THE LNM INSTITUTE OF INFORMATION TECHNOLOGY

COA MID-TERM (2015-2016)

MAX MARKS: 30 **TIME: 90 MINS**

Q1. A computer has three I/O devices: a scanner, a flash drive and an RS232 (serial) line, with priorities 2, 3, and 6, respectively (6 being the highest). Initially, at $t = 0$ a user program is running. The three devices want to communicate with the system at the following times: scanner at $t = 15$, serial line at $t = 20$ and flash drive at $t = 25$. Assume that the ISR for each device runs for 15 time units. Show how the devices will be handled if the system uses a priority-based interrupt handling scheme. (4)

Q2. In a computer instruction format, the instruction length is 11 bits and the size of an address field is 4 bits. Is it possible to have

5 2-address instructions

45 1-address instructions

32 0-address instructions

using the format? Justify your answer.

b) Assume that a computer architect has already designed 6 two-address and 24 zero-address instructions using the instruction format given in the above problem. What is the maximum number of one-address instructions that can be added to the instruction set? (4+4=8)

Q3. The word-addressable memory unit of a computer has 256K words of 32 bits each. The computer has an instruction format with 4 fields: an opcode field; a mode field to specify 1 of 7 addressing modes; a register address field to specify 1 of 60 registers; and a memory address field. Assume an instruction is 32 bits long. Answer the following:

a) How large must the mode field be?

b) How large must the register field be?

c) How large must the address field be?

d) How large is the opcode field? (2+2+2+2=8)

Q4. How many bits would you need to address a 2M x 32 memory if

a) The memory is byte-addressable?

b) The memory is word-addressable? (2+2=4)

Q5. For a certain program, 4% of the code accounts for 70% of the execution time. Compare the following three strategies with respect to programming time and execution time. Assume that it would take 100 man-months to write it in C, and that assembly code is 10 times slower to write and four times more efficient.

a. Entire program in C.

b. Entire program in assembler.

c. First all in C, then the key 4% rewritten in assembler. (1+2+3=6)