1. Answer the following four questions:

(A) Compare **memory-mapped I/O and IO-mapped I/O** based on (a) R/W control lines (b) memory address space; (c) I/O instruction. (3%)

(B) Compare **programmed I/O and Interrupted I/O** based on (a) CPU utilization; (b) system complexity. (3%)

(C) Compare **hardwired control and microprogrammed control** based on (a) system flexibility; (b) speed of operation; (c) number of components used. (3%)

(D) A computer are executing data transfer instructions that each requires three bus cycles, one fetches the instruction and the other two transfer the data. Each bus cycle takes 50nsec. The computer also uses cycle stealing DMA to transfer data to/from the disk. The disk capacity is 16384 bytes/track. Disk rotation time is 8.192 msec/revolution. Assume that the computer allows 16-bits data bus transfer, to what percentage of its normal speed is reduced during a cycle stealing DMA transfer if each DMA takes one bus cycle ? (11%)

2. A cache system has a 95 percent hit ratio, an access time of 100 nsec on a cache hit and a access time of 800 nsec on a cache miss. what is the effective access time ? If we want to reduce the effective access time to 120 nsec, what is the cache system hit ratio ? (10%)

3. Find a method of encoding the microinstructions described by the following figure so that the minimum number of control bits is used and all inherent parallelism among the microinstructions is preserved. **(15%)**

Microinstruction	Control signals activate
I_1	a
I_2	c,d
I_3	b,d,e
I_4	b,d,h
I_5	d,e,g
I_6	d,g,h
I_7	e,f,g
I_8	f,g,h

4. Consider the 4-bit CPU as shown below. The functions of this ALU and shifter are explained as follows.

1. ALU: D= A arithmetic-add B, D= A logic-and B, D= complement A, D = A exclusive-or B, Z=1 if D=0, Z=0 if D ≠ 0, and Cin and Cout are used for addition.
2. Shifter: Shift right/left one bit or pass

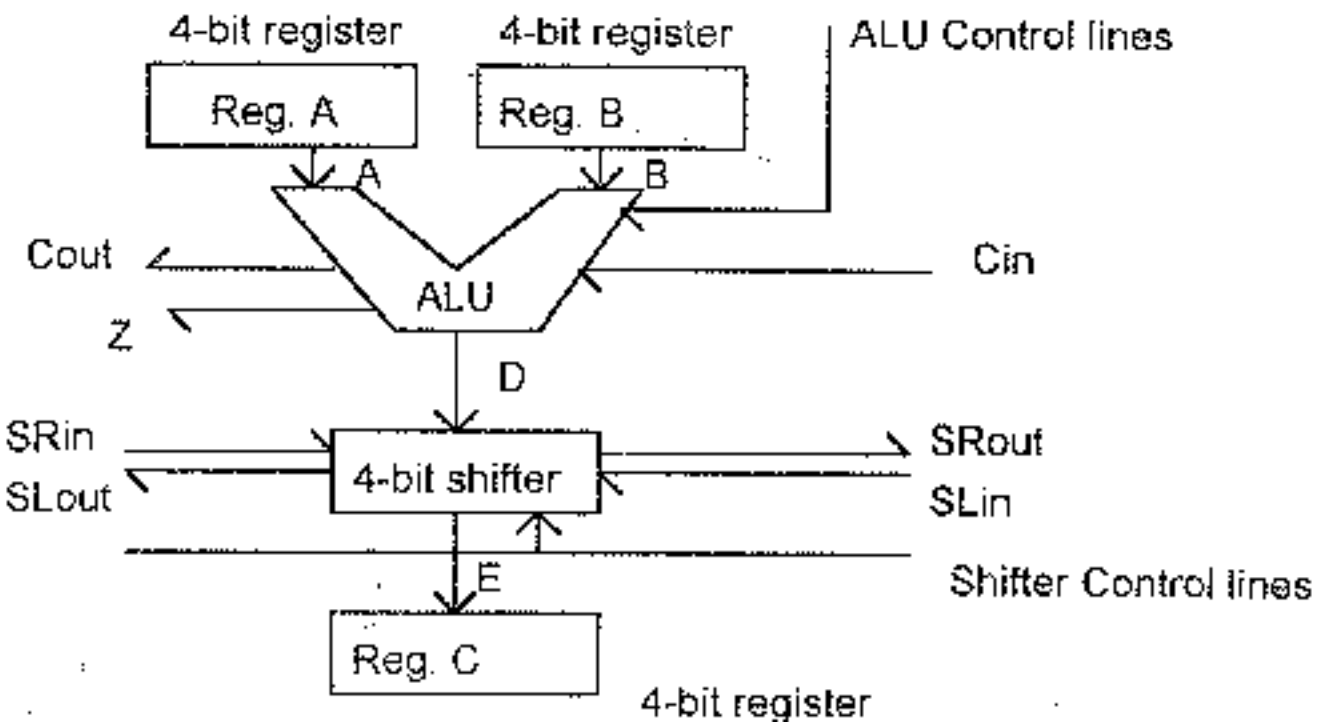
(a) Can you make a 16-bit CPU by using such 4-bit CPU? Note that two Flags are required:

1. Z flag: all zeros.
2. N flag: N=1 if Negative, otherwise N=0. (Two's complement number system is used)

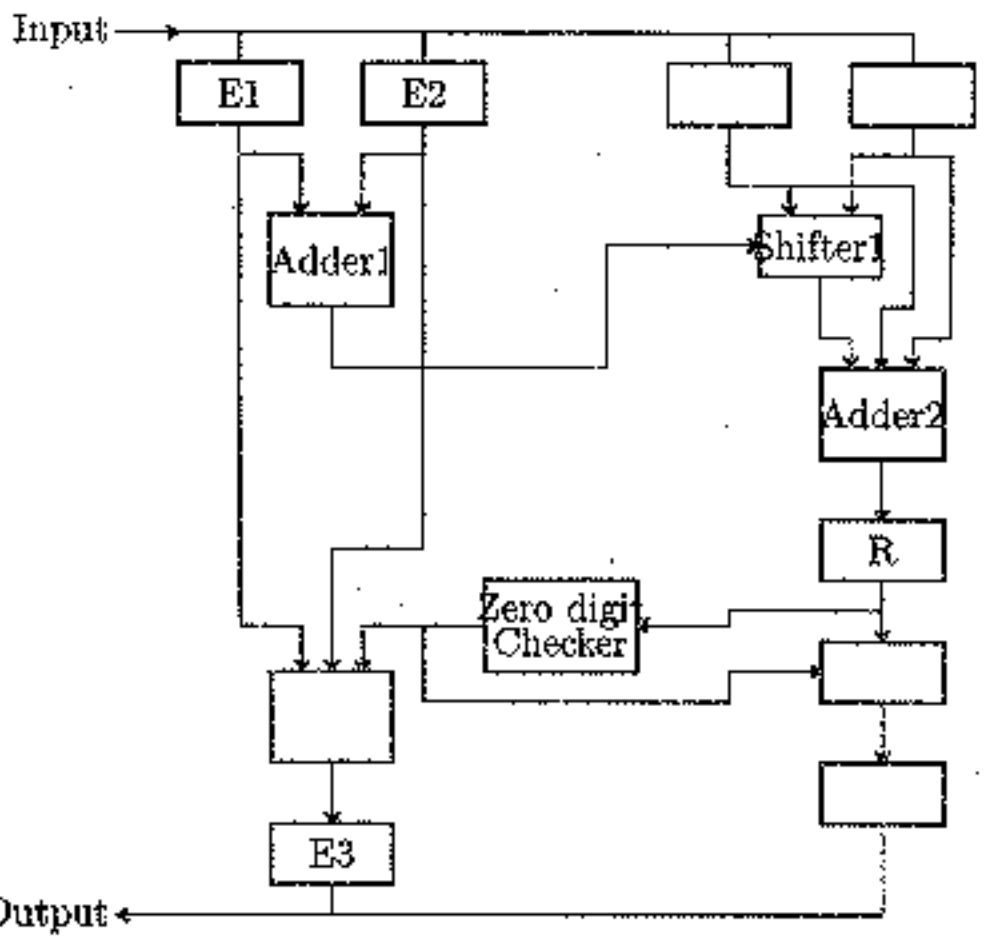
Assume that other registers are available and neglect the register transfer time. Also, you can modify the 4-bit CPU module, which includes adding extra gates, pins, etc., if necessary. **(10%)**

(b) How do you perform X "logic-or" Y in as few cycles as possible? **(5%)**

(c) How do you perform " X - Y" in two cycles? **(5%)**



5. The W&W Computer and Pizza Company has invented an advanced arithmetic circuit which can be used in accurately calculating the time needed for baking their pizza. Suppose you, probably from Pizza Hot, somehow spied the design which is shown below, with a few words missing. You have to figure out what the missing words stand for and what the circuit is in order to stay competitive in the market.



(a) Fill in the blank boxes in the figure and describe briefly their functions (you have to copy the figure to the answer sheet first). Note that it is the function of each box that is important, not its name. (10%)

(b) What can the circuit do? Give at least one *arithmetic* application and explain in detail how it is executed by this circuit. (5%)

6. (a) Show that a 2's-complement number can be converted to a representation with more bits by *sign extension*. That is, given an n-bit 2's-complement number X, show that the m-bit 2's-complement representation of X, where $m > n$, can be obtained by appending $m - n$ copies of X's sign bit to the left of the n-bit representation of X. (5%)

(b) We can define the 10's-complement/9's-complement of a decimal number in a similar way as the 2's-complement/1's-complement of a binary number. Assume that we have a *decimal* computer in which a register cell stores a decimal digit instead of a bit, and the length of all numbers in the computer is n digits. Prove that in such a digital computer, the subtraction of a decimal number B from another decimal number A can be done by taking the 10's-complement of B and adding the result to A. (5%)

7. The following assembly program is for a typical *one-address* machine, where LOAD transfers the content of a register or memory location to the accumulator (AC), STORE transfers the content of AC to a register or memory location.

```
LOAD A
MULTIPLY A
STORE M
LOAD B
MULTIPLY B
ADD M
STORE M
```

(a) Port the program to a typical *three-address* machine, i.e., rewrite the program using corresponding three-address instructions. (5%)

(b) Port the program to a typical *zero-address* (stack-based) machine, i.e., rewrite the program using corresponding zero-address instructions. (5%)