

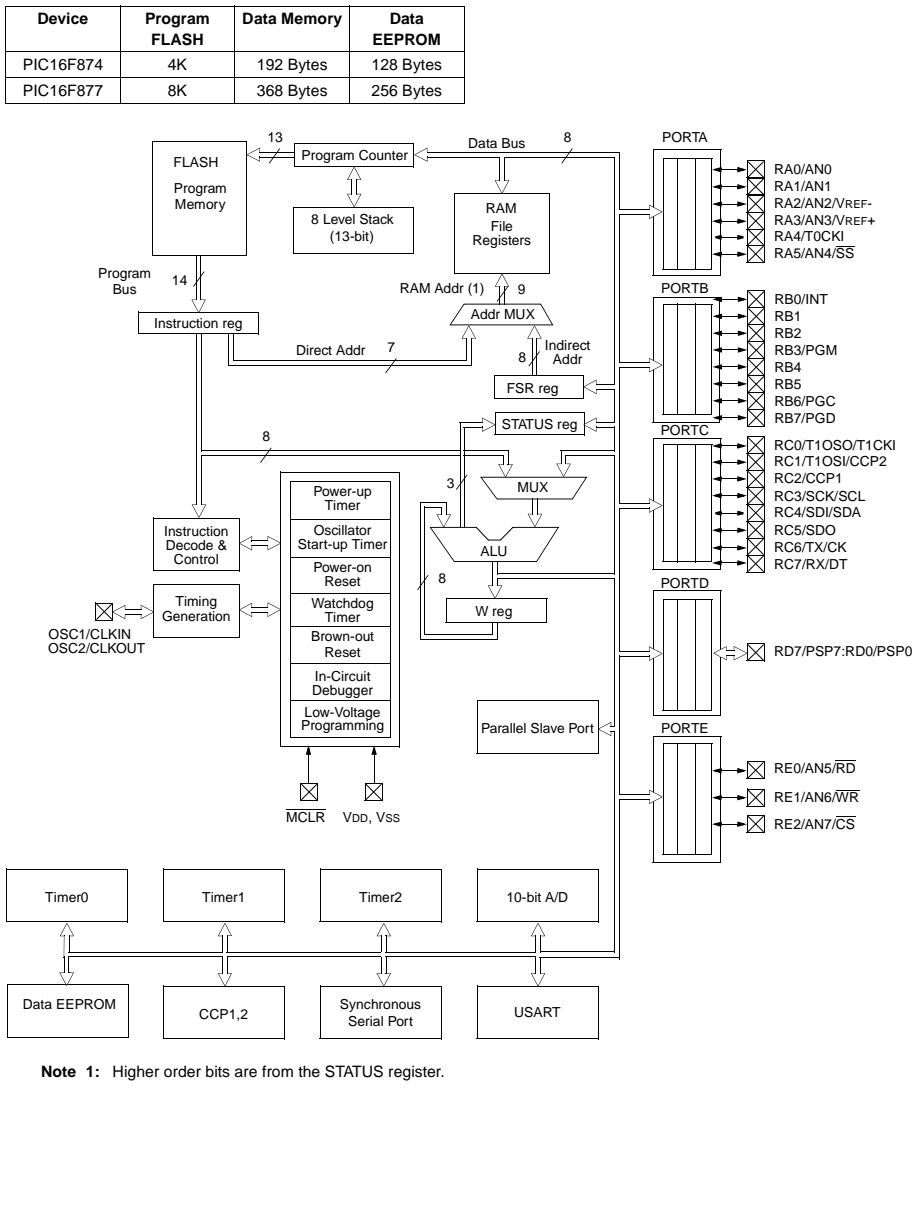
- Learning the architecture of the PIC16F877
- Learning the instruction set of the PIC16F877
 - ★ 35 instructions
 - ★ same as previous PIC versions (eg 16F74)
- Evaluation of example code
- Experimentation with lab software (today)
- Experimentation with lab hardware (Wednesday)

- 14 bit instructions
- 8 bit data values, byte addressed
- Harvard architecture (separate address/data memory)
- Storage capacity:
 - ★ 8k x 14 Flash EPROM (program storage)
 - ★ 256x8 EEPROM
 - ★ 368x8 SRAM (data storage)
- Universal Asynch./Synch. Receiver/Transmitter (UART)
- 8 channel, 1-bit A/D converter.

16F877 Data Path

PIC16F87X

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM



16F877 Program Memory Map

Figure 2-1 of 16F877 data sheet.

- Location 0 is first instruction executed (go to Start).
- Interrupt vector aids with spurious actions.
- Eight-deep program stack.

16F877 Data Memory and Register Files

PIC16F87X

FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP

						File Address	
Indirect addr. ^(*)	00h	Indirect addr. ^(*)	80h	Indirect addr. ^(*)	100h	Indirect addr. ^(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose Register 96 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes	
		accesses 70h-7Fh	EFh F0h	accesses 70h-7Fh	16Fh 170h	accesses 70h - 7Fh	1EFh 1F0h
Bank 0	7Fh	Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

■ Unimplemented data memory locations, read as '0'.
 * Not a physical register.
Note 1: These registers are not implemented on 28-pin devices.
Note 2: These registers are reserved, maintain these registers clear.

16F877 Status Register

PIC16F87X

2.2.2.1 STATUS REGISTER

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

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 or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the \overline{TO} and \overline{PD} bits are not writable, 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` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions not affecting any status bits, see the "Instruction Set Summary."

Note 1: The C and DC bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 2-1: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

	R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
bit7								bit0
bit 7:	IRP: Register Bank Select bit (used for indirect addressing) 1 = Bank 2, 3 (100h - 1FFh) 0 = Bank 0, 1 (00h - FFh)							
bit 6-5:	RP1:RP0: Register Bank Select bits (used for direct addressing) 11 = Bank 3 (180h - 1FFh) 10 = Bank 2 (100h - 17Fh) 01 = Bank 1 (80h - FFh) 00 = Bank 0 (00h - 7Fh) Each bank is 128 bytes							
bit 4:	\overline{TO}: Time-out bit 1 = After power-up, <code>CLRWDT</code> instruction, or <code>SLEEP</code> instruction 0 = A WDT time-out occurred							
bit 3:	\overline{PD}: Power-down bit 1 = After power-up or by the <code>CLRWDT</code> instruction 0 = By execution of the <code>SLEEP</code> instruction							
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/borrow bit (<code>ADDWF</code> , <code>ADDLW</code> , <code>SUBLW</code> , <code>SUBWF</code> instructions) (for borrow the polarity is reversed) 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							
bit 0:	C: Carry/borrow bit (<code>ADDWF</code> , <code>ADDLW</code> , <code>SUBLW</code> , <code>SUBWF</code> 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 borrow the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (<code>RRF</code> , <code>RLF</code>) instructions, this bit is loaded with either the high or low order bit of the source register.							

R = Readable bit
 W = Writable bit
 U = Unimplemented bit, read as '0'
 - n = Value at POR reset

16F877 Instruction Set

See Table 13-1 of the 16F877 data sheet.

See Figure 13-1 of the 16F877 data sheet.

3 types of instructions:

- Byte-oriented operations
- Bit-oriented operations
- Literal and control operations

16F877 Instruction Set

PIC16F87X

TABLE 13-2: PIC16CXXX INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	14-Bit Opcode		Status Affected	Notes
			MSb	LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111	dfff ffff	C,DC,Z 1,2
ANDWF	f, d	AND W with f	1	00 0101	dfff ffff	Z 1,2
CLRF	f	Clear f	1	00 0001	1fff ffff	Z 2
CLRWF	-	Clear W	1	00 0001	0xxx xxxx	Z
COMF	f, d	Complement f	1	00 1001	dfff ffff	Z 1,2
DECf	f, d	Decrement f	1	00 0011	dfff ffff	Z 1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011	dfff ffff	1,2,3
INCF	f, d	Increment f	1	00 1010	dfff ffff	Z 1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111	dfff ffff	1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100	dfff ffff	Z 1,2
MOVF	f, d	Move f	1	00 1000	dfff ffff	Z 1,2
MOVWF	f	Move W to f	1	00 0000	1fff ffff	
NOP	-	No Operation	1	00 0000	0xx0 0000	
RLF	f, d	Rotate Left f through Carry	1	00 1101	dfff ffff	C 1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100	dfff ffff	C 1,2
SUBWF	f, d	Subtract W from f	1	00 0010	dfff ffff	C,DC,Z 1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110	dfff ffff	1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110	dfff ffff	Z 1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb	bfff ffff	1,2
BSF	f, b	Bit Set f	1	01 01bb	bfff ffff	1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01 10bb	bfff ffff	3
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01 11bb	bfff ffff	3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11 111x	kkkk kkkk	C,DC,Z
ANDLW	k	AND literal with W	1	11 1001	kkkk kkkk	Z
CALL	k	Call subroutine	2	10 0kkk	kkkk kkkk	
CLRWD _T	-	Clear Watchdog Timer	1	00 0000	0110 0100	\overline{TO} , \overline{PD}
GOTO	k	Go to address	2	10 1kkk	kkkk kkkk	
IORLW	k	Inclusive OR literal with W	1	11 1000	kkkk kkkk	Z
MOVLW	k	Move literal to W	1	11 00xx	kkkk kkkk	
RETFIE	-	Return from interrupt	2	00 0000	0000 1001	
RETLW	k	Return with literal in W	2	11 01xx	kkkk kkkk	
RETURN	-	Return from Subroutine	2	00 0000	0000 1000	
SLEEP	-	Go into standby mode	1	00 0000	0110 0011	\overline{TO} , \overline{PD}
SUBLW	k	Subtract W from literal	1	11 110x	kkkk kkkk	C,DC,Z
XORLW	k	Exclusive OR literal with W	1	11 1010	kkkk kkkk	Z

Note 1: When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), 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 to the Timer0 Module.

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.

Note: Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

16F877 Instruction Set

See page 27 of Peatman book.

16F877 Instruction Set

See page 27 of Peatman book.

16F877 Development Tools

- A large amount of supplemental documentation available at www.microchip.com
- Sizable amount of development can be completed at home.
- MPASM - Assembler for PIC programs
- MPSIM - Simulator for PIC programs
 - * allows single-stepping.
 - * allows simulation of analog inputs.
 - * allows full evaluation of state.
- MPLAB - integrated hardware/software debugger
 - * front-end interface for MPASM and MPSIM
 - * configures MPLAD-ICD (hardware)
 - * allows for in-circuit testing.

Reading and Exercises

- 16F877 data sheet - pages 1-18, 26, 137-144
- Peatman book - Chapter 2.

Exercises:

- MPASM - learn to assemble and simulate a demo program (today)
- MPLAB-ICD - learn to download, single-step, and debug a program.