asynchronous 8-bit UART mode with 9th bit even parity
- 0010 = Asynchronous 8-bit UART mode with 9th bit odd parity
- 0001 = Asynchronous 7-bit UART mode
- 0000 = Asynchronous 8-bit UART mode

Note 1: Changing the UART MODE while ON = 1 may cause unexpected results.
2: Clearing TXEN or RXEN will not clear the corresponding buffers. Use TXBE or RXBE to clear the buffers.
3: When MODE = 100x, then ABDEN bit is ignored.
4: UART1 only.

REGISTER 31-4: UxERRIR: UART ERROR INTERRUPT FLAG REGISTER

R/S/C-1/1	R/S/C-0/0	R/W/S-0/0	R/W/S-0/0	R/S/C-0/0	R/W/S-0/0	R/W/S-0/0	R/W/S-0/0
TXMTIF	PERIF	ABDOVF	CERIF	FERIF	RXBKIF	RXFOIF	TXCIF
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	S = Hardware set	C = Hardware clear

bit 7 **TXMTIF:** Transmit Shift Register Empty Interrupt Flag bit
1 = Transmit shift register is empty (Set at end of Stop bits)
0 = Transmit shift register is actively shifting data

bit 6 **PERIF**: Parity Error Interrupt Flag bit
LIN and Parity modes:
1 = Unread byte at top of input FIFO has parity error
0 = Unread byte at top of input FIFO does not have parity error
DALI Device mode:
1 = Unread byte at top of input FIFO received as Forward Frame
0 = Unread byte at top of input FIFO received as Back Frame
Address mode:
1 = Unread byte at top of input FIFO received as address
0 = Unread byte at top of input FIFO received as data
Other modes:
Not used

bit 5 **ABDOVF:** Auto-baud Detect Overflow Interrupt Flag bit
DALI mode:
1 = Start bit measurement overflowed counter
0 = No overflow during Start bit measurement
Other modes:
1 = Baud rate generator overflowed during the auto detection sequence
0 = Baud rate generator has not overflowed

bit 4 **CERIF**: Checksum Error Interrupt Flag bit (LIN mode only)
1 = Checksum error
0 = No checksum error

bit 3 **FERIF:** Framing Error Interrupt Flag bit
1 = Unread byte at top of input FIFO has framing error
0 = Unread byte at top of input FIFO does not have framing error

bit 2 **RXBKIF:** Break Reception Interrupt Flag bit
1 = Break detected
0 = No Break detected

bit 1 **RXFOIF:** Receive FIFO Overflow Interrupt Flag bit
1 = Receive FIFO has overflowed
0 = Receive FIFO has not overflowed

bit 0 **TXCIF:** Transmit Collision Interrupt Flag bit
1 = Transmitted word is not equal to the word received during transmission
0 = Transmitted word equals the word received during transmission

The SPI transmit output (SDO_out) is available to the remappable PPS SDO pin and internally to the following peripherals:

- Configurable Logic Cell (CLC)
- Data Signal Modulator (DSM)

The SPI bus typically operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

Transmissions typically involve shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new bit is shifted into the device. Unlike older Microchip devices, the SPI on the PIC18(L)F2X/4X/5XK42 contains two separate registers for incoming and outgoing data. Both registers also have 2-byte FIFO buffers and allow for DMA bus connections.

Figure 32-2 shows a typical connection between two PIC18F2X/4X/5XK42 devices configured as master and slave devices.

Data is shifted out of the transmit FIFO on the programmed clock edge and into the receive shift register on the opposite edge of the clock.

The master device transmits information on its SDO output pin which is connected to, and received by, the slave's SDI input pin. The slave device transmits information on its SDO output pin, which is connected to, and received by, the master's SDI input pin.

The master device sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its output register (on its SDO pin) and the slave device is reading this bit and saving as the LSb of its input register, that the slave device is also sending out the MSb from its shift register (on its SDO pin) and the master device is reading this bit and saving it as the LSb of its input register.

After eight bits have been shifted out, the master and slave have exchanged register values and stored the incoming data into the receiver FIFOs.

If there is more data to exchange, the registers are loaded with new data and the process repeats itself.

Whether the data is meaningful or not (dummy data) depends on the application software. This leads to three scenarios for data transmission:

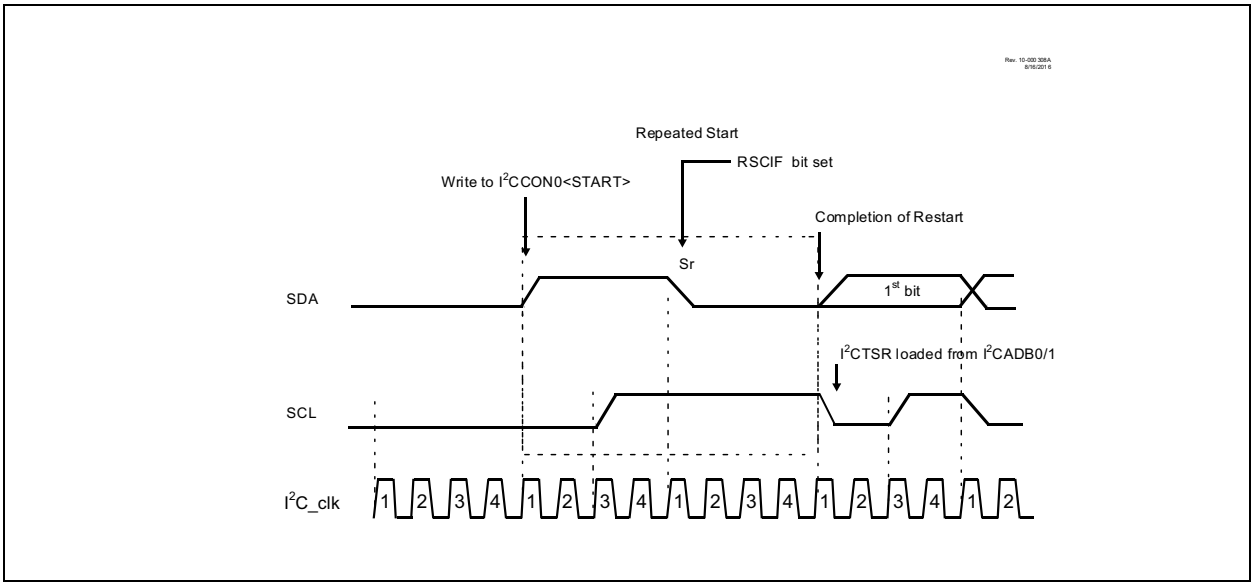
- Master sends useful data and slave sends dummy data
- Master sends useful data and slave sends useful data
- Master sends dummy data and slave sends useful data

In this particular SPI module, dummy data may be sent without software involvement, by clearing either the RXR bit (for receiving dummy data) or the TXR bit (for sending dummy data) (see Table 32-1 as well as **Section 32.5 "Master mode"** and **Section 32.6 "Slave Mode"** for further TXR/RXR setting details). This SPI module can send transmissions of any number of bits, and can send information in segments of varying size (from 1-8 bits in width). As such, transmissions may involve any number of clock cycles, depending on the amount of data to be transmitted.

When there is no more data to be transmitted, the master stops sending the clock signal and deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line disregards the clock and transmission signals and does not transmit out any data of its own.

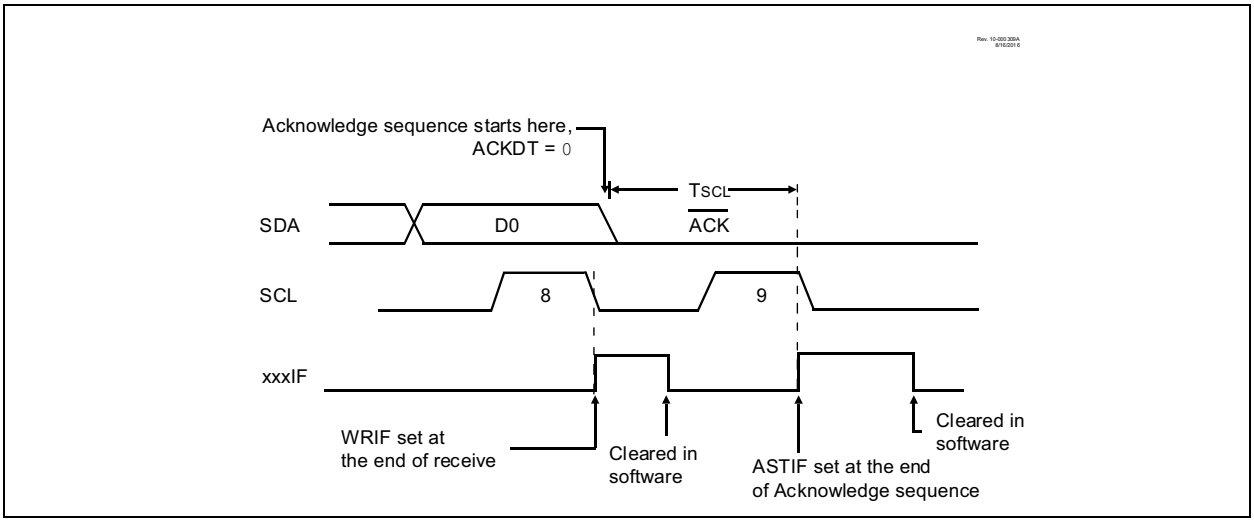
FIGURE 33-16: REPEATED START CONDITION TIMING



33.5.7 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled automatically following an address/data byte transmission. The SCL pin is pulled low and the contents of the Acknowledge Data bits (ACKDT/ACKCNT) are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The master then waits one clock period (TSCL) and the SCL pin is released high. When the SCL pin is sampled high (clock arbitration), the master counts another TSCL. The SCL pin is then pulled low. Figure 33-17 shows the timings for Acknowledge sequence.

FIGURE 33-17: ACKNOWLEDGE SEQUENCE TIMING



33.5.8 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of receive/transmit when I2CxCNT = 0. After the last byte of a receive/transmit sequence, the SCL line is held low. The master asserts the SDA line low. The SCL pin is then released high TSCL/2 later and is detected high. The SDA pin is then released. When the SDA pin transitions high while SCL is high, the PCIF bit of the I2CxIF register is set. Figure 33-18 shows the timings for a Stop condition.

REGISTER 33-2: I2CxCON1: I²C CONTROL REGISTER 1

R/W-0	R/W-0	R-0	R-0	U-0	R/W/HS-0	R/W/HS-0	R/W-0
ACKCNT(2)	ACKDT(1,2)	ACKSTAT	ACKT	—	RXO	TXU	CSD
bit 7							bit 0

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

bit 7 **ACKCNT:** Acknowledge End of Count bit(2)
Acknowledge value transmitted after received data, when I2CCNT = 0
1 = Not Acknowledge (copied to SDA output)
0 = Acknowledge (copied to SDA output)

bit 6 **ACKDT:** Acknowledge Data bit(1,2)
Acknowledge value transmitted after matching address
Acknowledge value transmitted after received data, when I2CCNT! = 0
1 = Not Acknowledge (copied to SDA output)
0 = Acknowledge (copied to SDA output)

bit 5 **ACKSTAT:** Acknowledge Status bit (Transmission only)
1 = Acknowledge was not received for most recent transmission
0 = Acknowledge was received for most recent transmission

bit 4 **ACKT:** Acknowledge Time Status bit
1 = Indicates the I²C bus is in an Acknowledge sequence, set on 8th falling edge of SCL clock
0 = Not in Acknowledge sequence, cleared on 9th rising edge of SCL

bit 3 **Unimplemented:** Read as 1'b0

bit 2 **RXO:** Receive Overflow Status bit (MODE<2:0> = 0xx & 11x)
This bit can only be set when CSD= 1
1 = Set when SMA = 1, and a master clocks in data when RXBF = 1
0 = No slave overflow condition

bit 1 **TXU:** Transmit Underflow Status bit (MODE<2:0> = 0xx & 11x)
This bit can only be set when CSTRDIS = 1
1 = Set when SMA = 1, and a master clocks out data when TXBE = 1
0 = No slave underflow condition

bit 0 **CSD:** Clock Stretching Disable bit (MODE<2:0> = 0xx & 11x)
1 = When SMA = 1, the CSTR bit will never be set
0 = Slave clock stretching proceeds normally

Note 1: Software writes to ACKDT bit must be followed by a minimum SDA data-setup time before clearing CSTR.
2: NACK may still be generated by I²C hardware when bus errors are indicated in the I2CxSTAT1 or I2CxERR registers.

REGISTER 36-32: ADLTHL: ADC LOWER THRESHOLD LOW BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LTH<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 **LTH<7:0>**: ADC Lower Threshold LSB. LTH and UTH are compared with ERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

REGISTER 36-33: ADUTHH: ADC UPPER THRESHOLD HIGH BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
UTH<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 **UTH<15:8>**: ADC Upper Threshold MSB. LTH and UTH are compared with ERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

REGISTER 36-34: ADUTHL: ADC UPPER THRESHOLD LOW BYTE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
UTH<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 **UTH<7:0>**: ADC Upper Threshold LSB. LTH and UTH are compared with ERR to set the ADUTHR and ADLTHR bits of ADSTAT. Depending on the setting of ADTMD, an interrupt may be triggered by the results of this comparison.

TABLE 41-2: INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word MSb	LSb	Status Affected	Notes
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 1111 ffff	ffff 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 1111 ffff 1111 gggg	0110 ffff ffgg 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.