**Important Note:** You need to show all your work clearly in deriving the answers. Just writing down the final answers is not enough.

**Suggestion:** Begin by reading all questions and do those first that you think you know best.

1. **[C.U. Design]**
   
   Draw a Moore FSM for that part of the control unit of the `myth8` processor that processes the following instruction:

   ```plaintext
   [LOADIN_AUTO_INCR ri rj rj blank] Semantics: ri = MEM[rj], rj=rj+1;
   ```

   The above instruction is used to access data arrays easily. You need to only provide the states after the decode state of the `myth8` control unit. If appropriate, you can show transitions from the state(s) you design to a `fetch0` state without giving details of the `fetch0` state. The `myth8` processor organization is provided on the next page.
MEMORY SYSTEM

Memory System Diagram:
- Memory Address Bus:
  - Write: \( \phi_1 \), \( \phi_2 \)
  - Read: \( \phi_2 \)
  - MAR: \( \phi_1 \), mar_sel
- Memory Data Bus:
  - CPU: \( \phi_1 \), \( \phi_2 \)
  - CPU Request:
    - ir0_sel, ir1_sel
  - Memory System:
    - To C.U.:
      - Opcode
    - MUX:
      - register file
      - ALU
    - Write Bus:
      - Write Bus A
      - Write Bus B
      - Write Bus
    - ALU:
      - alu_sel[0..2]
      - cin (carry in)
      - cout, m7, v

The Myth8 Processor
2. [2’s complement arithmetic]:
   (a) State the steps of the direct 2’s complement multiplication algorithm (where it is **not necessary** to convert either a negative multiplier or multiplicand into a positive number first). Give a simple 4-bit multiplication example using these steps in which both the multiplier and multiplicand are negative)—note that the result will be an 8-bit 2’s complement number.

   (b) Prove using basics of 2’s complement arithmetic that the 2’s complement multiplication algorithm you have described works correctly.
3. [Pipelining]
(a) Design the C.U. FSM for the Read stage of a pipelined version of the myth8 that has the following stages: Fetch, Decode, Read, Execute, Write. A schematic of a pipelined myth8 is given on the following page. The Read FSM has to interact with the Decode and Execute FSM so that:

(a) It starts processing a new instruction only when the Decode stage indicates that it has put a new instruction in Read’s input register (Read_Reg) by setting signal decode_avail = 1.
(b) It forwards its just completed instruction to the Execute stage’s input register (called Ex_Reg with a load signal Ex_Reg_load that is set by the Read stage) and loads in the read registers into the two output registers of Bus A and B (called BusA_Reg with load signal BusA_Reg_load and BusB_Reg with load signal BusB_Reg_load) only when the Execute stage signals to the Read FSM that it is done with its current instruction by setting signal Ex_done = 1. Assume that a direct path ia available from Read_Reg to Ex_Reg for forwarding the Read stage’s instruction to the Execute stage.
(c) It indicates that the above input registers to the Execute stage have been loaded by setting read_avail = 1.
(d) When it is done with its current instruction it sets read_done = 1 to indicate to the Decode stage that it can now receive a new instructions.

The inputs the Read stage gets from its input register are fields do_read (=1 means a read is needed for that instruction, = 0 means no read is needed), and the r_j and r_k fields. As in the myth8, read is preformed by setting r_j_sel = 1, r_k_sel = 1. Minimize the number of cc’s taken by Read to process an instruction, and the number of states in its FSM. Assume that reading a register can be done in 1 cc.

30

(b) A processor has a 5-stage pipeline F (3cc’s), D (1cc), R (1cc), Ex (variable: 1cc for Logic operations, 2cc’s for ADD/SUB, 4cc’s for MUL/Branch, 5cc’s for DIV), WB (1cc).

Consider a program A whose typical run results in 70K instructions being executed, with one out of every 7 instruction being a branch, and for which typically 60% of “True” branches are taken over all branch instructions. Assume that the distribution of the various operations are: 10% Logic, 40% ADD/SUB, 40% MUL/Branch, and 10% DIV. Determine the total number of cc’s taken to execute program A when a branch prediction scheme with 80% accuracy is used. 20
The Myth8 Pipelined Processor
4. [Memory Hierarchy]

(a) Briefly describe the different write miss strategies, and state with rationale which strategy is used at the cache-MM level and which at the MM-SS level.

(b) A computer system has a 4-level memory hierarchy: 1st-level cache (cache1), 2nd-level cache (cache2), main memory (MM), and secondary storage (SS). Each level has the following characteristics:

1. Cache1: hit time = 2 cc’s, hit rate = 0.9
2. Cache2: block replacement time to cache1 $t_{cache2}$ = 4 cc, hit rate = 0.95
3. MM: block replacement time to cache2 $t_{MM}$ = 40 cc, hit rate = $1 - 10^{-5}$
4. SS: block replacement time to MM $t_{SS}$ = $1.7 \times 10^5$ cc’s.

Determine the AMAT $t_{av}$ for this computer.