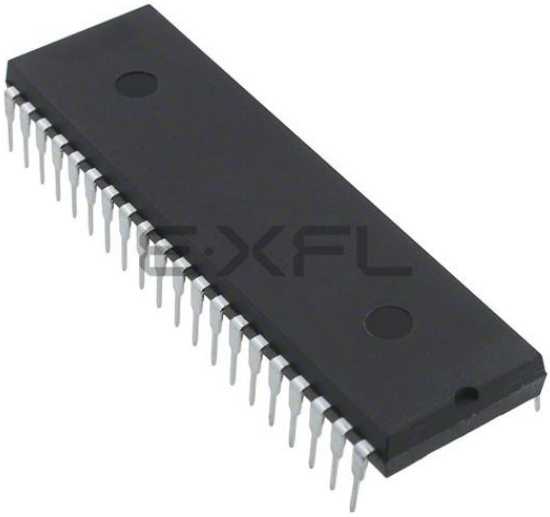


Welcome to E-XFL.COM

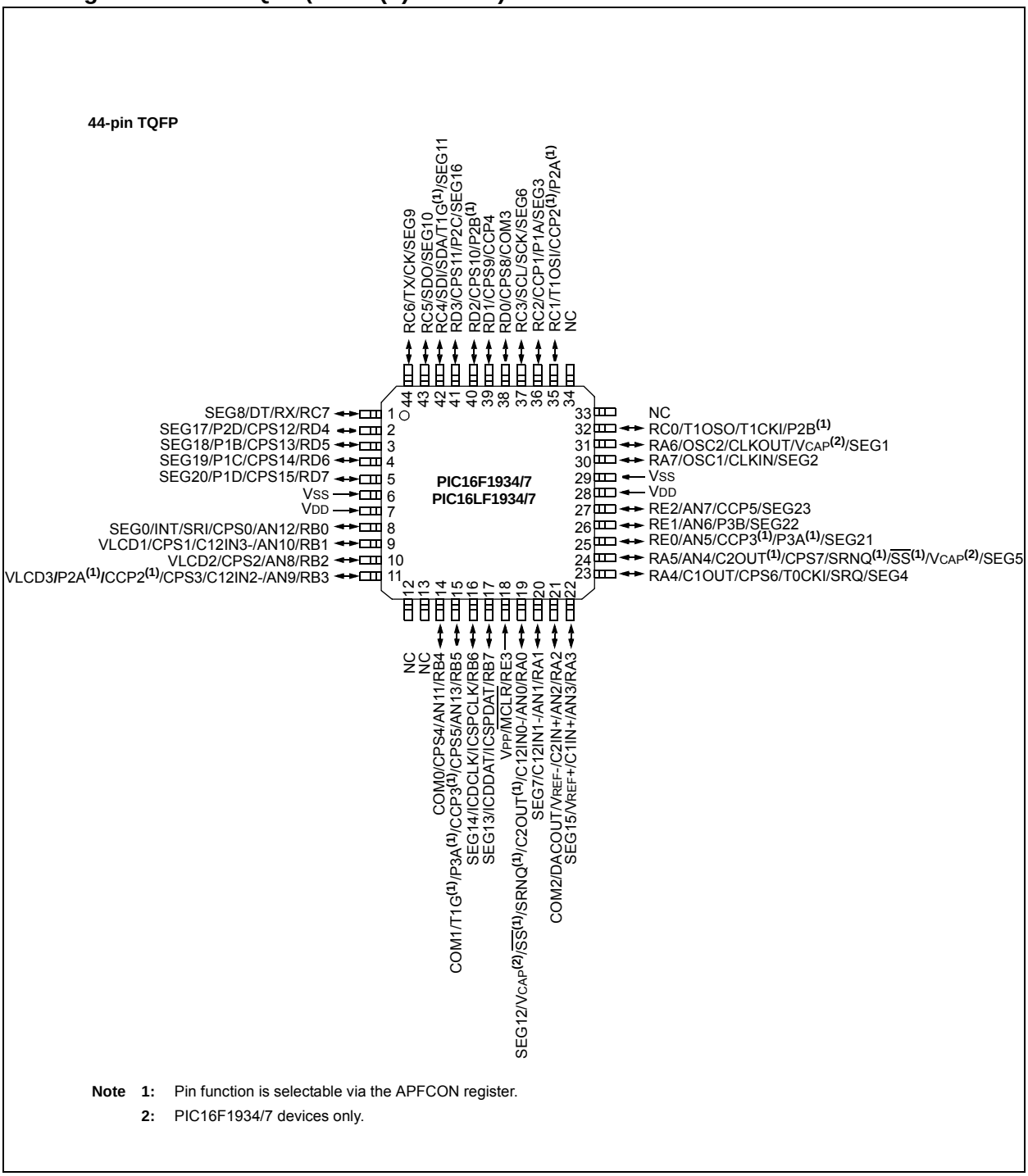
What is "Embedded - Microcontrollers"?

"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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 14x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1937-e-p

Pin Diagram – 44-Pin TQFP (PIC16(L)F1934/7)



Note 1: Pin function is selectable via the APFCON register.

2: PIC16F1934/7 devices only.

TABLE 2: 40/44-PIN SUMMARY(PIC16(L)F1934/7)

I/O	40-Pin PDIP	40-Pin UQFN	44-Pin TQFP	44-Pin QFN	ANSEL	A/D	Cap Sense	Comparator	SR Latch	Timers	CCP	EUSART	MSSP	LCD	Interrupt	Pull-up	Basic
RA0	2	17	19	19	Y	AN0	—	C12IN0-/ C2OUT(1)	SRNQ(1)	—	—	—	$\overline{SS}$(1)	SEG12	—	—	VCAP
RA1	3	18	20	20	Y	AN1	—	C12IN1-	—	—	—	—	—	SEG7	—	—	—
RA2	4	19	21	21	Y	AN2/ VREF-	—	C2IN+/ DACOUT	—	—	—	—	—	COM2	—	—	—
RA3	5	20	22	22	Y	AN3/ VREF+	—	C1IN+	—	—	—	—	—	SEG15	—	—	—
RA4	6	21	23	23	Y	—	CPS6	C1OUT	SRQ	T0CKI		—	—	SEG4	—	—	—
RA5	7	22	24	24	Y	AN4	CPS7	C2OUT(1)	SRNQ(1)	—	—	—	$\overline{SS}$(1)	SEG5	—	—	VCAP
RA6	14	29	31	33	—	—	—	—	—	—	—	—	—	SEG1	—	—	OSC2/ CLKOUT VCAP
RA7	13	28	30	32	—	—	—	—	—	—	—	—	—	SEG2	—	—	OSC1/ CLKIN
RB0	33	8	8	9	Y	AN12	CPS0	—	SRI	—	—	—	—	SEG0	INT/ IOC	Y	—
RB1	34	9	9	10	Y	AN10	CPS1	C12IN3-	—	—	—	—	—	VLCD1	IOC	Y	—
RB2	35	10	10	11	Y	AN8	CPS2	—	—	—	—	—	—	VLCD2	IOC	Y	—
RB3	36	11	11	12	Y	AN9	CPS3	C12IN2-	—	—	CCP2(1)/ P2A(1)	—	—	VLCD3	IOC	Y	—
RB4	37	12	14	14	Y	AN11	CPS4	—	—	—	—	—	—	COM0	IOC	Y	—
RB5	38	13	15	15	Y	AN13	CPS5	—	—	T1G(1)	CCP3(1)/ P3A(1)	—	—	COM1	IOC	Y	—
RB6	39	14	16	16	—	—	—	—	—	—	—	—	—	SEG14	IOC	Y	ICSPCLK/ ICDCLK
RB7	40	15	17	17	—	—	—	—	—	—	—	—	—	SEG13	IOC	Y	ICSPDAT/ ICDDAT
RC0	15	30	32	34	—	—	—	—	—	T1OSO/ T1CKI	P2B(1)	—	—	—	—	—	—
RC1	16	31	35	35	—	—	—	—	—	T1OSI	CCP2(1)/ P2A(1)	—	—	—	—	—	—
RC2	17	32	36	36	—	—	—	—	—	—	CCP1/ P1A	—	—	SEG3	—	—	—
RC3	18	33	37	37	—	—	—	—	—	—	—	—	SCK/SCL	SEG6	—	—	—
RC4	23	38	42	42	—	—	—	—	—	T1G(1)	—	—	SDI/SDA	SEG11	—	—	—
RC5	24	39	43	43	—	—	—	—	—	—	—	—	SDO	SEG10	—	—	—
RC6	25	40	44	44	—	—	—	—	—	—	—	TX/CK	—	SEG9	—	—	—
RC7	26	1	1	1	—	—	—	—	—	—	—	RX/DT	—	SEG8	—	—	—
RD0	19	34	38	38	Y	—	CPS8	—	—	—	—	—	—	COM3	—	—	—
RD1	20	35	39	39	Y	—	CPS9	—	—	—	CCP4	—	—	—	—	—	—
RD2	21	36	40	40	Y	—	CPS10	—	—	—	P2B(1)	—	—	—	—	—	—
RD3	22	37	41	41	Y	—	CPS11	—	—	—	P2C	—	—	SEG16	—	—	—
RD4	27	2	2	2	Y	—	CPS12	—	—	—	P2D	—	—	SEG17	—	—	—
RD5	28	3	3	3	Y	—	CPS13	—	—	—	P1B	—	—	SEG18	—	—	—
RD6	29	4	4	4	Y	—	CPS14	—	—	—	P1C	—	—	SEG19	—	—	—
RD7	30	5	5	5	Y	—	CPS15	—	—	—	P1D	—	—	SEG20	—	—	—
RE0	8	23	25	25	Y	AN5	—	—	—	—	CCP3(1) P3A(1)	—	—	SEG21	—	—	—
RE1	9	24	26	26	Y	AN6	—	—	—	—	P3B	—	—	SEG22	—	—	—
RE2	10	25	27	27	Y	AN7	—	—	—	—	CCP5	—	—	SEG23	—	—	—
RE3	1	16	18	18	—	—	—	—	—	—	—	—	—	—	—	Y	$\overline{MCLR}$/VPP
VDD	11, 32	7, 26	7, 28	7,8, 28	—	—	—	—	—	—	—	—	—	—	—	—	VDD
Vss	12, 31	6, 27	6, 29	6,30, 31	—	—	—	—	—	—	—	—	—	—	—	—	VSS

Note 1: Pin functions can be moved using the APFCON register.

3.1.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRxH register and reading the matching INDFx register. The MOVIW instruction will place the lower 8 bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDF registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The HIGH directive will set bit<7> if a label points to a location in program memory.

EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSR

```
constants
    RETLW DATA0        ;Index0 data
    RETLW DATA1        ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;… LOTS OF CODE…
    MOVLW  LOW constants
    MOVWF  FSR1L
    MOVLW  HIGH constants
    MOVWF  FSR1H
    MOVIW  0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

3.2 Data Memory Organization

The data memory is partitioned in 32 memory banks with 128 bytes in a bank. Each bank consists of (Figure 3-3):

- 12 core registers
- 20 Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.5 "Indirect Addressing"** for more information.

3.2.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation of the PIC16(L)F1934/6/7. These registers are listed below:

- INDF0
- INDF1
- PCL
- STATUS
- FSR0 Low
- FSR0 High
- FSR1 Low
- FSR1 High
- BSR
- WREG
- PCLATH
- INTCON

Note:	The core registers are the first 12 addresses of every data memory bank.

3.2.2 SPECIAL FUNCTION REGISTER

The Special Function Registers (SFR) are registers used by the application to control the desired operation of peripheral functions in the device. The registers associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

3.2.3 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank.

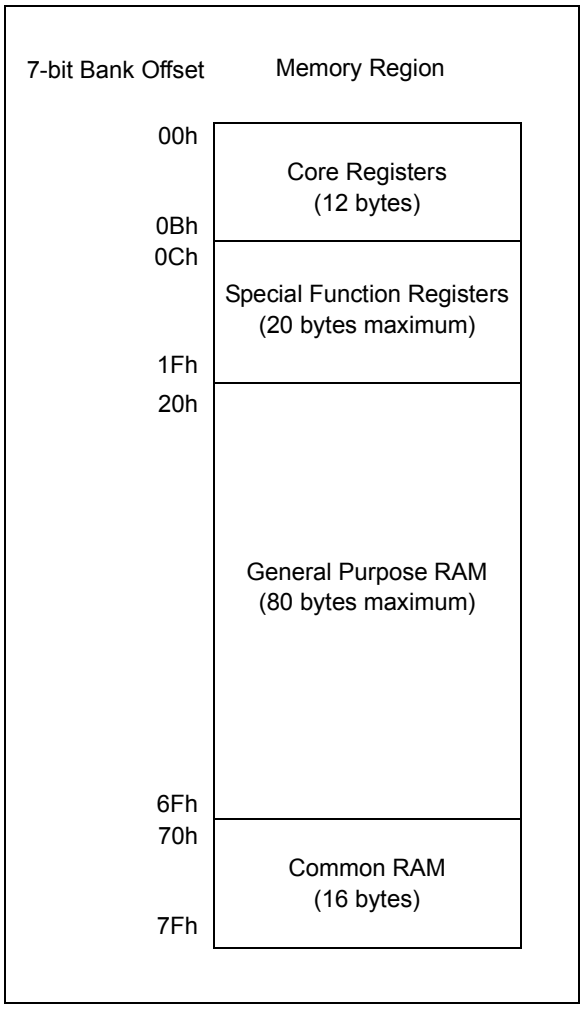
3.2.3.1 Linear Access to GPR

The general purpose RAM can be accessed in a non-banked method via the FSRs. This can simplify access to large memory structures. See **Section 3.5.2 "Linear Data Memory"** for more information.

3.2.4 COMMON RAM

There are 16 bytes of common RAM accessible from all banks.

FIGURE 3-3: BANKED MEMORY PARTITIONING



3.2.5 DEVICE MEMORY MAPS

The memory maps for the device family are as shown in Table 3-2.

TABLE 3-2: MEMORY MAP TABLES

Device	Banks	Table No.
PIC16F1934 PIC16LF1934	0-7	Table 3-3
	8-15	Table 3-4,Table 3-10
	16-23	Table 3-7
	23-31	Table 3-8, Table 3-11
PIC16F1936 PIC16LF1936	0-7	Table 3-5
	8-15	Table 3-6, Table 3-9
	16-23	Table 3-7
	23-31	Table 3-8, Table 3-11
PIC16F1937 PIC16LF1937	0-7	Table 3-5
	8-15	Table 3-6, Table 3-10
	16-23	Table 3-7
	23-31	Table 3-8, Table 3-11

TABLE 3-12: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 15											
780h(2)	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
781h(2)	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
782h(2)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
783h(2)	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
784h(2)	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
785h(2)	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
786h(2)	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
787h(2)	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
788h(2)	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000
789h(2)	WREG	Working Register								0000 0000	uuuu uuuu
78Ah(1, 2)	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
78Bh(2)	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000
78Ch	—	Unimplemented								—	—
78Dh	—	Unimplemented								—	—
78Eh	—	Unimplemented								—	—
78Fh	—	Unimplemented								—	—
790h	—	Unimplemented								—	—
791h	LCDCON	LCDEN	SLPEN	WERR	—	CS<1:0>		LMUX<1:0>		000- 0011	000- 0011
792h	LCDPS	WFT	BIASMD	LCDA	WA	LP<3:0>				0000 0000	0000 0000
793h	LCDREF	LCDIRE	LCDIRS	LCDIRI	—	VLCD3PE	VLCD2PE	VLCD1PE	—	000- 000-	000- 000-
794h	LCDCST	—	—	—	—	—	LCDCST<2:0>			---- -000	---- -000
795h	LCDRL	LRLAP<1:0>		LRLBP<1:0>		—	LRLAT<2:0>			0000 -000	0000 -000
796h	—	Unimplemented								—	—
797h	—	Unimplemented								—	—
798h	LCDSE0	SE<7:0>								0000 0000	uuuu uuuu
799h	LCDSE1	SE<15:8>								0000 0000	uuuu uuuu
79Ah	LCDSE2(3)	SE<23:16>								0000 0000	uuuu uuuu
79Bh	—	Unimplemented								—	—
79Ch	—	Unimplemented								—	—
79Dh	—	Unimplemented								—	—
79Eh	—	Unimplemented								—	—
79Fh	—	Unimplemented								—	—
7A0h	LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	xxxx xxxx	uuuu uuuu
7A1h	LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	xxxx xxxx	uuuu uuuu
7A2h	LCDDATA2(3)	SEG23 COM0	SEG22 COM0	SEG21 COM0	SEG20 COM0	SEG19 COM0	SEG18 COM0	SEG17 COM0	SEG16 COM0	xxxx xxxx	uuuu uuuu
7A3h	LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	xxxx xxxx	uuuu uuuu
7A4h	LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	xxxx xxxx	uuuu uuuu
7A5h	LCDDATA5(3)	SEG23 COM1	SEG22 COM1	SEG21 COM1	SEG20 COM1	SEG19 COM1	SEG18 COM1	SEG17 COM1	SEG16 COM1	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.

2: These registers can be addressed from any bank.

3: These registers/bits are not implemented on PIC16(L)F1936 devices, read as '0'.

4: Unimplemented, read as '1'.

REGISTER 5-3: OSCTUNE: OSCILLATOR TUNING 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
—	—	TUN<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 **TUN<4:0>:** Frequency Tuning bits
011111 = Maximum frequency
011110 =
•
•
•
000001 =
000000 = Oscillator module is running at the factory-calibrated frequency.
111111 =
•
•
•
100000 = Minimum frequency

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		81
OSCSTAT	T1OSCR	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	82
OSCTUNE	—	—	TUN<5:0>						83
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE(1)	100
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF(1)	103
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	203

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.
Note 1: PIC16F1934 only.

TABLE 5-3: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	FCMEN	IESO	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		$\overline{\text{CPD}}$	62
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		FOSC<2:0>			
CONFIG2	13:8	—	—	LVP	$\overline{\text{DEBUG}}$	—	BORV	STVREN	PLLEN	64
	7:0	—	—	VCAPEN<1:0>(1)		—	—	WRT<1:0>		

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.
Note 1: PIC16F1934/6/7 only.

7.6.2 PIE1 REGISTER

The PIE1 register contains the interrupt enable bits, as shown in Register 7-2.

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 7-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

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
TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	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 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
1 = Enables the Timer1 Gate Acquisition interrupt
0 = Disables the Timer1 Gate Acquisition interrupt

bit 6 **ADIE:** A/D Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **RCIE:** USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4 **TXIE:** USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3 **SSPIE:** Synchronous Serial Port (MSSP) Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

10.1 Independent Clock Source

The WDT derives its time base from the 31 kHz LFINTOSC internal oscillator. Time intervals in this chapter are based on a nominal interval of 1 ms. See the Electrical Specifications Chapters for the LFINTOSC tolerances.

10.2 WDT Operating Modes

The Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Word 1. See Table 10-1.

10.2.1 WDT IS ALWAYS ON

When the WDTE bits of Configuration Word 1 are set to '11', the WDT is always on.

WDT protection is active during Sleep.

10.2.2 WDT IS OFF IN SLEEP

When the WDTE bits of Configuration Word 1 are set to '10', the WDT is on, except in Sleep.

WDT protection is not active during Sleep.

10.2.3 WDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Word 1 are set to '01', the WDT is controlled by the SWDTEN bit of the WDTCON register.

WDT protection is unchanged by Sleep. See Table 10-1 for more details.

TABLE 10-1: WDT OPERATING MODES

WDTE<1:0>	SWDTEN	Device Mode	WDT Mode
11	X	X	Active
10	X	Awake	Active
		Sleep	Disabled
01	1	X	Active
	0		Disabled
00	X	X	Disabled

10.3 Time-Out Period

The WDTPS bits of the WDTCON register set the time-out period from 1 ms to 256 seconds (nominal). After a Reset, the default time-out period is 2 seconds.

10.4 Clearing the WDT

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

- Any Reset
- CLRWDT instruction is executed
- Device enters Sleep
- Device wakes up from Sleep
- Oscillator fail event
- WDT is disabled
- Oscillator Start-up TImer (OST) is running

See Table 10-2 for more information.

10.5 Operation During Sleep

When the device enters Sleep, the WDT is cleared. If the WDT is enabled during Sleep, the WDT resumes counting.

When the device exits Sleep, the WDT is cleared again. The WDT remains clear until the OST, if enabled, completes. See **Section 5.0 "Oscillator Module (With Fail-Safe Clock Monitor)"** for more information on the OST.

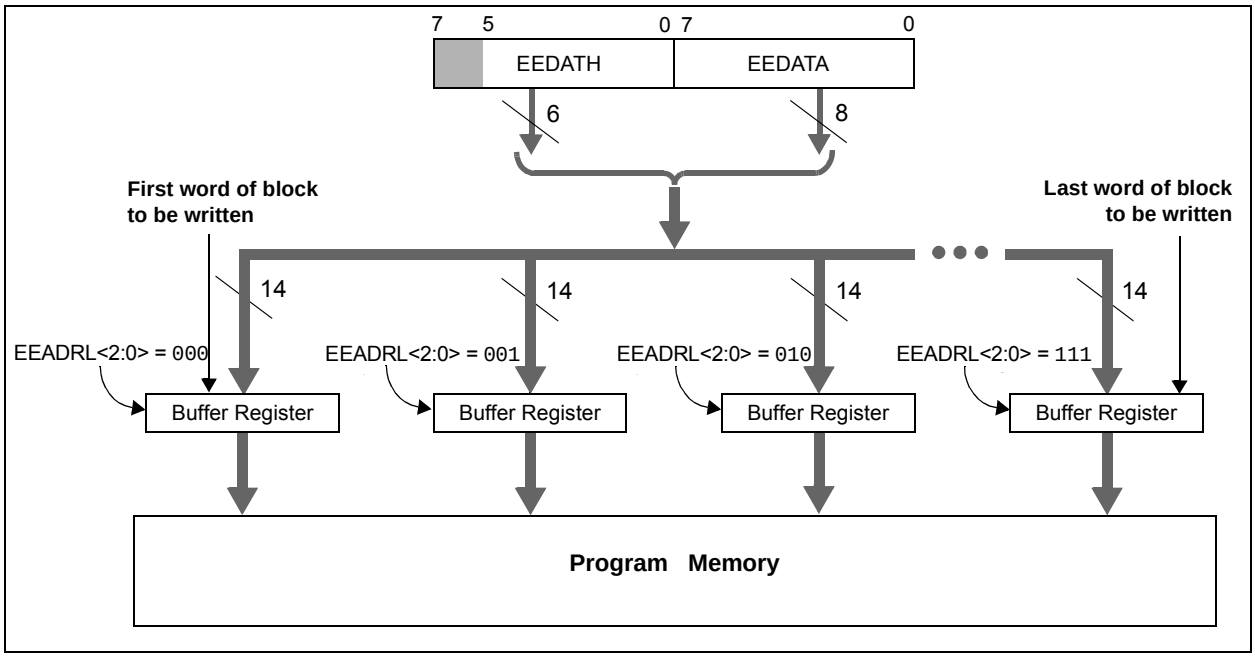
When a WDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the device wakes up and resumes operation. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the event. See **Section 3.0 "Memory Organization"** and STATUS register (**Register 3-1)** for more information.

TABLE 10-2: WDT CLEARING CONDITIONS

Conditions	WDT
WDTE<1:0> = 00	Cleared
WDTE<1:0> = 01 and SWDTEN = 0	
WDTE<1:0> = 10 and enter Sleep	
CLRWDT Command	
Oscillator Fail Detected	
Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK	
Exit Sleep + System Clock = XT, HS, LP	Cleared until the end of OST
Change INTOSC divider (IRCF bits)	Unaffected

After the "BSF EECON1,WR" instruction, the processor requires two cycles to set up the write operation. The user must place two NOP instructions after the WR bit is set. The processor will halt internal operations for the typical 2 ms, only during the cycle in which the write takes place (i.e., the last word of the block write). This is not Sleep mode as the clocks and peripherals will continue to run. The processor does not stall when LWLO = 1, loading the write latches. After the write cycle, the processor will resume operation with the third instruction after the EECON1 write instruction.

FIGURE 11-2: BLOCK WRITES TO FLASH PROGRAM MEMORY WITH 8 WRITE LATCHES



12.0 I/O PORTS

Depending on the device selected and peripherals enabled, there are up to five ports available. In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

Each port has three standard registers for its operation. These registers are:

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (analog select)
- WPUx (weak pull-up)
- INLVLx (input level control)

TABLE 12-1: PORT AVAILABILITY PER DEVICE

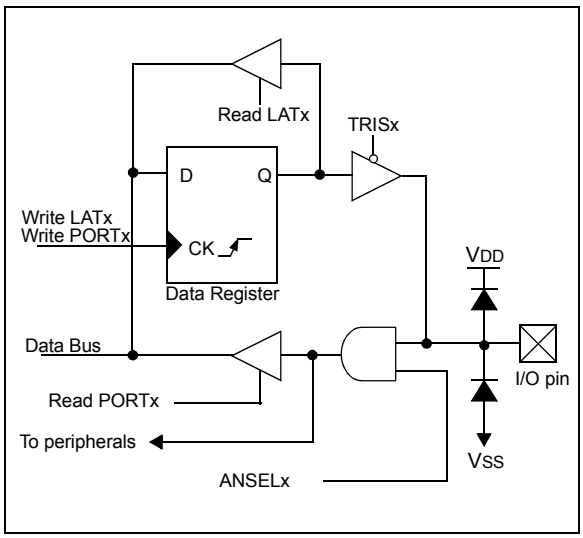
Device	PORTA	PORTB	PORTC	PORTD	PORTE
PIC16(L)F1934	●	●	●	●	●
PIC16(L)F1936	●	●	●		●
PIC16(L)F1937	●	●	●	●	●

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 12-1.

FIGURE 12-1: GENERIC I/O PORT OPERATION



EXAMPLE 12-1: INITIALIZING PORTA

```
; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL  PORTA        ;
CLRF     PORTA        ;Init PORTA
BANKSEL  LATA         ;Data Latch
CLRF     LATA         ;
BANKSEL  ANSELA       ;
CLRF     ANSELA       ;digital I/O
BANKSEL  TRISA        ;
MOVLW    B'00111000'  ;Set RA<5:3> as inputs
MOVWF    TRISA        ;and set RA<2:0> as
                      ;outputs
```

REGISTER 12-2: PORTA: PORTA REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
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 **RA<7:0>**: PORTA I/O Value bits(1)
1 = Port pin is > VIH
0 = Port pin is < VIL

Note 1: Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA register is return of actual I/O pin values.

REGISTER 12-3: TRISA: PORTA TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
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 **TRISA<7:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

REGISTER 12-4: LATA: PORTA DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
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 **LATA<7:0>**: PORTA Output Latch Value bits(1)

Note 1: Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA register is return of actual I/O pin values.

15.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting

15.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 12.0 "I/O Ports"** for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

15.1.2 CHANNEL SELECTION

There are 17 channel selections available:

- AN<13:0> pins
- Temperature Indicator
- DAC Output
- FVR (Fixed Voltage Reference) Output

Refer to **Section 16.0 "Temperature Indicator Module"**, **Section 17.0 "Digital-to-Analog Converter (DAC) Module"** and **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information on these channel selections.

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 15.2 "ADC Operation"** for more information.

15.1.3 ADC VOLTAGE REFERENCE

The ADPREF bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- VDD
- FVR 2.048V
- FVR 4.096V (Not available on LF devices)

The ADNREF bits of the ADCON1 register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF- pin
- VSS

See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more details on the fixed voltage reference.

15.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON1 register. There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11.5 TAD periods as shown in Figure 15-2.

For correct conversion, the appropriate TAD specification must be met. Refer to the A/D conversion requirements in the applicable Electrical Specifications Chapter for more information. Table 15-1 gives examples of appropriate ADC clock selections.

Note:	Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

24.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPCON3 register enables additional clock stretching and interrupt generation after the 8th falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPIF interrupt is set.

Figure 24-18 displays a standard waveform of a 7-bit Address Slave Transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with R/$\overline{W}$ bit set. After the 8th falling edge of the SCL line the CKP bit is cleared and SSPIF interrupt is generated.
4. Slave software clears SSPIF.
5. Slave software reads ACKTIM bit of SSPCON3 register, and R/$\overline{W}$ and D/$\overline{A}$ of the SSPSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{ACK}$ or not $\overline{ACK}$ and sets ACKDT bit of the SSPCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the $\overline{ACK}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPIF after the $\overline{ACK}$ if the R/$\overline{W}$ bit is set.
11. Slave software clears SSPIF.
12. Slave loads value to transmit to the master into SSPBUF setting the BF bit.

Note: SSPBUF cannot be loaded until after the $\overline{ACK}$.

13. Slave sets CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{ACK}$ value on the 9th SCL pulse.
15. Slave hardware copies the $\overline{ACK}$ value into the ACKSTAT bit of the SSPCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{ACK}$ the slave releases the bus allowing the master to send a Stop and end the communication.

Note: Master must send a not $\overline{ACK}$ on the last byte to ensure that the slave releases the SCL line to receive a Stop.

25.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 25-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all 8 or 9 bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

25.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

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

Setting the CREN bit of the RCSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART. The programmer must set the corresponding TRIS bit to configure the RX/DT I/O pin as an input.

Note 1:	If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

25.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See **Section 25.1.2.4 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREG register.

Note:	If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See **Section 25.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

25.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCIF interrupts are enabled by setting all of the following bits:

- RCIE interrupt enable bit of the PIE1 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RCIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

FIGURE 27-5: LCD INTERNAL REFERENCE LADDER POWER MODE SWITCHING DIAGRAM – TYPE A WAVEFORM (1/2 MUX, 1/2 BIAS DRIVE)

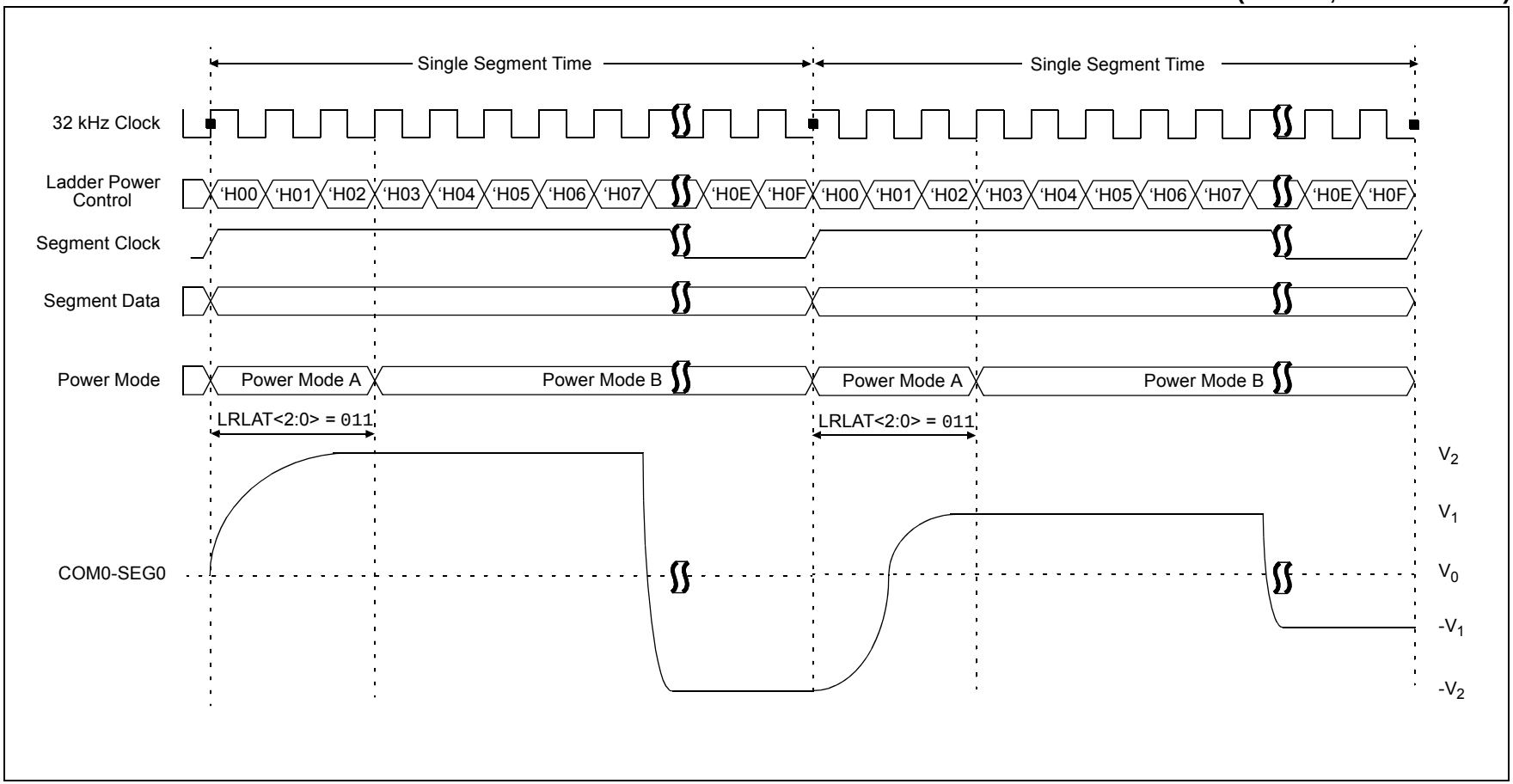


TABLE 29-3: PIC16(L)F1938/9 ENHANCED INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb	LSb	Status Affected	Notes
CONTROL OPERATIONS							
BRA	k	Relative Branch	2	11 001k kkkk kkkk			
BRW	–	Relative Branch with W	2	00 0000 0000 1011			
CALL	k	Call Subroutine	2	10 0kkk kkkk kkkk			
CALLW	–	Call Subroutine with W	2	00 0000 0000 1010			
GOTO	k	Go to address	2	10 1kkk kkkk kkkk			
RETFIE	k	Return from interrupt	2	00 0000 0000 1001			
RETLW	k	Return with literal in W	2	11 0100 kkkk kkkk			
RETURN	–	Return from Subroutine	2	00 0000 0000 1000			
INHERENT OPERATIONS							
CLRWDT	–	Clear Watchdog Timer	1	00 0000 0110 0100		$\overline{TO}$, $\overline{PD}$	
NOP	–	No Operation	1	00 0000 0000 0000			
OPTION	–	Load OPTION_REG register with W	1	00 0000 0110 0010			
RESET	–	Software device Reset	1	00 0000 0000 0001			
SLEEP	–	Go into Standby mode	1	00 0000 0110 0011		$\overline{TO}$, $\overline{PD}$	
TRIS	f	Load TRIS register with W	1	00 0000 0110 0fff			
C-COMPILER OPTIMIZED							
ADDFSR	n, k	Add Literal k to FSRn	1	11 0001 0nkk kkkk			
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00 0000 0001 0nmm		Z	**2, 3**
	k[n]	Move INDFn to W, Indexed Indirect.	1	11 1111 0nkk kkkk		Z	**2**
MOVWI	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	00 0000 0001 1nmm			**2, 3**
	k[n]	Move W to INDFn, Indexed Indirect.	1	11 1111 1nkk kkkk			**2**

Note 1: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

3: See Table in the MOVIW and MOVWI instruction descriptions.

FIGURE 30-3: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE

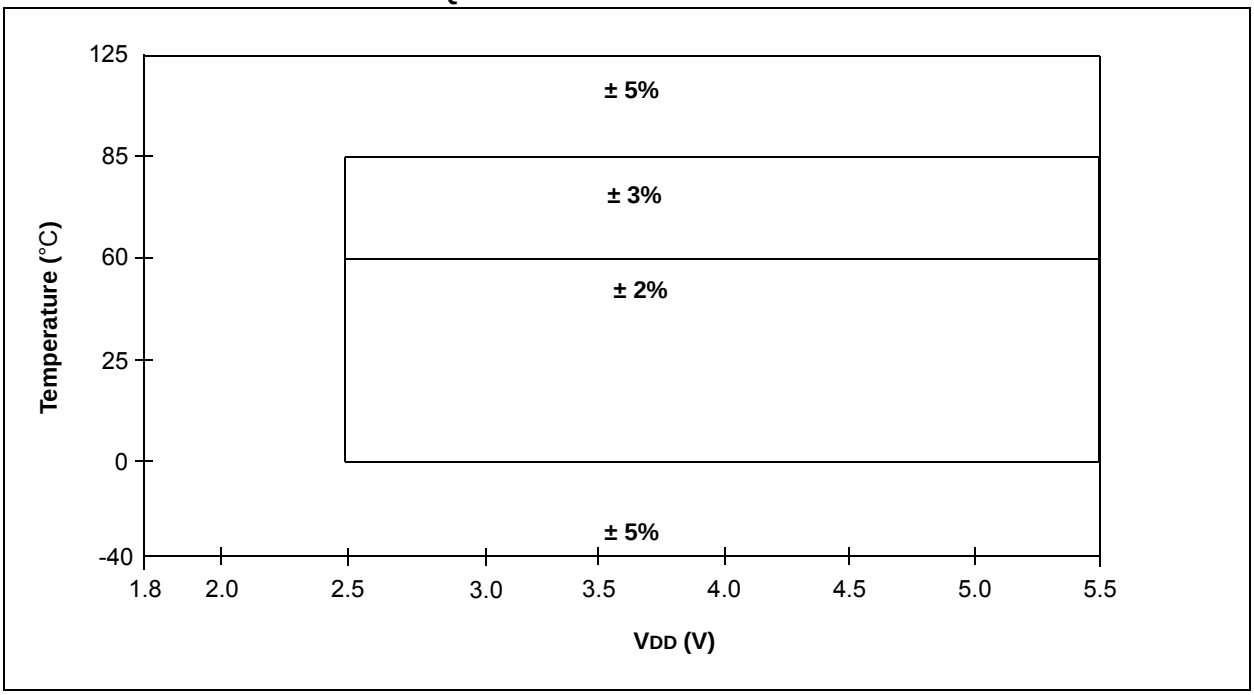


FIGURE 31-20: PIC16F1937 HF INTOSC MODE, FOSC = 16 MHz

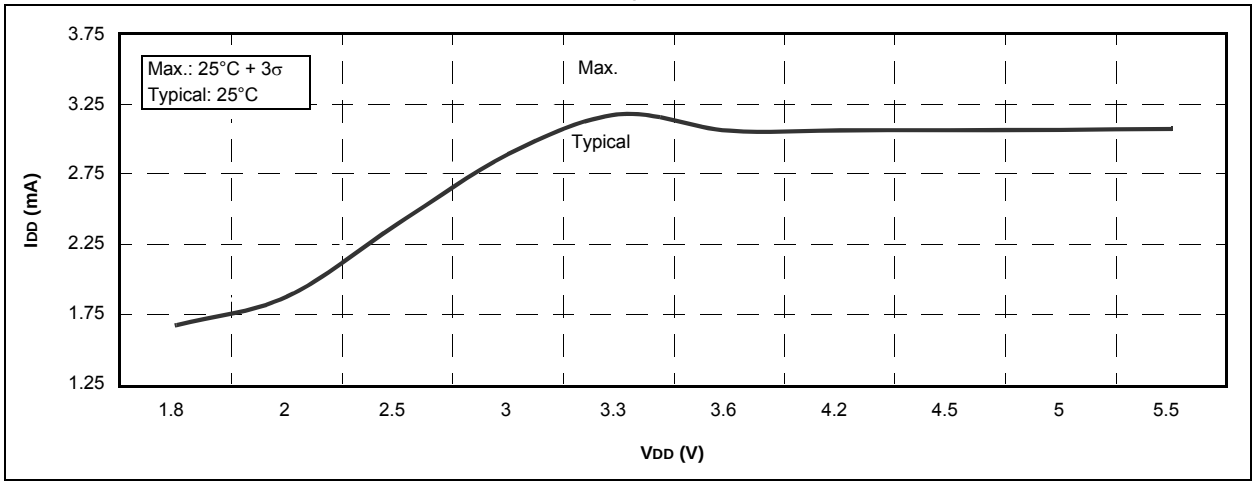


FIGURE 31-21: PIC16F1937 HF INTOSC MODE, FOSC = 8 MHz

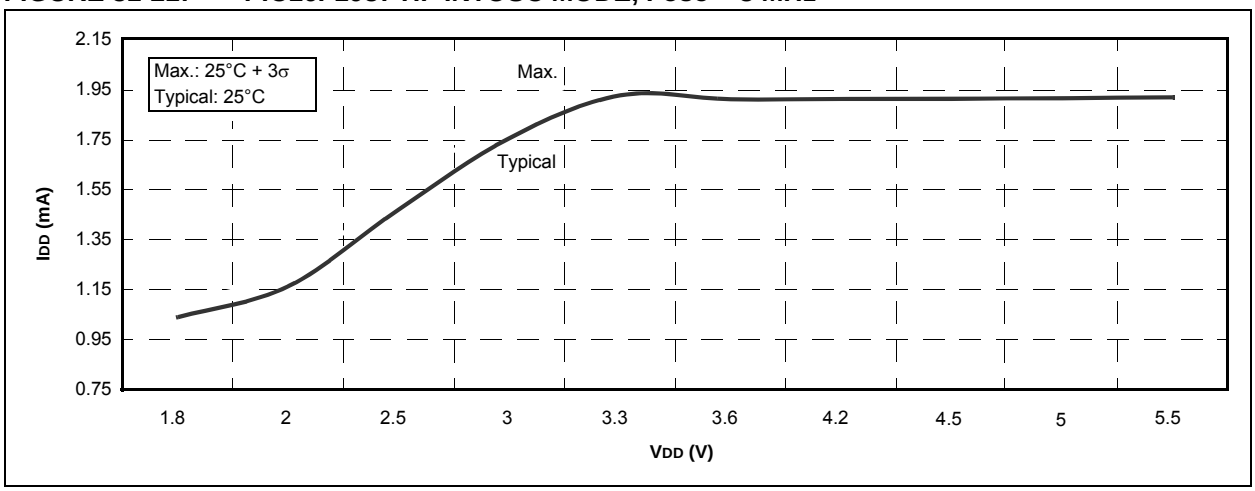


FIGURE 31-22: PIC16F1937 LF INTOSC MODE, FOSC = 32 kHz

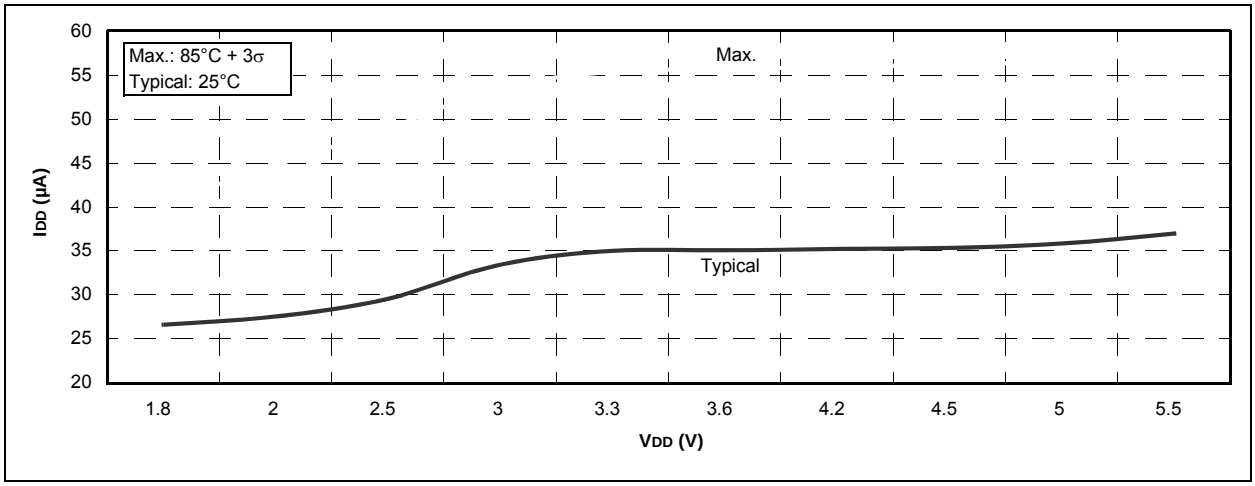


FIGURE 31-41: PIC16LF1937 LCD, LOW POWER

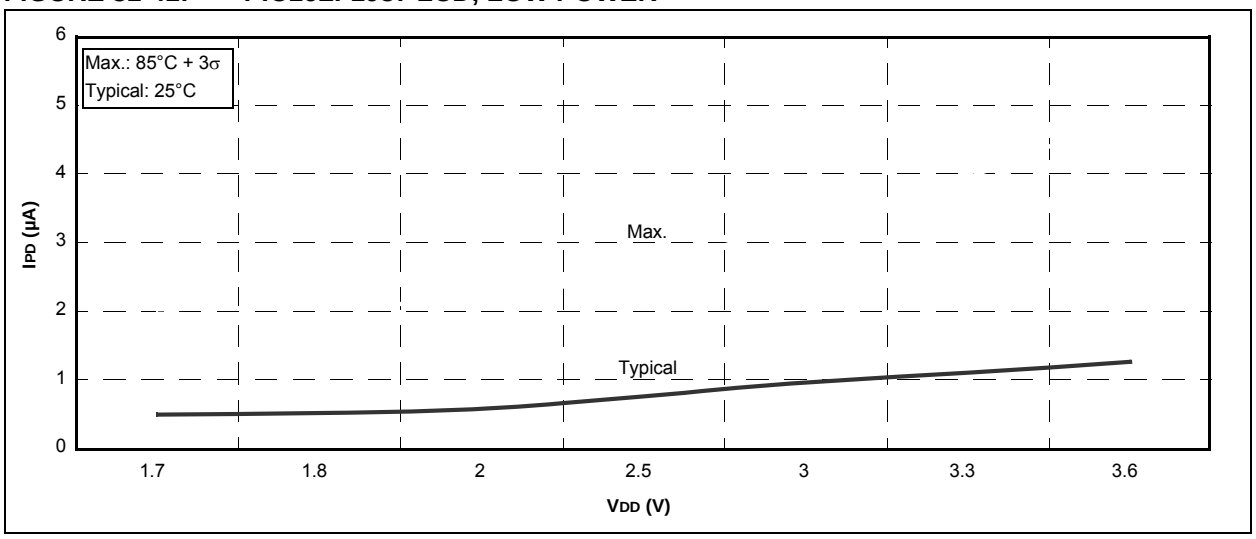


FIGURE 31-42: PIC16LF1937 LCD, MEDIUM POWER

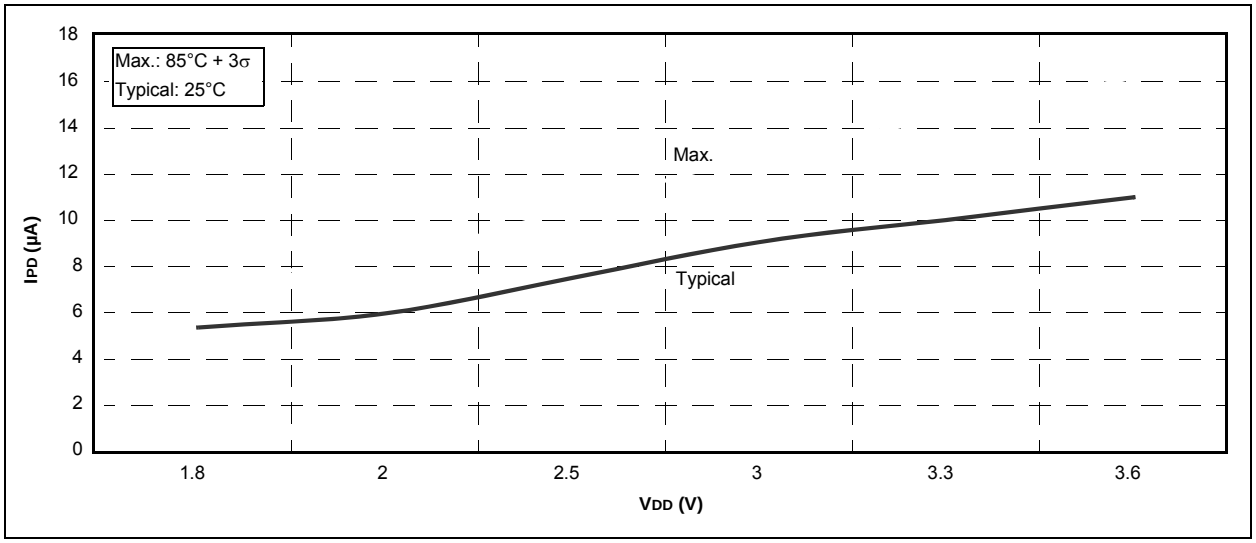


FIGURE 31-67: PIC16F1937 FVR_ADC

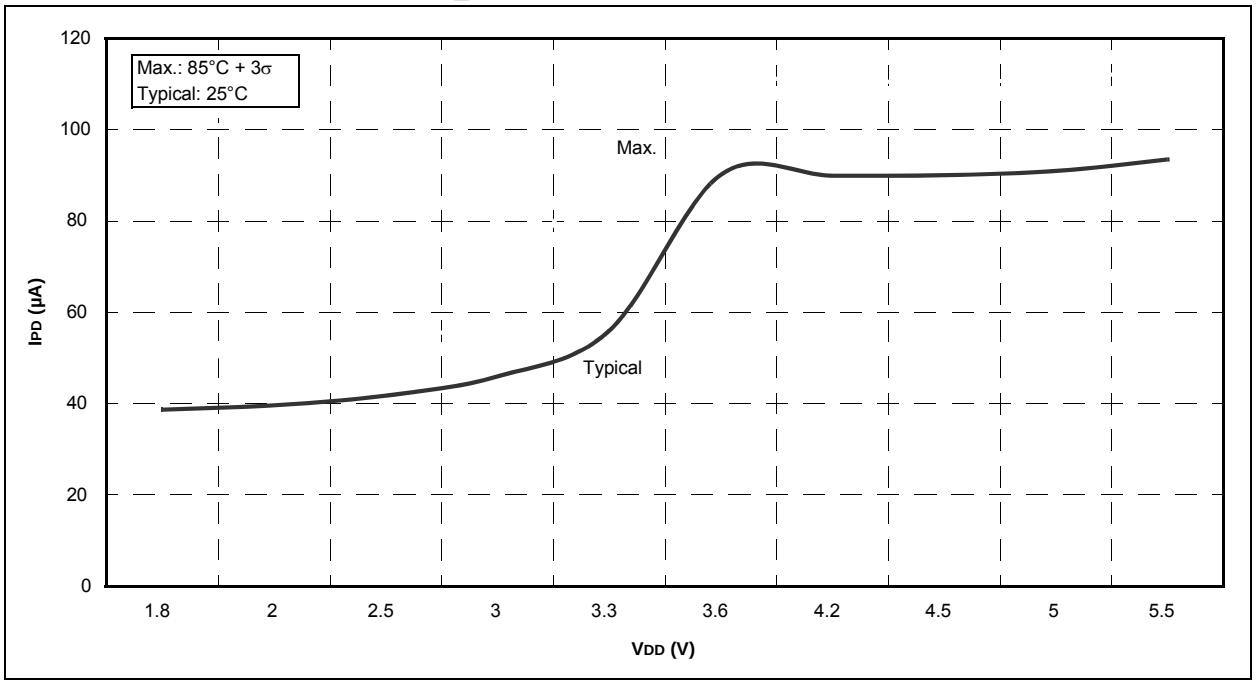


FIGURE 31-68: PIC16LF1937 WDT