Vivian Vijay Ludrick
9914 SE Comps-A
DR. CONCLICAO RODRIGUES COLLEGE OF ENGINEERING

DICOA - Assignment 2

Q.1]

A. MBR ← PC: This operation appears to be loading the value of the Program Counter(PC) into the memory Buffer Register(MBR). This might be part of a fetch operation where the current program counter value is being read from memory.

B. MAR ← X: This operation assigns the value in register X to the memory Address Register (MAR). The MAR typically holds the address of the memory location to be accessed, so this operation is setting the memory address to the value in register X.

C. PC ← Y: Here, the program counter (PC) is being updated with the value from register Y. This is likely a control transfer operation where the program is changing its execution flow by setting the PC to a new value.

d. Memory ← MBR:
This operation suggests that the value to the memory Buffer Register (MBR) is being written to memory. It's common in many computer architectures to use the MBR as an intermediate step for data transfer between memory and the processor.

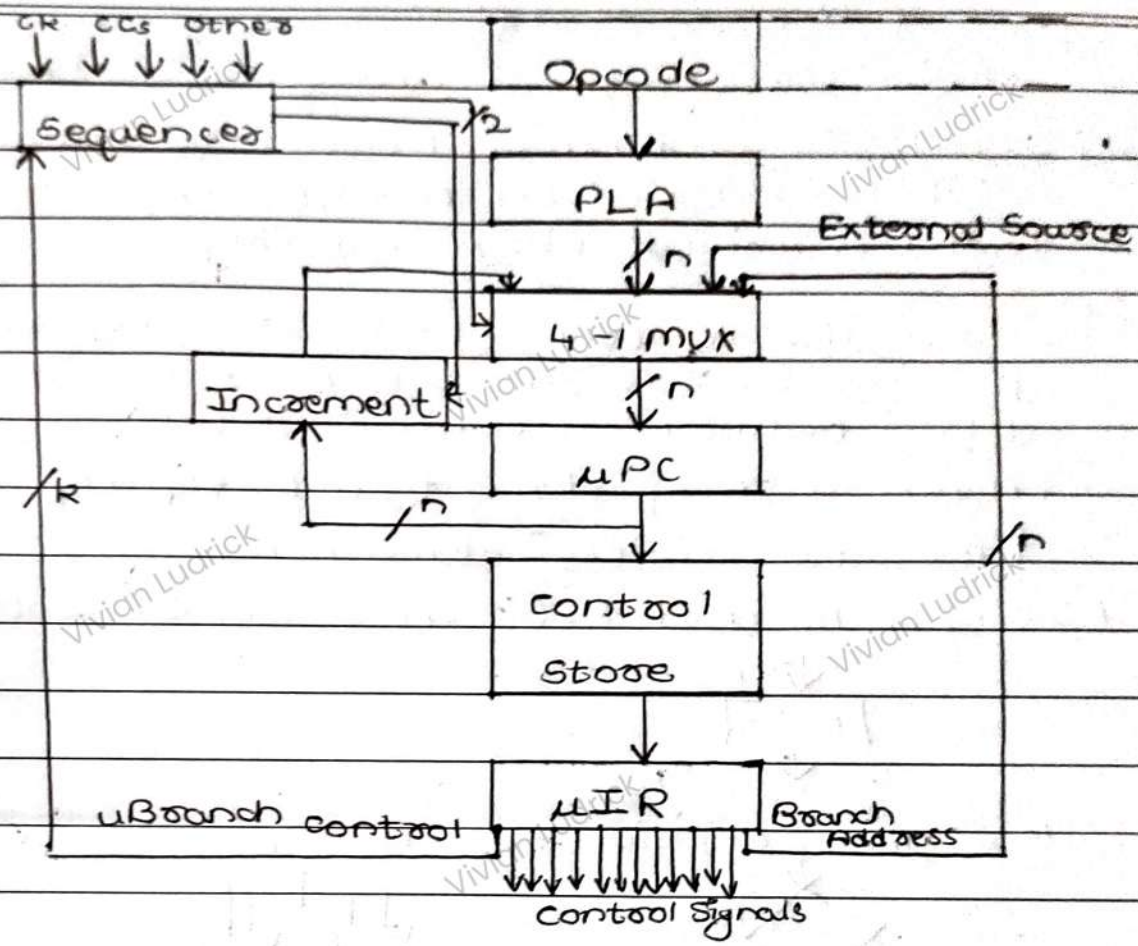
Q.2. MUX has 8 state bits as input lines.
So we require 3 select inputs to select input lines
No. of bits in control memory = 26 - 13 - 3 = 10
address

∴ 10 bit address
∴ 2^{10} memory size
∴ X, Y size = 10, 3

Q.3]

T - state	Operation	Microinstruction
T_1	PC → MAR	PC_{out}, MAR_{in}, Read, Clear, Set C_{in}, Add Z_{in}
T_2	M → MBR, PC ← PC+1	Z_{out}, PC_{in}, Wait for memory fetch cycle
T_3	MBR → IR	MBR_{out}, IR_{in}
T_4	R_3 → x	R_{3out}, X_{in}, CLRC
T_5	R_1 → ALU	R_{1out}, ADD, Z_{in}
T_6	Z → R_3	Z_{out}, R_{3in}
T_7	Check for intr	Assumption enabled intr pending CLRX, SET C, SP_{out}, SUB, Z_{in}
T_8	SP → SP - 1	Z_{out}, SP_{in}, MAR_{in}
T_9	PC → MDR	PC_{out}, MDR_{in}, WRITE
T_{10}	MDR → [SP]	Wait for mem access
T_{11}	PC ← is Raddr	PCin is Raddr out.

Q.4. i]



ii] a. μPC: Holds the address of next control word to be fetched from the control store.

b. incrementer: to increment μPC.

c. Control Store: to store the microroutines of all microinstructions.

d. MicroInstruction Register (μIR): To hold the fetched microinstruction

e. Programmable Logic array (PLA): mapping opcode filled of IR to starting address of microroutine of instruction.

f. 4x1 MUX: μPC can be loaded from:

i] The incremented μPC

ii] The output of PLA

iii] An external source: This allows the uPC to be initialised to a starting value to begin instruction fetch, interrupt services, on reset.

iv] Branch address field from a current microinstruction. This allows unconditional and conditional microbranches

8] Sequencer: Combinational circuit to control 4x1 mux select lines based on microbranch control signals from microinstruction and flags.

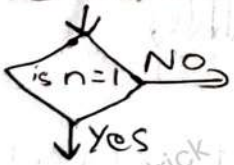
Q 5.] i] Since the control signals are to be activated in a proper sequence, there is need for specific time delay between activation of two conjugate signals

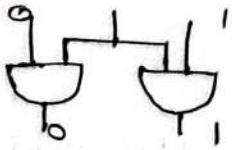
ii] Hence, a delay element, a D flip-flip is used.

iii] The use of delay element has some rules

Rule 1: To be used between 2 sets of control signals

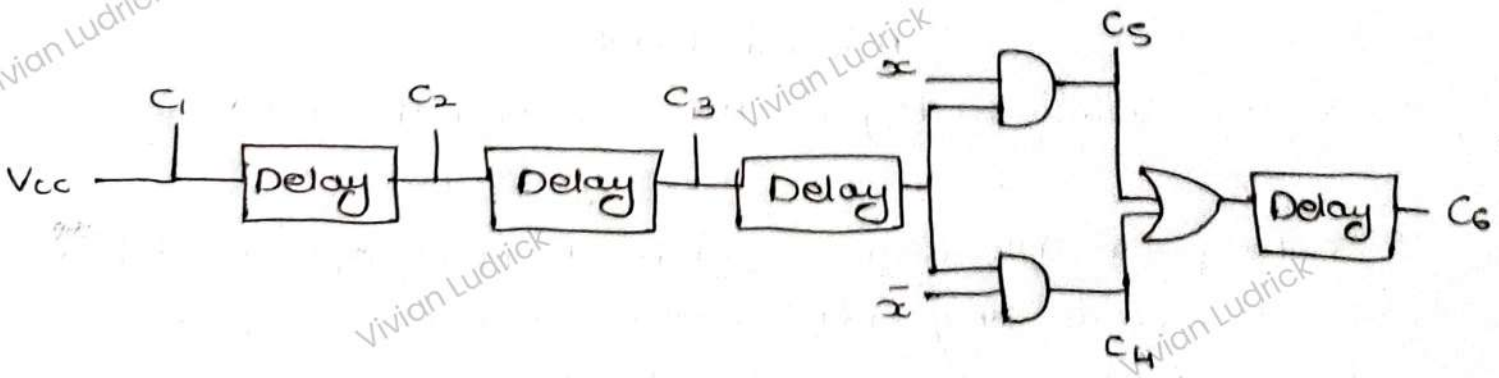
Rule 2: The decision box is implemented using 2 ANDs.





Rule 3: n signals are merged using OR.

iv] Example:



Q.6] 2-way Set.

CM = 16Kb CM = 8Kb

BS = 256 bytes MM = 128 Kb

a) Number of bits in tag

$$2^{tag} = \frac{MM}{CM} = \frac{128Kb}{8Kb} = 2^4$$

∴ tag = 4 bits.

b.) Tag size = 2^4 bytes.

Q.7. Direct mapped

CM = 16Kb BS = 256 bytes MM = 128Kb

tag	block	word

a] Number of bits in tag

$$2^{tag} = MM/CM = 128Kb/16Kb = 2^3$$

tag = 3bits

b] Tag size = 2^3 bytes

Q.8] 8-way sets.

CM = 64Kb CM' = 8Kb MM = 2^{32} bytes

$$2^{tag} = \frac{MM}{CM} = \frac{2^{32}}{2^3 \times 2^{10}} = 2^{19}$$

∴ tag = 19 bits