UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2011/2012

COURSE NAME	:	MICROPROCESSOR AND MICROCONTROLLER
COURSE CODE	:	BEX 32003 / BEE 3233
PROGRAMME	:	BEE
EXAMINATION DATE	:	JUNE 2012
DURATION	:	3 HOURS
INSTRUCTION	:	ANSWER **ALL** QUESTIONS IN **PART A** AND ANY **THREE (3)** QUESTIONS IN **PART B**

THIS PAPER CONSISTS OF **SIXTEEN** (16) PAGES

PART A

Q1 (a) Stack is one of the basic features in all modern computers. List down and explain the features and functions of stack in PIC16F84A.

(4 marks)

(b) Employ a sequence of instructions that will cause the changes of value in the stack on PIC16F84A.

(3 marks)

(c) Explain how conditional program jumps are implemented in the PIC MCU.

(4 marks)

(d) Draw a block diagram of the circuit with a keypad, 7-segment display (active high), PIC16F84A and a suitable decoder. The block diagram must have complete indicating of the main components and signals in the system.

(9 marks)

Q2 (a) Consider the following program segment written in Easy 68K assembly language:

```
INPUT  DC.B     %01010011
       MOVE.B   INPUT,D0
       BCLR     #7,D0
       MOVE.B   D0,D1
       MOVEQ    #0,D2
       MOVEQ    #0,D3
LOOP   LSR.B    #1,D1
       ADDX.B   D3,D2
       TST.B    D1
       BNE.S    LOOP
       BTST     #0,D2
       BEQ      CLEANUP
       BSET     #7,D0
```

(i) Determine value in D0, D1, D2 and D3 after instruction code BTST#0,D2.

(8 marks)

(ii) Identify the value in D0 after the above program is executed.

(2 marks)

(b) Below is a program segment written in Easy 68K assembly language. Assume value in **D0 = $ 0005 3F02** and **D1 = $ FFFF 0110**

```
            ORG       $1000
            ADD.L     #DATA,D0
            MOVE.W    DATA,D1

            ORG       $2000
DATA        DC.B      $0A, $EE, $83, $82
            DC.B      $0A, $EE, $30, $00
```

(i) Show the value in D0 after the instruction ADD.L #DATA,D0 is executed.

(2 marks)

(ii) Trace the value in D1 after the instruction MOVE.W DATA,D1 is executed.

(2 marks)

(c) Consider the following program segment:

```
MOVE.L    #NUMBER,D0
DIVU      #3,D0
```

(i) Determine the value of D0 after instruction executed if **NUMBER EQU $50005533**.

(3 marks)

(ii) Identify the value of D0 after instruction executed if **NUMBER EQU $00010000**.

(3 marks)

PART B

Q3 (a) By referring ASCII table in **Figure Q3**, the ASCII code for the letters 'E' 'N' and 'D' are to be transmitted using asynchronous serial protocol at 9600 baud with eight data bits, no parity frame with one stop bit (logic high) and start bit (logic low).

(i) Determine the value of data transmission for letters "END".

(3 marks)

(ii) Calculate the duration of stop bit 'T' and the maximum character that can be send in one second if transmission speed is 9600 baud.

(2 marks)

(iii) Produce a waveform, which showing what you would see if an oscilloscope were monitoring during transmission process.

(6 marks)

(b) Below is a segment of code written in PIC16F84 assembly language:

```
        movlw   d'20'
        movwf   h'3F'
LOOP1   movlw   d'250'
LOOP2   addlw   -1
        btfss   STATUS,Z
        goto    LOOP2
        decf    h'3F',f
        btfss   STATUS,Z
        goto    LOOP1
        sleep
```

(i) Explain what this routine does.

(3 marks)

(ii) Determine the execution time of the complete program segment, assuming a clock rate of 4 MHz.

(4 marks)

(iii) Modify the program so that the execution time is now approximately 0.1 second with a 4MHz clock.

(2 marks)

Q4 PIC16F84 based microcontroller is to be designed with eight LEDs and two switches to executes the following task:

If switch 1 (SW1) is closed, then all LEDs should blink ON/OFF with delay.
If switch 2 (SW2) is closed, then LEDs should count up in binary format.
If both switches are closed, then the priority is given to switch 1 (SW1).
The program should check the switches continuously.

(a) Based on the given specification draw the diagram to show the switches and the LEDs should be connected.

(5 marks)

(b) Construct the system by using a flowchart.

(5 marks)

(c) Produce a complete program to implement the application based on your flowchart in **Q4(b)**. Your program must be as compact as possible.

(10 marks)

Q5 (a) Determine the values of the X, N, Z, V and C flags after each of the following instructions is executed independently. Assume the flags are all zero and the register contents are as shown immediately prior to executing each instruction.

register contents	memory contents:
(D0) = $FFFFFFFF	($10000.W) = $1234
(D1) = $00001000	($10010.W) = $31D0
(A0) = $00010010	($10020.W) = $0D0A
(A1) = $00010020	($10030.W) = $1234
(A3) = $00010030	

Instruction:	X	N	Z	V	C
CMP.B $10000,D0					
CMPA.W D1,A0					
CMPI.W #$1234,(A0)					
CMPM.B (A1)+,(A3)+					

(6 marks)

(b) Design a partial address decoder for a 68000 based system that contains of

2MB of EPROM at a starting address $00 0000 using 512Kx8 chips
2MB of RAM at a starting address $10 0000 using 256Kx8 chips
64KB I/O space starting at $FF0000.

(i) Create the full memory table showing the address range and all the address lines for the devices above.

(7 marks)

(ii) Draw the full memory address decoder circuit showing all devices and decoder.

(7 marks)

Q6 **Figure Q6** shows the MC68000's assembly program written in EASY68K simulation software for stack. Answer the following questions.

(a) Draw the memory map which consist of main program, subroutines and stack.

(6 marks)

(b) Describe why label of ABC is assigned with value 50.

(2 marks)

(c) Identify the value of memory to stop the program.

(2 marks)

(d) Predict the value of stack pointer before command JSR SUB_1.

(2 marks)

(e) By using memory map, illustrate the value of stack after command jump to subroutine (JSR) and after return from subroutine (RTS) in right order.

(8 marks)

Q7 The Angry Bird games need some modification. As a programming engineer you need to add some function in Angry Bird games. The function you need to add is "Sit-down" and "Stand-Up". The value $AAAAAAAA is data for "Sit-down" and $55555555 is data for Stand-Up".

(a) To write a program you have to assume that the function of Sit-Down and Stand-Up is toggle function for Data Register. In this program you are not allow to use mnemonic NOT. Write a MC68000's assembly program to toggle Data Register for 1000 times.

(12 marks)

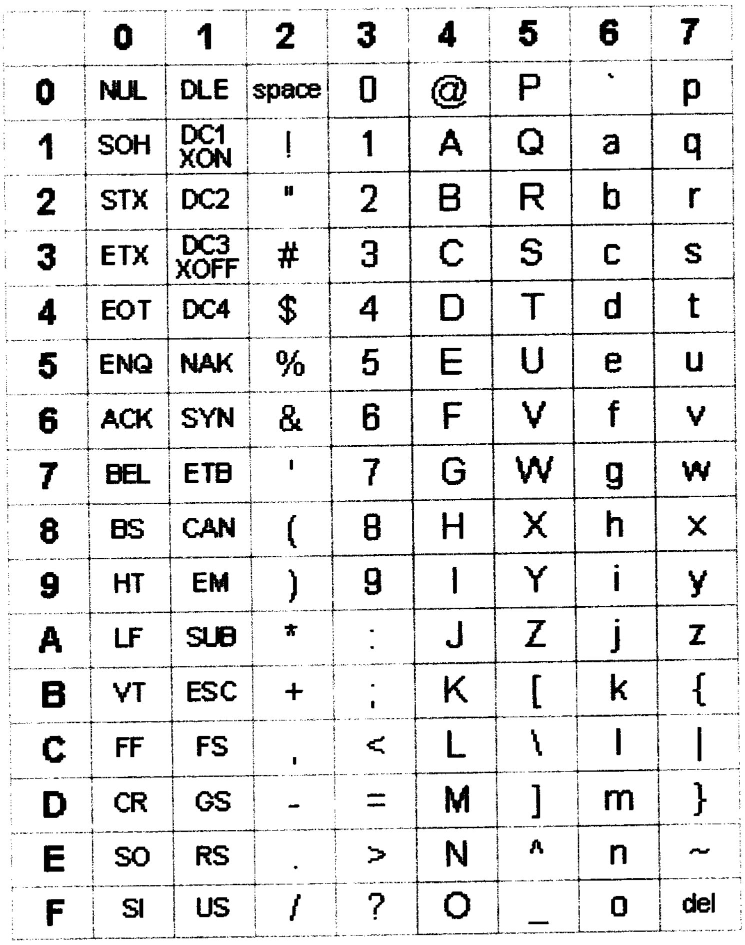
(b) Design a flowchart based on programs **Q7(a)**.

(8 marks)

FINAL EXAMINATION

SEMESTER/SESSION : II/2011/2012
COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE
COURSE CODE : BEX 32003/BEE3233



	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

FIGURE 3: ASCII Table

FINAL EXAMINATION

SEMESTER/SESSION : II/2011/2012
COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE
COURSE CODE : BEX 32003/BEE3233

```
00001000 Starting Address
Assembler used: EASy68K Editor/Assembler v5.3.3
Created On: 11/9/2011 1:16:43 PM

00000000                        1  *-------------------------------------------
-----------------
00000000                        2  * Program    :
00000000                        3  * Written by :
00000000                        4  * Date       :
00000000                        5  * Description:
00000000                        6  *-------------------------------------------
-----------------
00001000                        7          ORG     $1000
00001000                        8
00001000  3E7C 10A8             9  START:  MOVEA #STACK,SP
00001004  4E71                 10          NOP
00001006  4EB9 00001018        11          JSR     SUB_1
0000100C  4E71                 12  RET_1:  NOP
0000100E  4EB9 00001026        13          JSR     SUB_2
00001014  4E72 2700            14  RET_2   STOP    #$2700
00001018                       15
00001018                       16
00001018  4E71                 17  SUB_1:  NOP
0000101A  4E71                 18          NOP
0000101C  4EB9 0000102C        19          JSR     SUB_3
00001022  4E71                 20  RET_3:  NOP
00001024  4E75                 21          RTS
00001026                       22
00001026                       23
00001026  4E71                 24  SUB_2:  NOP
00001028  4E71                 25          NOP
0000102A  4E75                 26          RTS
0000102C                       27
0000102C  4E71                 28  SUB_3:  NOP
0000102E  4E71                 29          NOP
00001030  4EB9 0000103A        30          JSR     SUB_4
00001036  4E71                 31  RET_4   NOP
00001038  4E75                 32          RTS
0000103A                       33
0000103A  4E71                 34  SUB_4:  NOP
0000103C  4EB8 1026            35          JSR     SUB_2
00001040  4E71                 36  RET_5   NOP
00001042  4E75                 37          RTS
00001044                       38
00001044                       39  ABC:    DS.W    50
000010A8                       40
000010A8  =000010A8            41  STACK:  EQU *
000010A8                       42          END     $1000
```

FIGURE Q6

FINAL EXAMINATION

SEMESTER/SESSION : II/2011/2012
COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE
COURSE CODE : BEX 32003/BEE3233

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	$\overline{\text{TO}}$	$\overline{\text{PD}}$	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	$\overline{\text{TO}}$	$\overline{\text{PD}}$	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 $\overline{\text{TO}}$ and $\overline{\text{PD}}$ status bits in the STATUS register are not affected by a $\overline{\text{MCLR}}$ Reset.

3: Other (non power-up) RESETS include: external RESET through $\overline{\text{MCLR}}$ 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 : II/2011/2012

COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE

COURSE CODE : BEX 32003/BEE3233

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

bit 3 **$\overline{\text{PD}}$**: 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/$\overline{\text{borrow}}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for $\overline{\text{borrow}}$, 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/$\overline{\text{borrow}}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for $\overline{\text{borrow}}$, 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 : II/2011/2012
COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE
COURSE CODE : BEX 32003/BEE3233

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 **$\overline{\mathrm{RBPU}}$**: 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 : II/2011/2012
COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE
COURSE CODE : BEX 32003/BEE3233

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 : II/2011/2012
COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE
COURSE CODE : BEX 32003/BEE3233

PIC16F84A Instruction Set Summary

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	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	$\overline{\text{TO}}$,$\overline{\text{PD}}$	
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	$\overline{\text{TO}}$,$\overline{\text{PD}}$	
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 : II/2011/2012
COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE
COURSE CODE : BEX 32003/BEE3233

Data Transfer Instruction MC68000

Mnemonic	Meaning	Type	Operand Size	Operations
MOVE	Move	MOVE EAs, EAd MOVE EA,CCR MOVE EA,SR MOVE SR, EA MOVE USP,An MOVE An,USP MOVEA EA,An MOVEQ #XXX,Dn	8, 16, 32 8 16 16 32 32 16, 32 8	(EAs) → EAd (EA) → CCR (EA) → SR SR → EA USP → An An → USP (EA) → An #XXX → Dn
MOVEM	Move multiple	MOVEM Reg_list,EA MOVEM EA,Reg_list	16, 32 16, 32	Reg_list → EA (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 CMPA EA,An CMPI #XXX,EA CMPM $(Ay)^+$,$(AY)^+$	8, 16, 32 16, 32 8, 16, 32 8, 16, 32	N, Z, V, C N, Z, V, C N, Z, V, C N, Z, V, C
TST	Test	TST EA	8, 16, 32	N, Z, V, C

FINAL EXAMINATION

SEMESTER/SESSION : II/2011/2012
COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE
COURSE CODE : BEX 32003/BEE3233

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

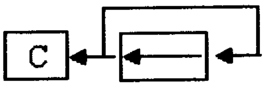
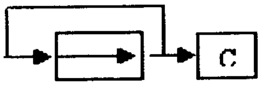
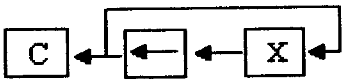
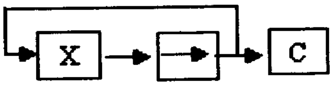
Mnemonic	Meaning	Type	Operand Size	Operation
BTST	Test a bit	BTST #XXX,EA	8, 32	
		BTST Dn,EA	8, 32	$\overline{EA}$ bit → Z
BSET	Test a bit and set	BSET #XXX,EA	8, 32	$\overline{EA}$ bit → Z
		BSET Dn,EA	8, 32	1 → EA bit
BCLR	Test a bit and clear	BCLR #XXX,EA	8, 32	$\overline{EA}$ bit → Z
		BCLR Dn,EA	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 : II/2011/2012
COURSE : MICROPROCESSOR AND MICROCONTROLLER

PROGRAM : BEE
COURSE CODE : BEX 32003/BEE3233

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