UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER I
SESSION 2012/2013

COURSE NAME	:	MICROCONTROLLER & MICROPROCESSOR
COURSE CODE	:	BEE 3233 / BEX 32003
PROGRAMME	:	BEE
EXAMINATION DATE	:	JANUARY 2013
DURATION	:	3 HOURS
INSTRUCTION	:	ANSWER **ALL** QUESTIONS IN **PART A** AND **FOUR (4)** QUESTIONS IN **PART B**

THIS QUESTION PAPER CONSISTS OF **EIGHTTEEN (18)** PAGES

PART A

Q1 (a) Reset is used for putting the microcontroller into a 'known' condition. Microcontroller can behave rather inaccurately under certain undesirable conditions. In order to continue its proper functioning it has to be reset, meaning all registers would be placed in a starting position. Microcontroller PIC16F84 knows several sources of resets. Give four examples of these sources of resets.

(4 marks)

(b) Microcontroller PIC16F84 can work with four different configurations of an oscillator. Figure Q1 shows one of the oscillation mode types that can be operated by this microcontroller. Based on the figure given:

(i) Point out the name of oscillation mode

(2 mark)

(ii) Analyze its architecture

(6 marks)

(c) There are basically two types of digital computer architectures. The first one is called Von Neumann architecture and later Harvard architecture was adopted for designing digital computers. The Von Neumann architecture was developed at Princeton University, while the Harvard architecture was the work of Harvard University. Illustrate and explain both architectures.

(8 marks)

PART B

Q2 (a) Program memory has been carried out in FLASH technology. The size of program memory is 1024 locations with 14 bits width. Define the function of program memory in PIC16F84.

(2 marks)

(b) Based on the coding as follows:

```
; Main Program
BUTTON    btfss   PORTA, 0
          goto    ON
          goto    OFF
ON        bsf     PORTB, 0
          goto    BUTTON
OFF       clrf    PORTB
          goto    BUTTON
end
```

(i) Analyze each of the instruction by giving a suitable comment for every row.

(7 marks)

(ii) Construct the flowchart for the above program.
(Note : the initialization part of the program is given as below).

(4 marks)

```
LIST P = 16F84
Include "P1684.inc"
bsf     STATUS, RP0
movlw   b '00000001'    ; RA0 is input
movwf   TRISA
movlw   b '00000000'    ; RB0 is output
movwf   TRISB
bcf     STATUS, RP0
```

(c) In PIC16F84 microcontroller, there is no specific command for multiplication operation. Hence, the *adding*, *decrement* and *bit manipulation* instruction need to be used for this purpose. Apply these three instructions concept to produce a program to multiply 9 with 4.

(5 marks)

(d) Compare the location to store the final result for the following instructions:

(i) COMF 0Ch, 1
(ii) COMF 0Eh, 0

(2 marks)

Q3 (a) Define the meaning of control directive *EQU* in assembly language element of PIC 16F84 microcontroller.

(2 marks)

(b) Based on figure Q3, produce a program that can interpret the flow of the flowchart given.

(5 marks)

(c) Based on the coding below, assuming that a crystal frequency of 7MHz is used as a clock generator.

```
DELAY          movlw   TIMEHIGH
               movwf   COUNTERHIGH
               movlw   TIMELOW
               movwf   COUNTERLOW

DELAYCOUNTER   decfsz  COUNTERLOW
               goto    DELAYCOUNTER

               movlw   TIMELOW
               movwf   COUNTERLOW
               decfsz  COUNTERHIGH
               goto    DELAYCOUNTER

               return
```

(i) decide a suitable time taken to process 1 instruction (Note: consider the frequency given and the normal cycles for microcontroller to process 1 instruction).

(2 marks)

(ii) predict its total time delay.

(4 marks)

(d) Based on the coding in A and B as follow;

A	B
bsf STATUS,RP0 movlw b '00001000' movwf TRISA	bcf STATUS,RP0 movlw b '00001111' movwf TRISB

(i) compare the purpose of opening the bank0 and bank1.

(1 mark)

(ii) differentiate the meaning of every instruction in A and B.

(6 marks)

Q4 (a) A 32KHz crystal frequency will produce a time out every 1 second if the prescaler value of 32 is selected. Defend this statement through calculation.
(2 marks)

(b) Figure Q4(b) shows the output of LED on Port B that is always ON. It is interrupted by an active low input from RB0/INT. The Interrupt Service Routine (ISR) will set the RB7=1 and LED will ON. Then, the program will return form interrupt (RETFIE). On the other hand, if the RB7=0, the LED will OFF and the program will RETFIE.

(i) Define the function of ISR.
(2 marks)

(ii) Predict which bit (in Interrupt Control Register (INTCON)) that need to be set as '1' to allow the interrupt happen for the above condition.
(4 marks)

(iii) By referring Figure Q4(b)(ii), produce a program for ISR part as below;
(5 marks)

```
#include "P16F84.INC"
            ORG     0x00
            GOTO    MAIN
            ORG     0x04
            GOTO ISR

MAIN        MOVLW   b '00000001'
            TRIS    PORT B
            MOVLW   b '10000000'
            OPTION
            CLRF    PORTB
            BSF     PORTB,7
            BSF     INTCON, INTE
            BSF     INTCON,GIE

LOOP        GOTO    LOOP
            END

ISR         _____________________
            _____________________
            _____________________
            _____________________
            _____________________

LAB1        BSF     PORTB,7
            RETFIE
```

(iv) Analyze the answer in (iii) by giving a suitable comment for every row.
(7 marks)

Q5 (a) Define the function of program counter (PC) in MC68000 microprocessor.
(2 marks)

(b) In the following sequence of instructions, produce the value of D4 after the execution of each instruction.

```
MOVEQ  # -2, D4
NEG.W  D4
EXT.L  D4
MULS   D4. D4
```

(4 marks)

(c) Differentiate the overflow flag bit in addition operation for both of the situation in table as follow:

source	**destination**
Positive number	Positive number
Negative number	Negative number

(2 marks)

(d) Given D0 = 0000 FFFF, D1 = 0000 0001, D2 = FFFF 8000 and D3 = 0000 7FFF. List the content of the register and status register CCR (XNZVC) after each arithmetic instruction below. (Note : each instruction is independent.)

(i) ADD.W D0 , D1
(ii) ADD.W D0 , D2
(iii) SUB.W D0 , D1
(iv) ADD.B D0 , D1

(7 marks)

(e) The coding below shows the program to calculate the mean for 5 numbers. Produce the instruction for LOOP part.

```
            ORG     $001000
            CLR.L   D0
            CLR.L   D1
            CLR.L   D2
            MOVE.B  #5, D1
            MOVEA.W #$00200, A0
LOOP        _____________
            _____________
            _____________
            _____________
MEAN        DIVU #5, D2
            BRA *
            ORG   $002000
            DC.B  $9,$AA,$BB,$CC,$DD
            END
```

(5 marks)

Q6 (a) Asynchronous bus control signal in MC68000 microprocessor contains five signals that needed for external peripheral operation. There are AS*, R/W*, UDS*, LDS* and DTACK*. Define the function for DTACK* signal and identify whether it is an output signal or an input signal.

(2 marks)

(b) Compare the states of UDS* and R/W* when the MC68000 microprocessor is involved in the following memory accesses:

(i) a word read from address $003000
(ii) a byte write to address $003001

(2 marks)

(c) Based on the specifications as follow:

Device	Design Space Size	Base address
U1 EPROM	8K words (16KB)	$000000
U2 EPROM	8K words (16KB)	$200000
U3 RAM	8K words (16KB)	$340000

(i) List the address range and the address lines for these 3 devices.

(6 marks)

(ii) Construct the memory table for all the address lines balance and show the minimum address lines that can be used as input to the decoder.

(5 marks)

(iii) Design the memory address decoder circuit based on (ii) that can be used to activate one device at a time by referring on the address line balance and total of device.

(4 marks)

(iv) Based on (iii), classify whether this decoder is a full decoder or a partial decoder.

(1 mark)

- END OF QUESTION –

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013
COURSE : MICROPROCESSOR & MICROCONTROLLER

PROGRAMME : BEE
COURSE CODE : BEE3233/BEX 32003

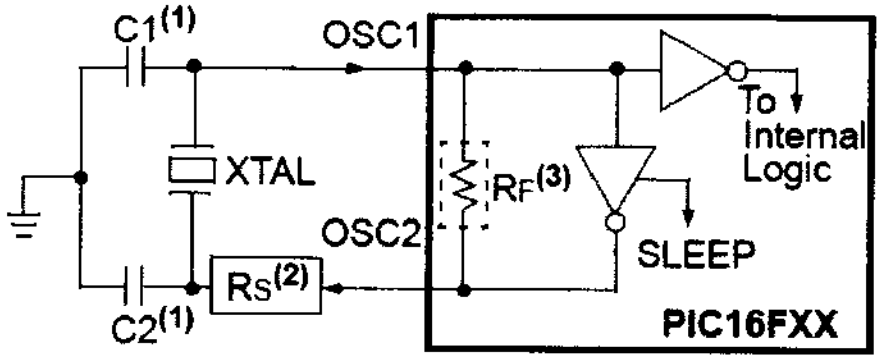


FIGURE Q1

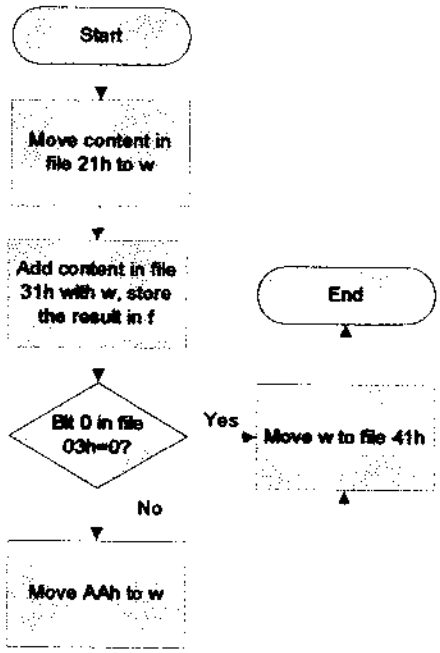


FIGURE Q3

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013
COURSE : MICROPROCESSOR & MICROCONTROLLER

PROGRAMME : BEE
COURSE CODE : BEE3233/BEX 32003

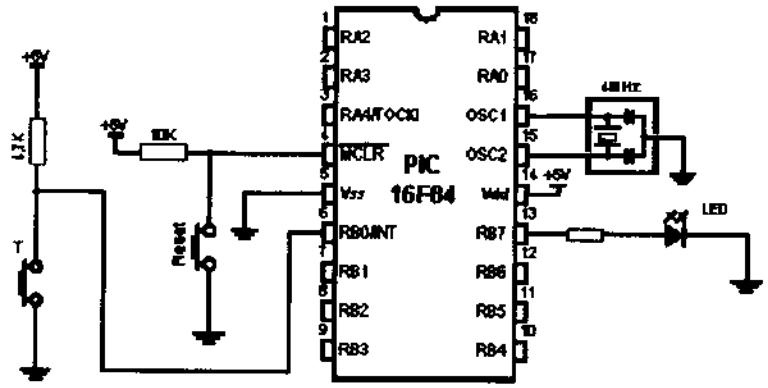


FIGURE Q4(b)

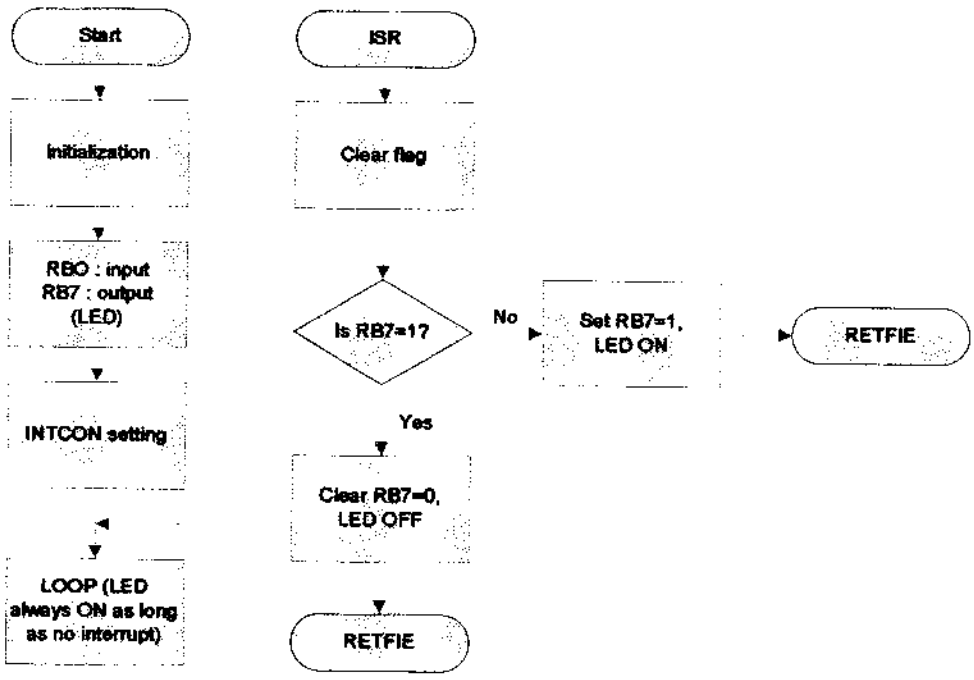


FIGURE Q4(b)(ii)

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013
COURSE : MICROPROCESSOR & MICROCONTROLLER

PROGRAMME : BEE
COURSE CODE : BEE3233/BEX 32003

Special Function Register (SFR) File Summary for PIC16F84A

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	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
04h	FSR	Indirect Data Memory Address Pointer 0								xxxx xxxx	11
05h	PORTA(4)	—	—	—	RA4/T0CKI	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	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10
Bank 1											
80h	INDF	Uses Contents of FSR to address Data Memory (not a physical register)								---- ----	11
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	9
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	11
83h	STATUS (2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	11
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	16
86h	TRISB	PORTB Data Direction Register								1111 1111	18
87h	—	Unimplemented location, read as '0'								—	—
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	13
89h	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	T0IE	INTE	RBIE	T0IF	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 $\overline{TO}$ and $\overline{PD}$ status bits in the STATUS register are not affected by a $\overline{MCLR}$ Reset.

3: Other (non power-up) RESETS include: external RESET through $\overline{MCLR}$ 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.

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013
COURSE : MICROPROCESSOR & MICROCONTROLLER

PROGRAMME : BEE
COURSE CODE : BEE3233/BEX 32003

STATUS Register of PIC16F84A

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

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

bit 3 **$\overline{PD}$**: Power-down bit
1 = After power-up or by the CLRWDT 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/$\overline{borrow}$ 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

bit 0 **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.

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013
COURSE : MICROPROCESSOR & MICROCONTROLLER

PROGRAMME : BEE
COURSE CODE : BEE3233/BEX 32003

OPTION Register of PIC16F84A / PIC16C71

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

bit 7 **$\overline{\text{RBPU}}$**: 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

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013
COURSE : MICROPROCESSOR & MICROCONTROLLER

PROGRAMME : BEE
COURSE CODE : BEE3233/BEX 32003

INTCON Register of PIC16F84A

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

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 **T0IE**: 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 **T0IF**: 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

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013
COURSE : MICROPROCESSOR & MICROCONTROLLER

PROGRAMME : BEE
COURSE CODE : BEE3233/BEX 32003

PIC16F84A Instruction Set Summary

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	

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013
COURSE : MICROPROCESSOR & MICROCONTROLLER

PROGRAMME : BEE
COURSE CODE : BEE3233/BEX 32003

Data Transfer Instruction MC68000

Mnemonic	Meaning	Type	Operand Size	Operations
MOVE	Move	MOVE EAs, EAd	8, 16, 32	(EAs) → EAd
		MOVE EA,CCR	8	(EA) → CCR
		MOVE EA,SR	16	(EA) → SR
		MOVE SR, EA	16	SR → EA
		MOVE USP,An	32	USP → An
		MOVE An,USP	32	An → USP
		MOVEA EA,An	16, 32	(EA) → An
		MOVEQ #XXX,Dn	8	#XXX → Dn
MOVEM	Move multiple	MOVEM Reg_list,EA	16, 32	Reg_list → EA
		MOVEM EA,Reg_list	16, 32	(EA) → Reg_list
LEA	Load effective address	LEA EA,An	32	EA → An
	Exchange			
EXG	Swap	EXG Rx,Ry	32	Rx ↔ Ry
SWAP	Clear	SWAP Dn	16	Dn31:16 ↔ Dn15:0
CLR		CLR EA	8, 16, 32	0 → EA

Compare and Test MC68000 Instruction

Mnemonic	Meaning	Type	Operand Size	Operation
CMP	Compare	CMP EA,Dn	8, 16, 32	N, Z, V, C
		CMPA EA,An	16, 32	N, Z, V, C
		CMPI #XXX,EA	8, 16, 32	N, Z, V, C
		CMPM (Ay)+,(AY)+	8, 16, 32	N, Z, V, C
TST	Test	TST EA	8, 16, 32	N, Z, V, C

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013 PROGRAMME : BEE
COURSE : MICROPROCESSOR & MICROCONTROLLER COURSE CODE : BEE3233/BEX 32003

Logical MC68000 Instruction

Mnemonic	Meaning	Type	Operand Size	Operation
AND	Logical AND	AND EA,Dn	8, 16, 32	(EA) · Dn → Dn
		AND Dn,EA	8, 16, 32	Dn · (EA) → EA
		ANDI #XXX,EA	8, 16, 32	#XXX · (EA) → EA
		ANDI #XXX,CCR	8	#XXX · CCR → CCR
		ANDI #XXX,SR	16	#XXX · SR → SR
OR	Logical OR	OR EA,Dn	8, 16, 32	(EA) + Dn → Dn
		OR Dn,EA	8, 16, 32	Dn + (EA) → EA
		ORI #XXX,EA	8, 16, 32	#XXX + (EA) → EA
		ORI #XXX,CCR	8	#XXX + CCR → CCR
		ORI #XXX,SR	16	#XXX + SR → SR
EOR	Logical exclusive-OR	EOR Dn,EA	8, 16, 32	Dn ⊕ (EA) → EA
		EORI #XXX,EA	8, 16, 32	#XXX ⊕ (EA) → EA
		EORI #XXX,CCR	8	#XXX ⊕ CCR → CCR
		EORI #XXX,SR	16	#XXX ⊕ SR → SR
NOT	Logical NOT	NOT EA	8, 16, 32	$(\overline{EA}) \rightarrow EA$

Bit Manipulation MC68000 Instruction

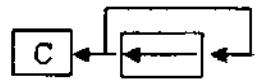
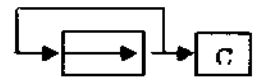
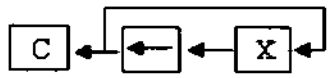
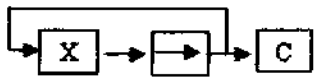
Mnemonic	Meaning	Type	Operand Size	Operation
BTST	Test a bit	BTST #XXX,EA	8, 32	
		BTST Dn,EA	8, 32	$\overline{EA}$ bit → Z
BSET	Test a bit and set	BSET #XXX,EA	8, 32	$\overline{EA}$ bit → Z
		BSET Dn,EA	8, 32	1 → EA bit
BCLR	Test a bit and clear	BCLR #XXX,EA	8, 32	$\overline{EA}$ bit → Z
		BCLR Dn,EA	8, 32	0 → EA bit
BCHG	Test a bit and change	BCHG #XXX,EA	8, 32	$\overline{EA}$ bit → Z
		BCHG Dn,EA	8, 32	$\overline{EA}$ bit → EA bit

FINAL EXAMINATION

SEMESTER / SESSION : I/20122013
COURSE : MICROPROCESSOR & MICROCONTROLLER

PROGRAMME : BEE
COURSE CODE : BEE3233/BEX 32003

Rotate MC68000 Instruction

Mnemonic	Meaning	Type	Operand Size	Operation
ROL	Rotate left	ROL #XXX,Dy ROL Dx,Dy ROL EA	8, 16, 32 8, 16, 32 8, 16, 32	
ROR	Rotate right	ROR #XXX,Dy ROR Dx,Dy ROR EA	8, 16, 32 8, 16, 32 8, 16, 32	
ROXL	Rotate left through extend	ROXL #XXX,Dy ROXL Dx,Dy ROXL EA	8, 16, 32 8, 16, 32 8, 16, 32	
ROXR	Rotate right through extend	ROXR #XXX,Dy ROXR Dx,Dy ROXR EA	8, 16, 32 8, 16, 32 8, 16, 32	

Bcc Instruction

Instruction	Meaning	Arithmetic	If the test is true
BEQ	EQual to zero	U	Z=1
BNE	Not Equal to zero	U	Z=0
BMI	Minus	U	N=1
BPL	Plus	U	N=0
BCS/LO	Carry Set/LOwer	U	C=1
BCC/HS	Carry Clear/Higher or Same	U	C=0
BVS	oVerflow Set	S	V=1
BVC	oVerflow Clear	S	V=0
BGT	GreaTer than	S	Z+(N⊕V)=0
BLT	Less Than	S	N⊕V=1
BGE	Greater than or Equal	S	N⊕V=0
BLE	Less than or Equal	S	Z+(N⊕V)=0
BHI	Higher	U	C+Z=0
BLS	Lower than or Same	U	C+Z=1