August 13, 2007

COMPUTER ENGINEERING DEPARTMENT

ICS 233

COMPUTER ARCHITECTURE & ASSEMBLY LANGUAGE

Major Exam II

Summer Semester (063)

Time: 7:00-9:30 PM

Student Name : ____

Student ID.   : ____

Question	Max Points	Score
Q1	20	
Q2	16	
Q3	16	
Q4	14	
Q5	8	
Q6	16	
Q7	10	
Total	100	

Dr. Aiman El-Maleh

[20 Points]

(Q1) Suppose that you are given a positive integer. You can add individual digits of this number to get another integer. If we repeat this procedure, eventually we will end up with a single digit. Here is an example:

7391928 = 7+3+9+1+9+2+8 = 39
39 = 3+9 = 12
12 = 1+2= 3

Write a procedure, **ToSDigit**, that receives a positive integer in $a0 and returns a single digit in register $v0 according to the method described above. It is required that the procedure **preserves the content of all used registers** by saving and restoring them on the stack. Then, write a program to read a positive integer from the user and display the **single digit** obtained by the above procedure.

A **sample execution** of the program is:

Enter a number: 7391928
Result is: 3

[16 Points]

(Q2) Given that **Multiplicand=1010** and **Multiplier=1001**.

(i) Using the **refined unsigned multiplication** hardware, show the **unsigned** multiplication of **Multiplicand** by **Multiplier**. The result of the multiplication should be an 8 bit **unsigned** number in HI and LO registers. Show the steps of your work.

(ii) Using the **refined signed multiplication** hardware, show the **signed** multiplication of **Multiplicand** by **Multiplier**. The result of the multiplication should be an 8 bit **signed** number in HI and LO registers. Show the steps of your work.

[
16 Points]

(Q3) Given that **Dividend=1010** and **Divisor=0100**.

(i) Using the **refined unsigned division** hardware, show the **unsigned** division of **Dividend** by **Divisor**. The result of division should be stored in the Remainder and Quotient registers. Show the steps of your work.

(ii) Using the **refined unsigned division** hardware, show the **signed** division of **Dividend** by **Divisor**. The result of division should be stored in the Remainder and Quotient registers. Show the steps of your work.

[14 Points]

(Q4) Given the following two floating-point numbers in single-precision format:

X=1100 0011 1100 0000 0000 0000 0000 0001
Y=0011 1110 1000 0000 0000 0010 0011 0001

(i) Perform the floating-point operation **X-Y** rounding the result to the nearest even, using guard, round and sticky bits. Represent the result in single-precision format.

(ii) Perform the floating-point operation **X*Y** rounding the result to the nearest even. Represent the result in single-precision format.

[8 Points]

(Q5) Suppose that a program runs in 150 seconds on a machine, with multiply operations responsible for 40 seconds of this time, divide operations responsible for 60 seconds of this time. The remaining time is taken by the remaining operations. Suppose that a new implementation of the machine has improved the execution time of the multiplier by a factor of 3 and the execution time of the divider by a factor of 2. Determine the new execution time and the speedup of the program on the new implementation.

[16 Points]

(Q6) Given the following instruction mix of a program on a RISC processor:

Class	CPI	Frequency
ALU	3	20%
Branch	2	30%
Jump	1	25%
Load	5	15%
Store	4	10%

- **(i)** What is the average CPI?
- **(ii)** Assuming that the processor has a clock rate of 3 GHz, determine MIPS.
- **(iii)** What is the percent of time used by each instruction class?
- **(iv)** How much faster would the program run if load and store time are reduced to 3 cycles, and two ALU instructions could be executed at once, assuming that the cycle time has increased by 10% and the instruction count has increased by 15%?

[10 Points]

(Q7) Assume that a processor has four 8-bit registers: R0, R1, R2, and R3. You are required to design a register file that allows reading the value of one of the registers specified by the field RS[1:0] and writing into one of the registers specified by the field RD[1:0]. Show all required control signals for the register file.