Welcome to E-XFL.COM

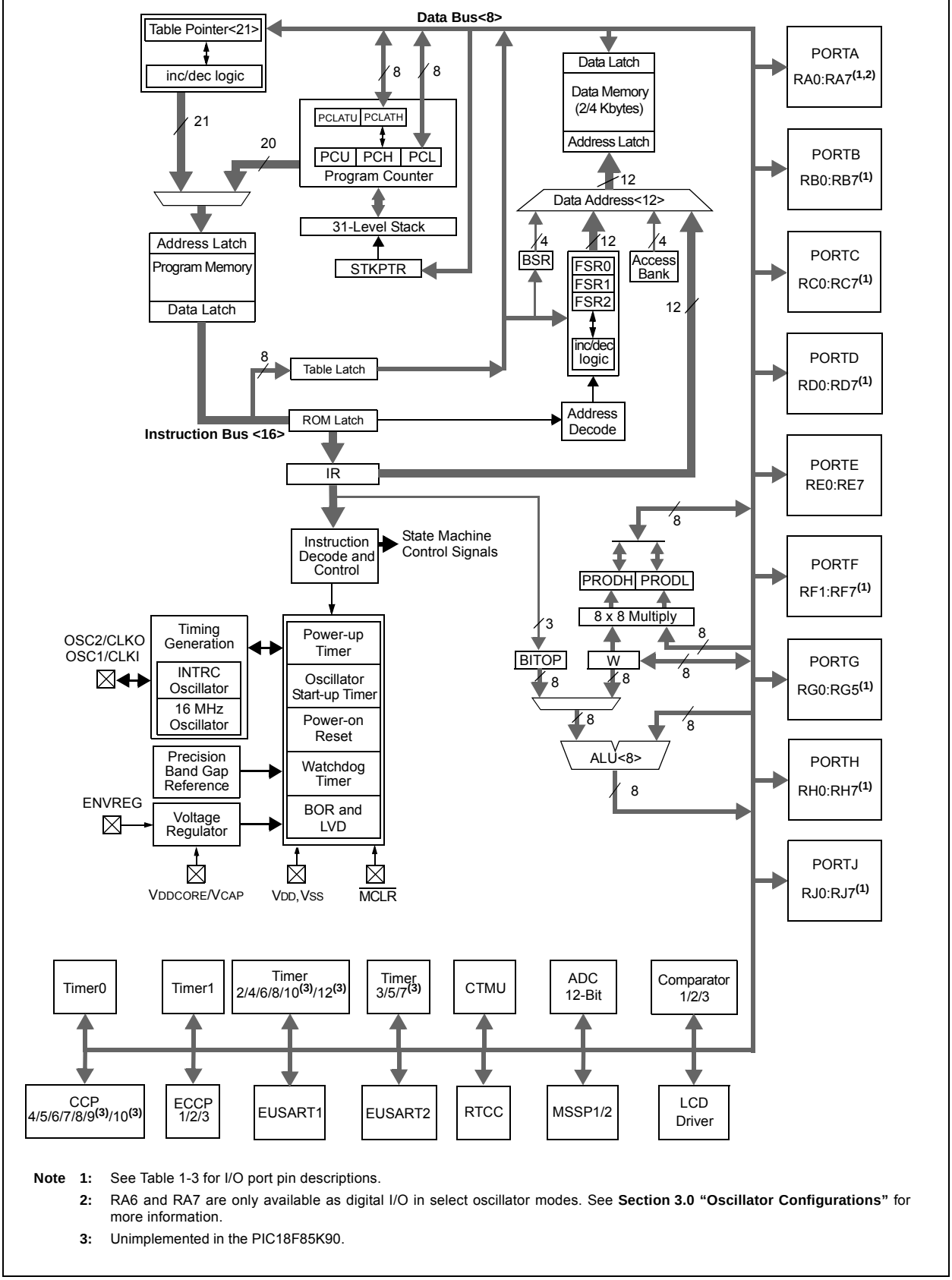
What is "Embedded - Microcontrollers"?

"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, LCD, POR, PWM, WDT
Number of I/O	69
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f87k90t-i-pt

FIGURE 1-2: PIC18F8XK90 (80-PIN) BLOCK DIAGRAM



Note 1: See Table 1-3 for I/O port pin descriptions.

2: RA6 and RA7 are only available as digital I/O in select oscillator modes. See **Section 3.0 "Oscillator Configurations"** for more information.

3: Unimplemented in the PIC18F85K90.

6.0 MEMORY ORGANIZATION

PIC18F87K90 family devices have these types of memory:

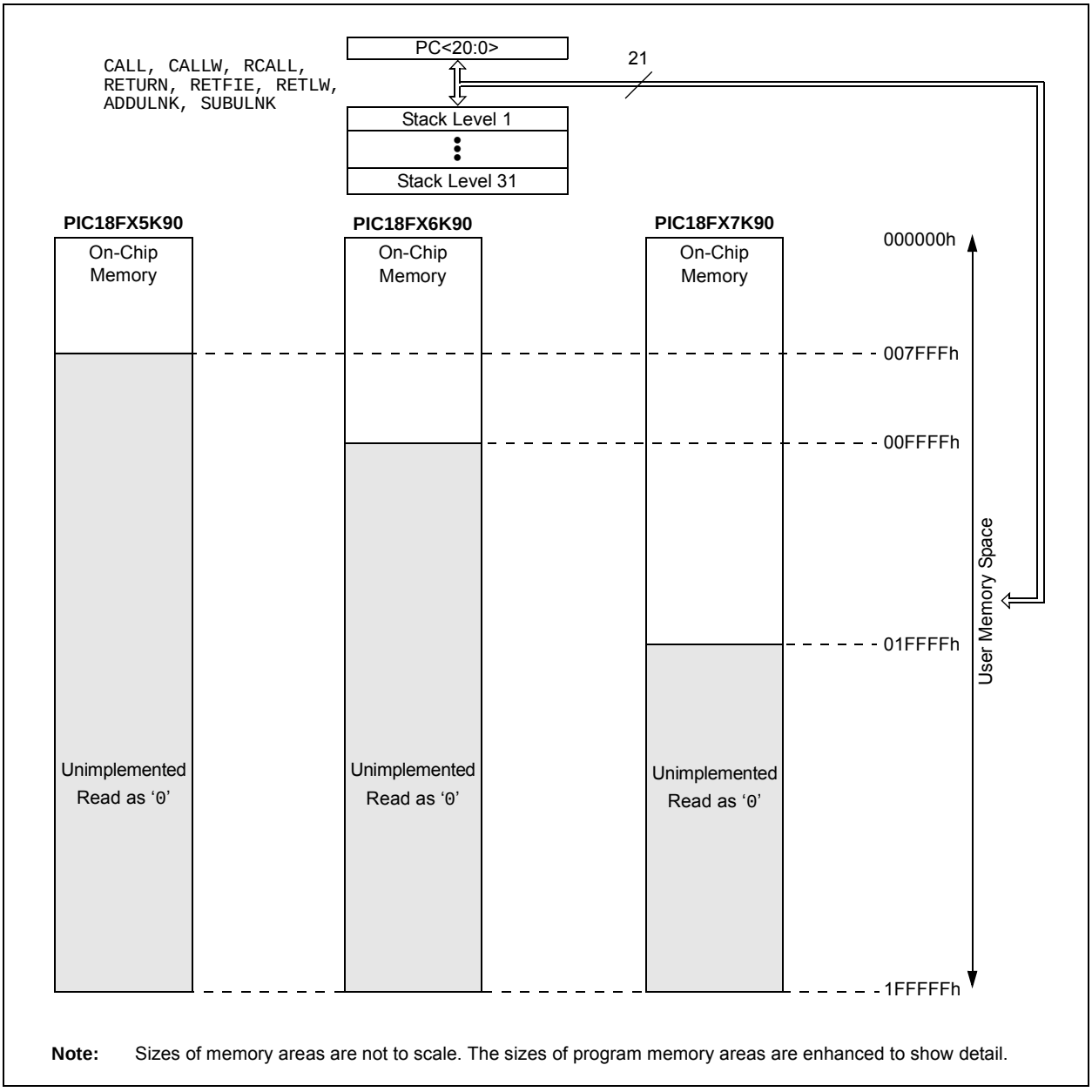
- Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate busses. This enables concurrent access of the two memory spaces.

The data EEPROM, for practical purposes, can be regarded as a peripheral device because it is addressed and accessed through a set of control registers.

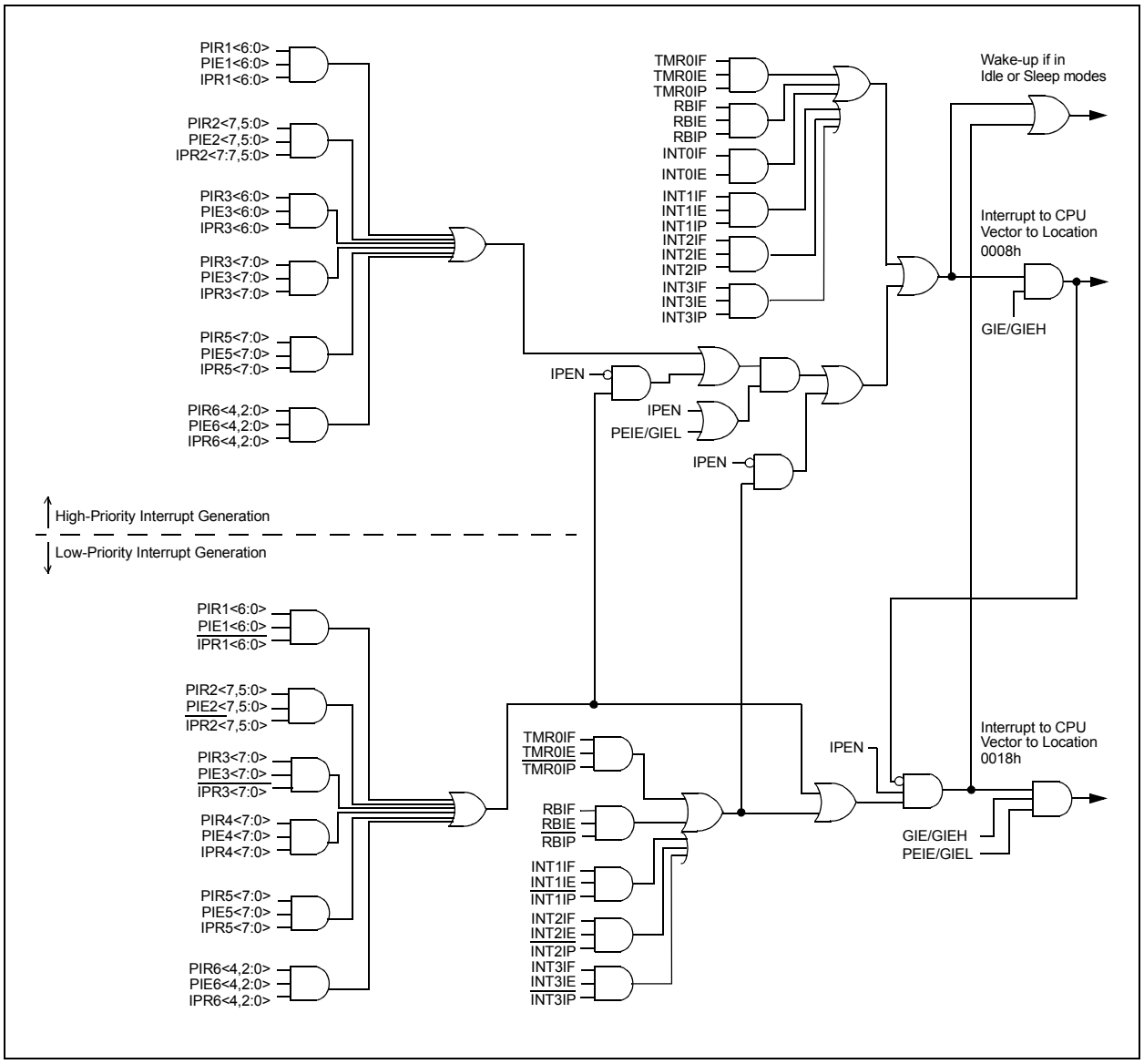
Additional detailed information on the operation of the Flash program memory is provided in **Section 7.0 "Flash Program Memory"**. The data EEPROM is discussed separately in **Section 8.0 "Data EEPROM Memory"**.

FIGURE 6-1: MEMORY MAPS FOR PIC18F87K90 FAMILY DEVICES



Note: Sizes of memory areas are not to scale. The sizes of program memory areas are enhanced to show detail.

FIGURE 10-1: PIC18F87K90 FAMILY INTERRUPT LOGIC



10.5 RCON Register

The RCON register contains the bits used to determine the cause of the last Reset, or wake-up from Idle or Sleep modes. RCON also contains the bit that enables interrupt priorities (IPEN).

REGISTER 10-22: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	SBOREN	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

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

bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6 **SBOREN:** BOR Software Enable bit
If BOREN<1:0> = 01:
1 = BOR is enabled
0 = BOR is disabled
If BOREN<1:0> = 00, 10 or 11:
Bit is disabled and read as '0'.

bit 5 **$\overline{CM}$:** Configuration Mismatch Flag bit
1 = A Configuration Mismatch Reset has not occurred
0 = A Configuration Mismatch Reset has occurred (must be subsequently set in software)

bit 4 **$\overline{RI}$:** RESET Instruction Flag bit
For details of bit operation, see Register 5-1.

bit 3 **$\overline{TO}$:** Watchdog Timer Time-out Flag bit
For details of bit operation, see Register 5-1.

bit 2 **$\overline{PD}$:** Power-Down Detection Flag bit
For details of bit operation, see Register 5-1.

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
For details of bit operation, see Register 5-1.

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
For details of bit operation, see Register 5-1.

REGISTER 11-3: ODCON3: PERIPHERAL OPEN-DRAIN CONTROL REGISTER 3

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
U2OD	U1OD	—	—	—	—	—	CTMUDS
bit 7							bit 0

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

bit 7 **U2OD:** EUSART2 Open-Drain Output Enable bit
1 = Open-drain capability is enabled
0 = Open-drain capability is disabled

bit 6 **U1OD:** EUSART1 Open-Drain Output Enable bit
1 = Open-drain capability is enabled
0 = Open-drain capability is disabled

bit 5-1 **Unimplemented:** Read as '0'

bit 0 **CTMUDS:** CTMU Pulse Delay Enable bit
1 = Pulse delay input for CTMU is enabled on pin, RF1
0 = Pulse delay input for CTMU is disabled on pin, RF1

11.1.4 ANALOG AND DIGITAL PORTS

Many of the ports multiplex analog and digital functionality, providing a lot of flexibility for hardware designers. PIC18F87K90 family devices can make any analog pin, analog or digital, depending on an application's needs. The ports' analog/digital functionality is controlled by the registers: ANCON0, ANCON1 and ANCON2.

Setting these registers makes the corresponding pins analog and clearing the registers makes the ports digital. For details on these registers, see **Section 23.0 "12-Bit Analog-to-Digital Converter (A/D) Module"**.

REGISTER 16-1: TxCON: TIMER4/6/8/10/12 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TxOUTPS3	TxOUTPS2	TxOUTPS1	TxOUTPS0	TMRxON	TxCKPS1	TxCKPS0
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-3 **TxOUTPS<3:0>:** Timerx Output Postscale Select bits

0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMRxON**: Timerx On bit

1 = Timerx is on
0 = Timerx is off

bit 1-0 **TxCKPS<1:0>:** Timerx Clock Prescale Select bits

00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

REGISTER 19-1: CCPxCON: ENHANCED CAPTURE/COMPARE/PWM x CONTROL

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

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

bit 7-6 **PxM<1:0>:** Enhanced PWM Output Configuration bits

If CCPxM<3:2> = 00, 01, 10:

xx = PxA is assigned as a capture/compare input/output; PxB, PxC and PxD are assigned as PORT pins

If CCPxM<3:2> = 11:

00 = Single output: PxA, PxB, PxC and PxD are controlled by steering (see **Section 19.4.7 "Pulse Steering Mode"**)

01 = Full-bridge output forward: PxD is modulated; PxA is active; PxB, PxC are inactive

10 = Half-bridge output: PxA, PxB are modulated with dead-band control; PxC and PxD are assigned as PORT pins

11 = Full-bridge output reverse: PxB is modulated; PxC is active; PxA and PxD are inactive

bit 5-4 **DCxB<1:0>**: PWM Duty Cycle Bit 1 and Bit 0

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPRxL.

bit 3-0 **CCPxM<3:0>**: ECCPx Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCPx module)

0001 = Reserved

0010 = Compare mode: toggle output on match

0011 = Capture mode

0100 = Capture mode: every falling edge

0101 = Capture mode: every rising edge

0110 = Capture mode: every fourth rising edge

0111 = Capture mode: every 16th rising edge

1000 = Compare mode: initialize the ECCPx pin low; set the output on a compare match (set CCPxIF)

1001 = Compare mode: initialize the ECCPx pin high; clear the output on a compare match (set CCPxIF)

1010 = Compare mode: generate a software interrupt only; ECCPx pin reverts to an I/O state

1011 = Compare mode: trigger special event (ECCPx resets TMR1 or TMR3, starts A/D conversion, sets CCxIF bit)

1100 = PWM mode: PxA and PxC are active-high; PxB and PxD are active-high

1101 = PWM mode: PxA and PxC are active-high; PxB and PxD are active-low

1110 = PWM mode: PxA and PxC are active-low; PxB and PxD are active-high

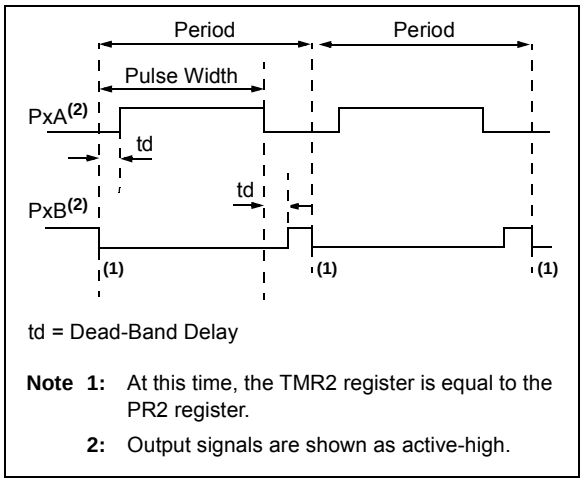
1111 = PWM mode: PxA and PxC are active-low; PxB and PxD are active-low

19.4.6 PROGRAMMABLE DEAD-BAND DELAY MODE

In half-bridge applications, where all power switches are modulated at the PWM frequency, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on and the other turned off), both switches may be on for a short period until one switch completely turns off. During this brief interval, a very high current (shoot-through current) will flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In Half-Bridge mode, a digitally programmable dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. For an illustration, see Figure 19-14. The lower seven bits of the associated ECCPxDEL register (Register 19-4) set the delay period in terms of microcontroller instruction cycles (TCY or 4 TOSC).

FIGURE 19-14: EXAMPLE OF HALF-BRIDGE PWM OUTPUT



td = Dead-Band Delay

Note 1: At this time, the TMR2 register is equal to the PR2 register.

2: Output signals are shown as active-high.

FIGURE 19-15: EXAMPLE OF HALF-BRIDGE APPLICATIONS

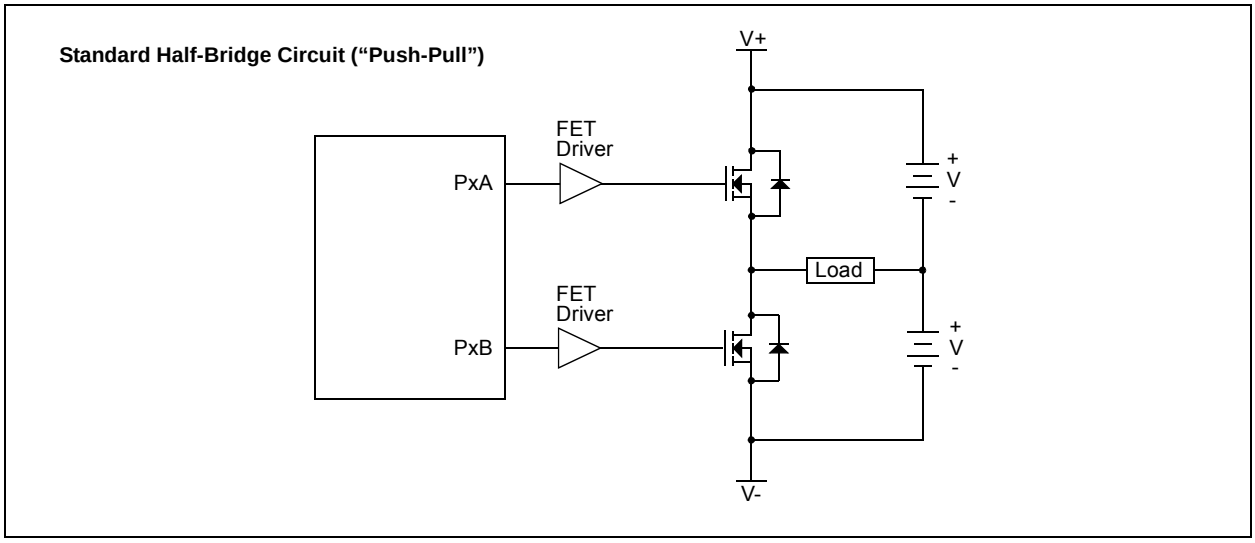
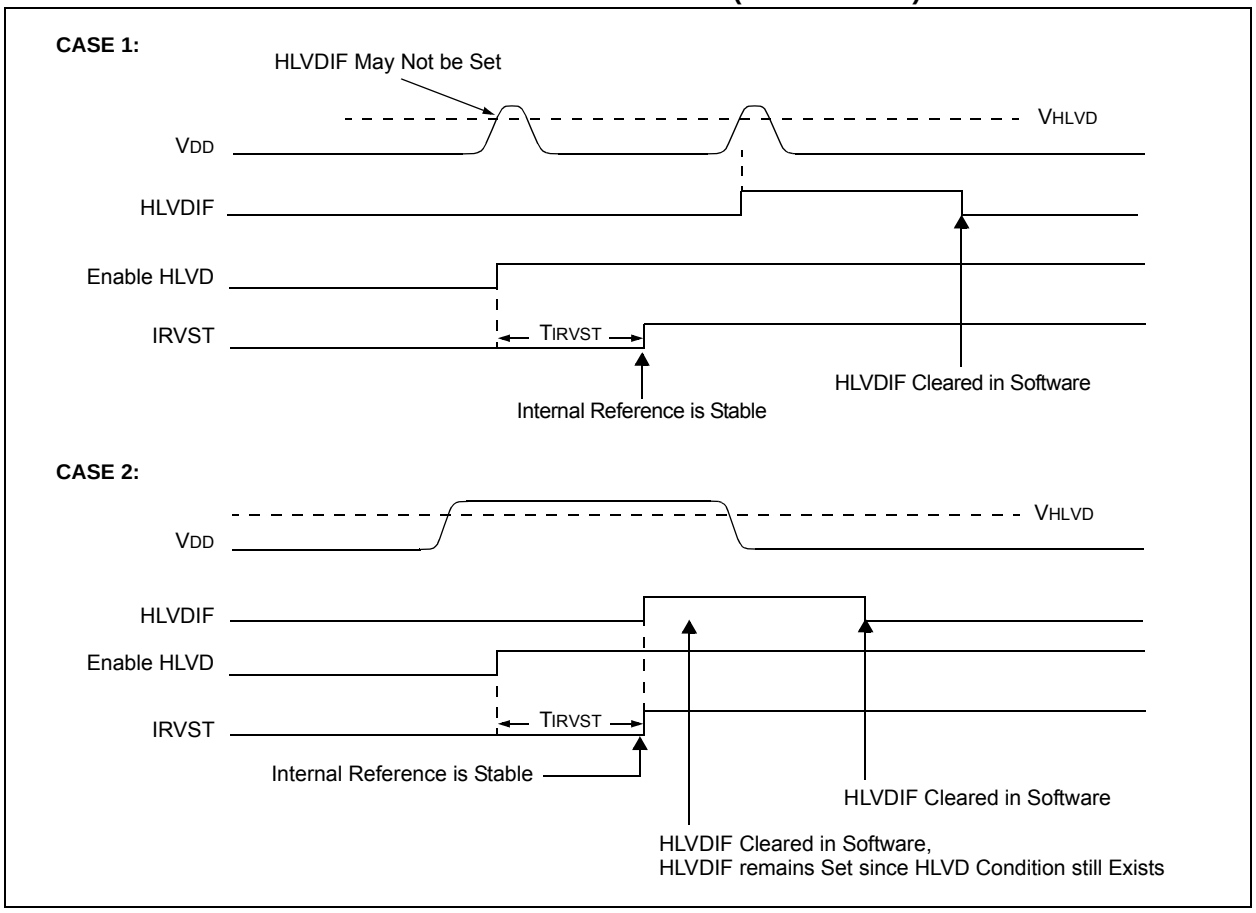


FIGURE 26-3: HIGH-VOLTAGE DETECT OPERATION (VDIRMAG = 1)

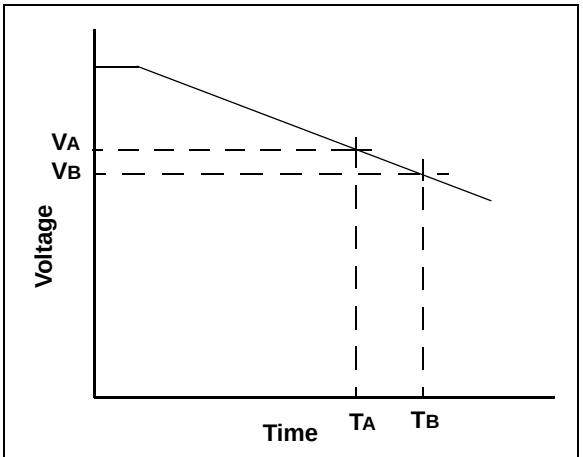


26.5 Applications

In many applications, it is desirable to detect a drop below, or rise above, a particular voltage threshold. For example, the HLVD module could be periodically enabled to detect Universal Serial Bus (USB) attach or detach. This assumes the device is powered by a lower voltage source than the USB when detached. An attach would indicate a High-Voltage Detect from, for example, 3.3V to 5V (the voltage on USB) and vice versa for a detach. This feature could save a design a few extra components and an attach signal (input pin).

For general battery applications, Figure 26-4 shows a possible voltage curve. Over time, the device voltage decreases. When the device voltage reaches voltage, VA, the HLVD logic generates an interrupt at time, TA. The interrupt could cause the execution of an ISR (Interrupt Service Routine), which would allow the application to perform “housekeeping tasks” and a controlled shutdown before the device voltage exits the valid operating range at TB. This would give the application a time window, represented by the difference between TA and TB, to safely exit.

FIGURE 26-4: TYPICAL LOW-VOLTAGE DETECT APPLICATION



Legend: VA = HLVD trip point
VB = Minimum valid device operating voltage

NOTES:

REGISTER 28-3: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	BORPWR1(1)	BORPWR0(1)	BORV1(1)	BORV0(1)	BOREN1(2)	BOREN0(2)	$\overline{\text{PWRTEN}}$(2)
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as ‘0’

bit 6-5 **BORPWR<1:0>:** BORMV Power Level bits(1)
- 11 = ZPBORVMV instead of BORMV is selected
- 10 = BORMV is set to high-power level
- 01 = BORMV is set to medium-power level
- 00 = BORMV is set to low-power level

bit 4-3 **BORV<1:0>:** Brown-out Reset Voltage bits(1)
- 11 = VBORMV is set to 1.8V
- 10 = VBORMV is set to 2.0V
- 01 = VBORMV is set to 2.7V
- 00 = VBORMV is set to 3.0V

bit 2-1 **BOREN<1:0>:** Brown-out Reset Enable bits(2)
- 11 = Brown-out Reset is enabled in hardware only (SBOREN is disabled)
- 10 = Brown-out Reset is enabled in hardware only and disabled in Sleep mode (SBOREN is disabled)
- 01 = Brown-out Reset is enabled and controlled by software (SBOREN is enabled)
- 00 = Brown-out Reset is disabled in hardware and software

bit 0 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit(2)
- 1 = PWRT is disabled
- 0 = PWRT is enabled

Note 1: For the specifications, see **Section 31.1 “DC Characteristics: Supply Voltage PIC18F87K90 Family (Industrial/Extended)”**.

2: The Power-up Timer is decoupled from Brown-out Reset, allowing these features to be independently controlled.

REGISTER 28-13: CONFIG7H: CONFIGURATION REGISTER 7 HIGH (BYTE ADDRESS 30000Dh)(1)

U-0	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB	—	—	—	—	—	—
bit 7							bit 0

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

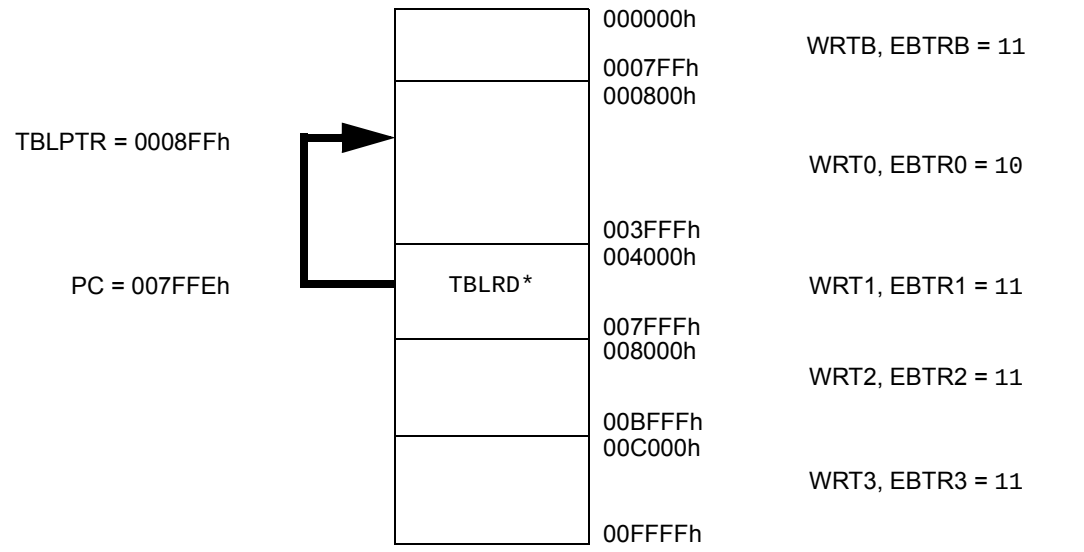
bit 7 **Unimplemented:** Read as '0'

bit 6 **EBTRB:** Boot Block Table Read Protection bit
1 = Boot block is not protected from table reads executed in other blocks
0 = Boot block is protected from table reads executed in other blocks

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

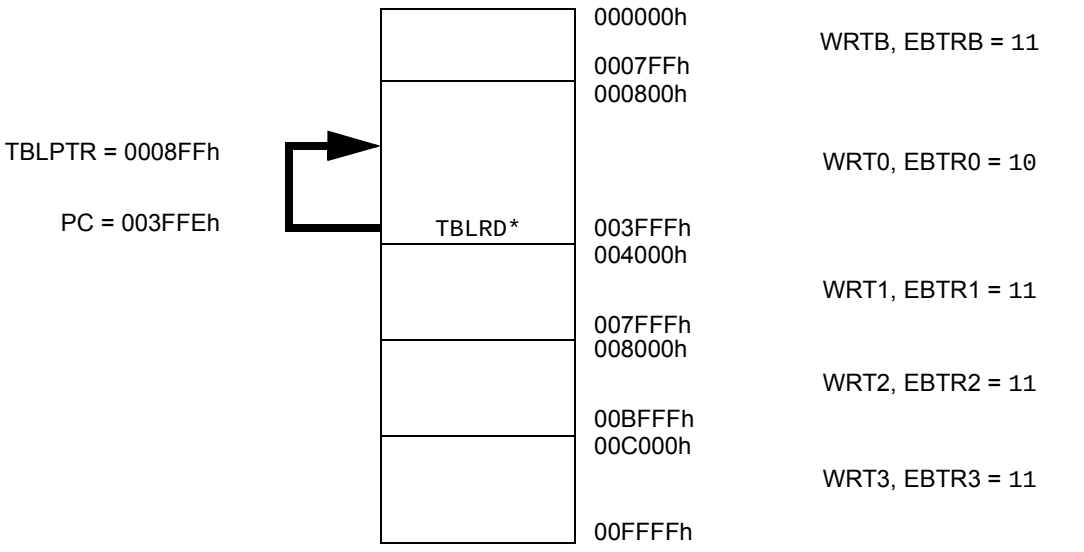
Note 1: For the memory size of the blocks, refer to Figure 28-6.

FIGURE 28-8: EXTERNAL BLOCK TABLE READ (EBTRn) DISALLOWED



Results: All table reads from external blocks to Blockn are disabled whenever EBTRn = 0.
The TABLAT register returns a value of '0'.

FIGURE 28-9: EXTERNAL BLOCK TABLE READ (EBTRn) ALLOWED



Results: Table reads are permitted within Blockn, even when EBTRBn = 0.
The TABLAT register returns the value of the data at the location, TBLPTR.

29.1.1 STANDARD INSTRUCTION SET

ADDLW **ADD Literal to W**

Syntax: ADDLW k

Operands: $0 \le k \le 255$

Operation: $(W) + k \rightarrow W$

Status Affected: N, OV, C, DC, Z

Encoding:

0000	1111	kkkk	kkkk

Description: The contents of W are added to the 8-bit literal 'k' and the result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: ADDLW 15h

Before Instruction
W = 10h
After Instruction
W = 25h

ADDWF **ADD W to f**

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

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

Operation: $(W) + (f) \rightarrow$ dest

Status Affected: N, OV, C, DC, Z

Encoding:

0010	01da	ffff	ffff

Description: Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: ADDWF REG, 0, 0

Before Instruction
W = 17h
REG = 0C2h
After Instruction
W = 0D9h
REG = 0C2h

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

CLRF **Clear f**

Syntax: CLRF f {,a}

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

Operation: 000h → f,
1 → Z

Status Affected: Z

Encoding:

0110	101a	ffff	ffff

Description: Clears the contents of the specified register.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: CLRF FLAG_REG,1

Before Instruction
FLAG_REG = 5Ah
After Instruction
FLAG_REG = 00h

CLRWDT **Clear Watchdog Timer**

Syntax: CLRWDT

Operands: None

Operation: 000h → WDT,
000h → WDT postscaler,
1 → $\overline{TO}$,
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0000	0100

Description: CLRWDT instruction resets the Watchdog Timer. It also resets the post-scaler of the WDT. Status bits, $\overline{TO}$ and $\overline{PD}$, are set.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	No operation

Example: CLRWDT

Before Instruction
WDT Counter = ?
After Instruction
WDT Counter = 00h
WDT Postscaler = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

SUBWFB **Subtract W from f with Borrow**

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

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

Operation: (f) – (W) – ($\overline{C}$) → dest

Status Affected: N, OV, C, DC, Z

Encoding:

0101	10da	ffff	ffff

Description: Subtract W and the Carry flag (borrow) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 1:
```
SUBWFB  REG, 1, 0
```
Before Instruction
```
REG  =  19h   (0001 1001)
W    =  0Dh   (0000 1101)
C    =  1
```
After Instruction
```
REG  =  0Ch   (0000 1011)
W    =  0Dh   (0000 1101)
C    =  1
Z    =  0
N    =  0     ; result is positive
```
Example 2:
```
SUBWFB  REG, 0, 0
```
Before Instruction
```
REG  =  1Bh   (0001 1011)
W    =  1Ah   (0001 1010)
C    =  0
```
After Instruction
```
REG  =  1Bh   (0001 1011)
W    =  00h
C    =  1
Z    =  1     ; result is zero
N    =  0
```
Example 3:
```
SUBWFB  REG, 1, 0
```
Before Instruction
```
REG  =  03h   (0000 0011)
W    =  0Eh   (0000 1101)
C    =  1
```
After Instruction
```
REG  =  F5h   (1111 0100)
                ; [2's comp]
W    =  0Eh   (0000 1101)
C    =  0
Z    =  0
N    =  1     ; result is negative
```

SWAPF **Swap f**

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

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

Operation: (f<3:0>) → dest<7:4>,
(f<7:4>) → dest<3:0>

Status Affected: None

Encoding:

0011	10da	ffff	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in register 'f'.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:
```
SWAPF   REG, 1, 0
```
Before Instruction
```
REG  =  53h
```
After Instruction
```
REG  =  35h
```

30.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

30.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

FIGURE 31-14: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)

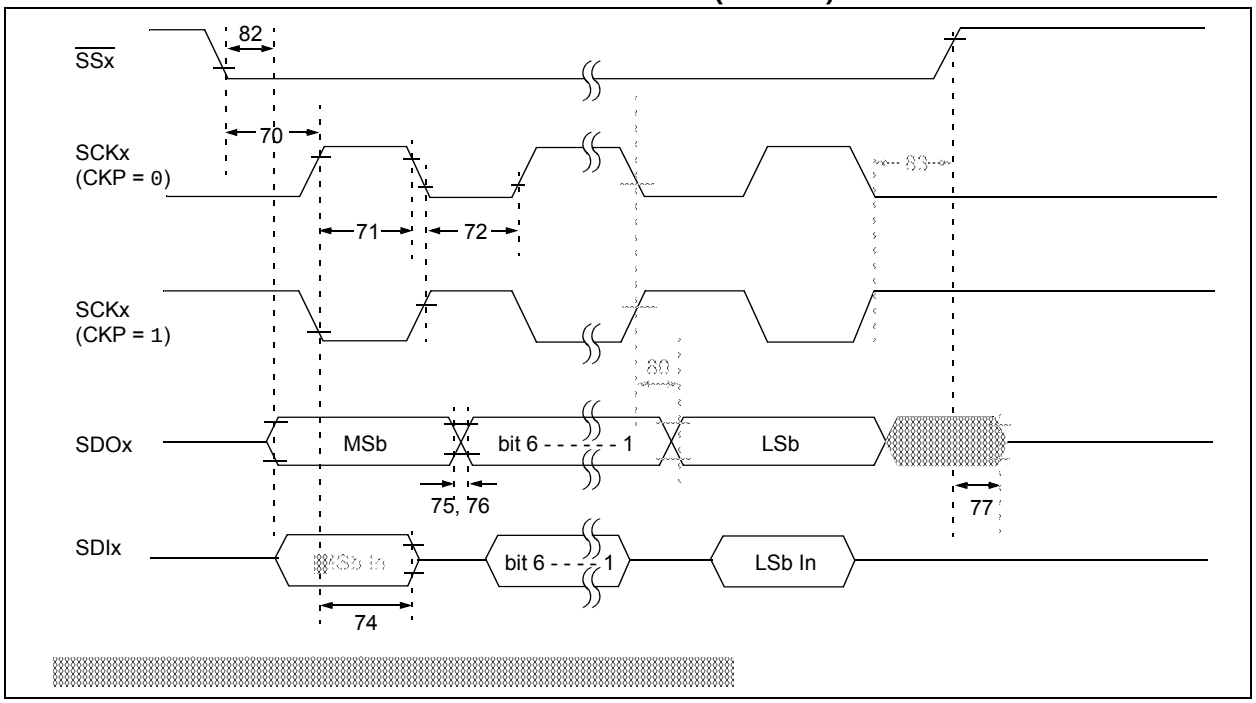


TABLE 31-17: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SSx}$ ↓ to SCKx ↓ or SCKx ↑ Input		3 TCY	—	ns	
70A	TssL2WB	$\overline{SSx}$ to write to SSPxBUF		3 TCY	—	ns	
71	TscH	SCKx Input High Time (Slave mode)	Continuous	1.25 TCY + 30	—	ns	
71A			Single Byte	40	—	ns	**(Note 1)**
72	TscL	SCKx Input Low Time (Slave mode)	Continuous	1.25 TCY + 30	—	ns	
72A			Single Byte	40	—	ns	**(Note 1)**
73A	TB2B	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2		1.5 TCY + 40	—	ns	**(Note 2)**
74	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge		40	—	ns	
75	TdoR	SDOx Data Output Rise Time		—	25	ns	
76	TdoF	SDOx Data Output Fall Time		—	25	ns	
77	TssH2doZ	$\overline{SSx}$ ↑ to SDOx Output High-Impedance		10	50	ns	
78	TscR	SCKx Output Rise Time (Master mode)		—	25	ns	
79	TscF	SCKx Output Fall Time (Master mode)		—	25	ns	
80	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge		—	50	ns	
82	TssL2doV	SDOx Data Output Valid after $\overline{SSx}$ ↓ Edge		—	50	ns	
83	TscH2ssH, TscL2ssH	$\overline{SSx}$ ↑ after SCKx Edge		1.5 TCY + 40	—	ns	

Note 1: Requires the use of Parameter #73A.

2: Only if Parameter #71A and #72A are used.

S

SCKx 303
SDIx 303
SDOx 303
SEC_IDLE Mode 59
SEC_RUN Mode 54
Secondary Oscillator (SOSC) 45
Selective Peripheral Module Control 60
Serial Clock, SCKx 303
Serial Data In (SDIx) 303
Serial Data Out (SDOx) 303
Serial Peripheral Interface. *See* SPI Mode.
SETF 485
Shoot-Through Current 267
Slave Select (SSx) 303
SLEEP 486
Software Simulator (MPLAB SIM) 503
Special Event Trigger. *See* Compare (CCP Module).
Special Event Trigger. *See* Compare (ECCP Mode).
Specifications
    AC (Timing) Characteristics
        Temperature and Voltage 524
    Capture/Compare/PWM Requirements
        (ECCP1, ECCP2) 532
    CLKO and I/O Requirements 527
    Comparator 522
    EUSART/AUSART Synchronous Receive
        Requirements 541
    EUSART/AUSART Synchronous Transmission
        Requirements 541
    Example SPI Mode Requirements (Master Mode,
        CKE = 0) 533
    Example SPI Mode Requirements (Master Mode,
        CKE = 1) 534
    Example SPI Mode Requirements (Slave Mode,
        CKE = 0) 535
    Example SPI Slave Mode Requirements
        (CKE = 1) 536
    External Clock Requirements 525
    High/Low-Voltage Detect Characteristics 530
    I^2C Bus Data Requirements (Slave Mode) 538
    I^2C Bus Start/Stop Bits Requirements
        (Slave Mode) 537
    Internal RC Accuracy (INTOSC) 526
    Internal Voltage Regulator 522
    Memory Programming Requirements 521
    MSSP I^2C Bus Data Requirements 540
    MSSP I^2C Bus Start/Stop Bits Requirements 539
    PLL Clock Timing 526
    Timer0 and Timer1 External Clock
        Requirements 531
    Ultra Low-Power Wake-up 541
    Voltage Reference 522
SSPOV 339
SSPOV Status Flag 339
SSPxSTAT Register
    R/W Bit 318, 321
SSx 303
Stack Full/Underflow Resets 89
SUBFSR 497
SUBFWB 486
SUBLW 487
SUBULNK 497
SUBWF 487
SUBWFB 488
SWAPF 488

T

Table Pointer Operations (table) 114
Table Reads/Table Writes 89
TBLRD 489
TBLWT 490
Timer0 183
    Associated Registers 185
    Operation 184
    Overflow Interrupt 185
    Prescaler 185
        Switching Assignment 185
    Prescaler Assignment (PSA Bit) 185
    Prescaler Select (T0PS2:T0PS0 Bits) 185
    Reads and Writes in 16-Bit Mode 184
    Source Edge Select (T0SE Bit) 184
    Source Select (T0CS Bit) 184
Timer1 187
    16-Bit Read/Write Mode 191
    Associated Registers 197
    Clock Source Selection 189
    Gate 193
    Interrupt 192
    Operation 189
    Oscillator 187
    Resetting, Using the ECCP Special
        Event Trigger 193
    SOSC Oscillator 191
        Layout Considerations 192
        Use as a Clock Source 192
    TMR1H Register 187
    TMR1L Register 187
Timer2 199
    Associated Registers 200
    Interrupt 200
    Operation 199
    Output 200
    PR2 Register 248
    TMR12 to PR12 Match Flag (TMR12IF Bit) 138
    TMR2 to PR2 Match Interrupt 248
Timer3/5/7 201
    16-Bit Read/Write Mode 206
    Associated Registers 211
    Gates 206
    Operation 205
    Overflow Interrupt 201, 210
    SOSC Oscillator 201
        Use as a Clock Source 206
    Special Event Trigger (ECCP) 210
    TMRxH Register 201
    TMRxL Register 201
Timer4/6/8/10/12 213
    Associated Registers 215
    Interrupt 215
    Operation 213
    Output 215
    Outputs, PWM Time Base for ECCP 215
    Postscaler. *See* Postscaler, Timer4/6/8/10/12.
    Prescaler. *See* Prescaler, Timer4/6/8/10/12.
    PRx Register 213
    TMRx Register 213
    TMRx to PRx Match Interrupt 213, 215

Timing Diagrams
A/D Conversion ... 543
Automatic Baud Rate Calculation ... 358
Auto-Wake-up Bit (WUE) During Normal Operation ... 365
Auto-Wake-up Bit (WUE) During Sleep ... 365
Baud Rate Generator with Clock Arbitration ... 336
BRG Overflow Sequence ... 358
BRG Reset Due to SDAx Arbitration During Start Condition ... 345
Brown-out Reset (BOR) ... 528
Bus Collision During a Repeated Start Condition (Case 1) ... 346
Bus Collision During a Repeated Start Condition (Case 2) ... 346
Bus Collision During a Start Condition (SCLx = 0) ... 345
Bus Collision During a Stop Condition (Case 1) ... 347
Bus Collision During a Stop Condition (Case 2) ... 347
Bus Collision During Start Condition (SDAx Only) ... 344
Bus Collision for Transmit and Acknowledge ... 343
Capture/Compare/PWM (ECCP1, ECCP2) ... 532
CLKO and I/O ... 527
Clock Synchronization ... 329
Clock/Instruction Cycle ... 90
EUSART Asynchronous Reception ... 363
EUSART Asynchronous Transmission ... 360
EUSART Asynchronous Transmission (Back-to-Back) ... 360
EUSART Synchronous Master Transmission ... 367
EUSART Synchronous Transmission (Master/Slave) ... 541
EUSART/AUSART Synchronous Receive (Master/Slave) ... 541
Example SPI Master Mode (CKE = 0) ... 533
Example SPI Master Mode (CKE = 1) ... 534
Example SPI Slave Mode (CKE = 0) ... 535
Example SPI Slave Mode (CKE = 1) ... 536
External Clock ... 525
Fail-Safe Clock Monitor (FSCM) ... 446
First Start Bit Timing ... 337
Full-Bridge PWM Output ... 262
Half-Bridge PWM Output ... 260, 267
High/Low-Voltage Detect Characteristics ... 530
High-Voltage Detect Operation (VDIRMAG = 1) ... 404
I2C Acknowledge Sequence ... 342
I2C Bus Data ... 538
I2C Bus Start/Stop Bits ... 537
I2C Master Mode (7 or 10-Bit Transmission) ... 340
I2C Master Mode (7-Bit Reception) ... 341
I2C Slave Mode (10-Bit Reception, SEN = 0, ADMSK = 01001) ... 325
I2C Slave Mode (10-Bit Reception, SEN = 0) ... 326
I2C Slave Mode (10-Bit Reception, SEN = 1) ... 331
I2C Slave Mode (10-Bit Transmission) ... 327
I2C Slave Mode (7-Bit Reception, SEN = 0, ADMSK = 01011) ... 323
I2C Slave Mode (7-Bit Reception, SEN = 0) ... 322
I2C Slave Mode (7-Bit Reception, SEN = 1) ... 330
I2C Slave Mode (7-Bit Transmission) ... 324
I2C Slave Mode General Call Address Sequence (7 or 10-Bit Addressing Mode) ... 332
I2C Stop Condition Receive or Transmit Mode ... 342
LCD Interrupt Timing in Quarter Duty Cycle Drive ... 298
LCD Reference Ladder Power Mode Switching ... 283
LCD Sleep Entry/Exit When SLPEN = 1 or CS = 00 ... 299
Low-Voltage Detect Operation (VDIRMAG = 0) ... 403
MSSP I2C Bus Data ... 539
MSSP I2C Bus Start/Stop Bits ... 539
PWM Auto-Shutdown with Auto-Restart Enabled, PxRSEN = 1 ... 266
PWM Auto-Shutdown with Firmware Restart, PxRSEN = 0 ... 266
PWM Direction Change ... 263
PWM Direction Change at Near 100% Duty Cycle ... 264
PWM Output ... 248
PWM Output (Active-High) ... 258
PWM Output (Active-Low) ... 259
Repeated Start Condition ... 338
Reset, Watchdog Timer (WDT), Oscillator Start-up Timer (OST) and Power-up Timer (PWRT) ... 528
Send Break Character Sequence ... 366
Slave Synchronization ... 309
Slow Rise Time (MCLR Tied to VDD, VDD Rise > TPWRT) ... 73
SPI Mode (Master Mode) ... 308
SPI Mode (Slave Mode, CKE = 0) ... 310
SPI Mode (Slave Mode, CKE = 1) ... 310
Steering Event at Beginning of Instruction (STRSYNC = 1) ... 270
Steering Event at End of Instruction (STRSYNC = 0) ... 270
Synchronous Master Transmission (Through TXEN) ... 368
Synchronous Reception (Master Mode, SREN) ... 369
Time-out Sequence on Power-up (MCLR Not Tied to VDD), Case 1 ... 73
Time-out Sequence on Power-up (MCLR Not Tied to VDD), Case 2 ... 73
Time-out Sequence on Power-up (MCLR Tied to VDD, VDD Rise TPWRT) ... 72
Timer Pulse Generation ... 234
Timer0 and Timer1 External Clock ... 531
Timer1 Gate Count Enable Mode ... 193
Timer1 Gate Single Pulse Mode ... 196
Timer1 Gate Single Pulse/Toggle Combined Mode ... 197
Timer1 Gate Toggle Mode ... 195
Timer3/5/7 Gate Count Enable Mode ... 206
Timer3/5/7 Gate Single Pulse Mode ... 208
Timer3/5/7 Gate Single Pulse/Toggle Combined Mode ... 209
Timer3/5/7 Gate Toggle Mode ... 207
Transition for Entry to Idle Mode ... 59
Transition for Entry to SEC_RUN Mode ... 55
Transition for Entry to Sleep Mode ... 58
Transition for Two-Speed Start-up (INTOSC to HSPLL) ... 444
Transition for Wake from Idle to Run Mode ... 59
Transition for Wake from Sleep (HSPLL) ... 58
Transition from RC_RUN Mode to PRI_RUN Mode ... 57
Transition from SEC_RUN Mode to PRI_RUN Mode (HSPLL) ... 55
Transition to RC_RUN Mode ... 57
Type-A in 1/2 MUX, 1/2 Bias Drive ... 288