St. Name:

St. Number:

Final Exam, First Semester 2009/2010

<b>Course Title:</b>	Embedded Systems Design	Date:	26/1/2010
<b>Course No:</b>	0630470	Time Allowed:	2 hours
<b>Instructor:</b>	Prof. Kasim Al-Aubidy	No. of Pages:	6

## **Instructions:**

- Write your name and number on each page of the exam.
- Consider the PIC 16F84 Microcontroller running from 8 MHz crystal frequency.
- Use the free space after each assignment for your answer.
- Write your answer readable in English.

## Question 1:

8.

## Objectives: Basic concepts of Embedded Systems.

[25%]

## [A]. Choose the correct answer:

- 1. The baud rate is;
  - (a). the total number of bits that can be transmitted.
  - (b). the bandwidth of the serial interface.
  - (c). the total number of bits per unit time that can be transmitted.
- 2. CISC processors normally have;
  - (a). a large number of general-purpose registers.
  - (b). variable instruction length.
  - (c). many addressing modes.
- 3. Each PIC instruction cycle takes
  (a). 2 clock periods.
  (b). 4 clock periods.
  (c). single clock period.
- 4. The RISC architecture executes most instructions in
  - (a). one clock cycle.
  - (b). two clock cycles.
  - (c). three or more clock cycles.
- 5. The program counter of the PIC 16F84 is
- (a). 10-bit wide. (b). 12-bit wide. (c). 14-bit wide.
- 6. The PIC 16F84 program is stored in a

(a). flash RAM at address  $00C_H$ .

(b). flash RAM at address  $000_{H}$ .

- (c). flash ROM at address  $000_{H}$ .
- 7. To make PortB an output port, you must load register TRISB with

(a).  $80_H$  (b).  $00_H$  (c).  $FF_H$ 

To rest the microcontroller if it is ever allowed to overflow use (a). reset (b). sleep (c). wdt **[B].** Embedded systems can be implemented either using single chip microcontrollers or field programmable chips (such as FPGA or FPAA). List the differences between the following:

Using Microcontroller	Using Field Programmable Chip
	Using Microcontroller

	Using FPGA	Using FPAA
1		
2		

**[C].** The Harvard structure gives some clear advantages over conventional von Neumann structure. Can you think of any disadvantages? Show that with the help of diagrams?

	Harvard Structure	von Neumann structure
Diagram		
Advantage		
Disadvantage		

#### [25%]

Objectives: Basic concepts of Embedded Systems.

[A]. Three microcontrollers (A, B & C) have maximum clock speeds 10MHz, 20MHz, 24MHz respectively. Microcontroller A divides its clock by 4 to give one machine cycle, microcontroller B by 8, and microcontroller C by 12. Microcontrollers A & C take 2 machine cycles to perform an instruction, while microcontroller B takes three cycles.

Place the microcontrollers in order of the speed in which they can perform that instruction?

- **[B].** If the INTCON register is set to A8<sub>H</sub>, then 1. Determine which interrupts are enabled?
  - 2. An interrupt occurs, and the INTCON register is found to have changed to  $A9_{H}$ , which interrupt source has called?
  - 3. Which bit must the designer change before the end of the ISR, and why?
- [C]. A small embedded system has an 8-bit microcontroller requires a regular interrupt every 128 μs.
  - 1. Show how this requirement can be achieved to the best accuracy possible?

	7	6	5	4	3	2	1	0
OPTION Reg								
INTCON Reg								

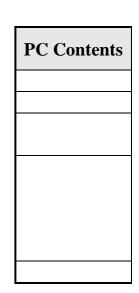
2. How should the requirement be met if the interrupt is to occur every 500  $\mu$ s?

	7	6	5	4	3	2	1	0
<b>OPTION Reg</b>								
INTCON Reg								
TMR0 Reg								

## **Question 3:**

## Objectives: Microcontroller Programming

- [A]. Consider this program:
  - 1. Give the size of sub1?
  - 2. List the changes (Full the table) in the contents of the program counter between the time of execution of the instruction before the call and the instruction following the call?
- start EQU 0365 sub1 EQU 079F start: instruction call sub1 instruction
- sub1: instruction 1 instruction 2 instruction 3 instruction 4 instruction 5 return



delay:	movlw	fa
	movwf	delayreg
loop:	nop	
	nop	
	decfsz	delayreg,1
	goto	loop
	return	

- **[B].** Consider the following delay subroutine: 1. Calculate the delay it actually introduces,
  - taking into account the time taken by every instruction?
  - 2. What number should be loaded into delay register to make the generated delay as close as possible to 500µs?
  - 3. Modify the given delay subroutine to introduce a time delay of 10 ms?

**[C].** Outline the procedures of installing a program in the PIC microcontroller during your project design?

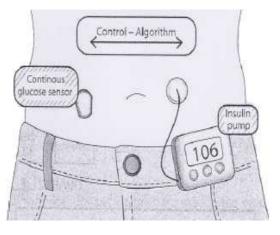
[20%]

### **Question 4:**

# Objectives: Hardware and Software Design of an Embedded System

[A]. The counter/timer of the microcontroller is used to measure the speed of an object falling through a column of water. When the object passes an upper sensor the controller starts the counter from zero. When it passes the lower sensor, it stops the counter and reads its value. The microcontroller is clocked from a **1.024** MHz crystal oscillator, and the counter is required to increment every **1 ms**. How should be the **OPTION** register be set?

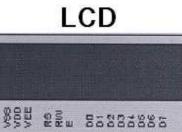
**[B].** You have been asked to design an embedded system to regulate glucose levels in the body of someone with diabetes by continuously measuring the level of glucose and dispensing insulin doses of based on those measurements. The chemical glucose sensor generates 4-bit digital signal. The insulin infusion pump is controlled by a **PWM** signal. The system has **ON/OFF** switch, **RESET** switch, alarm speaker and **LCD** for blood glucose monitoring.



- 1. Show the general layout of the required system?
- 2. Give the detailed hardware design of each unit in your design?
- 3. Draw a flowchart to demonstrate the operation of such a system?
- 4. If the microcontroller has no enough input/output lines, show how you can interface the LCD serially?

Beschption         Cycles         Msb         Lsb         Affected         Notes           BYTE-ORIENTED FILE REGISTER OPERATIONS	Mnem	onic.			Τ	14-Bit	Opcod	e	Status		]				
AND W with f       1       00       0101       dfff       ffff       Z       1,2         f       Clear f       1       00       0001       1fff       fff       Z       2         id       Complement f       1       00       0001       0011       dff       fff       Z       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,2       1,0       0,0       0010	Opera		Description	Cycles	MSb			LSb		Notes		ຼ [	TOIF		
AND W with f       1       00       0101       dfff       ffff       Z       1,2         f       Clear f       1       00       0001       1fff       ffff       Z       2         .d       Complement f       1       00       0001       dfff       ffff       Z       2         .d       Decrement f       1       00       0011       dfff       ffff       Z       1,2         .d       Decrement f       1       00       0011       dfff       ffff       Z       1,2         .d       Decrement f       1       00       0011       dfff       ffff       Z       1,2         .d       Increment f       1       00       0011       dfff       ffff       Z       1,2         .d       Increment f, Skip if 0       1 (2)       00       111       00       11,2       1,2       1,2         .d       Inclusive OR W with f       1       00       0000       0xx0       0000       1,2       1,2         .d       Rotate Left through Carry       1       00       100       0110       dfff       1,2       1,2         .d       Swap nibbles in f <t< th=""><th></th><th></th><th>BYTE-ORIENTED FILE RE</th><th>GISTER OPE</th><th colspan="6">TER OPERATIONS</th><th>Bu</th><th colspan="4"></th></t<>			BYTE-ORIENTED FILE RE	GISTER OPE	TER OPERATIONS						Bu				
f       Clear f       1       00       0001       1fff       fff       Z       Z         -       Clear W       1       00       0001       0xxx xxxx       Z       Z       L       L         -       Clear W       1       00       0001       0xx xxxx       Z       L	ADDWF	f, d		· ·							∕at	$\rightarrow 1$	N Se Se		
1       00       100       100       12       12       12         1       00       000       11       12       12       12       12         1       00       000       11       12       12       12       12       12         1       00       010       11       10       000       111       12       12       12       12         1       00       12       00       111       11       12       12       12       12       12         1       1       00       100       111       111       12	ANDWF	f, d									-		5		
1       00       1001       0101       12       1,2       1,2         1       00       0011       dfff       ffff       Z       1,2       1,2         1       00       0111       dfff       ffff       Z       1,2       1,2         1,d       Increment f       1       00       0111       dfff       ffff       Z       1,2         1,d       Increment f       1       00       1010       dfff       ffff       Z       1,2         1,d       Increment f       1       00       1010       dfff       ffff       Z       1,2         1,d       Inclusive OR W with f       1       00       1000       dfff       ffff       Z       1,2         1,d       Move f       1       00       1000       dfff       ffff       Z       1,2         1,d       Rotate Right through Carry       1       00       1000       0110       dfff       1,2       1,2         1,d       Subtract W from f       1       00       100       0110       dfff       1,2       1,2       1,2         1,d       Bit Clear f       1       1       01       100b b b	CLRF	,								2			T g		
1       00       1001       0101       12       1,2       1,2         1       00       0011       dfff       ffff       Z       1,2       1,2         1       00       0111       dfff       ffff       Z       1,2       1,2         1,d       Increment f       1       00       0111       dfff       ffff       Z       1,2         1,d       Increment f       1       00       1010       dfff       ffff       Z       1,2         1,d       Increment f       1       00       1010       dfff       ffff       Z       1,2         1,d       Inclusive OR W with f       1       00       1000       dfff       ffff       Z       1,2         1,d       Move f       1       00       1000       dfff       ffff       Z       1,2         1,d       Rotate Right through Carry       1       00       1000       0110       dfff       1,2       1,2         1,d       Subtract W from f       1       00       100       0110       dfff       1,2       1,2       1,2         1,d       Bit Clear f       1       1       01       100b b b	CLRW												<u>d</u>		
Inclusive OR W with f       1       00       0100       dfff       ffff       Z       1,2         f       Move f       1       00       1000       dfff       ffff       Z       1,2         f       Move W to f       1       00       0000       lfff       ffff       Z       1,2         f       Move W to f       1       00       0000       0xx0       0000       lff         -       No Operation       1       00       1001       dfff       ffff       C       1,2         id       Rotate Right f through Carry       1       00       1000       dfff       ffff       C       1,2         id       Subtract W from f       1       00       0100       dfff       ffff       1,2         id       Exclusive OR W with f       1       00       0110       dfff       ffff       1,2         id       Bit Clear f       1       01       00bb       bfff       ffff       1,2         id       Bit Set f       1       01       01bb       bfff       ffff       1,2         id       Bit Test f, Skip if Set       1 (2)       01       10bb       bfff <td< td=""><td>COMF</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	COMF														
Inclusive OR W with f       1       00       0100       dfff       ffff       Z       1,2         f       Move f       1       00       1000       dfff       ffff       Z       1,2         f       Move W to f       1       00       0000       lfff       ffff       Z       1,2         f       Move W to f       1       00       0000       0xx0       0000       lff         -       No Operation       1       00       1001       dfff       fff       C       1,2         id       Rotate Right f through Carry       1       00       1000       dfff       ffff       C       1,2         id       Subtract W from f       1       00       0100       dfff       ffff       1,2         id       Swap nibbles in f       1       00       0110       dfff       ffff       1,2         id       Exclusive OR W with f       1       01       00 bbb bfff       ffff       1,2         id       Bit Clear f       1       01       01bb bfff       ffff       1,2       1,2         ib       Bit Set f       1       1       10       10bb bfff       1,2 <t< td=""><td>DECF</td><td></td><td></td><td>[ ·</td><td></td><td></td><td></td><td></td><td>Z</td><td></td><td></td><td>E E</td><td> lä</td></t<>	DECF			[ ·					Z			E E	lä		
Inclusive OR W with f       1       00       0100       dfff       ffff       Z       1,2         f       Move f       1       00       1000       dfff       ffff       Z       1,2         f       Move W to f       1       00       0000       lfff       ffff       Z       1,2         f       Move W to f       1       00       0000       0xx0       0000       lff         -       No Operation       1       00       1001       dfff       fff       C       1,2         id       Rotate Right f through Carry       1       00       1000       dfff       ffff       C       1,2         id       Subtract W from f       1       00       0100       dfff       ffff       1,2         id       Swap nibbles in f       1       00       0110       dfff       ffff       1,2         id       Exclusive OR W with f       1       01       00 bbb bfff       ffff       1,2         id       Bit Clear f       1       01       01bb bfff       ffff       1,2       1,2         ib       Bit Set f       1       1       10       10bb bfff       1,2 <t< td=""><td>DECFSZ</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>-</td><td></td><td> </td><td>2</td><td>ËŽ 🛛</td></t<>	DECFSZ								-			2	ËŽ 🛛		
Inclusive OR W with f       1       00       0100       dfff       ffff       Z       1,2         f       Move f       1       00       1000       dfff       ffff       Z       1,2         f       Move W to f       1       00       0000       lfff       ffff       Z       1,2         f       Move W to f       1       00       0000       0xx0       0000       lff         -       No Operation       1       00       1001       dfff       fff       C       1,2         id       Rotate Right f through Carry       1       00       1000       dfff       ffff       C       1,2         id       Subtract W from f       1       00       0100       dfff       ffff       1,2         id       Swap nibbles in f       1       00       0110       dfff       ffff       1,2         id       Exclusive OR W with f       1       01       00 bbb bfff       ffff       1,2         id       Bit Clear f       1       01       01bb bfff       ffff       1,2       1,2         ib       Bit Set f       1       1       10       10bb bfff       1,2 <t< td=""><td>NCF NCFSZ</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>4</td><td></td><td></td><td>E E</td><td>월응 [호</td></t<>	NCF NCFSZ								4			E E	월응 [호		
i.d       Move f       1       00       1000       dff ffff       1,2         f       Move W to f       1       00       0000       0xx0       0000       0xx0       0000         i.d       Rotate Left fthrough Carry       1       00       1000       dfff fffff       C       1,2         i.d       Rotate Left fthrough Carry       1       00       1000       dfff fffff       C       1,2         i.d       Rotate Right fthrough Carry       1       00       100       dfff fffff       C       1,2         i.d       Subtract W from f       1       00       0010       dfff fffff       C,DC,Z       1,2         i.d       Swap nibbles in f       1       00       0110       dfff fffff       Z       1,2         i.d       Bit Clear f       1       01       00bb bfff fffff       1,2       1,2         i.d       Bit Set f       1       01       01bb bfff fffff       1,2       1,2         i.b       Bit Test f, Skip if Clear       1 (2)       01       10bb bfff fffff       3       3         i.b       Bit Test f, Skip if Set       1 (2)       01       11bb bfff fffff       3       3 <t< td=""><td>ORWF</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>-</td><td></td><td></td><td>6</td><td>=0  0</td></t<>	ORWF								-			6	=0  0		
f       Move W to f       1       00       0000       1fff       fiff         -       No Operation       1       00       0000       0xx0       0000       0xx0       0000         i, d       Rotate Left f through Carry       1       00       1100       dfff       ffff       C       1,2         i, d       Rotate Right f through Carry       1       00       1100       dfff       ffff       C       1,2         i, d       Subtract W from f       1       00       1100       dfff       ffff       1,2       1,2         i, d       Swap nibbles in f       1       00       0110       dfff       ffff       1,2       1,2         i, d       Exclusive OR W with f       1       00       0110       dfff       ffff       1,2       1,2         i, d       Exclusive OR W with f       1       01       00bb       bfff       ffff       1,2       1,2         i, b       Bit Clear f       1       01       00bb       bfff       1,2       3       3         i, b       Bit Test f, Skip if Clear       1 (2)       01       10bb       bfff       fffff       3       3	NOVE								_			: 	©		
-       No Operation       1       00       0000       0xx0       0x00       0x0	NOVE								~	1,2					
Id       Rotate Left f through Carry       1       00       1101       dfff       ffff       C       1,2         Id       Rotate Right f through Carry       1       00       1100       dfff       fff       C       1,2         Id       Subtract W from f       1       00       1100       dfff       fff       C       1,2         Id       Swap nibbles in f       1       00       1100       dfff       ffff       1,2         Id       Exclusive OR W with f       1       00       110       dfff       ffff       1,2         Id       Exclusive OR W with f       1       00       0110       dfff       ffff       1,2         Id       Bit Clear f       1       01       00bb       bfff       ffff       1,2         Id       Bit Set f       1       01       01bb       bfff       ffff       1,2         Id       Bit Test f, Skip if Clear       1 (2)       01       10bb       bfff       1,2       3         Id       D       10bb       bfff       ffff       3       3       3       3         Id       D       1       11       111x       kkkkk	NOP											/	Å		
i.d       Rotate Right f through Carry       1       00       1100       dtf f ffff       C       1.2         i.d       Subtract W from f       1       00       0010       dfff       ffff       1.2       1.2         i.d       Swap nibbles in f       1       00       0110       dfff       ffff       1.2       1.2         i.d       Exclusive OR W with f       1       00       0110       dfff       ffff       1.2       1.2         BIT-ORIENTED FILE REGISTER OPERATIONS         Store for the second secon	RLF							****	0	10	4		ő		
i.d       Subtract W from f       1       00       0010       dfff       ffff       C,DC,Z       1,2         i.d       Swap nibbles in f       1       00       0110       dfff       ffff       1,2       1,2         i.d       Exclusive OR W with f       1       00       0110       dfff       ffff       1,2       1,2         BIT-ORIENTED FILE REGISTER OPERATIONS         Site Clear f         i.b       Bit Clear f       1       01       00bb       bfff       ffff       1,2         j.b       Bit Set f       1       01       01bb       bfff       ffff       1,2         j.b       Bit Test f, Skip if Clear       1 (2)       01       10bb       bfff       ffff         j.b       Bit Test f, Skip if Set       1 (2)       01       11bb       bfff       ffff       3         LITERAL AND CONTROL OPERATIONS       I       11       111       111x       kkkkk       kkkkk       C,DC,Z       I	RF														
b     Bit Set f     1     01     01bb     bfff     1,2       i, b     Bit Test f, Skip if Clear     1 (2)     01     10bb     bfff     ffff       i, b     Bit Test f, Skip if Set     1 (2)     01     10bb     bfff     ffff       i, b     Bit Test f, Skip if Set     1 (2)     01     11bb     bfff     ffff       i     LITERAL AND CONTROL OPERATIONS     I     11     111x     kkkk     kkkk     C,DC,Z	SUBWF								-				9		
b     Bit Set f     1     01     01bb     bfff     1,2       i, b     Bit Test f, Skip if Clear     1 (2)     01     10bb     bfff     ffff       i, b     Bit Test f, Skip if Set     1 (2)     01     10bb     bfff     ffff       i, b     Bit Test f, Skip if Set     1 (2)     01     11bb     bfff     ffff       i     LITERAL AND CONTROL OPERATIONS     I     11     111x     kkkk     kkkk     C,DC,Z	SWAPF								0,00,2		1		SC leta		
b     Bit Set f     1     01     01bb     bfff     1,2       i, b     Bit Test f, Skip if Clear     1 (2)     01     10bb     bfff     ffff       i, b     Bit Test f, Skip if Set     1 (2)     01     10bb     bfff     ffff       i, b     Bit Test f, Skip if Set     1 (2)     01     11bb     bfff     ffff       i     LITERAL AND CONTROL OPERATIONS     I     11     111x     kkkk     kkkk     C,DC,Z	KORWF								7						
b     Bit Set f     1     01     01bb     bfff     1,2       i, b     Bit Test f, Skip if Clear     1 (2)     01     10bb     bfff     ffff       i, b     Bit Test f, Skip if Set     1 (2)     01     10bb     bfff     ffff       i, b     Bit Test f, Skip if Set     1 (2)     01     11bb     bfff     ffff       i     LITERAL AND CONTROL OPERATIONS     I     11     111x     kkkk     kkkk     C,DC,Z		1, 0		ISTER OPER	-		0111	1111	-	1,2	1		PS1		
b         Bit Set f         1         01         01bb         bfff         1,2         3           i, b         Bit Test f, Skip if Clear         1 (2)         01         10bb         bfff         1         3           i, b         Bit Test f, Skip if Set         1 (2)         01         11bb         bfff         1         3           LITERAL AND CONTROL OPERATIONS           k         Add literal and W         1         11         11x         kkkk         kkkk         C,DC,Z         C	BCF	1.5			1					1.0	-		18 <sup>-</sup> 8'		
b         Bit Test f, Skip if Clear         1 (2)         01         10bb         bfff         3         3         4         3         3         3         4         1	3CF 3SF												μ ŭ		
b         Bit Test f, Skip if Set         1 (2)         01         11bb         bfff         3         3           LITERAL AND CONTROL OPERATIONS           k         Add literal and W         1         11         111x         kkkk         C,DC,Z         Image: C,DC,Z <td>BTFSC</td> <td></td> <td>- <b>-</b></td> <td></td>	BTFSC											- <b>-</b>			
k Add literal and W 1 11 111x kkkk kkkk C,DC,Z	BTFSC														
k Add literal and W 1 11 111x kkkk kkkk C,DC,Z	biraa	1, 0	La construction of the second se			1100	DIII	IIII		3		>	ë		
	ADDLW	k				1111x	kkkk	le le le le	C.DC.Z	1	- 1		τ Γ		
k Call subrating	ANDLW											(			
					1				-			ち レ			
- Clear Watchdog Timer 1 00 0000 0110 0100 TO PD 8		-							TO.PD		2	<u> </u>			
		k		2						1	ŭ	:	L 13		
		k	Inclusive OR literal with W	1	11	1000		kkkk	z				ž ř		
K Inclusive OH Interal with W I 1   11 1000 kkkk kkkk   Z   I   X F	NOVLW	k	Move literal to W	1	11	00xx		kkkk	_	1			28		
k         Inclusive OR literal with W         1         11         1000         kkk         kkkk         Z          S          S          S          S          S          S          S           S           S           S           S           S            S		-	Return from interrupt	2	00	0000	0000	1001							
k         Inclusive OR literal with W         1         11         1000         kkkk         kkkk         Z         Implementation         Second state         Sec		k	Return with literal in W	2	11			kkkk		1			-⊿ ¥		
k         Inclusive OR literal with W         1         11         1000         kkkk         kkkk         Z           k         Move literal to W         1         11         1000x         kkkk         kkkk         Z         Implementation         Second to the second to			Return from Subroutine	2	00	0000	0000	1000		1			E.		
k         Move literal to W         1         11         1000         kkkk         Kkkkkkkkkkkkk         Kkkkkkkkkkkkkkkkkkkkkkkkkkkkk	SLEEP	-	Go into standby mode	1	00	0000	0110	0011	TO,PD	1					
k         Move literal to W         1         11         1000         kkkk         kkkk         2           -         Return from interrupt         2         0         0000         0000         1001         Image: Constraint of the second s	SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z						
k         Move literal to W         1         11         1000         kkkk         kkkk         2           -         Return from interrupt         2         0         0000         0001         1000	XORLW	k	Exclusive OR Iteral with W	1	11	1010	kkkk	kkkk	Z	1					
- Clear Watchdog Timer 1 00 0000 0110 0100 TO,PD 2	ANDLW CALL CLRWDT GOTO IORLW MOVLW RETFIE RETLW RETFIE RETLW RETURN SLEEP SUBLW XORLW		Call subroutine Clear Watchdog Timer Go to address Inclusive OR literal with W Move literal to W Return from Interrupt Return with literal in W Return from Subroutine Go into standby mode Subtract W from literal	2 1 2 1 2 2 2 1 1	10 00 10 11 11 00 00 11 00 00 11	0%kk 0000 1%kk 1000 00%x 0000 01%x 0000 01%x	kkkk 0110 kkkk kkkk kkkk 0000 kkkk 0000 0110 kkkk	kkkk 0100 kkkk kkkk 1001 kkkk 1000 0011 kkkk	Z TO,PD Z TO,PD C,DC,Z		Energy				
									z				1 K		
K Inclusive OH Ineral with W 1 111 1000 kkkk kkkk Z		k										$\sim$	/1¢		
K     Inclusive OH interal with W     1     11     1000 kkkk kkkk     Z       k     Move literal to W     1     11     1000 x kkkk kkkk     Z													73		
k         Inclusive OH Interal with W         1         11         1000 kkkk kkkk         Z           k         Move literal to W         1         11         1000 xkkkk kkkk         Z         Image: Comparison of the state of the sta		ĸ								1		_			
k         Move literal to W         1         11         1000         kkkk         kkkkkkkkkkkk         kkkk         kkkkkkkk		•			1				70.00	1					
k         Move literal to W         1         11         1000 kkkk kkkk         2           -         Return from interrupt         2         00         0000         0001         1           k         Return from Subroutine         2         11         01xx         kkkk kkkk         1         1         0000         0000         1001         1         1         1         0000         0000         1         1         1         00000         1         1         1         1         00000         1         1         1         1         1         00000         0000         1 </td <td></td> <td>-</td> <td></td> <td></td> <td></td> <td>****</td> <td></td> <td></td> <td></td> <td>1</td> <td></td> <td></td> <td></td>		-				****				1					
k         Move literal to W         1         11         1000         kkkk         kkkk         2           -         Return from interrupt         2         00         0000         1001         1         1         1000x         kkkk         kkkkk         kkkkk															
k         Move literal to W         1         11         1000         k kkk         k kkk         2           -         Return from interrupt         2         00         0000         0001         001           k         Return from Subroutine         2         11         01xx         kkkk         kkkkk	NOTION .					1010	ABBA	haaa	-		1				
k         Move literal to W         1         11         1000         k kkk         k kkk         2           -         Return from interrupt         2         00         0000         0001         001           k         Return from Subroutine         2         00         0000         0001         0001           -         Return from Subroutine         2         00         0000         0100         0001           -         Go into standby mode         1         00         0000         011         TO,PD           k         Subtract W from literal         1         11         10x         kkkk         kkkk         G,DC,Z													_		
k     Move literal to W     1     11     1000     kkkk     kkkk     L       -     Return from interrupt     2     00     0000     1001       k     Return from Subroutine     2     11     01xx     kkkk     kkkk       -     Return from Subroutine     2     00     0000     1000       -     Go into standby mode     1     00     0000     011     TO,PD       k     Subtract W from literal     1     11     10x     kkkk     kkkk	RA2 🗲		18 🔤 + RA1				GI	F FFII			TOIE		INTCON		
k       Move Reral to W       1       11       1000       kkkk       kkkk       L         -       Return from interrupt       2       00       0000       1001       L	RA3 🔶	+0					U								
k       Move Reral to W       1       11       1000 KKK KKK       2         K       Move Reral to W       1       11       000 kK KKK       kKkk       2         Return from interrupt       2       00       0000       1000       1000       1         k       Return with literal in W       2       11       01xx       kkkk       kkkk       kkkk         -       Return from Subroutine       2       00       0000       0000       1000       1         -       Go into standby mode       1       00       0000       0110       0011       TO,PD         -       Go into standby mode       1       11       110x kkkk kkkk       Z       Z       D         -       Subtract W from literal       1       11       1100 kkkk kkkk       Z       Z       D         -       Return with W       1       11       1010 kkkk kkkk       Z       Z       D         -       Betwire OR Reral with W       1       11       1010 kkkk kkkk       Z       Z       D         -       18       Heral with W       1       11       1010 kkkk kkkk       Z       Z       D       D         - <td>RA4 🗲</td> <td>+0</td> <td></td> <td></td> <td></td> <td></td> <td>IR</td> <td>P RP1</td> <td>RP0 T</td> <td>O PD</td> <td>7 D</td> <td>C C</td> <td>STATUS</td>	RA4 🗲	+0					IR	P RP1	RP0 T	O PD	7 D	C C	STATUS		
R       Installate Of Initial With W       1       11       1000 KKK KKKK KKKK       2         K       Move Retard to W       1       11       1000 KKK KKKK KKKK       2       1         Return from interrupt       2       00       0000       0000       10001       1         k       Return with literal in W       2       11       01xx kkkk kkkk       1       1         -       Return from Subroutine       2       00       0000       0000       1000       1000         -       Go into standby mode       1       00       0000       0110       0011       TO,PD         -       Go into standby mode       1       11       110x kkkk kkkk       Z       Z         -       Subtract W from literal       1       11       1100 kkkk kkkk       Z       Z         -       Retari with W       1       11       1010 kkkk kkkk       Z       Z       Z         -       18       -       RA1       -       1       1010 kkkk kkkk       Z       Z       INTCON         -       18       -       RA1       -       -       -       -       -       -       -       -       -       <	MCLR -	+0										•			
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	Vss —	-	16F84			г									

RBPU	NTES	TOCS	TOSE	PSA	PS2	PS1	PS0	OPTION



Embedded Systems Design (0630470), Final Exam 26/1/2010

Vdd

RB7

RB6

RB5

++

10 **□**↔ RB4

RB0 ↔

RB1 ↔

RB2 ↔

RB3 ↔ 9