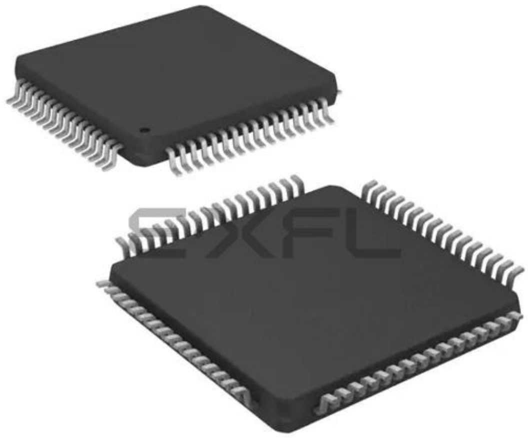
Microchip Technology - PIC18F6680-E/PT Datasheet



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	40MHz
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.25K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f6680-e-pt

FIGURE 3-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO VDD VIA 1 kΩ RESISTOR)

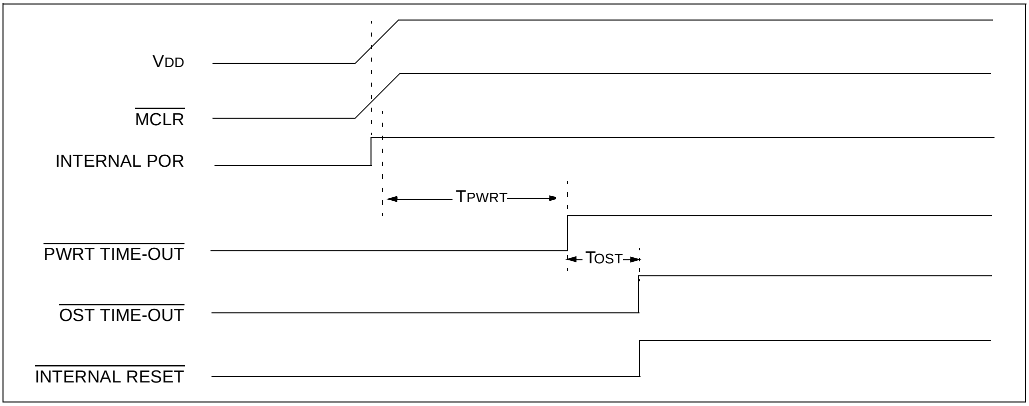


FIGURE 3-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO VDD): CASE 1

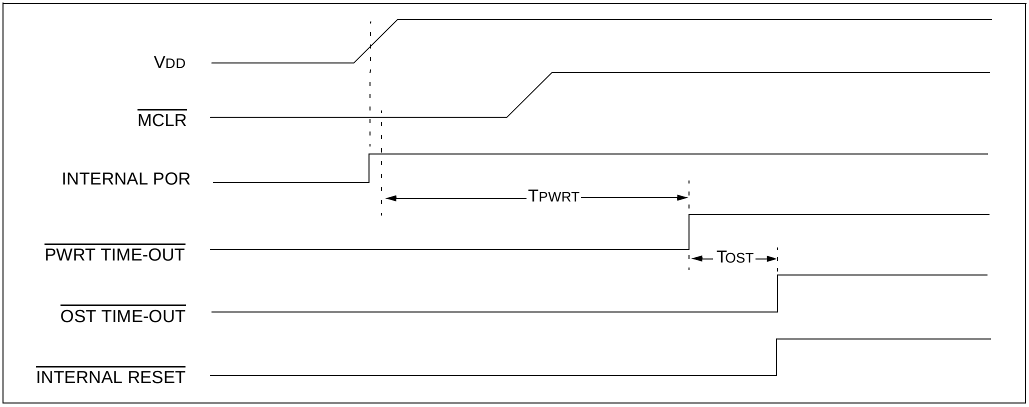


FIGURE 3-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO VDD): CASE 2

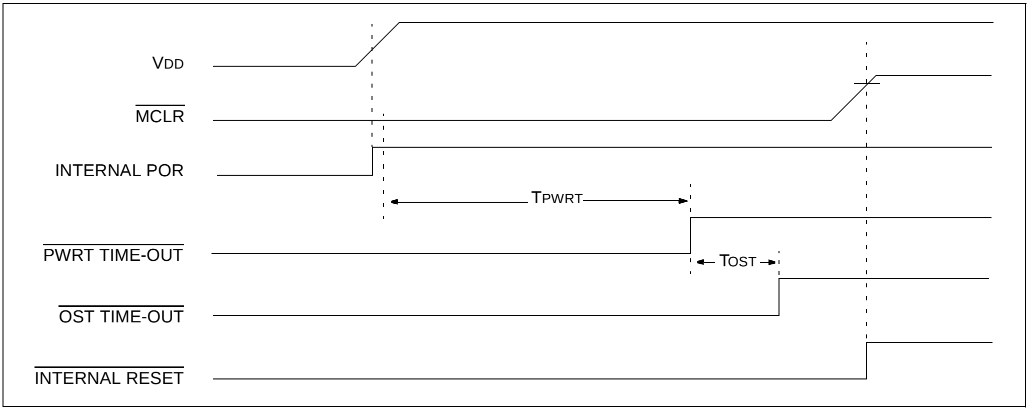


TABLE 4-2: SPECIAL FUNCTION REGISTER MAP (CONTINUED)

Address	Name	Address	Name	Address	Name	Address	Name
F7Fh	SPBRGH	F5Fh	CANCON_RO0	F3Fh	CANCON_RO2	F1Fh	RXM1EIDL
F7Eh	BAUDCON	F5Eh	CANSTAT_RO0	F3Eh	CANSTAT_RO2	F1Eh	RXM1EIDH
F7Dh	—[1]	F5Dh	RXB1D7	F3Dh	TXB1D7	F1Dh	RXM1SIDL
F7Ch	—[1]	F5Ch	RXB1D6	F3Ch	TXB1D6	F1Ch	RXM1SIDH
F7Bh	—[1]	F5Bh	RXB1D5	F3Bh	TXB1D5	F1Bh	RXM0EIDL
F7Ah	—[1]	F5Ah	RXB1D4	F3Ah	TXB1D4	F1Ah	RXM0EIDH
F79h	ECCP1DEL	F59h	RXB1D3	F39h	TXB1D3	F19h	RXM0SIDL
F78h	—[1]	F58h	RXB1D2	F38h	TXB1D2	F18h	RXM0SIDH
F77h	ECANCON	F57h	RXB1D1	F37h	TXB1D1	F17h	RXF5EIDL
F76h	TXERRCNT	F56h	RXB1D0	F36h	TXB1D0	F16h	RXF5EIDH
F75h	RXERRCNT	F55h	RXB1DLC	F35h	TXB1DLC	F15h	RXF5SIDL
F74h	COMSTAT	F54h	RXB1EIDL	F34h	TXB1EIDL	F14h	RXF5SIDH
F73h	CIOCON	F53h	RXB1EIDH	F33h	TXB1EIDH	F13h	RXF4EIDL
F72h	BRGCON3	F52h	RXB1SIDL	F32h	TXB1SIDL	F12h	RXF4EIDH
F71h	BRGCON2	F51h	RXB1SIDH	F31h	TXB1SIDH	F11h	RXF4SIDL
F70h	BRGCON1	F50h	RXB1CON	F30h	TXB1CON	F10h	RXF4SIDH
F6Fh	CANCON	F4Fh	CANCON_RO1	F2Fh	CANCON_RO3	F0Fh	RXF3EIDL
F6Eh	CANSTAT	F4Eh	CANSTAT_RO1	F2Eh	CANSTAT_RO3	F0Eh	RXF3EIDH
F6Dh	RXB0D7	F4Dh	TXB0D7	F2Dh	TXB2D7	F0Dh	RXF3SIDL
F6Ch	RXB0D6	F4Ch	TXB0D6	F2Ch	TXB2D6	F0Ch	RXF3SIDH
F6Bh	RXB0D5	F4Bh	TXB0D5	F2Bh	TXB2D5	F0Bh	RXF2EIDL
F6Ah	RXB0D4	F4Ah	TXB0D4	F2Ah	TXB2D4	F0Ah	RXF2EIDH
F69h	RXB0D3	F49h	TXB0D3	F29h	TXB2D3	F09h	RXF2SIDL
F68h	RXB0D2	F48h	TXB0D2	F28h	TXB2D2	F08h	RXF2SIDH
F67h	RXB0D1	F47h	TXB0D1	F27h	TXB2D1	F07h	RXF1EIDL
F66h	RXB0D0	F46h	TXB0D0	F26h	TXB2D0	F06h	RXF1EIDH
F65h	RXB0DLC	F45h	TXB0DLC	F25h	TXB2DLC	F05h	RXF1SIDL
F64h	RXB0EIDL	F44h	TXB0EIDL	F24h	TXB2EIDL	F04h	RXF1SIDH
F63h	RXB0EIDH	F43h	TXB0EIDH	F23h	TXB2EIDH	F03h	RXF0EIDL
F62h	RXB0SIDL	F42h	TXB0SIDL	F22h	TXB2SIDL	F02h	RXF0EIDH
F61h	RXB0SIDH	F41h	TXB0SIDH	F21h	TXB2SIDH	F01h	RXF0SIDL
F60h	RXB0CON	F40h	TXB0CON	F20h	TXB2CON	F00h	RXF0SIDH

Note 1: Unimplemented registers are read as '0'.

2: This register is not available on PIC18F6X8X devices.

3: This is not a physical register.

TABLE 4-2: SPECIAL FUNCTION REGISTER MAP (CONTINUED)

Address	Name	Address	Name	Address	Name	Address	Name
EFFh	—(1)	EDFh	—(1)	EBFh	—(1)	E9Fh	—(1)
EFEh	—(1)	EDEh	—(1)	EBEh	—(1)	E9Eh	—(1)
EFDh	—(1)	EDDh	—(1)	EBDh	—(1)	E9Dh	—(1)
EFCh	—(1)	EDCh	—(1)	EBCh	—(1)	E9Ch	—(1)
EFBh	—(1)	EDBh	—(1)	EBBh	—(1)	E9Bh	—(1)
EFAh	—(1)	EDAh	—(1)	EBAh	—(1)	E9Ah	—(1)
EF9h	—(1)	ED9h	—(1)	EB9h	—(1)	E99h	—(1)
EF8h	—(1)	ED8h	—(1)	EB8h	—(1)	E98h	—(1)
EF7h	—(1)	ED7h	—(1)	EB7h	—(1)	E97h	—(1)
EF6h	—(1)	ED6h	—(1)	EB6h	—(1)	E96h	—(1)
EF5h	—(1)	ED5h	—(1)	EB5h	—(1)	E95h	—(1)
EF4h	—(1)	ED4h	—(1)	EB4h	—(1)	E94h	—(1)
EF3h	—(1)	ED3h	—(1)	EB3h	—(1)	E93h	—(1)
EF2h	—(1)	ED2h	—(1)	EB2h	—(1)	E92h	—(1)
EF1h	—(1)	ED1h	—(1)	EB1h	—(1)	E91h	—(1)
EF0h	—(1)	ED0h	—(1)	EB0h	—(1)	E90h	—(1)
EEFh	—(1)	ECFh	—(1)	EAFh	—(1)	E8Fh	—(1)
EEEh	—(1)	ECEh	—(1)	EAEh	—(1)	E8Eh	—(1)
EEDh	—(1)	ECDh	—(1)	EADh	—(1)	E8Dh	—(1)
EECh	—(1)	ECCh	—(1)	EACh	—(1)	E8Ch	—(1)
EEBh	—(1)	ECBh	—(1)	EABh	—(1)	E8Bh	—(1)
EEAh	—(1)	ECAh	—(1)	EAAh	—(1)	E8Ah	—(1)
EE9h	—(1)	EC9h	—(1)	EA9h	—(1)	E89h	—(1)
EE8h	—(1)	EC8h	—(1)	EA8h	—(1)	E88h	—(1)
EE7h	—(1)	EC7h	—(1)	EA7h	—(1)	E87h	—(1)
EE6h	—(1)	EC6h	—(1)	EA6h	—(1)	E86h	—(1)
EE5h	—(1)	EC5h	—(1)	EA5h	—(1)	E85h	—(1)
EE4h	—(1)	EC4h	—(1)	EA4h	—(1)	E84h	—(1)
EE3h	—(1)	EC3h	—(1)	EA3h	—(1)	E83h	—(1)
EE2h	—(1)	EC2h	—(1)	EA2h	—(1)	E82h	—(1)
EE1h	—(1)	EC1h	—(1)	EA1h	—(1)	E81h	—(1)
EE0h	—(1)	EC0h	—(1)	EA0h	—(1)	E80h	—(1)

Note 1: Unimplemented registers are read as ‘0’.
2: This register is not available on PIC18F6X8X devices.
3: This is not a physical register.

TABLE 4-3: REGISTER FILE SUMMARY (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxx	37, 81
TMR0H	Timer0 Register High Byte								0000 0000	37, 157
TMR0L	Timer0 Register Low Byte								xxxx xxxx	37, 157
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	37, 155
OSCCON	—	—	—	—	LOCK	PLLEN	SCS1	SCS	---- 0000	27, 37
LVDCON	—	—	IRVST	LVDEN	LVDL3	LVDL2	LVDL1	LVDL0	--00 0101	37, 271
WDTCON	—	—	—	—	—	—	—	SWDTE	---- ---0	37, 355
RCON	IPEN	—	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	0--1 11qq	37, 82, 123
TMR1H	Timer1 Register High Byte								xxxx xxxx	37, 159
TMR1L	Timer1 Register Low Byte								xxxx xxxx	37, 159
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	0-00 0000	37, 159
TMR2	Timer2 Register								0000 0000	37, 162
PR2	Timer2 Period Register								1111 1111	37, 163
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	37, 162
SSPBUF	SSP Receive Buffer/Transmit Register								xxxx xxxx	37, 189
SSPADD	SSP Address Register in I^{2}C Slave mode. SSP Baud Rate Reload Register in I^{2}C Master mode.								0000 0000	37, 198
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	37, 199
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	37, 191
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	37, 201
ADRESH	A/D Result Register High Byte								xxxx xxxx	38, 257
ADRESL	A/D Result Register Low Byte								xxxx xxxx	38, 257
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	ADON	--00 0000	38, 249
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0000	38, 257
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	38, 251
CCPR1H	Enhanced Capture/Compare/PWM Register 1 High Byte								xxxx xxxx	38, 173
CCPR1L	Enhanced Capture/Compare/PWM Register 1 Low Byte								xxxx xxxx	38, 172
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	38, 172
CCPR2H	Capture/Compare/PWM Register 2 High Byte								xxxx xxxx	38, 172
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								xxxx xxxx	38, 172
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	38, 172
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	38, 172
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	38, 265
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	38, 259
TMR3H	Timer3 Register High Byte								xxxx xxxx	38, 164
TMR3L	Timer3 Register Low Byte								xxxx xxxx	38, 164
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{T3SYNC}$	TMR3CS	TMR3ON	0000 0000	38, 164
PSPCON	IBF	OBF	IBOV	PSPMODE	—	—	—	—	0000 ----	38, 153

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

Note 1: RA6 and associated bits are configured as port pins in RCIO and ECIO Oscillator mode only and read ‘0’ in all other oscillator modes.

2: Bit 21 of the TBLPTRU allows access to the device configuration bits.

3: These registers are unused on PIC18F6X80 devices; always maintain these clear.

4: These bits have multiple functions depending on the CAN module mode selection.

5: Meaning of this register depends on whether this buffer is configured as transmit or receive.

6: RG5 is available as an input when $\overline{MCLR}$ is disabled.

7: This register reads all ‘0’s until the ECAN module is set up in Mode 1 or Mode 2.

4.13 Status Register

The Status register, shown in Register 4-3, contains the arithmetic status of the ALU. The Status register can be the destination for any instruction as with any other register. If the Status register is the destination for an instruction that affects the Z, DC, C, OV or N bits, then the write to these five bits is disabled. These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the Status register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the Status register as `000u u1uu` (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF`, `MOVFF` and `MOVWF` instructions are used to alter the Status register because these instructions do not affect the Z, C, DC, OV or N bits from the Status register. For other instructions not affecting any status bits, see Table 25-2.

Note: The C and DC bits operate as a $\overline{\text{borrow}}$ and digit borrow bit respectively, in subtraction.

REGISTER 4-3: STATUS REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	N	OV	Z	DC	C
bit 7							bit 0

bit 7-5 **Unimplemented:** Read as '`0`'

bit 4 **N:** Negative bit

This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = `1`).

`1` = Result was negative
`0` = Result was positive

bit 3 **OV:** Overflow bit

This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7) to change state.

`1` = Overflow occurred for signed arithmetic (in this arithmetic operation)
`0` = No overflow occurred

bit 2 **Z:** Zero bit

`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit carry/$\overline{\text{borrow}}$ bit

For `ADDWF`, `ADDLW`, `SUBLW`, and `SUBWF` instructions:

`1` = A carry-out from the 4th low order bit of the result occurred
`0` = No carry-out from the 4th low order bit of the result

Note: For $\overline{\text{borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the bit 4 or bit 3 of the source register.

bit 0 **C:** Carry/$\overline{\text{borrow}}$ bit

For `ADDWF`, `ADDLW`, `SUBLW`, and `SUBWF` instructions:

`1` = A carry-out from the Most Significant bit of the result occurred
`0` = No carry-out from the Most Significant bit of the result occurred

Note: For $\overline{\text{borrow}}$, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low-order bit of the source register.

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

9.6 INT0 Interrupt

External interrupts on the RB0/INT0, RB1/INT1, RB2/INT2 and RB3/INT3 pins are edge-triggered: either rising if the corresponding INTEDGx bit is set in the INTCON2 register, or falling if the INTEDGx bit is clear. When a valid edge appears on the RBx/INTx pin, the corresponding flag bit, INTxF, is set. This interrupt can be disabled by clearing the corresponding enable bit, INTxE. Flag bit, INTxF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt. All external interrupts (INT0, INT1, INT2 and INT3) can wake-up the processor from Sleep if bit INTxIE was set prior to going into Sleep. If the global interrupt enable bit GIE is set, the processor will branch to the interrupt vector following wake-up.

The interrupt priority for INT, INT2 and INT3 is determined by the value contained in the interrupt priority bits: INT1IP (INTCON3<6>), INT2IP (INTCON3<7>) and INT3IP (INTCON2<1>). There is no priority bit associated with INT0; it is always a high priority interrupt source.

9.7 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow in the TMR0 register (0FFh → 00h) will set flag bit TMR0IF. In 16-bit mode, an overflow in the TMR0H:TMR0L registers (0FFFFh → 0000h) will set flag bit, TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit, TMR0IP (INTCON2<2>). See **Section 11.0 “Timer0 Module”** for further details on the Timer0 module.

9.8 PORTB Interrupt-on-Change

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit, RBIE (INTCON<3>). Interrupt priority for PORTB interrupt-on-change is determined by the value contained in the interrupt priority bit, RBIP (INTCON2<0>).

9.9 Context Saving During Interrupts

During an interrupt, the return PC value is saved on the stack. Additionally, the WREG, Status and BSR registers are saved on the fast return stack. If a fast return from interrupt is not used (See **Section 4.3 “Fast Register Stack”**), the user may need to save the WREG, Status and BSR registers in software. Depending on the user’s application, other registers may also need to be saved. Example 9-1 saves and restores the WREG, Status and BSR registers during an Interrupt Service Routine.

EXAMPLE 9-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

```
MOVWF     W_TEMP                          ; W_TEMP is in virtual bank
MOVFF     STATUS, STATUS_TEMP             ; STATUS_TEMP located anywhere
MOVFF     BSR, BSR_TEMP                   ; BSR located anywhere
;
; USER ISR CODE
;
MOVFF     BSR_TEMP, BSR                   ; Restore BSR
MOVF      W_TEMP, W                       ; Restore WREG
MOVFF     STATUS_TEMP, STATUS             ; Restore STATUS
```

TABLE 10-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT0	bit 0	TTL/ST[(1)]	Input/output pin or external interrupt input 0. Internal software programmable weak pull-up.
RB1/INT1	bit 1	TTL/ST[(1)]	Input/output pin or external interrupt input 1. Internal software programmable weak pull-up.
RB2/INT2	bit 2	TTL/ST[(1)]	Input/output pin or external interrupt input 2. Internal software programmable weak pull-up.
RB3/INT3/CCP2[(3)]	bit 3	TTL/ST[(4)]	Input/output pin or external interrupt input 3. Capture 2 input/ Compare 2 output/PWM output (when CCP2MX configuration bit is enabled, all PIC18FXX85 operating modes except Microcontroller mode). Internal software programmable weak pull-up.
RB4/KBI0	bit 4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5/KBI1/PGM	bit 5	TTL/ST[(2)]	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Low-voltage ICSP enable pin.
RB6/KBI2/PGC	bit 6	TTL/ST[(2)]	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming clock.
RB7/KBI3/PGD	bit 7	TTL/ST[(2)]	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: RC1 is the alternate assignment for CCP2 when CCP2MX is not set (all operating modes except Microcontroller mode).

4: This buffer is a Schmitt Trigger input when configured as the CCP2 input.

TABLE 10-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

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
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
LATB	LATB Data Output Register								xxxx xxxx	uuuu uuuu
TRISB	PORTB Data Direction Register								1111 1111	1111 1111
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP	1111 1111	1111 1111
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	1100 0000	1100 0000

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

FIGURE 17-27: BUS COLLISION DURING START CONDITION (SCL = 0)

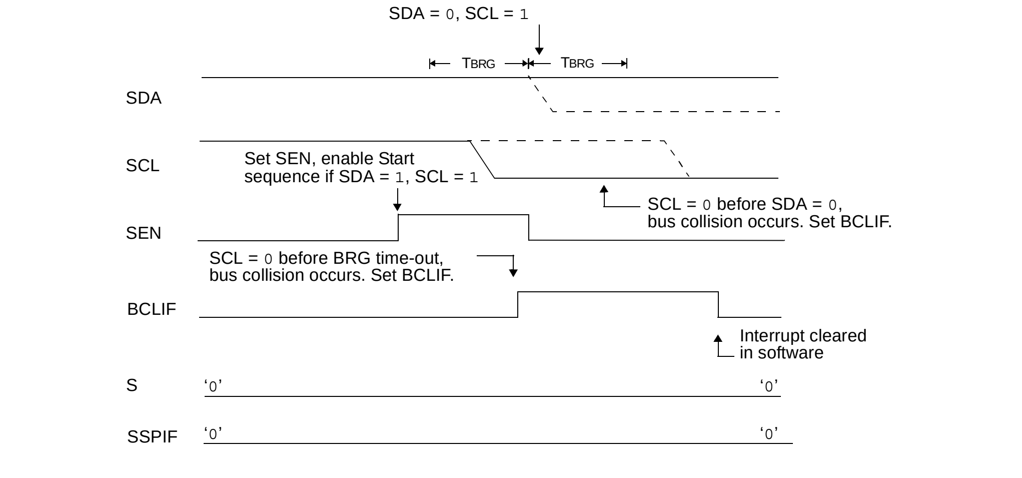
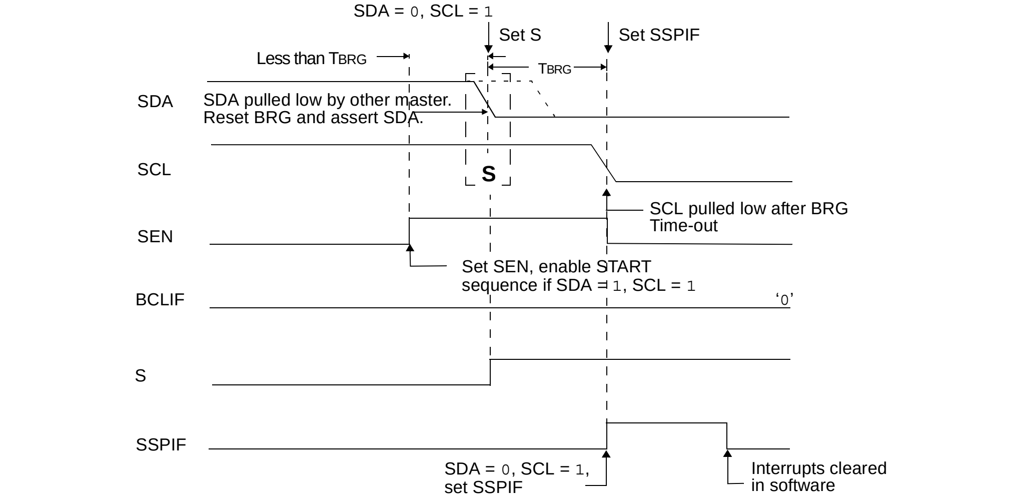

SDA = 0, SCL = 1
TBRG
TBRG
SDA
SCL
Set SEN, enable Start sequence if SDA = 1, SCL = 1
SCL = 0 before SDA = 0, bus collision occurs. Set BCLIF.
SEN
SCL = 0 before BRG time-out, bus collision occurs. Set BCLIF.
BCLIF
Interrupt cleared in software
S
'0'
'0'
SSPIF
'0'
'0'

FIGURE 17-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION


SDA = 0, SCL = 1
Set S
Set SSPIF
Less than TBRG
TBRG
SDA
SDA pulled low by other master. Reset BRG and assert SDA.
SCL
S
SCL pulled low after BRG Time-out
SEN
Set SEN, enable START sequence if SDA = 1, SCL = 1
BCLIF
'0'
S
SSPIF
SDA = 0, SCL = 1, set SSPIF
Interrupts cleared in software

19.0 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) converter module has 12 inputs for the PIC18F6X8X devices and 16 inputs for the PIC18F8X8X devices. This module allows conversion of an analog input signal to a corresponding 10-bit digital number.

A new feature for the A/D converter is the addition of programmable acquisition time. This feature allows the user to select a new channel for conversion and to set the GO/$\overline{\text{DONE}}$ bit immediately. When the GO/$\overline{\text{DONE}}$ bit is set, the selected channel is sampled for the programmed acquisition time before a conversion is actually started. This removes the firmware overhead that may have been required to allow for an acquisition (sampling) period (see Register 19-3 and **Section 19.4 "Selecting the A/D Conversion Clock"**).

The module has five registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

The ADCON0 register, shown in Register 19-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 19-2, configures the functions of the port pins. The ADCON2 register, shown in Register 19-3, configures the A/D clock source, programmed acquisition time and justification.

REGISTER 19-1: ADCON0 REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-2 **CHS3:CHS0:** Analog Channel Select bits

0000 = Channel 0 (AN0)
0001 = Channel 1 (AN1)
0010 = Channel 2 (AN2)
0011 = Channel 3 (AN3)
0100 = Channel 4 (AN4)
0101 = Channel 5 (AN5)
0110 = Channel 6 (AN6)
0111 = Channel 7 (AN7)
1000 = Channel 8 (AN8)
1001 = Channel 9 (AN9)
1010 = Channel 10 (AN10)
1011 = Channel 11 (AN11)
1100 = Channel 12 (AN12)[(1)]
1101 = Channel 13 (AN13)[(1)]
1110 = Channel 14 (AN14)[(1)]
1111 = Channel 15 (AN15)[(1)]

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

When ADON = 1:
1 = A/D conversion in progress. This bit is automatically cleared when the A/D conversion is complete.
0 = A/D Idle

bit 0 **ADON:** A/D On bit

1 = A/D converter module is enabled
0 = A/D converter module is disabled and consumes no current

Note 1: These channels are only available on PIC18F8X8X devices.

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

21.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference is a 16-tap resistor ladder network that provides a selectable voltage reference. The resistor ladder is segmented to provide two ranges of CVREF values and has a power-down function to conserve power when the reference is not being used. The CVRCON register controls the operation of the reference as shown in Register 21-1. The block diagram is given in Figure 21-1.

The comparator reference supply voltage can come from either VDD or VSS, or the external VREF+ and VREF- that are multiplexed with RA3 and RA2. The comparator reference supply voltage is controlled by the CVRSS bit.

21.1 Configuring the Comparator Voltage Reference

The comparator voltage reference can output 16 distinct voltage levels for each range. The equations used to calculate the output of the comparator voltage reference are as follows:

If CVRR = 1:
CVREF = (CVR<3:0>/24) x CVRSRC

If CVRR = 0:
CVREF = (CVDD x 1/4) + (CVR<3:0>/32) x CVRSRC

The settling time of the comparator voltage reference must be considered when changing the CVREF output (**Section 27.0 "Electrical Characteristics"**).

REGISTER 21-1: CVRCON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

bit 7 **CVREN**: Comparator Voltage Reference Enable bit
1 = CVREF circuit powered on
0 = CVREF circuit powered down

bit 6 **CVROE**: Comparator VREF Output Enable bit[(1)]
1 = CVREF voltage level is also output on the RF5/AN10/C1IN+/CVREF pin
0 = CVREF voltage is disconnected from the RF5/AN10/C1IN+/CVREF pin

bit 5 **CVRR**: Comparator VREF Range Selection bit
1 = 0.00 CVRSRC to 0.625 CVRSRC with CVRSRC/24 step size
0 = 0.25 CVRSRC to 0.71875 CVRSRC with CVRSRC/32 step size

bit 4 **CVRSS**: Comparator VREF Source Selection bit
1 = Comparator reference source, CVRSRC = VREF+ – VREF-
0 = Comparator reference source, CVRSRC = VDD – VSS

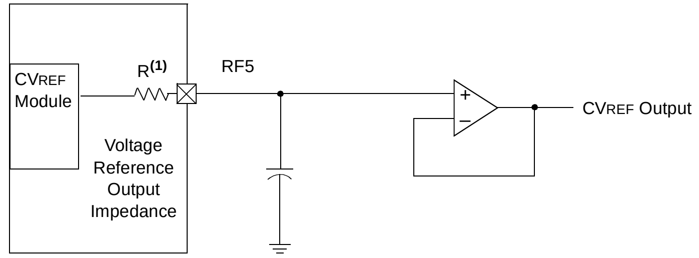
Note: To select (VREF+ – VREF-) as the comparator voltage reference source, the voltage reference configuration bits in the ADCON1 register (ADCON1<5:4>) must also be set to '11'.

bit 3-0 **CVR3:CVR0:** Comparator VREF Value Selection bits ($0 \leq$ VR3:VR0 ≤ 15)
When CVRR = 1:
CVREF = (CVR<3:0>/24) • (CVRSRC)
When CVRR = 0:
CVREF = 1/4 • (CVRSRC) + (CVR3:CVR0/32) • (CVRSRC)

Note 1: If enabled for output, RF5 must also be configured as an input by setting TRISF<5> to '1'.

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

FIGURE 21-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



Note 1: R is dependent upon the voltage reference configuration bits CVRCON<3:0> and CVRCON<5>.

TABLE 21-1: REGISTERS ASSOCIATED WITH COMPARATOR VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	0000 0000
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'.
Shaded cells are not used with the comparator voltage reference.

NOTES:

EXAMPLE 23-2: WIN AND ICODE BITS USAGE IN INTERRUPT SERVICE ROUTINE TO ACCESS TX/RX BUFFERS (CONTINUED)

```
ErrorInterrupt
    BCF    PIR3, ERRIF                   ; Clear the interrupt flag
    …                                    ; Handle error.
    RETFIE
TXB2Interrupt
    BCF    PIR3, TXB2IF                  ; Clear the interrupt flag
    GOTO   AccessBuffer
TXB1Interrupt
    BCF    PIR3, TXB1IF                  ; Clear the interrupt flag
    GOTO   AccessBuffer
TXB0Interrupt
    BCF    PIR3, TXB0IF                  ; Clear the interrupt flag
    GOTO   AccessBuffer
RXB1Interrupt
    BCF    PIR3, RXB1IF                  ; Clear the interrupt flag
    GOTO   Accessbuffer
RXB0Interrupt
    BCF    PIR3, RXB0IF                  ; Clear the interrupt flag
    GOTO   AccessBuffer
AccessBuffer                             ; This is either TX or RX interrupt
    ; Copy CANSTAT.ICODE bits to CANCON.WIN bits
    MOVF   TempCANCON, W                 ; Clear CANCON.WIN bits before copying
                                         ; new ones.
    ANDLW  B’11110001’                   ; Use previously saved CANCON value to
                                         ; make sure same value.
    MOVWF  TempCANCON                    ; Copy masked value back to TempCANCON
    MOVF   TempCANSTAT, W                ; Retrieve ICODE bits
    ANDLW  B’00001110’                   ; Use previously saved CANSTAT value
                                         ; to make sure same value.
    IORWF  TempCANCON                    ; Copy ICODE bits to WIN bits.
    MOVFF  TempCANCON, CANCON            ; Copy the result to actual CANCON
    ; Access current buffer…
    ; User code
    ; Restore CANCON.WIN bits
    MOVF   CANCON, W                     ; Preserve current non WIN bits
    ANDLW  B’11110001’
    IORWF  TempCANCON                    ; Restore original WIN bits
    ; Do not need to restore CANSTAT - it is read-only register.
    ; Return from interrupt or check for another module interrupt source
```

EXAMPLE 23-4: TRANSMITTING A CAN MESSAGE USING WIN BITS

```
; Need to transmit Standard Identifier message 123h using TXB0 buffer.
; To successfully transmit, CAN module must be either in Normal or Loopback mode.
; TXB0 buffer is not in access bank.  Use WIN bits to map it to RXB0 area.
MOVF    CANCON, W                     ; WIN bits are in lower 4 bits only.  Read CANCON
                                      ; register to preserve all other bits.  If operation
                                      ; mode is already known, there is no need to preserve
                                      ; other bits.
ANDLW   B’11110000’                   ; Clear WIN bits.
IORLW   B’00001000’                   ; Select Transmit Buffer 0
MOVWF   CANCON                        ; Apply the changes.
; Now TXB0 is mapped in place of RXB0.  All future access to RXB0 registers will actually
; yield TXB0 register values.

; Load transmit data into TXB0 buffer.
MOVLW   MY_DATA_BYTE1                 ; Load first data byte into buffer
MOVWF   RXB0D0                        ; Access TXB0D0 via RXB0D0 address.
; Load rest of the data bytes - up to 8 bytes into “TXB0” buffer using RXB0 registers.
...
; Load message identifier
MOVLW   60H                           ; Load SID2:SID0, EXIDE = 0
MOVWF   RXB0SIDL
MOVLW   24H                           ; Load SID10:SID3
MOVWF   RXB0SIDH
; No need to load RXB0EIDL:RXB0EIDH, as we are transmitting Standard Identifier Message only.

; Now that all data bytes are loaded, mark it for transmission.
MOVLW   B’00001000’                   ; Normal priority; Request transmission
MOVWF   RXB0CON

; If required, wait for message to get transmitted
BTFSC   RXB0CON, TXREQ                ; Is it transmitted?
BRA     $-2                           ; No.  Continue to wait...

; Message is transmitted.
; If required, reset the WIN bits to default state.
```

REGISTER 23-53: BRGCON2: BAUD RATE CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SEG2PHTS	SAM	SEG1PH2	SEG1PH1	SEG1PH0	PRSEG2	PRSEG1	PRSEG0
bit 7							bit 0

bit 7 **SEG2PHTS:** Phase Segment 2 Time Select bit

1 = Freely programmable
0 = Maximum of PHEG1 or Information Processing Time (IPT), whichever is greater

bit 6 **SAM:** Sample of the CAN bus Line bit

1 = Bus line is sampled three times prior to the sample point
0 = Bus line is sampled once at the sample point

bit 5-3 **SEG1PH2:SEG1PH0:** Phase Segment 1 bits

111 = Phase Segment 1 time = 8 x T_Q
110 = Phase Segment 1 time = 7 x T_Q
101 = Phase Segment 1 time = 6 x T_Q
100 = Phase Segment 1 time = 5 x T_Q
011 = Phase Segment 1 time = 4 x T_Q
010 = Phase Segment 1 time = 3 x T_Q
001 = Phase Segment 1 time = 2 x T_Q
000 = Phase Segment 1 time = 1 x T_Q

bit 2-0 **PRSEG2:PRSEG0:** Propagation Time Select bits

111 = Propagation time = 8 x T_Q
110 = Propagation time = 7 x T_Q
101 = Propagation time = 6 x T_Q
100 = Propagation time = 5 x T_Q
011 = Propagation time = 4 x T_Q
010 = Propagation time = 3 x T_Q
001 = Propagation time = 2 x T_Q
000 = Propagation time = 1 x T_Q

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

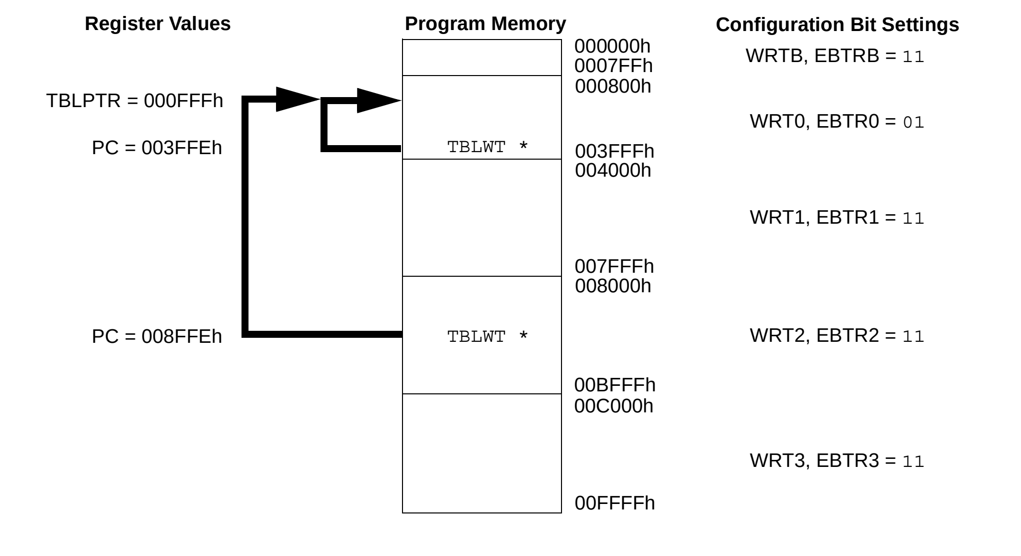
24.4.1 PROGRAM MEMORY CODE PROTECTION

The user memory may be read to or written from any location using the table read and table write instructions. The device ID may be read with table reads. The Configuration registers may be read and written with the table read and table write instructions.

In User mode, the CPn bits have no direct effect. CPn bits inhibit external reads and writes. A block of user memory may be protected from table writes if the WRTn configuration bit is '0'. The EBTRn bits control table reads. For a block of user memory with the EBTRn bit set to '0', a table read instruction that executes from within that block is allowed to read. A table read instruction that executes from a location outside of that block is not allowed to read and will result in reading '0's. Figures 24-4 through 24-6 illustrate table write and table read protection.

Note: Code protection bits may only be written to a '0' from a '1' state. It is not possible to write a '1' to a bit in the '0' state. Code protection bits are only set to '1' by a full chip erase or block erase function. The full chip erase and block erase functions can only be initiated via ICSP or an external programmer.

FIGURE 24-4: TABLE WRITE (WRTn) DISALLOWED



Results: All table writes disabled to Block n whenever WRTn = 0.

LFSR — Load FSR

Syntax: [*label*] LFSR f,k

Operands: $0 \leq f \leq 2$
$0 \leq k \leq 4095$

Operation: k → FSRf

Status Affected: None

Encoding:

1110	1110	00ff	k_{11}kkk
1111	0000	$k_7$kkk	kkkk

Description: The 12-bit literal ‘k’ is loaded into the file select register pointed to by ‘f’.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’ MSB	Process Data	Write literal ‘k’ MSB to FSRfH
Decode	Read literal ‘k’ LSB	Process Data	Write literal ‘k’ to FSRfL

Example: `LFSR 2, 0x3AB`

After Instruction

FSR2H	=	0x03
FSR2L	=	0xAB

MOVF — Move f

Syntax: [*label*] MOVF f [,d [,a]]

Operands: $0 \leq f \leq 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: f → dest

Status Affected: N, Z

Encoding:

0101	00da	ffff	ffff

Description: The contents of register ‘f’ are moved to a destination dependent upon the status of ‘d’. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is placed back in register ‘f’ (default). Location ‘f’ can be anywhere in the 256-byte bank. If ‘a’ is ‘0’, the Access Bank will be selected, overriding the BSR value. If ‘a’ = 1, then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `MOVF REG, 0, 0`

Before Instruction

REG	=	0x22
W	=	0xFF

After Instruction

REG	=	0x22
W	=	0x22

RLNCF — Rotate Left f (no carry)

Syntax: [*label*] RLNCF f [,d [,a]]

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

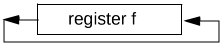
Operation: (f<n>) → dest<n+1>,
(f<7>) → dest<0>

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 will be selected, overriding the BSR value. If ‘a’ is ‘`1`’, then the bank will be selected as per the BSR value (default).



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: [*label*] RRCF f [,d [,a]]

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

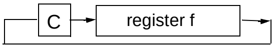
Operation: (f<n>) → dest<n-1>,
(f<0>) → C,
(C) → dest<7>

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 will be selected, overriding the BSR value. If ‘a’ is ‘`1`’, then the bank will be selected as per the BSR value (default).



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

FIGURE 27-3: PIC18F6585/8585/6680/8680 VOLTAGE-FREQUENCY GRAPH (EXTENDED)

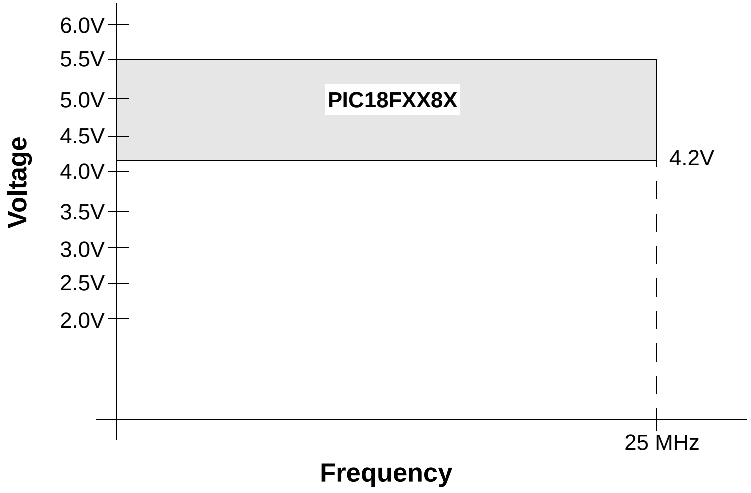


FIGURE 27-4: LOW-VOLTAGE DETECT CHARACTERISTICS

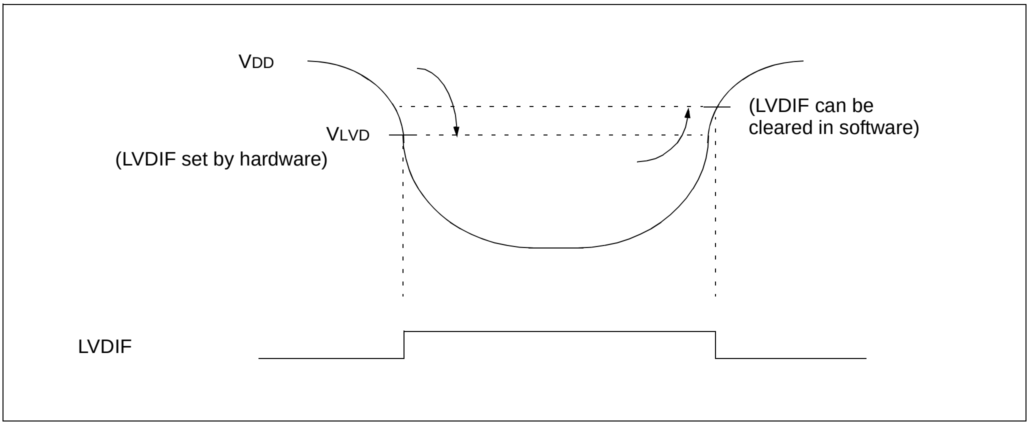


TABLE 27-3: LOW-VOLTAGE DETECT CHARACTERISTICS

				Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
D420		LVD Voltage on VDD transition high to low	LVV = 0000	—	—	—	V	
			LVV = 0001	1.96	2.06	2.16	V	
			LVV = 0010	2.16	2.27	2.38	V	
			LVV = 0011	2.35	2.47	2.59	V	
			LVV = 0100	2.46	2.58	2.71	V	
			LVV = 0101	2.64	2.78	2.92	V	
			LVV = 0110	2.75	2.89	3.03	V	
			LVV = 0111	2.95	3.1	3.26	V	
			LVV = 1000	3.24	3.41	3.58	V	
			LVV = 1001	3.43	3.61	3.79	V	
			LVV = 1010	3.53	3.72	3.91	V	
			LVV = 1011	3.72	3.92	4.12	V	
			LVV = 1100	3.92	4.13	4.33	V	
			LVV = 1101	4.11	4.33	4.55	V	
			LVV = 1110	4.41	4.64	4.87	V	
D423	VBG	Band Gap Reference Voltage Value		—	1.22	—	V	

† Production tested at TAMB = 25°C. Specifications over temperature limits ensured by characterization.