SUBJECT: COMPUTER ARCHITECTURE
COURSE CODE: ECPC-211

Maximum Marks: **50** Time: **3 Hrs**

- **Attempt one question out of Q.5 and Q.6**
- **Assume necessary info. if not given**

****

1. a) Consider a 16-bit register used to store floating point numbers. The mantissa is normalized signed fraction number. Exponent is represented in excess-32 form.
Determine: **[3]**
   I. Represent $-(19.25)_{10}$ in this register.
   II. The maximum value that can be represented in register.
   III. The minimum value that can be represented in register.

   b) Calculate effective address and operand loaded to Accumulator (AC) for 9 different addressing modes as shown in **Table 1.** **[5]**

   c) Consider a processor with 64 registers and an instruction set of size twelve. Each instruction has five distinct fields, namely, opcode, two source register identifiers, one destination register identifier and a twelve-bit immediate value. Each instruction must be stored in memory in a byte-aligned fashion. If a program has 100 instructions, then what is the amount of memory (in bytes) consumed by the program text? **[2]**

Table 1

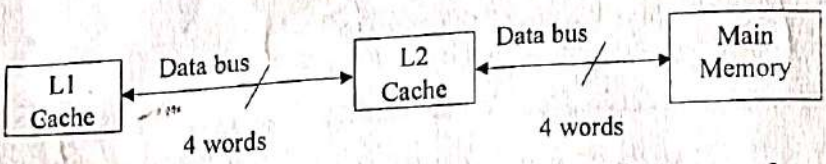
PC = 300 R1 = 425 Index Reg = 100	Address	Memory	Mode
	300	Load to (AC)	
	301	Address = 350	
	302	Next Instruction	
	350	780	
	424	900	
	425	500	
	450	600	
	652	425	
	700	100	
	780	200	

2. a) Explain microprogrammed approach of control unit design and elaborate the superiority of this approach as compared to hardwired control unit with example. **[3]**
   b) Discuss Amdahl's Law for multiprocessor systems. For a given application is running on 32-processor machine and 65% of the application is parallelizable. If processor decreasing from 32 to 16 the find out speed up in both the cases. **[3]**
   c) Explain the Translation Look-Aside Buffer (TLB) and Paging mechanism in Virtual Memory. **[4]**

3. a) Explain different methods to solve data Hazards with examples. Design a processor datapath that incorporates operand forwarding in a pipelined processor. **[4]**
   b) Describe Fast page mode in DRAM using suitable block diagram. **[3]**
   c) The width of the physical address on a machine is 40 bits. What is the width of the tag field in a 512 KB 8-way set associative cache in bits? Find Tag directory size, number of comparators and types of comparators. **[3]**

data insufficient

4. a) A computer system has an L1 cache, an L2 cache, and a main memory unit connected as shown in Figure 1. The block size in L1 cache is 4 words. The block size in L2 cache is 16 words. The memory access times are 2 ns, 20 ns and 200 ns for L1 cache, L2 cache and the main memory unit, respectively.



When there is a miss in both L1 cache and L2 cache, first a block is transferred from main memory to L2 cache and then a block is transferred from L2 cache to L1 cache. What is the total time taken for the transfers? [3]

b) Consider a fully associate cache with 8 cache blocks (numbered 0-7) and the following sequence of memory block requests:

4, 3, 25, 8, 19, 6, 25, 8, 16, 35, 45, 22, 8, 3, 16, 25, 7

Which cache block will have memory block 7, if LRU replacement policy is used? Also, calculate the hit and miss ratio. [3]

c) Consider an instruction pipeline with five stages without any branch prediction: Fetch Instruction (FI), Decode Instruction (DI), Fetch Operand (FO), Execute Instruction (EI) and Write Operand (WO). The stage delay for FI, DI, FO, EI and WO are 5 ns, 7 ns, 10 ns, 8 ns and 6 ns, respectively. There are intermediate storage buffer after each stage and the delay of each buffer is 1 ns. A program consisting of 12 instructions $I1$, $I2$, $I3$ ..... , $I12$ is executed in this pipelined processor. Instruction $I4$ is only the branch instruction and its branch target if $I9$. If the branch is taken during the execution of this program, what is the time ($in\ ns$) needed to complete the program? [4]

5. a) Discuss and explain a non-ideal Synchronous & Asynchronous bus using timing diagram. [5]

b) Explain programmed I/O. Consider proper steps, diagram for the discussion along with application. [5]

OR

6. Find solutions for following questions with respect to Interrupt-driven I/O. [10]

a) How can the processor recognize device requesting an interrupt?
b) Given that different devices are likely to require different interrupt-service routines, how can the processor obtain the starting address of the appropriate routine in each case?
c) Should a device be allowed to interrupt the processor while another interrupt is being serviced?
d) How should two or more simultaneous interrupt requests be handled?

****