



UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 6 Examination in Engineering: November 2017

Module Number: EE6304

Module Name: Embedded Systems Design

[3 Hours]

[Answer all questions]

Q1 Figure Q1 shows part of the PIC1684A internal architecture. Study Figure Q1 and answer the following questions.

- a) What is the purpose of the Stack block? [2 Marks]
- b) What is the size of each memory location in the Program Memory? [2 Marks]
- c) What is the purpose of the 1-bit dashed wire? [2 Marks]
- d) Explain why the Program Counter is connected to the Data Bus? [2 Marks]
- e) What happens inside the microcontroller hardware when the instruction `call 34` is executed? [2 Marks]

- Q2
- a) State the function of the EQU directive. [1 Mark]
 - b) State the difference between a subroutine and a macro, and one advantage of each. [1 Mark]
 - c) Describe the function of the standard header file "P16F877A.INC". [1 Mark]
 - d) State the assembler directive essential to any program, and its function. [1 Mark]
 - e) Identify the five main symbols used in a flowchart. [1 Mark]
 - f) Explain why a pull-up resistor needed with a switch input. [1 Mark]

- g) State three methods of switch de-bouncing.
[1 Mark]
- h) Briefly explain the scanning process used to read a keypad button.
[1 Mark]
- i) Draw a block diagram of a circuit with a keypad and 7-segment display, indicating the main components and signals in the system.
[2 Mark]

- Q3
- a) Explain how conditional program jumps are implemented in the PIC MCU.
[1 Mark]
 - b) How many bits does the 8k MCU program memory contain?
[0.5 Mark]
 - c) Describe briefly the process of fetching an instruction in a microcontroller.
[1 Mark]
 - d) State the advantages of flash read only memory (ROM), compared to other memory types.
[1 Marks]
 - e) Briefly compare the operation of a subroutine and an interrupt, explaining the role of the stack, return address, interrupt flag and the special significance of address 004 in the P16XXX.
[1.5 Marks]
 - f) Describe the advantages of in-circuit programming and debugging over the corresponding conventional development process.
[1 Mark]
 - g) Identify two instructions, one of which must be placed last in the PIC source code. What happens if one of these is not used?
[1 Mark]
 - h) Explain how a pre-scaler extends the timer period.
[1 Mark]
 - i) Explain why the plain 7-segment LED display needs a code table and why a BCD encoded display does not need a code table.
[2 Marks]

Q4 Answer the following questions by referring to the given program in Figure Q4 (a) which is written in assembly language. This program initiates the PIC ports according to requirement to power on a light emitting diode (LED) array by connecting push buttons to the PIC 16F877 microcontroller. (Refer the supplementary Figures of Q4 (b) and Q4 (c))

- Propose a title for the given program in Figure Q4 (a).
[1 Mark]
- Draw a flow chart to demonstrate the given program in Figure Q3 (a).
[1 Mark]
- Draw a flowchart to demonstrate the operation of the system?
[3 Marks]
- Propose appropriate comments and fill the blank space shown in the program. (Refer instructions given in the Table Q4) (a)
[2 Marks]
- Draw a circuit diagram to implement the hardware part connecting all resources and component.
[3 Marks]

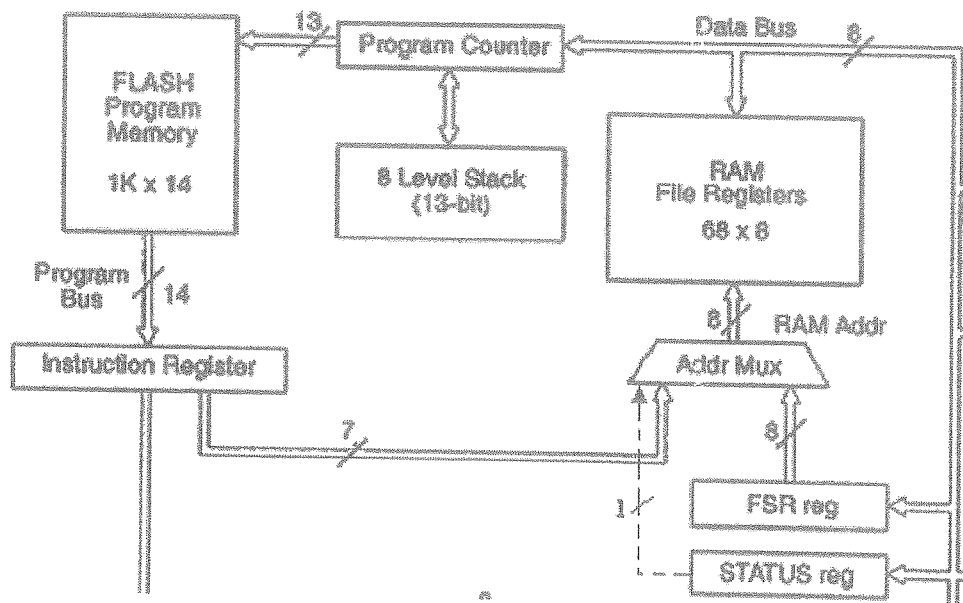


Figure Q1: PIC 16F84A architecture.

```

; FOR PIC 16F877      40 PIN DEVICE
; RESONATOR          10MHz
; WATCHDOG           DISABLED

```

```

TITLE "(1).....
list -p = 16f877
include <p16f877.inc>

```

```

BANK0 macro
bcf STATUS,RP0
bcf STATUS,RP1
endm

```

```

BANK1 macro
bsf STATUS,RP0
bcf STATUS,RP1
endm

```

```

BANK2 macro
bcf STATUS,RP0
bsf STATUS,RP1
endm

```

```

BANK3 macro
bsf STATUS,RP0
bsf STATUS,RP1
endm

```

```

ORG 0 ;START ADDRESS
NOP
GOTO START ;PROGRAM START

```

```

;***** THE ROUTINES START HERE *****

```

```

ORG 0x20
START CALL INITP ;INITIALISE PORTS

```

```

REPEAT
MOVWF PORTA,W;(2).....
MOVWF PORTD;(3).....
GOTO REPEAT ;REPEAT FOREVER

```

```

; PORT INIT ROUTINE
INITP BANK1
MOVLW 06;(4).....
MOVWF ADCON1
MOVLW 0xFF
MOVWF TRISA;(5).....
BANK0;(6).....
RETURN
END

```

Figure Q4 (a): Assembly program.

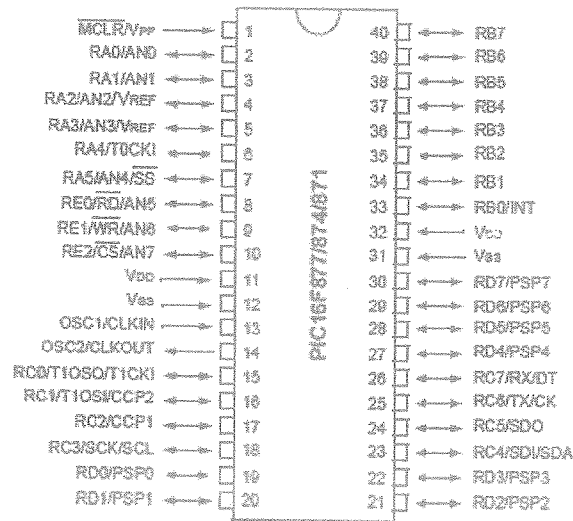


Figure Q4 (b): Pin diagram of 16F87X.

File Address	File Address	File Address	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 ^(†) 08h	TRISD ^(†) 88h	108h	188h
PORTE ^(†) 09h	TRISE ^(†) 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 ^(†) 18Eh
TMR1H 0Fh	8Fh	EEADRH 10Fh	Reserved ^(†) 18Fh
T1CON 10h	90h	110h	190h
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADQ 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h	95h		
CCPR1H 16h	96h		
CCP1CON 17h	97h		
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah	9Ah		
CCPR2L 1Bh	9Bh		
CCPR2H 1Ch	9Ch		
CCP2CON 1Dh	9Dh		
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
20h	A0h	120h	1A0h
General Purpose Register 96 Bytes	General Purpose Register 96 Bytes	accesses 20h-7Fh	accesses A0h - FFh
Bank 0 7Fh	Bank 1 FFh	Bank 2 17Fh	Bank 3 1FFh
		16Fh 170h	1EFh 1F0h

Figure Q4 (c): Data memory organization.

Table 1: 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
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	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	$\overline{\text{TO,PD}}$	
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	$\overline{\text{TO,PD}}$	
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.

Table 2: Special Function Registrars

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on RESET	Details on page
Bank 0											
00h	INDF	Uses contents of FSR to address Data Memory (not a physical register)								---- --	11
01h	TMR0	8-bit Real-Time Clock/Counter								XXXX XXXX	20
02h	PCL	Low Order 8 bits of the Program Counter (PC)								0000 0000	11
03h	STATUS ⁽²⁾	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1XXX	8
04h	FSR	Indirect Data Memory Address Pointer 0								XXXX XXXX	11
05h	PORTA ⁽⁴⁾	—	—	—	RA4/T0CK1	RA3	RA2	RA1	RA0	---x XXXX	16
06h	PORTB ⁽⁵⁾	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	XXXX XXXX	18
07h	—	Unimplemented location, read as '0'								—	—
08h	EEDATA	EEPROM Data Register								XXXX XXXX	13,14
09h	EEADR	EEPROM Address Register								XXXX XXXX	13,14
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of the PC ⁽¹⁾				---0 0000	11	
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	10
Bank 1											
90h	INDF	Uses Contents of FSR to address Data Memory (not a physical register)								---- --	11
91h	OPTION_REG	RBFU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	9
92h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	11
93h	STATUS ⁽²⁾	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1XXX	8
94h	FSR	Indirect data memory address pointer 0								XXXX XXXX	11
95h	TRISA	—	—	—	PORTA Data Direction Register				---1 1111	16	
96h	TRISB	PORTB Data Direction Register								1111 1111	18
97h	—	Unimplemented location, read as '0'								—	—
98h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	13
99h	EECON2	EEPROM Control Register 2 (not a physical register)								---- --	14
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC ⁽¹⁾				---0 0000	11	
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	10

Legend: x = unknown, u = unchanged, — = unimplemented, read as '0', q = value depends on condition

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> are never transferred to PCLATH.

2: The TO and PD status bits in the STATUS register are not affected by a MCLR Reset.

3: Other (non power-up) RESETS include: external RESET through MCLR and the Watchdog Timer Reset.

4: On any device RESET, these pins are configured as inputs.

5: This is the value that will be in the port output latch.

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
RBPUP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

bit 7

bit 0

bit 7 **RBPUP**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **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

bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **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

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

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	TO	PD	Z	DC	C

bit 7

bit 0

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

bit 5 **RP0**: Register Bank Select bits (used for direct addressing)
01 = Bank 1 (80h - FFh)
00 = Bank 0 (00h - 7Fh)

bit 4 **TO**: Time-out bit
1 = After power-up, CLRWDI instruction, or SLEEP instruction
0 = A WDT time-out occurred

bit 3 **PD**: Power-down bit
1 = After power-up or by the CLRWDI instruction
0 = By execution of the SLEEP 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 (ADDWF, ADDLW, SUBLW, SUBWF 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 (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for 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.

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

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	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7							bit 0

- bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts
- bit 6 **EEIE:** EE Write Complete Interrupt Enable bit
1 = Enables the EE Write Complete Interrupts
0 = Disables the EE Write Complete interrupt
- bit 5 **TOIE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt
- bit 4 **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt
- bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt
- bit 2 **TOIF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow
- bit 1 **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
- bit 0 **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

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