BIRZEIT UNIVERSITY

ENCS336 – 1st Exam

2nd Semester 07/08
Date: 20/4/2008

ID :____________________

Name :____

□ Section 1 Dr. Emad Hamadeh

□ Section 2 Eng. Mamoun Nawahdah

Instructions:

- **You have 100 minutes, so budget your time carefully!**
- **Turn OFF your mobile.**
- **To make sure you receive credit, please write clearly and show your work.**
- **We will not answer questions regarding course material.**

Question	Maximum	Your Score
1	15	
2	15	
3	15	
4	15	
5	20	
6	20	
Total	100%	

Question 1 (15 marks) :

a) TRUE or FALSE :

TRUE	FALSE	
●	○	Design of common bus for 16 registers of 32-bits each requires 32 multiplexers
○	●	Performance of x machine is better than y machine if x has **more** execution time than y
○	●	Extended PCI bus has **32-bit** data/address lines and twice the speed of standard PCI
○	●	Number of address lines for a 512K x 16 memory organization is **16**
○	●	The clock rate for a clock with a period of 250 nanosecond is 4 **G**Hz
○	●	MIPS stands for **Multi** Instructions Per Second
●	○	In Booth's Algorithm, the number of Shift Arithmetic Right operations is equal to the number of bits in M or Q
●	○	The ASCII code for the character '7' is 55 in decimal
○	●	All Intel x86 family share the same basic **organization** in order to maintain code backwards compatibility
○	●	ENIAC was a **binary** machine programmed manually by switches

Question 2 (15 marks) :

a) State whether the following none related register transfer statements are legal or not?

If not, why?

1. **D.T : AR ← AR' , AR ← 0**

Illegal, can't assign to operations to AR at once

2. **X : PC ← AR**

Legal

b) Use register transfer statements with required control to represent the following pseudo code:

```
 If (x = 1)   then
        R1 gets R2
else
    If (y = 1) then
           R1 gets R3
```

X: R1 ← R2
X’Y: R1 ← R3

c) If operand **A** is **01001_b**, and operand **B** is unknown, what logic operation and **B** value would you use to obtain a **10001_b** result? Draw any needed logic blocks.

A XOR B with B = 11000b,
Draw an XOR gate

Question 3 (15 marks):

a) A system with three I/O devices: printer, disk, and communication line. The devices have interrupt priority of **3** for printer, **6** for disk, and **8** for communication line such that the higher the number the higher the priority, and vv. It takes any ISR **10** time-units to process its interrupt. A User Program starts executing at **t = 0**. If a disk interrupt occurs at **t =20**, and a communication line interrupt occurs at **t = 28**, and a printer interrupt occurs at **t = 35**, when will each ISR completes its execution of its interrupt?

Communication Line Interrupt completes at t = 38
Disk Interrupt completes at t = 40
Printer Interrupt completes at t = 50

b) Computer A, running at **500 MHz**, has a program with the following instruction classes, CPI budget, and number of instructions distribution.

Instruction Class	CPI	Number of Instructions
A	3	50
B	2	20
C	4	30

1. Find the Average CPI for running the above program?

Ave CPI = 3x(50/100) + 2x(20/100) + 4x(40/100)
= 3.1

2. Find the CPU execution time for computer A?

CPU Time = (# of Instructions) x (Ave. CPI) x (Clock Cycle Time)
= 100 x 3.1 x (1/(500x10^6)) = 620 ns

Question 4 (15 marks):

A **2K x 16** memory, shown below, is used to store instructions set and data for a basic computer architecture based machine. If first instruction is stored at memory address location **400h**, fill in the table below with the values for **PC**, **IR**, and **AC**? Initial values mean the values of the registers prior to fetching the first instruction from memory.

Opcode **1h** is to load AC from memory

Opcode **3h** is to store AC to memory, while

Opcode **5h** is to add to AC from memory.

(All numbers are in hex representation)

Instruction Phase	PC	IR	AC
Initial Values	400	---	---
After 1st Instruction Fetch Cycle	401	1 F F 0	---
After 1st Instruction Execution Cycle	401	1 F F 0	0 1 0 0
After 2nd Instruction Fetch Cycle	402	5 F F 3	0 1 0 0
After 2nd Instruction Execution Cycle	402	5 F F 3	0 F 0 F

Memory Address	Instructions & Data
:	:
400	1 F F 0
401	5 F F 3
402	3 F F 2
:	:
FF0	0 1 0 0
FF1	5 0 0 0
FF2	3 1 0 0
FF3	0 E 0 F
:	:

Question 5 (20 marks):

a) What is the range of two's complement integers that can be represented using **12 bits**? Give your answers in decimal.

-2^{11} ←→ $+2^{11}-1$

-2048 ←→ +2047

b) Given the bit pattern:

1011 1111 1110 0000 0000 0000 0000 0000

What is the value (in *decimal*) that this pattern represent, assuming that it is:

1. A two's complement integer?

```
    0100 0000 0001 1111 1111 1111 1111 1111
 +  0000 0000 0000 0000 0000 0000 0000 0001
 ------------------------------------------
    0100 0000 0010 0000 0000 0000 0000 0000
```

$$\begin{aligned} -2^{30}+2^{21} &= -(2^9+1)\times 2^{21} \\ &= -(513\times 2\times 1024\times 1024) \\ &= -1.075\times 10^9 \end{aligned}$$

2. A single format floating-point number?

1	0111 1111	110 0000 0000 0000 0000 0000

$$\begin{aligned} S &= 1 \\ exponent &= 2^7-1 = 128-1 = 127 \\ fraction &= 2^{-1}+2^{-2} = 0.5+0.25 = 0.75 \\ N &= (-1)^1\times 1.75\times 2^{127-127} = -1\times 1.75\times 1 = -1.75 \end{aligned}$$

d) Consider the division of a **dividend X=8** and a **Divisor D=3** using unsigned algorithm. Show your work step by step in the following table?

Q ← Dividend (8d ➔ 1000b)
M ← Divisor (3d ➔ 0011b)
-M ← (-3d ➔ 1101b)

	A	Q	M	Comments
	0000	1000	0011	Initial value
N=4	0001	0000		Shift left A,Q
	1110	0000		A ← A - M
	0001	000**0**		A < 0 ? YES Restore
N=3	0010	0000		Shift left A,Q
	1111	0000		A ← A - M
	0010	00**00**		A < 0 ? YES Restore
N=2	0100	0000		Shift left A,Q
	0001	0000		A ← A - M
	0001	0**001**		A < 0 ? NO Restore
N=1	0010	0010		Shift left A,Q
	1111	0010		A ← A - M
	0010	**0010**		A < 0 ? YES Restore

Reminder **Quotient**

Question 6 (20 marks):

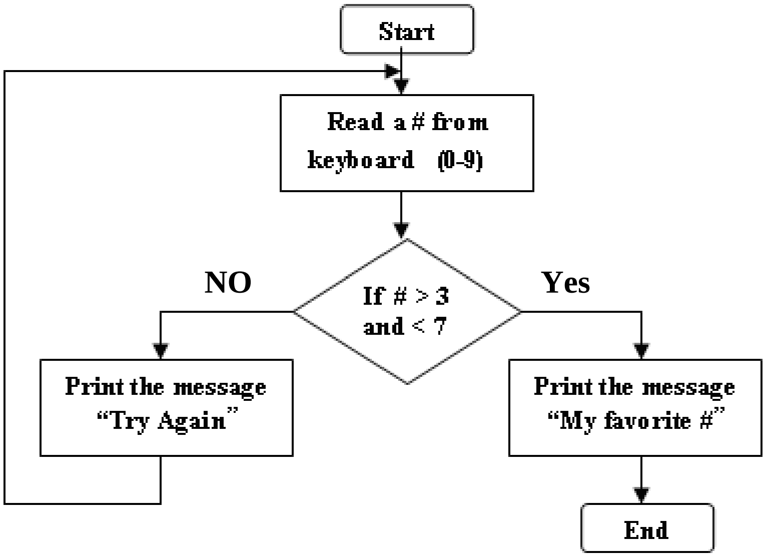
a) Write an assembly code to find the average of two numbers stored in **AL** and **DL** using *ONLY* the following assembly instructions:

MOV ➔ move
ADD ➔ add
ADC ➔ add with carry
SAR ➔ shift arithmetic right

```
MOV AH, 00H
ADD AL, DL
ADC AH, 00H
SAR AX, 1
```

Result in AL is the average of AL and DL

b) Convert the following flow chart to an assembly program


Start
Read a # from keyboard (0-9)
NO
If # > 3 and < 7
Yes
Print the message "Try Again"
Print the message "My favorite #"
End

```
ideal
p286n
model tiny
codeseg
          org 100h
          JMP ABAS
          mesg2 db “Try Again $”
          mesg1 db “My favorite # $”


ABAS:
          ; Read character from keyboard
          MOV DL, 0FFH
          MOV AH, 06H
          INT 21H
          JZ ABAS
          SUB AL, 30H     ;convert ASCII # to a binary number

          ; IF
          CMP AL, 3
          JLE NO
          CMP AL, 7
          JGE NO

          ; THEN    print the message “My Favorite #”
          MOV DX, OFFSET mesg1
          MOV AH, 09
          INT 21H
          JMP EXIT

          ; ELSE    print the message “Try Again”
NO:       MOV DX, OFFSET mesg2
          MOV AH, 09
          INT 21H
          JMP ABAS

EXIT:
          mov ax, 4c00h
          int 21h
end
```

c) Show the absolute addresses formed by **SP** contains **0040h** and **SS** contains **B42Ah**

SS:SP ➔ B42A:0040 ➔ B42Ah * 16d + 0040h ➔ B42E0h
B42Ah * 10h + 0040h ➔ B42E0h