UNIVERSITY OF SWAZILAND
MAIN EXAMINATION, SECOND SEMESTER MAY 2016

FACULTY OF SCIENCE AND ENGINEERING

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

TITLE OF PAPER: MICROCONTROLLERS AND MICROCOMPUTER SYSTEMS

COURSE CODE: EE423

TIME ALLOWED: THREE HOURS

INSTRUCTIONS:

1. There are five questions in this paper. Answer any FOUR questions. Each question carries 25 marks.
2. If you think not enough data has been given in any question, you may assume any reasonable values stating your assumptions in each case.

THIS PAPER SHOULD NOT BE OPENED UNTIL PERMISSION HAS BEEN GIVEN BY THE INVIGILATOR.

THIS PAPER CONTAINS SEVEN (6) PAGES INCLUDING THIS PAGE.

QUESTION ONE *(25 marks)*

(a) Draw a typical block diagram of a microprocessor based microcomputer system capable of storing user inputs, event counting and timing, interfacing analog inputs, interfacing digital I/O, serial communications and handling interrupts.

(5 marks)

(b) (i) With respect to a basic processor architecture, briefly state the function of its *ALU*, control unit and few of the dedicated processor registers. (6 marks)

(ii) Differentiate between *CISC* and *RISC* systems. (2 marks)

(c) Consider the program segment of a microcontroller program shown in Figure-Q1.

```
Loop_1    EQU    0100h
Loop_2    EQU    0200h
          ........
          ........
          goto   Loop_1
          .........
          ..........
Loop_1    bsf    portA,0
          call   Loop_2
          bcf    portA,0
          ........
          ........
Loop_2    nop
          bcf    portB,0
          bsf    portB,1
          retlw  .34
          btfss  portA,3
          goto   check_4
          ..........
```

Figure-Q1

(i) Determine the memory locations occupied in the program memory by the subroutine section. (3 marks)

(ii) Tabulate the program counter contents, between the execution of the instruction before the *CALL* and the instruction after the *CALL*. Indicate the changes in the *STACK* at relevant points.

(5 marks)

(d) State the following information with respect to 16F84A microcontroller; instruction width, program memory size, data memory size, data EEPROM size, type of architecture and on-chip modules.

(4 marks)

QUESTION TWO *(25 marks)*

An ultrasonic ranging module used in a remotely controlled vehicle is initiated by an input pulse (pulse width $> 50\mu s$). The module in response generates an output pulse having a pulse width equal to the time taken by the ultrasonic burst to travel to an object and back to the module. Assume that the speed of sound in air is $300\frac{m}{s}$. The input of the module is connected to a 16F84A, PortB(0) and the output of the module is connected to PortA(0). Motor control of the vehicle is connected to PortB(1) and the microcontroller is running on a $10MHz$ crystal oscillator.

(i) Draw a diagram to show the interconnections between the microcontroller and the ultrasonic module.

(2 marks)

(ii) Draw the flow chart of a program, which decides if an object is within $50cm$ to switch off the motors. You may consider the program section from the initiation of the input pulse to the module and up to the switching off of the motors.
Show the calculations of any data values and state your assumptions.

(7 marks)

(iii) Show the configuration of the registers TRISA, TRISB and OPTION to meet the conditions in (ii) above.

(4 marks)

(iv) Write the code in assembly for (ii) above.

(10 marks)

(v) Show how you are going to connect a crystal oscillator to the microcontroller.

(2 marks)

QUESTION THREE ***(25 marks)***

An egg incubator system is implemented using a 16F84A. A heater is used to keep the temperature at the desired value. Humidity is controlled operating a humiditifier and a de-humiditifier as required. The signals used in the system are TTL compatible and listed below.

Signal	*Description*
T	Output of the temperature sensor. Returns '1' if the temperature is above the set limit.
HH	'High' output of the humidity sensor. Returns '1' if the humidity is above the set value.
HL	'Low' output of the humidity sensor. Returns '1' if the humidity is less than the set value.
H	Control signal of the heater. Logic '1' applied turns ON the heater.
HM	Control signal of the humiditifier. Logic '1' applied turns ON the humiditifier.
DH	Control signal of the de-humiditifier. Logic '1' applied turns ON the de-humiditifier.

(i) Draw a circuit diagram for this system showing signal interconnections, marking microcontroller pins used. Sensors and the controlled components can be shown as blocks. Clock oscillator connections can be omitted.

(5 marks)

(ii) Explain briefly a control logic that can be implemented by a microcontroller program.

(7 marks)

(iii) Draw a flowchart of a program based on (ii) above.

(9 marks)

(iv) Using assembly instructions show how you are going to configure ports in your program.

(4 marks)

QUESTION FOUR (*25 marks*)

(a) Assume that a 16F84A is used in the following cases.

(i) An external device generates a low to high interrupt signal. How do you connect this signal to the microcontroller? Show the configuration of the registers involved. Assume that no other types of interrupts are used.

(3 marks)

(ii) Show the contents of the interrupt control register just after an interrupt mentioned in (i) above.

(2 marks)

(iii) In the case of an interrupt, state the events that will take place in the program execution. Indicate the essential features to be included in an interrupt service routine.

(5 marks)

(iv) The microcontroller is running on a 8MHz crystal oscillator and it is required to generate an interrupt every $500\mu s$. Briefly state a simple way to do this and show the contents of the relevant registers before and after the interrupt. Comment on timing errors if there are any.

(5 marks)

(b) Assume that a 16F877 microcontroller is used in the following cases.

(i) The microcontroller is used to monitor five analog sensors connected to its ADC. Show the configuration of the relevant registers for this operation assuming ADC is turned ON, channel 2 is selected and ready to convert. State your assumptions if there are any.

(5 marks)

(ii) After the conversion is just started and running, indicate the changes in the registers mentioned in (i) above.

(2 marks)

(iii) If a $4MHz$ crystal oscillator is to be used, show a suitable option for $ADCS1$ and $ADCS0$ bits giving justification.

(3 marks)

QUESTION FIVE (25 *marks*)

(a) A serial SPI digital temperature sensor is shown in Figure-Q5. It is to be used with a 16F877 microcontroller.

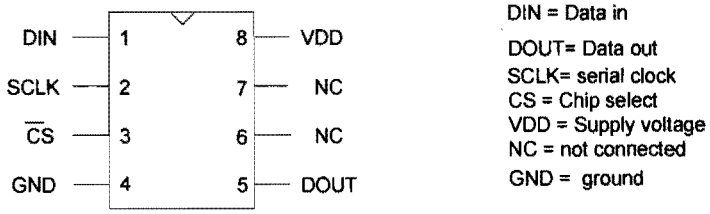


Figure-Q5

(i) Draw a block diagram to show the interconnections between the sensor and the microcontroller. Show only the pins that are interconnected giving identification including the pin numbers. State the assumptions used if there are any.

(6 marks)

(ii) Explain briefly the way data is transferred from the sensor to the microcontroller.

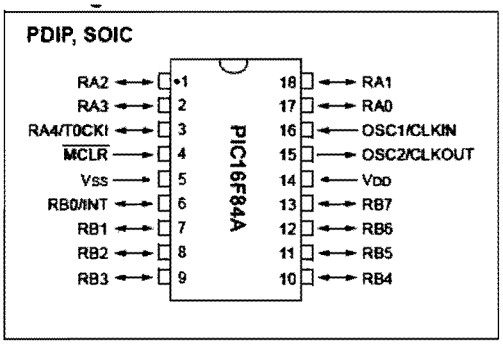
(9 marks)

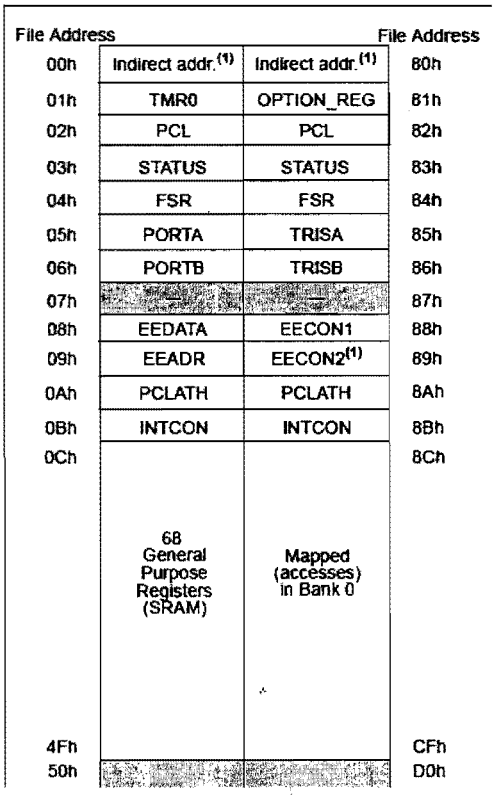
(b) Assume that for the case in (a) above, the microcontroller is running on a $4MHz$ crystal oscillator. The maximum data rate of the sensor is $300\frac{kb}{s}$.

Show the configuration of the necessary registers of the microcontroller which can be used for this interfacing. In addition to the usual settings, it requires positive idle state of the clock, data transfer on the falling edge of the clock and data sampling at the middle of the bit period. Show the settings of $SSPCON$, $SSPSTAT$ and any other register involved in the configuration.

(10 marks)

PIC 16F84A





STATUS REGISTER (ADDRESS 03h, 83h)

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
bit 7							bit 0

Unimplemented: Maintain as '0'

RP0: Register Bank Select bits (used for direct addressing)
01 = Bank 1 (80h - FFh)
00 = Bank 0 (00h - 7Fh)

$\overline{TO}$: Time-out bit
1 = After power-up, CLRWDT instruction, or SLEEP instruction
0 = A WDT time-out occurred

$\overline{PD}$: Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

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

DC: Digit carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for $\overline{borrow}$, 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

C: Carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for $\overline{borrow}$, the polarity is reversed)
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: A subtraction is executed by adding the two'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.

OPTION REGISTER (ADDRESS 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

RBPU: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

INTEDG: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

T0CS: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

T0SE: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

PSA: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

PS2:PS0: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

INTCON REGISTER (ADDRESS 0Bh, 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7							bit 0

GIE: Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

EEIE: EE Write Complete Interrupt Enable bit
1 = Enables the EE Write Complete interrupts
0 = Disables the EE Write Complete interrupt

T0IE: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

INTE: RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

RBIE: RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

T0IF: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

INTF: RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

RBIF: RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
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	lfff	ffff	Z	2
CLRW	-	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	lfff	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		
CLRWDT	-	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	

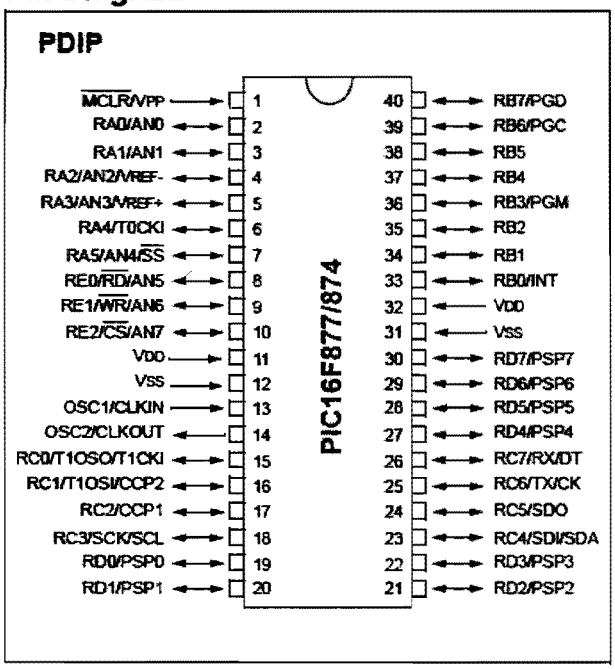
Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '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`.

PIC 16F877

Pin Diagram



SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS: 94h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

SMP: Sample bit
SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
SPI Slave mode:
SMP must be cleared when SPI is used in slave mode
In I^2C Master or Slave mode:
1 = Slew rate control disabled for standard speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for high speed mode (400 kHz)

CKE: SPI Clock Edge Select (Figure 9-2, Figure 9-3 and Figure 9-4)
SPI mode:
For CKP = 0
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK
For CKP = 1
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK
In I^2C Master or Slave mode:
1 = Input levels conform to SMBus spec
0 = Input levels conform to I^2C specs

D/A: Data/$\overline{\text{Address}}$ bit (I^2C mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

P: STOP bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a STOP bit has been detected last (this bit is '0' on RESET)
0 = STOP bit was not detected last

S: START bit
(I^2C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.)
1 = Indicates that a START bit has been detected last (this bit is '0' on RESET)
0 = START bit was not detected last

R/W: Read/Write bit Information (I^2C mode only)
This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next START bit, STOP bit or not $\overline{\text{ACK}}$ bit.
In I^2C Slave mode:
1 = Read
0 = Write
In I^2C Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress
Logical OR of this bit with SEN, RSEN, PEN, RCEN, or ACKEN will indicate if the MSSP is in IDLE mode.

UA: Update Address (10-bit I^2C mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

BF: Buffer Full Status bit
Receive (SPI and I^2C modes):
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty
Transmit (I^2C mode only):
1 = Data transmit in progress (does not include the $\overline{\text{ACK}}$ and STOP bits), SSPBUF is full
0 = Data transmit complete (does not include the $\overline{\text{ACK}}$ and STOP bits), SSPBUF is empty

SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

WCOL: Write Collision Detect bit
Master mode:
1 = A write to SSPBUF was attempted while the I2C conditions were not valid
0 = No collision
Slave mode:
1 = SSPBUF register is written while still transmitting the previous word (must be cleared in software)
0 = No collision

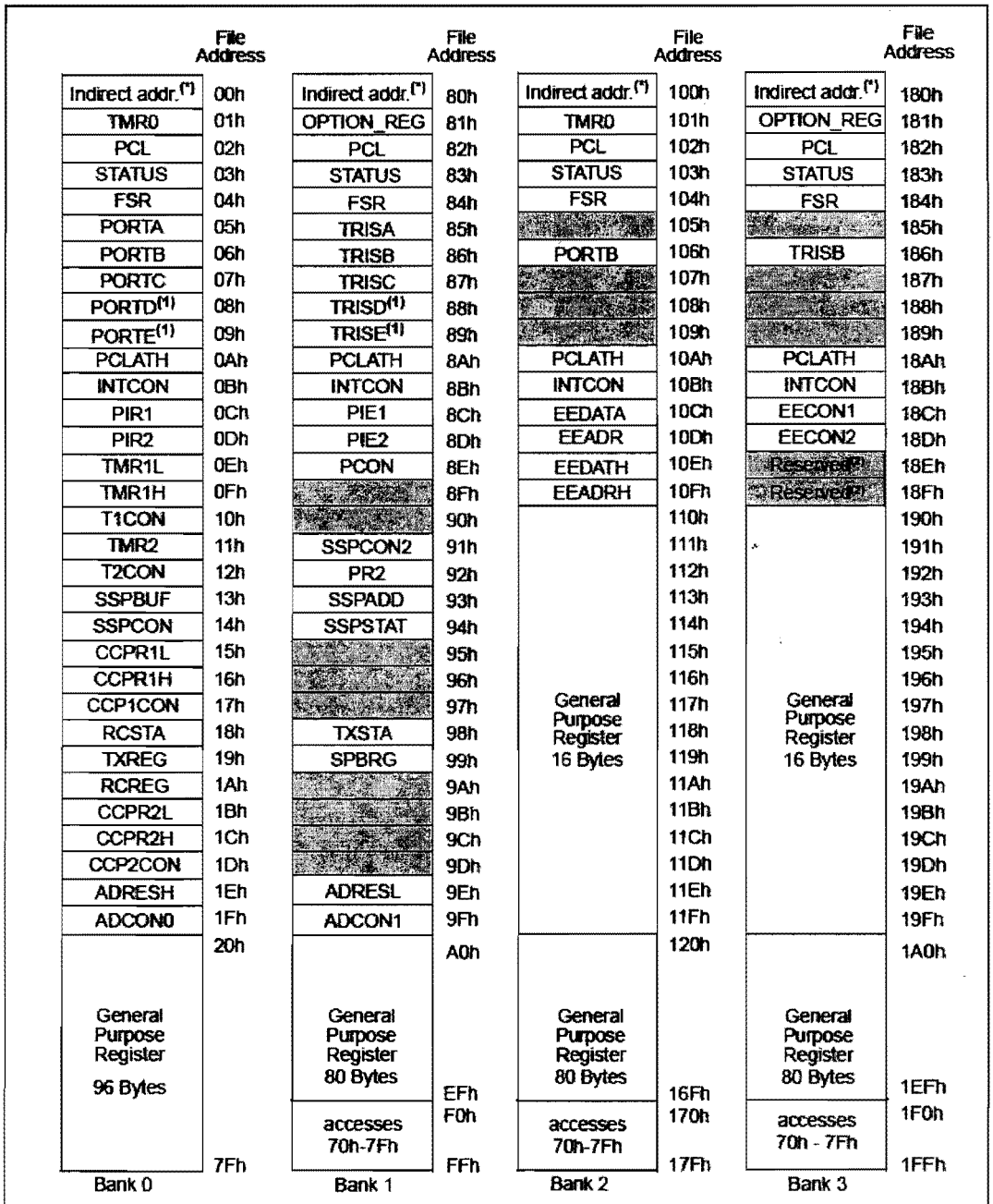
SSPOV: Receive Overflow Indicator bit
In SPI mode:
1 = A new byte is received while SSPBUF holds previous data. Data in SSPSR is lost on overflow. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid overflows. In Master mode, the overflow bit is not set, since each operation is initiated by writing to the SSPBUF register. (Must be cleared in software.)
0 = No overflow
In I^2C mode:
1 = A byte is received while the SSPBUF is holding the previous byte. SSPOV is a "don't care" in Transmit mode. (Must be cleared in software.)
0 = No overflow

SSPEN: Synchronous Serial Port Enable bit
In SPI mode,
When enabled, these pins must be properly configured as input or output
1 = Enables serial port and configures SCK, SDO, SDI, and SS as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins
In I^2C mode,
When enabled, these pins must be properly configured as input or output
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

CKP: Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In I^2C Slave mode:
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I^2C Master mode:
Unused in this mode

SSPM3:SSPM0: Synchronous Serial Port Mode Select bits
0000 = SPI Master mode, clock = Fosc/4
0001 = SPI Master mode, clock = Fosc/16
0010 = SPI Master mode, clock = Fosc/64
0011 = SPI Master mode, clock = TMR2 output/2
0100 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
0101 = SPI Slave mode, clock = SCK pin. SS pin control disabled. SS can be used as I/O pin.
0110 = I^2C Slave mode, 7-bit address
0111 = I^2C Slave mode, 10-bit address
1000 = I^2C Master mode, clock = Fosc / (4 * (SSPADD+1))
1011 = I^2C Firmware Controlled Master mode (slave idle)
1110 = I^2C Firmware Controlled Master mode, 7-bit address with START and STOP bit interrupts enabled
1111 = I^2C Firmware Controlled Master mode, 10-bit address with START and STOP bit interrupts enabled
1001, 1010, 1100, 1101 = Reserved

PIC16F877/876 REGISTER FILE MAP



Unimplemented data memory locations, read as '0'.
* Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.
2: These registers are reserved, maintain these registers clear.

ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$		ADON
bit 7							bit 0

ADCS1:ADCS0: A/D Conversion Clock Select bits
00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (clock derived from the internal A/D module RC oscillator)

CHS2:CHS0: Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 4, (RA5/AN4)
101 = channel 5, (RE0/AN5)[1]
110 = channel 6, (RE1/AN6)[1]
111 = channel 7, (RE2/AN7)[1]

GO/$\overline{\text{DONE}}$: A/D Conversion Status bit
If ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

Unimplemented: Read as '0'

ADON: A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM				PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

ADFM: A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

Unimplemented: Read as '0'

PCFG3:PCFG0: A/D Port Configuration Control bits:

PCFG3: PCFG0	AN7[1] RE2	AN6[1] RE1	AN5[1] RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs[2]
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input D = Digital I/O

Note 1: These channels are not available on PIC16F873/876 devices.
2: This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

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