CSE 321a: Computer Organization (I)
Third Year, Computer & Systems Engineering

Assignment #1

Due date: **Sunday, 15th October 2017**

1. The performance of a single-processor machine is evaluated using a two-program benchmark suite. Suppose only 50% of the first program and 87.5% of the second program can be executed in parallel.
   (a) What is the maximum factor of improvement that can be achieved in the benchmark score (*i.e.*, geometric mean) by adding more processors to that machine?
   (b) What is the minimum number of processors that need to be added to that machine in order to improve its benchmark score by a factor of three.

2. In a hypothetical computer, the processor has five registers: a 20-bit Program Counter (PC), an 18-bit Accumulator (AC), a 10-bit Counter (CTR), and two 13-bit Pointers (PTR1 and PTR2). The memory is divided into words each of which is 18-bit long. Each word can hold either an instruction or a piece of data. For each instruction X, the five most significant bits (denoted by $X_{17\text{-}13}$) represent an opcode. The rest of the instruction (denoted by $X_{12\text{-}0}$) can be either an address or a value of an operand. Operands that represent signed numbers are interpreted according to the sign-an-magnitude representation. The table below explains some of the instructions supported by the processor.

Opcode (binary)	Operation
00001	Load PTR1 with $X_{12\text{-}0}$.
00010	Load PTR2 with $X_{12\text{-}0}$.
00100	If X_0 is 1, increment the unsigned values of PTR1 and PTR2 (by one); otherwise, decrement the unsigned values of PTR1 and PTR2 (by one).
00110	Subtract the signed value of the memory location whose address is in PTR1 from the signed value of the memory location whose address is in PTR2 and store the result to AC.
01001	Load CTR with $X_{9\text{-}0}$.
01011	Decrement CTR (by one), and check if CTR is not 0, branch to an instruction whose address is obtained by subtracting $X_{12\text{-}0}$ from PC (*i.e.*, decrement PC by $X_{12\text{-}0}$); otherwise, continue normally (*i.e.*, do not change PC).
10100	Load AC from a memory location whose address is $X_{12\text{-}0}$.
10101	Store AC into a memory location whose address is $X_{12\text{-}0}$.
11010	Square the signed value of AC and add the result to the memory location whose address is $X_{12\text{-}0}$.
11011	Take the positive square root of the signed value of AC and store the result to the memory location whose address is $X_{12\text{-}0}$.
11111	Halt execution.

   (a) In this hypothetical computer:
      i. Which memory locations can be used to store instructions?
      ii. Which memory locations can be used to store data?
      iii. What is the maximum number of loop iterations (without nesting)?

   (b) Given the following program:

Address (Hexadecimal)	Contents (Hexadecimal)
B739C	13C02
B739D	03CE3
B739E	04B9F
B739F	0DFFF
B73A0	345D7
B73A1	09FFF
B73A2	16004
B73A3	285D7
B73A4	365D7
B73A5	3FFFF

Show, using the table below, the execution trace of that program by filling in the contents of every register and memory location after the fetch cycle and after the execute cycle of every instruction. All values are in hexadecimal.

Instruction	Cycle	PC	AC	CTR	PTR1	PTR2	Location: 005D7	Location: 00B9F	Location: 00BA0	Location: 01CE3	Location: 01CE4
Initially	---	B739C	3C0DE	2F1	1FFF	100D	00000	20005	20020	00007	2001B
13C02	Fetch	B739D	3C0DE	2F1	1FFF	100D	00000	20005	20020	00007	2001B
	Execute	B739D	3C0DE	002	1FFF	100D	00000	20005	20020	00007	2001B
03CE3	Fetch	…	…	…	…	…	…	…	…	…	…
	Execute	…	…	…	…	…	…	…	…	…	…
04B9F	Fetch	…	…	…	…	…	…	…	…	…	…
	Execute	…	…	…	…	…	…	…	…	…	…

(c) What does the program compute?

(d) Suppose that it takes 3 clock cycles in order for the processor to read a word from or write a word to the memory. Suppose further that the processor takes 2 clock cycles on average to execute an instruction; that is in addition to the time taken to fetch the operand(s) from the memory and/or write the result to the memory. Given that the processor is clocked at a rate of 50MHz, how much time does it take for the program to be executed? Justify your answer.