



Welcome to [E-XFL.COM](https://www.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	25MHz
Connectivity	EBI/EMI, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.75K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
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/pic18lf8720t-i-pt

PIC18F6520/8520/6620/8620/6720/8720

1.0 DEVICE OVERVIEW

This document contains device specific information for the following devices:

- PIC18F6520
- PIC18F8520
- PIC18F6620
- PIC18F8620
- PIC18F6720
- PIC18F8720

This family offers the same advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price – with the addition of high endurance Enhanced Flash program memory. The PIC18FXX20 family also provides an enhanced range of program memory options and versatile analog features that make it ideal for complex, high-performance applications.

1.1 Key Features

1.1.1 EXPANDED MEMORY

The PIC18FXX20 family introduces the widest range of on-chip, Enhanced Flash program memory available on PIC® microcontrollers – up to 128 Kbyte (or 65,536 words), the largest ever offered by Microchip. For users with more modest code requirements, the family also includes members with 32 Kbyte or 64 Kbyte.

Other memory features are:

- **Data RAM and Data EEPROM:** The PIC18FXX20 family also provides plenty of room for application data. Depending on the device, either 2048 or 3840 bytes of data RAM are available. All devices have 1024 bytes of data EEPROM for long-term retention of nonvolatile data.
- **Memory Endurance:** The Enhanced Flash cells for both program memory and data EEPROM are rated to last for many thousands of erase/write cycles – up to 100,000 for program memory and 1,000,000 for EEPROM. Data retention without refresh is conservatively estimated to be greater than 40 years.

1.1.2 EXTERNAL MEMORY INTERFACE

In the event that 128 Kbytes of program memory is inadequate for an application, the PIC18F8X20 members of the family also implement an External Memory Interface. This allows the controller's internal program counter to address a memory space of up to 2 Mbytes, permitting a level of data access that few 8-bit devices can claim.

With the addition of new operating modes, the External Memory Interface offers many new options, including:

- Operating the microcontroller entirely from external memory
- Using combinations of on-chip and external memory, up to the 2-Mbyte limit
- Using external Flash memory for reprogrammable application code, or large data tables
- Using external RAM devices for storing large amounts of variable data

1.1.3 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve.

The consistent pinout scheme used throughout the entire family also aids in migrating to the next larger device. This is true when moving between the 64-pin members, between the 80-pin members, or even jumping from 64-pin to 80-pin devices.

1.1.4 OTHER SPECIAL FEATURES

- **Communications:** The PIC18FXX20 family incorporates a range of serial communications peripherals, including 2 independent USARTs and a Master SSP module, capable of both SPI and I²C (Master and Slave) modes of operation. For PIC18F8X20 devices, one of the general purpose I/O ports can be reconfigured as an 8-bit Parallel Slave Port for direct processor-to-processor communications.
- **CCP Modules:** All devices in the family incorporate five Capture/Compare/PWM modules to maximize flexibility in control applications. Up to four different time bases may be used to perform several different operations at once.
- **Analog Features:** All devices in the family feature 10-bit A/D converters, with up to 16 input channels, as well as the ability to perform conversions during Sleep mode. Also included are dual analog comparators with programmable input and output configuration, a programmable Low-Voltage Detect module and a programmable Brown-out Reset module.
- **Self-programmability:** These devices can write to their own program memory spaces under internal software control. By using a bootloader routine located in the protected Boot Block at the top of program memory, it becomes possible to create an application that can update itself in the field.

PIC18F6520/8520/6620/8620/6720/8720

3.0 RESET

The PIC18FXX20 devices differentiate between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during Sleep
- Watchdog Timer (WDT) Reset (during normal operation)
- Programmable Brown-out Reset (PBOR)
- RESET Instruction
- Stack Full Reset
- Stack Underflow Reset

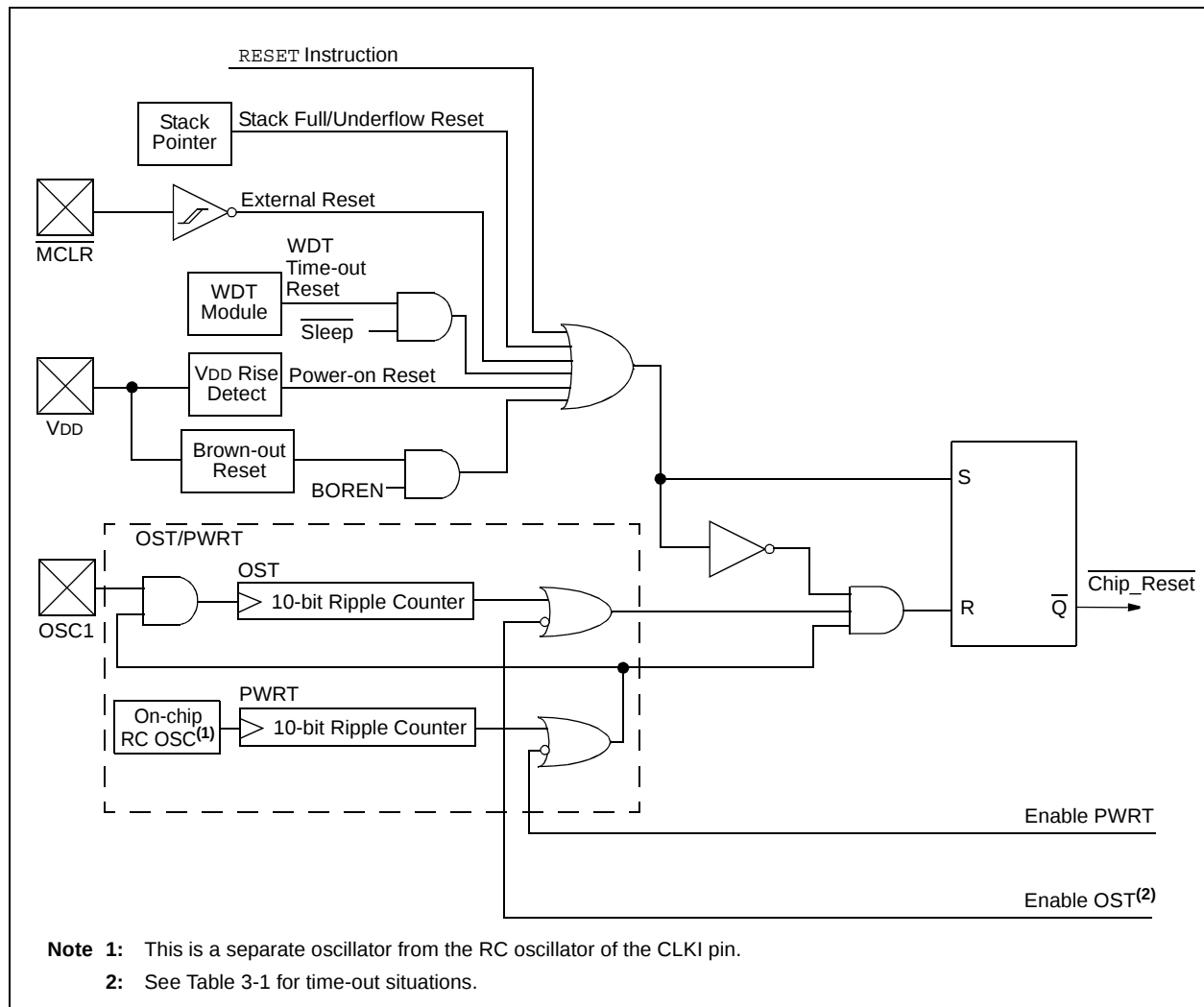
Most registers are unaffected by a Reset. Their status is unknown on POR and unchanged by all other Resets. The other registers are forced to a "Reset state" on Power-on Reset, $\overline{\text{MCLR}}$, WDT Reset, Brown-out Reset, $\overline{\text{MCLR}}$ Reset during Sleep and by the RESET instruction.

Most registers are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. Status bits from the RCON register, $\overline{\text{RI}}$, $\overline{\text{TO}}$, $\overline{\text{PD}}$, $\overline{\text{POR}}$ and $\overline{\text{BOR}}$, are set or cleared differently in different Reset situations, as indicated in Table 3-2. These bits are used in software to determine the nature of the Reset. See Table 3-3 for a full description of the Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 3-1.

The Enhanced MCU devices have a $\overline{\text{MCLR}}$ noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses. The $\overline{\text{MCLR}}$ pin is not driven low by any internal Resets, including the WDT.

FIGURE 3-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



PIC18F6520/8520/6620/8620/6720/8720

4.7.1 TWO-WORD INSTRUCTIONS

The PIC18FXX20 devices have four two-word instructions: `MOVFF`, `CALL`, `GOTO` and `LFSR`. The second word of these instructions has the 4 MSBs set to '1's and is a special kind of `NOP` instruction. The lower 12 bits of the second word contain data to be used by the instruction. If the first word of the instruction is executed, the data in the second word is accessed. If the second

word of the instruction is executed by itself (first word was skipped), it will execute as a `NOP`. This action is necessary when the two-word instruction is preceded by a conditional instruction that changes the PC. A program example that demonstrates this concept is shown in Example 4-3. Refer to **Section 24.0 "Instruction Set Summary"** for further details of the instruction set.

EXAMPLE 4-3: TWO-WORD INSTRUCTIONS

CASE 1:	
Object Code	Source Code
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; No, execute 2-word instruction
1111 0100 0101 0110	; 2nd operand holds address of REG2
0010 0100 0000 0000	ADDWF REG3 ; continue code
CASE 2:	
Object Code	Source Code
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; Yes
1111 0100 0101 0110	; 2nd operand becomes NOP
0010 0100 0000 0000	ADDWF REG3 ; continue code

4.8 Look-up Tables

Look-up tables are implemented two ways. These are:

- Computed `GOTO`
- Table Reads

4.8.1 COMPUTED GOTO

A computed `GOTO` is accomplished by adding an offset to the program counter (`ADDWF PCL`).

A look-up table can be formed with an `ADDWF PCL` instruction and a group of `RETLW 0xnn` instructions. `WREG` is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the `ADDWF PCL` instruction. The next instruction executed will be one of the `RETLW 0xnn` instructions, that returns the value `0xnn` to the calling function.

The offset value (value in `WREG`) specifies the number of bytes that the program counter should advance.

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

4.8.2 TABLE READS/TABLE WRITES

A better method of storing data in program memory allows 2 bytes of data to be stored in each instruction location.

Look-up table data may be stored 2 bytes per program word by using table reads and writes. The Table Pointer (`TBLPTR`) specifies the byte address and the Table Latch (`TBLAT`) contains the data that is read from, or written to program memory. Data is transferred to/from program memory, one byte at a time.

A description of the table read/table write operation is shown in **Section 5.0 "Flash Program Memory"**.

PIC18F6520/8520/6620/8620/6720/8720

FIGURE 4-7: DATA MEMORY MAP FOR PIC18FX620 AND PIC18FX720 DEVICES

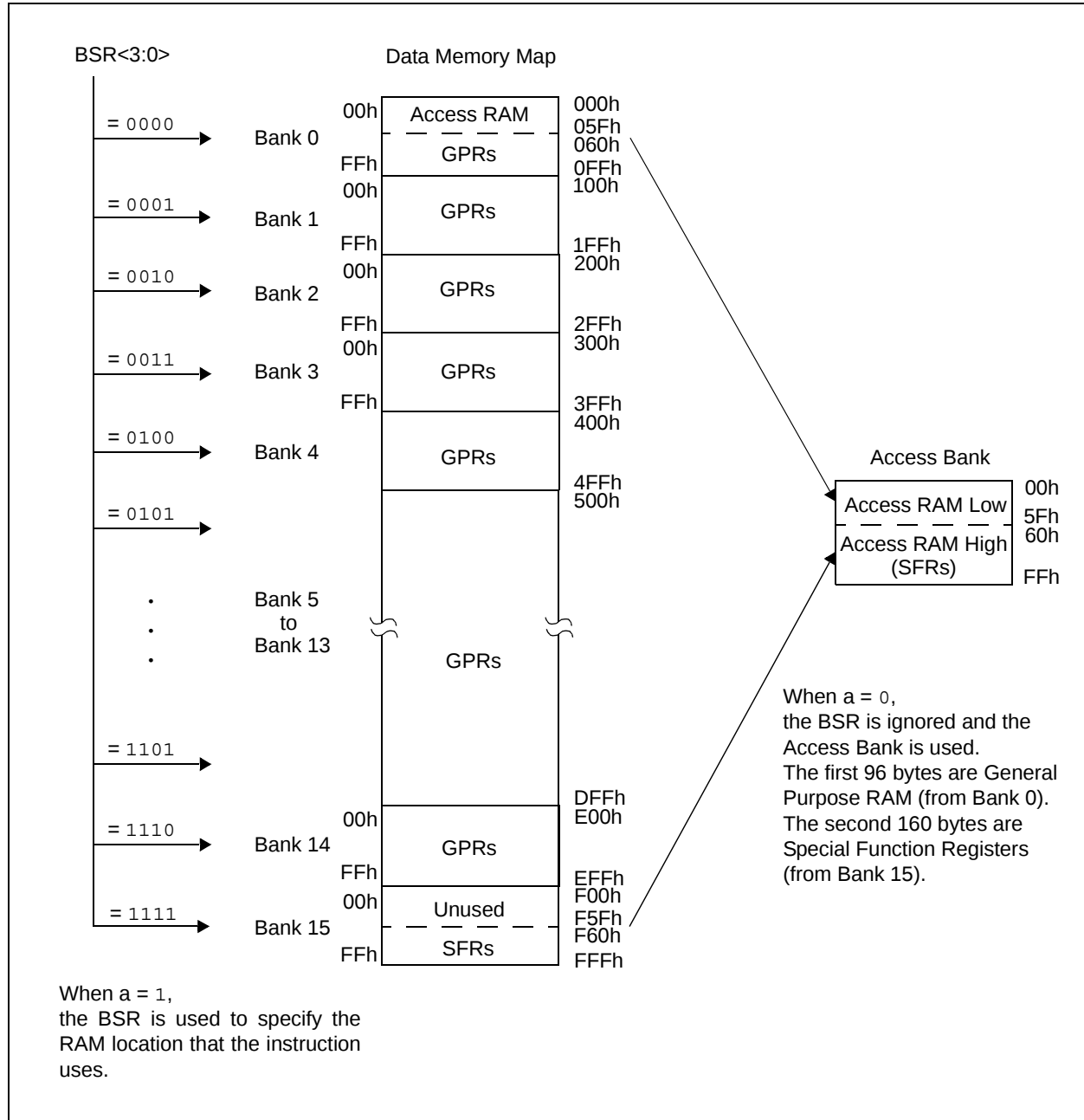
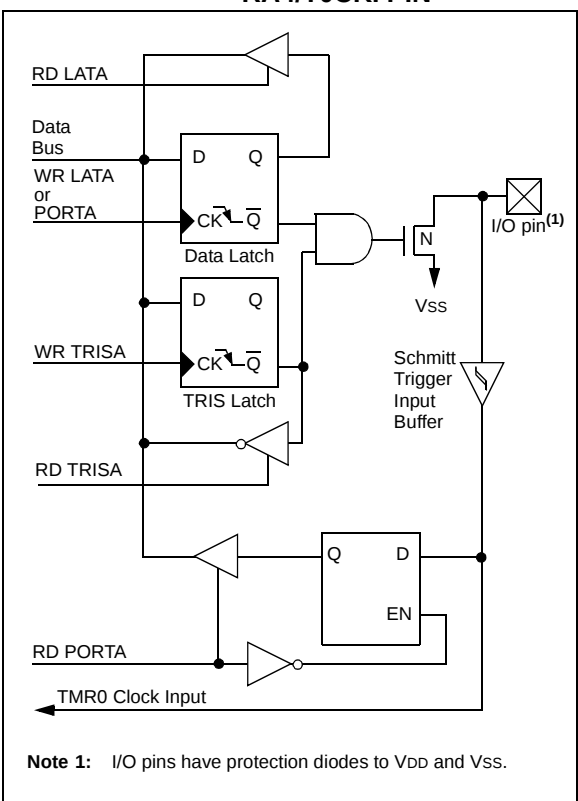


FIGURE 10-3: BLOCK DIAGRAM OF RA4/T0CKI PIN

[illegible]

PIC18F6520/8520/6620/8620/6720/8720

10.4 PORTD, TRISD and LATD Registers

PORTD is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., put the contents of the output latch on the selected pin).

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register, read and write the latched output value for PORTD.

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Note: On a Power-on Reset, these pins are configured as digital inputs.

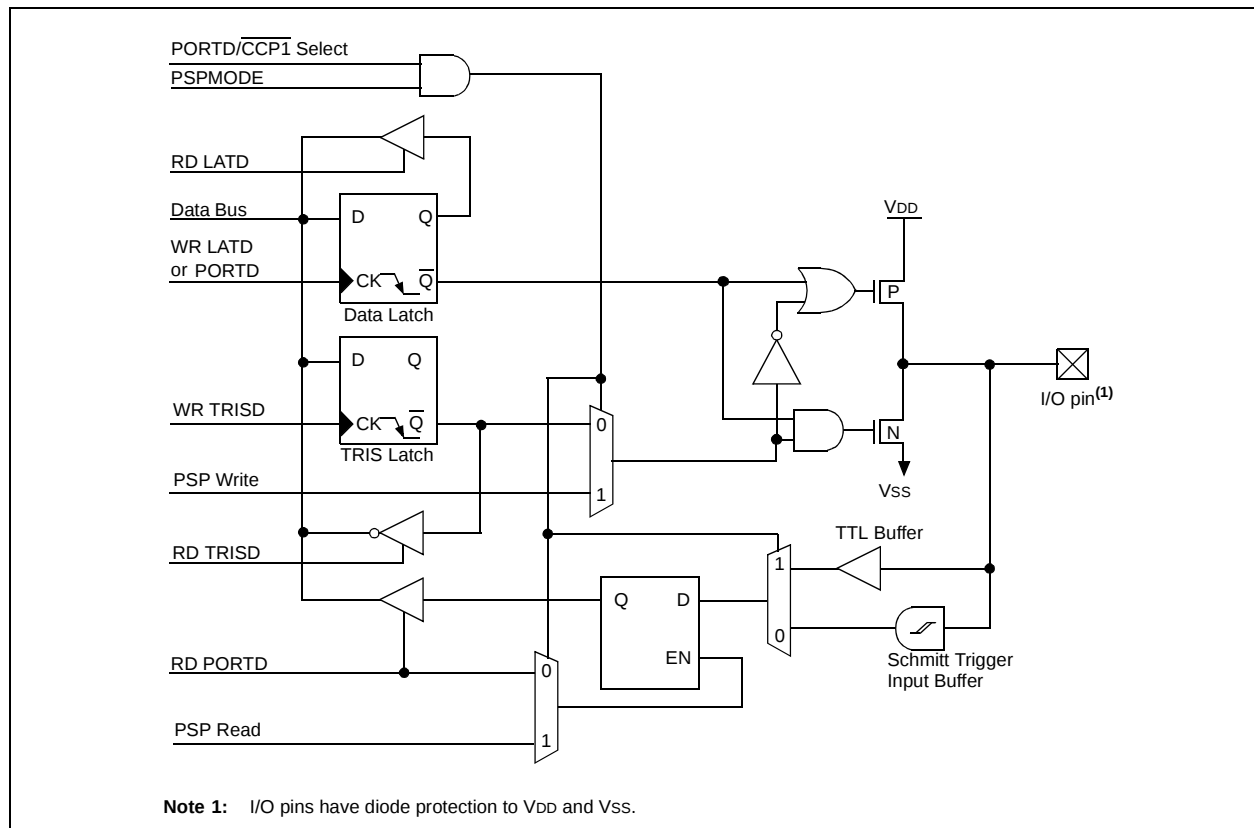
PORTD is multiplexed with the system bus as the external memory interface; I/O port functions are only available when the system bus is disabled, by setting the EBDIS bit in the MEMCOM register (MEMCON<7>). When operating as the external memory interface, PORTD is the low-order byte of the multiplexed address/data bus (AD7:AD0).

PORTD can also be configured as an 8-bit wide micro-processor port (Parallel Slave Port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL. See **Section 10.10 “Parallel Slave Port”** for additional information on the Parallel Slave Port (PSP).

EXAMPLE 10-4: INITIALIZING PORTD

```
CLRF    PORTD    ; Initialize PORTD by
                  ; clearing output
                  ; data latches
CLRF    LATD     ; Alternate method
                  ; to clear output
                  ; data latches
MOVLW   0xCF     ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISD    ; Set RD<3:0> as inputs
                  ; RD<5:4> as outputs
                  ; RD<7:6> as inputs
```

FIGURE 10-9: PORTD BLOCK DIAGRAM IN I/O PORT MODE



PIC18F6520/8520/6620/8620/6720/8720

TABLE 10-11: PORTF FUNCTIONS

Name	Bit#	Buffer Type	Function
RF0/AN5	bit 0	ST	Input/output port pin or analog input.
RF1/AN6/C2OUT	bit 1	ST	Input/output port pin, analog input or comparator 2 output.
RF2/AN7/C1OUT	bit 2	ST	Input/output port pin, analog input or comparator 1 output.
RF3/AN8	bit 3	ST	Input/output port pin or analog input/comparator input.
RF4/AN9	bit 4	ST	Input/output port pin or analog input/comparator input.
RF5/AN10/CVREF	bit 5	ST	Input/output port pin, analog input/comparator input or comparator reference output.
RF6/AN11	bit 6	ST	Input/output port pin or analog input/comparator input.
RF7/ $\overline{SS}$	bit 7	ST/TTL	Input/output port pin or slave select pin for synchronous serial port.

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

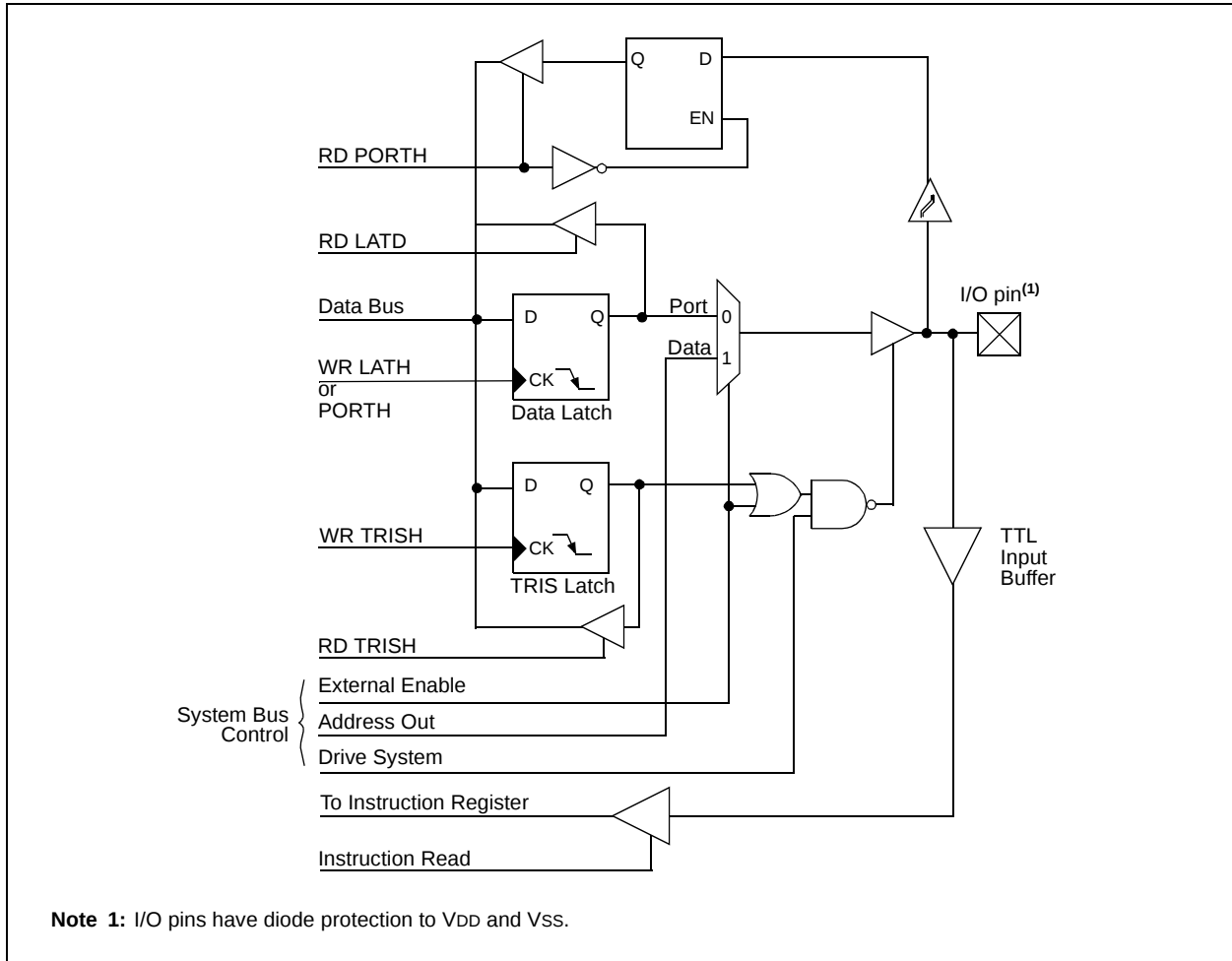
TABLE 10-12: SUMMARY OF REGISTERS ASSOCIATED WITH PORTF

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
TRISF	PORTF Data Direction Control Register								1111 1111	1111 1111
PORTF	Read PORTF pin/Write PORTF Data Latch								xxxx xxxx	uuuu uuuu
LATF	Read PORTF Data Latch/Write PORTF Data Latch								0000 0000	uuuu uuuu
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	--00 0000	--00 0000
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000 0000	0000 0000

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

PIC18F6520/8520/6620/8620/6720/8720

FIGURE 10-19: RH3:RH0 PINS BLOCK DIAGRAM IN SYSTEM BUS MODE



PIC18F6520/8520/6620/8620/6720/8720

17.3.3 ENABLING SPI I/O

To enable the serial port, SSP Enable bit, SSPEN (SSPCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{SS}$ must have TRISF<7> bit set

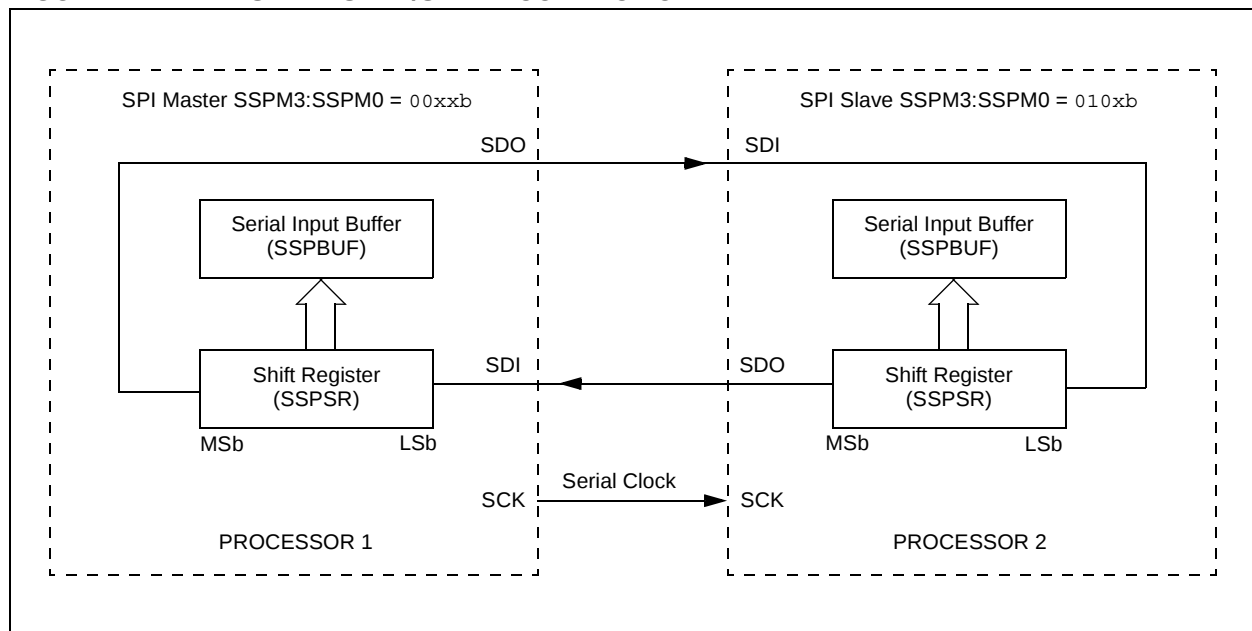
Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

17.3.4 TYPICAL CONNECTION

Figure 17-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

FIGURE 17-2: SPI MASTER/SLAVE CONNECTION



PIC18F6520/8520/6620/8620/6720/8720

FIGURE 17-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

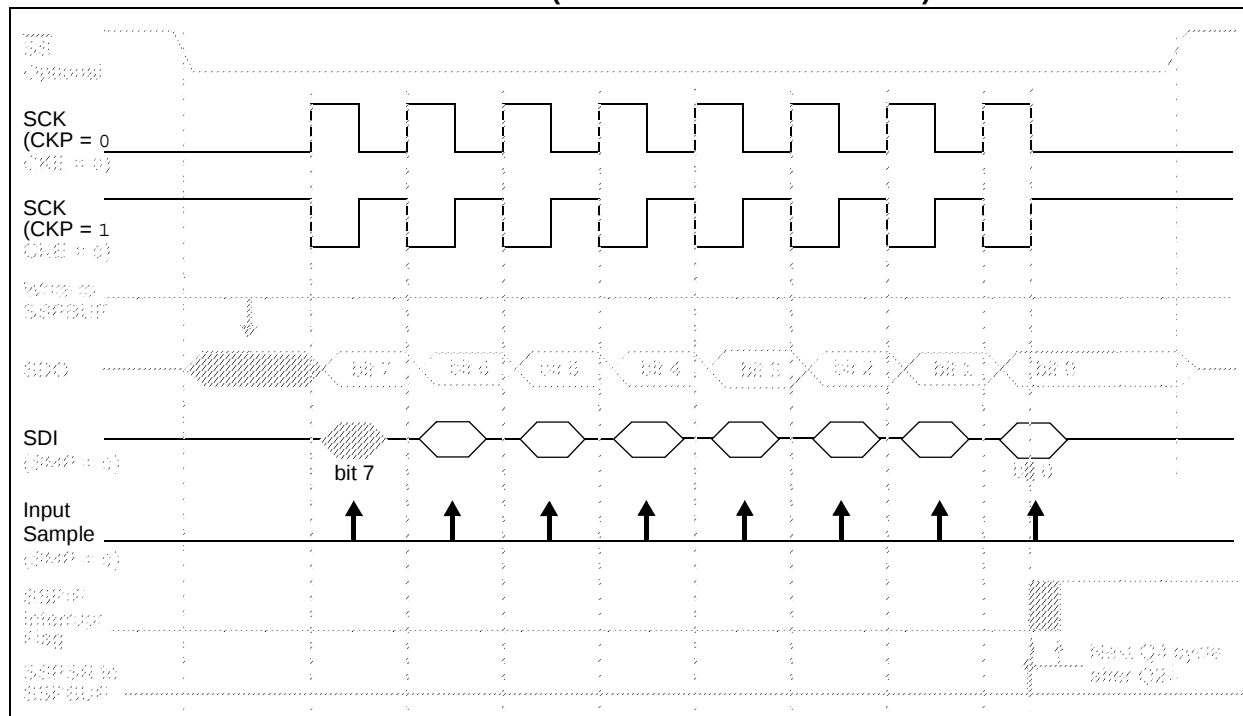
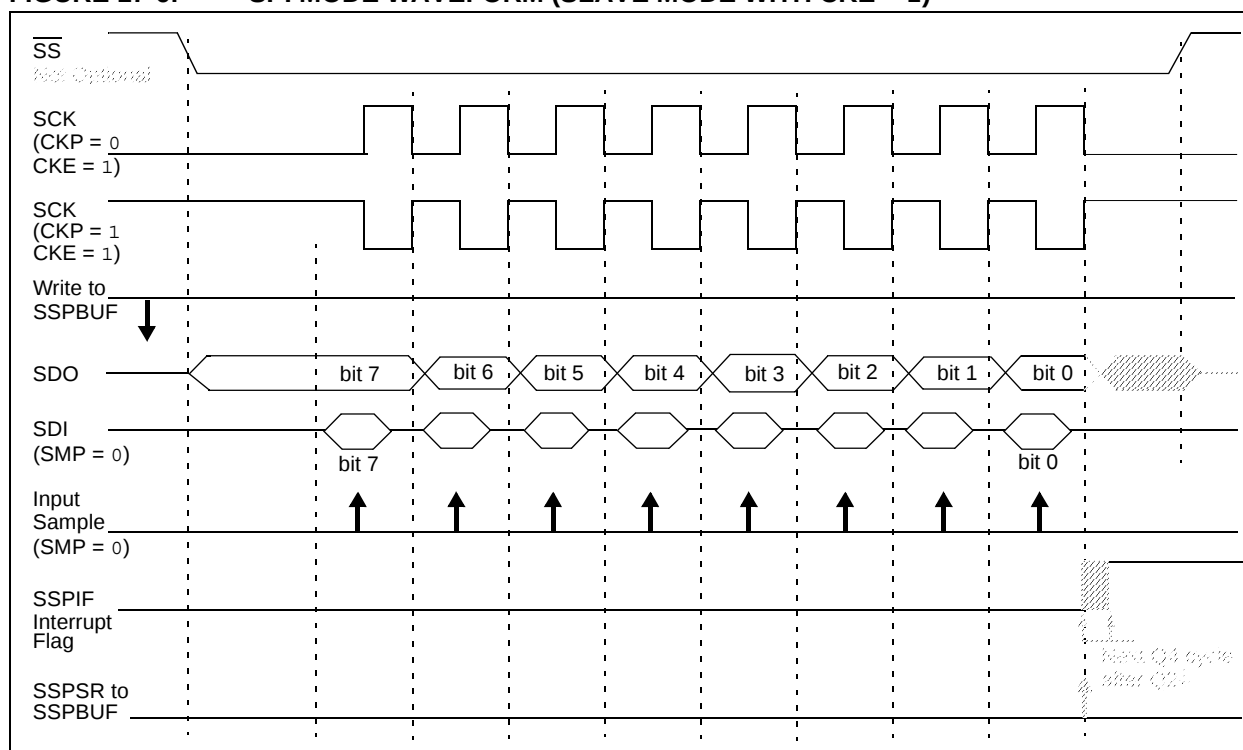


FIGURE 17-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



PIC18F6520/8520/6620/8620/6720/8720

FIGURE 18-2: ASYNCHRONOUS TRANSMISSION

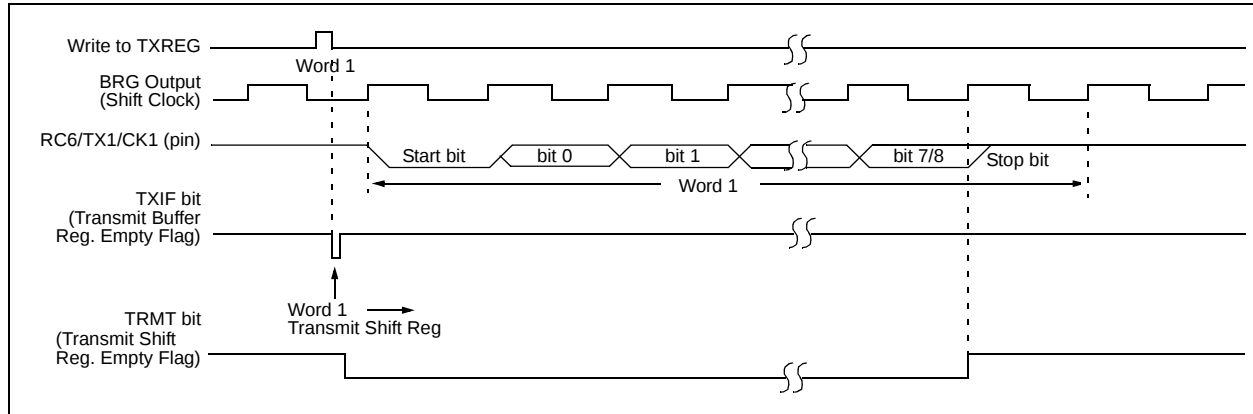


FIGURE 18-3: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)

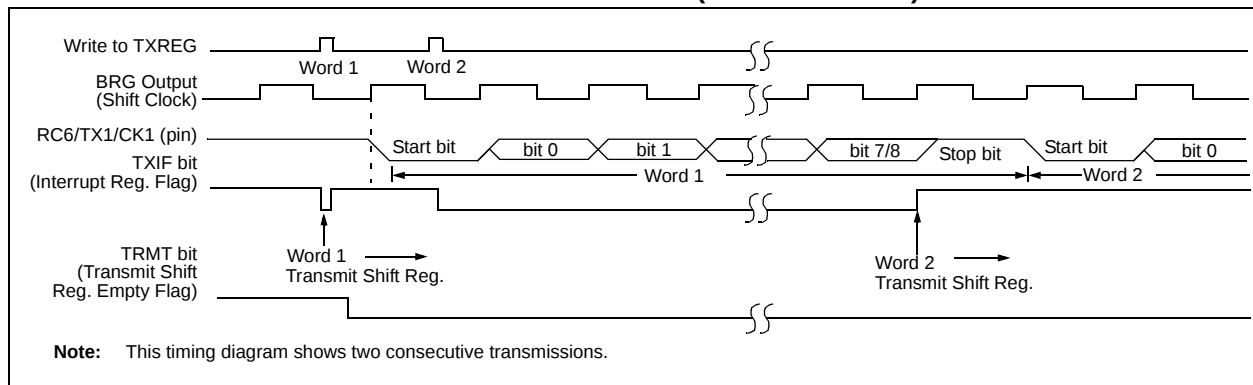


TABLE 18-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0111 1111	0111 1111
PIR3	—	—	RC2IF	TX2IF	TMR4IF	CCP5IF	CCP4IF	CCP3IF	--00 0000	--00 0000
PIE3	—	—	RC2IE	TX2IE	TMR4IE	CCP5IE	CCP4IE	CCP3IE	--00 0000	--00 0000
IPR3	—	—	RC2IP	TX2IP	TMR4IP	CCP5IP	CCP4IP	CCP3IP	--11 1111	--11 1111
RCSTAx ⁽¹⁾	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
TXREGx ⁽¹⁾	USART Transmit Register								0000 0000	0000 0000
TXSTAx ⁽¹⁾	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRGx ⁽¹⁾	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

Note 1: Register names generically refer to both of the identically named registers for the two USART modules, where 'x' indicates the particular module. Bit names and Reset values are identical between modules.

PIC18F6520/8520/6620/8620/6720/8720

20.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with the RF1 through RF6 pins. The on-chip voltage reference (Section 21.0 “Comparator Voltage Reference Module”) can also be an input to the comparators.

The CMCON register, shown as Register 20-1, controls the comparator input and output multiplexers. A block diagram of the various comparator configurations is shown in Figure 20-1.

REGISTER 20-1: CMCON REGISTER

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
bit 7				bit 0			

bit 7 **C2OUT**: Comparator 2 Output bit

When C2INV = 0:

1 = C2 VIN+ > C2 VIN-

0 = C2 VIN+ < C2 VIN-

When C2INV = 1:

1 = C2 VIN+ < C2 VIN-

0 = C2 VIN+ > C2 VIN-

bit 6 **C1OUT**: Comparator 1 Output bit

When C1INV = 0:

1 = C1 VIN+ > C1 VIN-

0 = C1 VIN+ < C1 VIN-

When C1INV = 1:

1 = C1 VIN+ < C1 VIN-

0 = C1 VIN+ > C1 VIN-

bit 5 **C2INV**: Comparator 2 Output Inversion bit

1 = C2 output inverted

0 = C2 output not inverted

bit 4 **C1INV**: Comparator 1 Output Inversion bit

1 = C1 output inverted

0 = C1 output not inverted

bit 3 **CIS**: Comparator Input Switch bit

When CM2:CM0 = 110:

1 = C1 VIN- connects to RF5/AN10

C2 VIN- connects to RF3/AN8

0 = C1 VIN- connects to RF6/AN11

C2 VIN- connects to RF4/AN9

bit 2-0 **CM2:CM0**: Comparator Mode bits

Figure 20-1 shows the Comparator modes and the CM2:CM0 bit settings.

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

PIC18F6520/8520/6620/8620/6720/8720

REGISTER 23-9: CONFIG6L: CONFIGURATION REGISTER 6 LOW (BYTE ADDRESS 30000Ah)

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
WRT7 ⁽¹⁾	WRT6 ⁽¹⁾	WRT5 ⁽¹⁾	WRT4 ⁽¹⁾	WRT3	WRT2	WRT1	WRT0
bit 7							bit 0

- bit 7 **WR7:** Write Protection bit⁽¹⁾
 1 = Block 7 (01C000-01FFFFh) not write-protected
 0 = Block 7 (01C000-01FFFFh) write-protected
- bit 6 **WR6:** Write Protection bit⁽¹⁾
 1 = Block 6 (018000-01BFFFh) not write-protected
 0 = Block 6 (018000-01BFFFh) write-protected
- bit 5 **WR5:** Write Protection bit⁽¹⁾
 1 = Block 5 (014000-017FFFh) not write-protected
 0 = Block 5 (014000-017FFFh) write-protected
- bit 4 **WR4:** Write Protection bit⁽¹⁾
 1 = Block 4 (010000-013FFFh) not write-protected
 0 = Block 4 (010000-013FFFh) write-protected
- bit 3 **WR3:** Write Protection bit
For PIC18FX520 devices:
 1 = Block 3 (006000-007FFFh) not write-protected
 0 = Block 3 (006000-007FFFh) write-protected
For PIC18FX620 and PIC18FX720 devices:
 1 = Block 3 (00C000-00FFFFh) not write-protected
 0 = Block 3 (00C000-00FFFFh) write-protected
- bit 2 **WR2:** Write Protection bit
For PIC18FX520 devices:
 1 = Block 2 (004000-005FFFh) not write-protected
 0 = Block 2 (004000-005FFFh) write-protected
For PIC18FX620 and PIC18FX720 devices:
 1 = Block 2 (008000-00BFFFh) not write-protected
 0 = Block 2 (008000-00BFFFh) write-protected
- bit 1 **WR1:** Write Protection bit
For PIC18FX520 devices:
 1 = Block 1 (002000-003FFFh) not write-protected
 0 = Block 1 (002000-003FFFh) write-protected
For PIC18FX620 and PIC18FX720 devices:
 1 = Block 1 (004000-007FFFh) not write-protected
 0 = Block 1 (004000-007FFFh) write-protected
- bit 0 **WR0:** Write Protection bit
For PIC18FX520 devices:
 1 = Block 0 (000800-001FFFh) not write-protected
 0 = Block 0 (000800-001FFFh) write-protected
For PIC18FX620 and PIC18FX720 devices:
 1 = Block 0 (000200-003FFFh) not write-protected
 0 = Block 0 (000200-003FFFh) write-protected

Note 1: Unimplemented in PIC18FX520 and PIC18FX620 devices; maintain this bit set.

Legend:

R = Readable bit P = Programmable bit U = Unimplemented bit, read as '0'
 - n = Value when device is unprogrammed u = Unchanged from programmed state

PIC18F6520/8520/6620/8620/6720/8720

REGISTER 23-12: CONFIG7H: CONFIGURATION REGISTER 7 HIGH (BYTE ADDRESS 30000Dh)

U-0	R/P-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB	—	—	—	—	—	—

bit 7

bit 0

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

bit 6 **EBTRB:** Boot Block Table Read Protection bit

For PIC18FX520 devices:

1 = Boot Block (000000-0007FFh) not protected from table reads executed in other blocks

0 = Boot Block (000000-0007FFh) protected from table reads executed in other blocks

For PIC18FX620 and PIC18FX720 devices:

1 = Boot Block (000000-0001FFh) not protected from table reads executed in other blocks

0 = Boot Block (000000-0001FFh) protected from table reads executed in other blocks

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

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

- n = Value when device is unprogrammed

u = Unchanged from programmed state

REGISTER 23-13: DEVICE ID REGISTER 1 FOR PIC18FXX20 DEVICES (ADDRESS 3FFFEh)

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0

bit 7

bit 0

bit 7-5 **DEV2:DEV0:** Device ID bits

000 = PIC18F8720

001 = PIC18F6720

010 = PIC18F8620

011 = PIC18F6620

bit 4-0 **REV4:REV0:** Revision ID bits

These bits are used to indicate the device revision.

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

- n = Value when device is unprogrammed

u = Unchanged from programmed state

REGISTER 23-14: DEVICE ID REGISTER 2 FOR PIC18FXX20 DEVICES (ADDRESS 3FFFFh)

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3

bit 7

bit 0

bit 7-0 **DEV10:DEV3:** Device ID bits

These bits are used with the DEV2:DEV0 bits in the Device ID Register 1 to identify the part number.

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '0'

- n = Value when device is unprogrammed

u = Unchanged from programmed state

PIC18F6520/8520/6620/8620/6720/8720

TABLE 24-1: PIC18FXXXX INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word				Status Affected	Notes
				MSb		LSb			
CONTROL OPERATIONS									
BC	n	Branch if Carry	1 (2)	1110	0010	nnnn	nnnn	None	4
BN	n	Branch if Negative	1 (2)	1110	0110	nnnn	nnnn	None	
BNC	n	Branch if Not Carry	1 (2)	1110	0011	nnnn	nnnn	None	
BNN	n	Branch if Not Negative	1 (2)	1110	0111	nnnn	nnnn	None	
BN OV	n	Branch if Not Overflow	1 (2)	1110	0101	nnnn	nnnn	None	
BNZ	n	Branch if Not Zero	1 (2)	1110	0001	nnnn	nnnn	None	
BOV	n	Branch if Overflow	1 (2)	1110	0100	nnnn	nnnn	None	
BRA	n	Branch Unconditionally	2	1101	0nnn	nnnn	nnnn	None	
BZ	n	Branch if Zero	1 (2)	1110	0000	nnnn	nnnn	None	
CALL	n, s	Call subroutine 1st word	2	1110	110s	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk	$\overline{TO}$, $\overline{PD}$	
CLRWDT	—	Clear Watchdog Timer	1	0000	0000	0000	0100	C	
DAW	—	Decimal Adjust WREG	1	0000	0000	0000	0111	None	
GOTO	n	Go to address 1st word	2	1110	1111	kkkk	kkkk	None	
		2nd word		1111	kkkk	kkkk	kkkk		
NOP	—	No Operation	1	0000	0000	0000	0000	None	
NOP	—	No Operation	1	1111	xxxx	xxxx	xxxx	None	
POP	—	Pop top of return stack (TOS)	1	0000	0000	0000	0110	None	
PUSH	—	Push top of return stack (TOS)	1	0000	0000	0000	0101	None	
RCALL	n	Relative Call	2	1101	1nnn	nnnn	nnnn	None	
RESET		Software device Reset	1	0000	0000	1111	1111	All	
RETFIE	s	Return from interrupt enable	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	Return with literal in WREG	2	0000	1100	kkkk	kkkk	None	
RETURN	s	Return from Subroutine	2	0000	0000	0001	001s	None	
SLEEP	—	Go into Standby mode	1	0000	0000	0000	0011	$\overline{TO}$, $\overline{PD}$	

- Note 1:** When a Port register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, $d = 1$), the prescaler will be cleared if assigned.
- 3:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are 2-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.
- 5:** If the table write starts the write cycle to internal memory, the write will continue until terminated.

PIC18F6520/8520/6620/8620/6720/8720

RLNCF Rotate Left f (no carry)

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

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

Operation: $(f<n>) \rightarrow \text{dest}<n+1>$,
 $(f<7>) \rightarrow \text{dest}<0>$

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

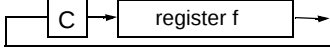
Operation: $(f<n>) \rightarrow \text{dest}<n-1>$,
 $(f<0>) \rightarrow C$,
 $(C) \rightarrow \text{dest}<7>$

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

25.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

25.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, pre-processor, and one-step driver, and can run on multiple platforms.

25.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

25.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

25.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

PIC18F6520/8520/6620/8620/6720/8720

26.3 DC Characteristics: PIC18F6520/8520/6620/8620/6720/8720 (Industrial, Extended) PIC18LF6520/8520/6620/8620/6720/8720 (Industrial)

DC 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.	Sym	Characteristic	Min	Max	Units	Conditions
D030 D030A D031 D032 D032A D033	V _{IL}	Input Low Voltage I/O ports: with TTL buffer with Schmitt Trigger buffer RC3 and RC4 $\overline{\text{MCLR}}$ OSC1 (in XT, HS and LP modes) and T1OSI OSC1 (in RC and EC mode) ⁽¹⁾	V _{SS} — V _{SS} V _{SS} V _{SS} V _{SS} V _{SS}	0.15 V _{DD} 0.8 0.2 V _{DD} 0.3 V _{DD} 0.2 V _{DD} 0.2 V _{DD} 0.2 V _{DD}	V V V V V V V	V _{DD} < 4.5V 4.5V ≤ V _{DD} ≤ 5.5V
D040 D040A D041 D042 D042A D043	V _{IH}	Input High Voltage I/O ports: with TTL buffer with Schmitt Trigger buffer RC3 and RC4 $\overline{\text{MCLR}}$, OSC1 (EC mode) OSC1 and T1OSI OSC1 (RC mode) ⁽¹⁾	0.25 V _{DD} + 0.8V 2.0 0.8 V _{DD} 0.7 V _{DD} 0.8 V _{DD} 1.6 0.9 V _{DD}	V _{DD} V _{DD} V _{DD} V _{DD} V _{DD} V _{DD} V _{DD}	V V V V V V V	V _{DD} < 4.5V 4.5V ≤ V _{DD} ≤ 5.5V LP, XT, HS, HSPLL modes ⁽¹⁾
D060 D061 D063	I _{IL}	Input Leakage Current^(2,3) I/O ports $\overline{\text{MCLR}}$ OSC1	— — —	±1 ±5 ±5	μA μA μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , Pin at high-impedance V _{SS} ≤ V _{PIN} ≤ V _{DD} V _{SS} ≤ V _{PIN} ≤ V _{DD}
D070	I _{PU} I _{PURB}	Weak Pull-up Current PORTB weak pull-up current	50	400	μA	V _{DD} = 5V, V _{PIN} = V _{SS}

Note 1: In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC device be driven with an external clock while in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: Parameter is characterized but not tested.

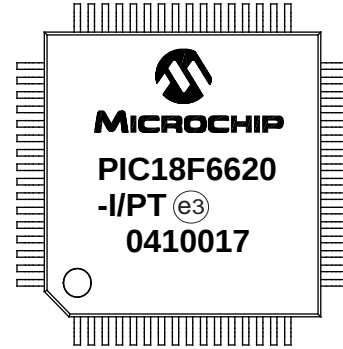
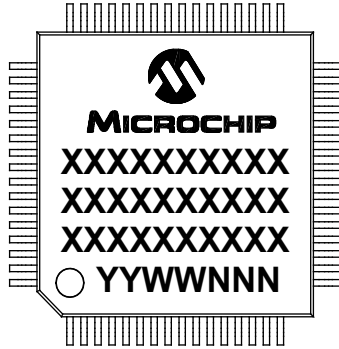
PIC18F6520/8520/6620/8620/6720/8720

28.0 PACKAGING INFORMATION

28.1 Package Marking Information

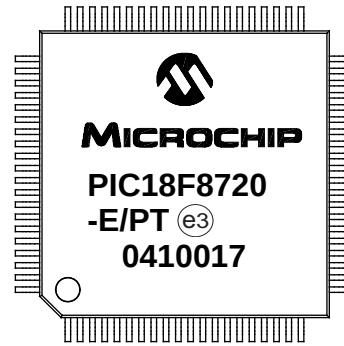
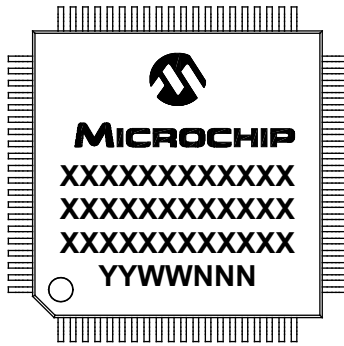
64-Lead TQFP (10x10x1 mm)

Example



80-Lead TQFP (12x12x1 mm)

Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: <http://microchip.com/support>