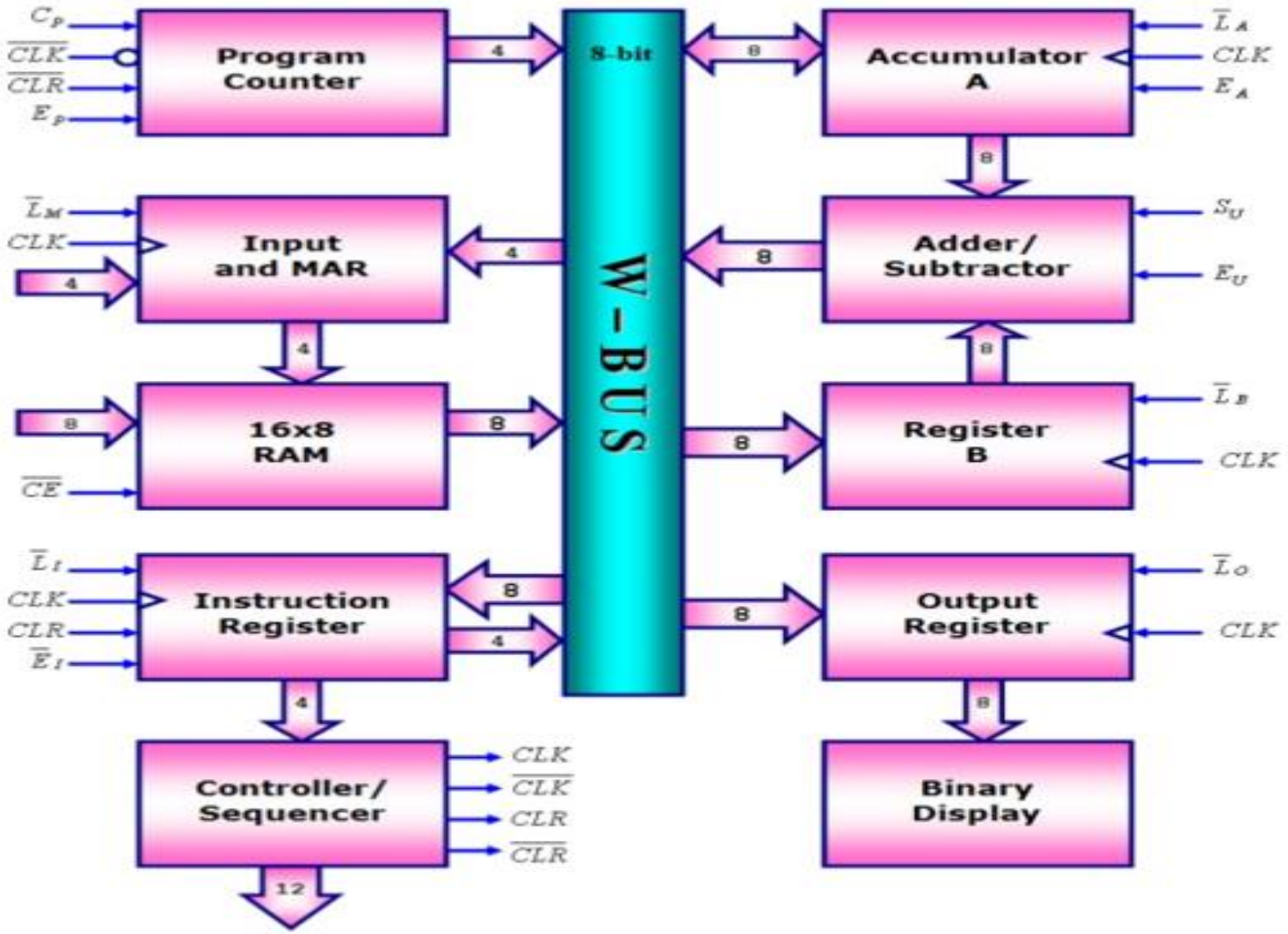
SAP-1(cont..)

SAP-1(Block Diagram)



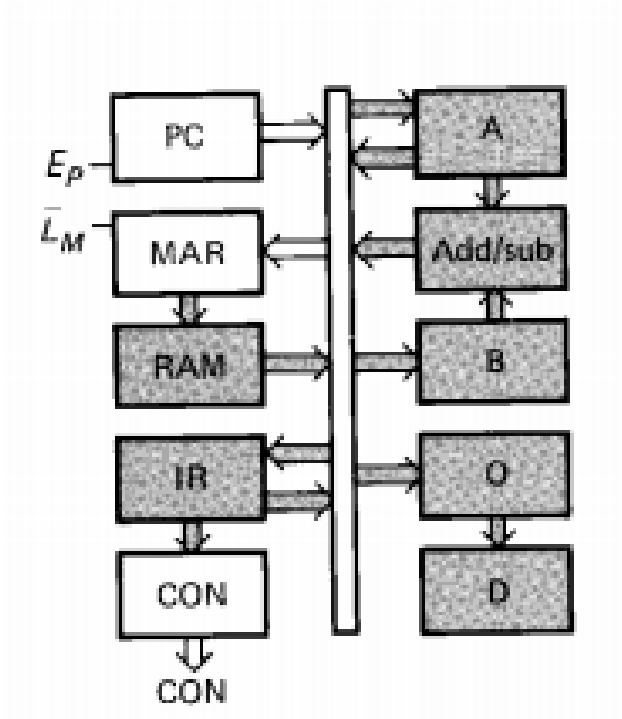
AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Fetch Cycle

- The Control Unit is the Key to a computer's automatic operation.
- The control unit generates the control words that fetch and execute each instruction.(two cycle Fetch and Execute)
- While each instruction is fetched and executed the computer is passes through different timing states (T states).
- SAP-1 contains 6 'T' states. T= T1 T2 T3 T4 T5 T6
- Address State(T1), Increment state(T2) and Memory state(T3) are called Fetch cycle
- T4, T5 and T6 are the execution cycle.

Fetch Cycle

- Address State
  - T1 is called the address state because the address in the PC is transferred to the MAR during this state. E_P and $\overline{L_M}$ active.

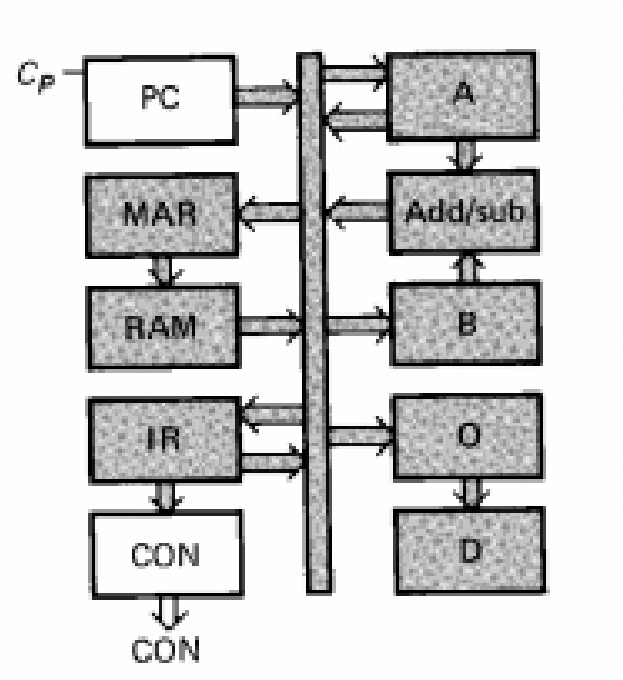


$$\begin{array}{lccccccccc} \text{CON} = & C_P & E_P & \overline{L}_M & \overline{CE} & \overline{L}_I & \overline{E}_I & \overline{L}_A & E_A & S_U & E_U & \overline{L}_B & \overline{L}_O \\ = & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{array}$$

T1 state

Fetch Cycle

- Increment State(T2)
  - This state is called increment state because the program counter is incremented. Here, C_p is active. During the increment state the controller-sequencer is producing a control word of

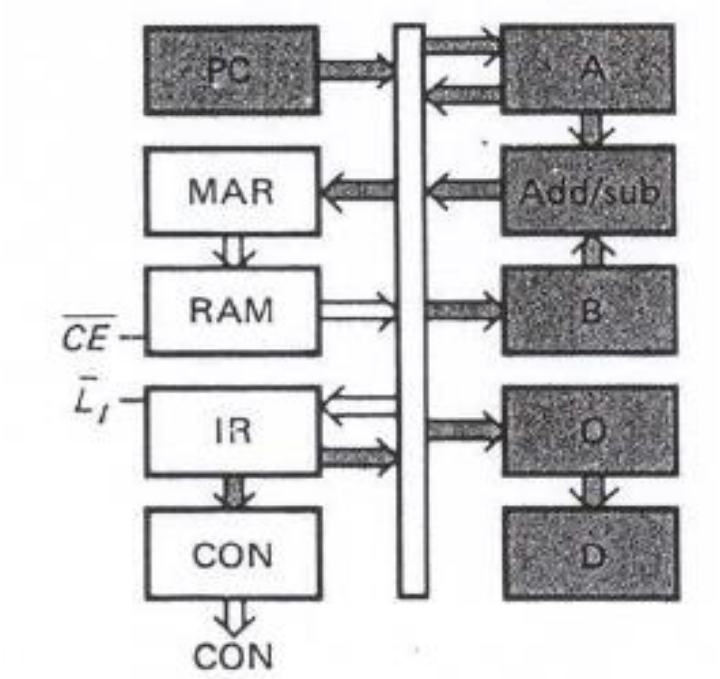


$$\begin{aligned} \text{CON} &= C_P E_P \bar{L}_M \overline{CE} \quad \bar{L}_I \bar{E}_I \bar{L}_A E_A \quad S_U E_U \bar{L}_B \bar{L}_O \\ &= 1\ 0\ 1\ 1 \quad 1\ 1\ 1\ 0 \quad 0\ 0\ 1\ 1 \end{aligned}$$

T2 state

Fetch Cycle

- Memory State
  - The T3 is called the memory state because the addressed RAM instruction is transferred from the memory to the instruction register. The only active control bits during the state are $\overline{CE}$ and $\overline{L_I}$.



T3 State

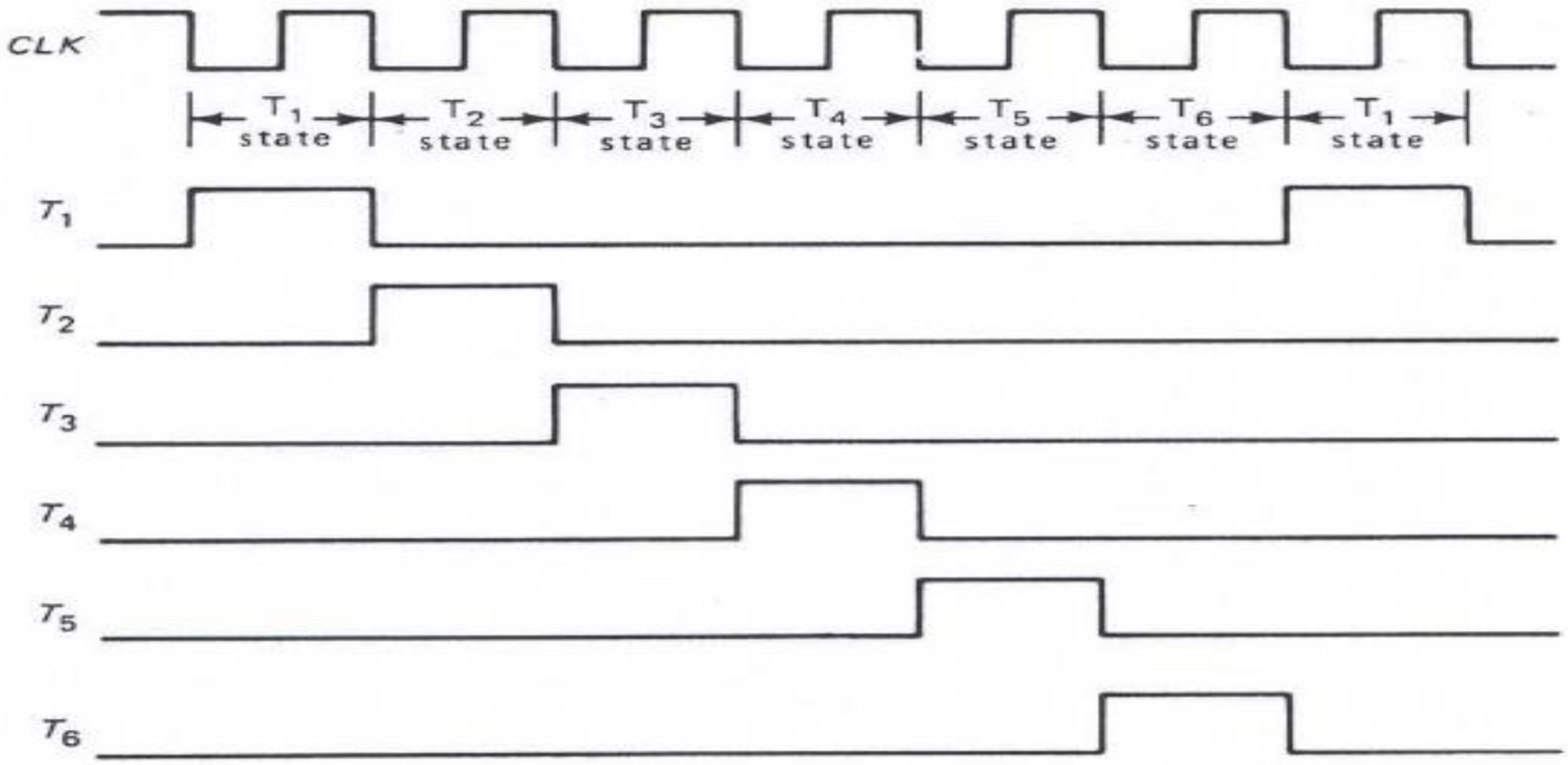
$$\begin{aligned} \text{CON} &= C_P E_P \overline{L}_M \overline{CE} \quad \overline{L}_I \overline{E}_I \overline{L}_A E_A \quad S_U E_U \overline{L}_B \overline{L}_O \\ &= 0\ 0\ 1\ 0 \quad 0\ 1\ 1\ 0 \quad 0\ 0\ 1\ 1 \end{aligned}$$

Overall Fetch Cycle

- During the address state E_P and $\overline{L_M}$ are active; that means the program counter sets up the MAR via W bus.
- C_p is the only control bit during the increment state. This sets up the program counter to count positive clock edges.
- $\overline{CE}$ and $\overline{L_I}$ is active during the memory states. Therefore, the addressed RAM word sets up the instruction register via W bus.
- All these three sates load data, increment and loading of the instruction done in Midway through the states as positive edge of the clock is in the midway of the states(Fig next slide).

Note: Add Diagrams of T1, T2 and T3 cycles.

Clock and Timing Diagram



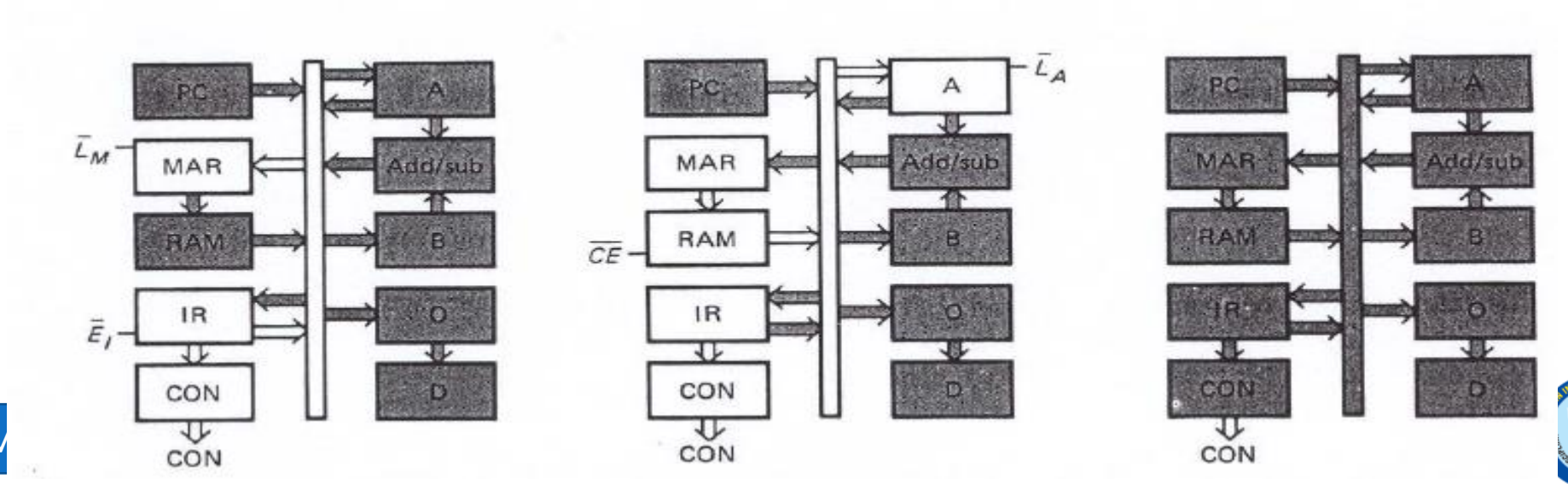
AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Execution Cycle

- The next three states(T4,T5,T6) are the execution cycle of SAP-1. The register transfers during execution cycle depend on the particular instruction being executed.
- For instance LDA 9H requires different register transfers than ADD BH.

LDA Routine

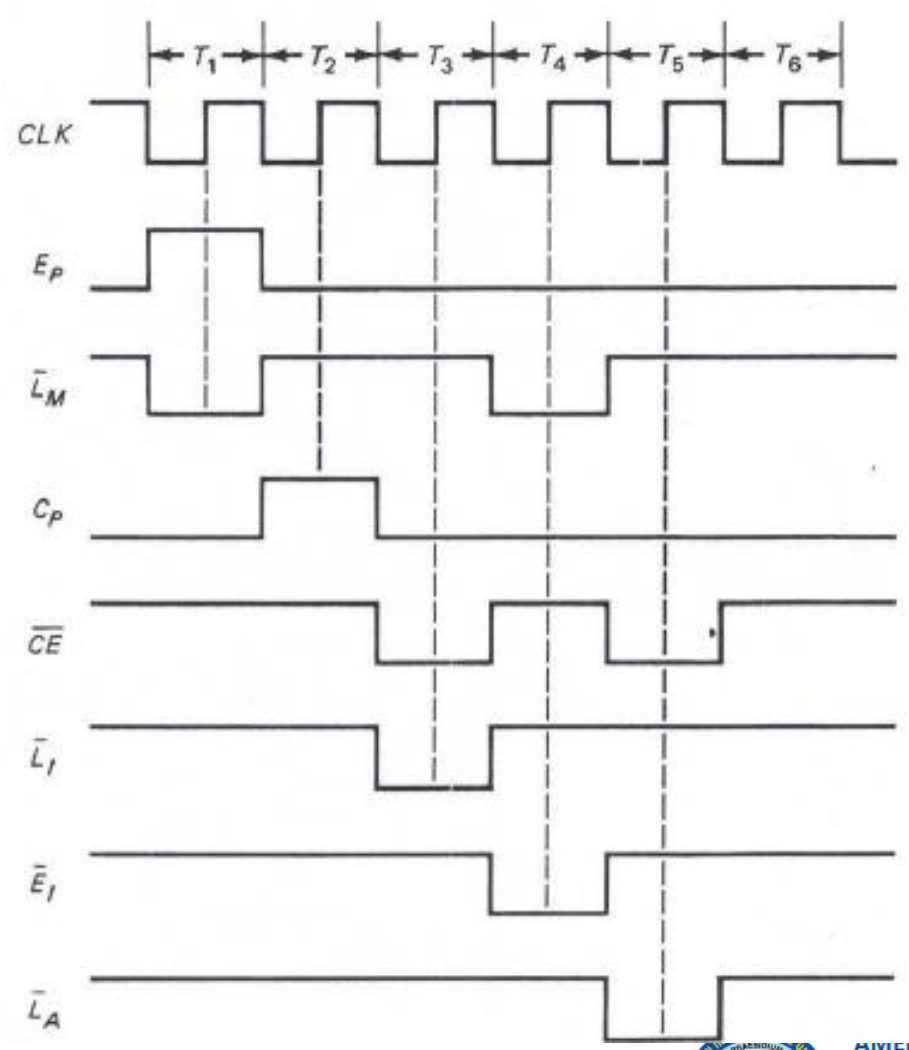
- Assume instruction register loaded with LDA 9H, So IR =0000 1001
- During T4 state, the instruction field 0000 goes to the controller-sequencer, where it is decoded; the address field 1001 is loaded into MAR. $\overline{E_I}$ and $\overline{L_M}$ are active.
- During T5 State, $\overline{CE}$ and $\overline{L_A}$ goes low. This means that the addressed data word in the RAM will be loaded into the accumulator on the next positive clock edge.
- T6 is no operation(NOP) state. All registers are inactive.



AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

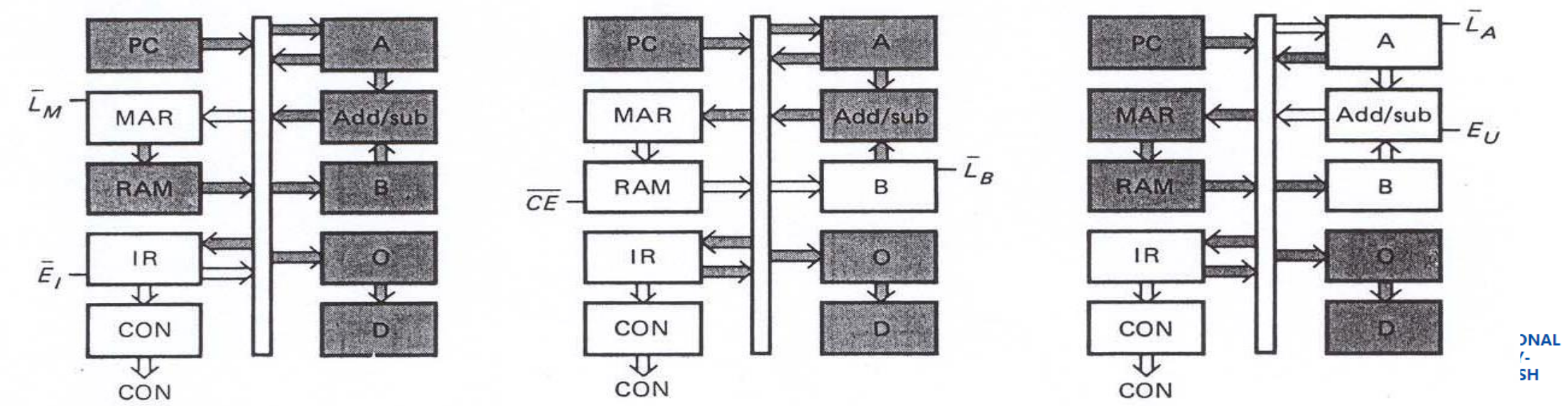
Fetch and LDA timing Diagram

- During T1 state, E_P and $\overline{L_M}$ are active. The +ve clock edge midway through the state transfer the address in the program counter to the MAR.
- During T2 state, C_p is active and the program counter incremented in the +ve clock cycle.
- During T3 state, $\overline{CE}$ and $\overline{L_I}$ are active; when the +ve clock edge occurs, the addressed RAM word is transferred to the instruction register.
- The LDA execution starts with T4 state, where $\overline{CE}$ and $\overline{L_A}$ are active; On the +ve edge of the clock the address field in the instruction register is transferred to the MAR.
- During the T5 state, $\overline{CE}$ and $\overline{L_A}$ are active; the addressed RAM data word is transferred to the accumulator on the +ve clock edge.
- T6 is NOP



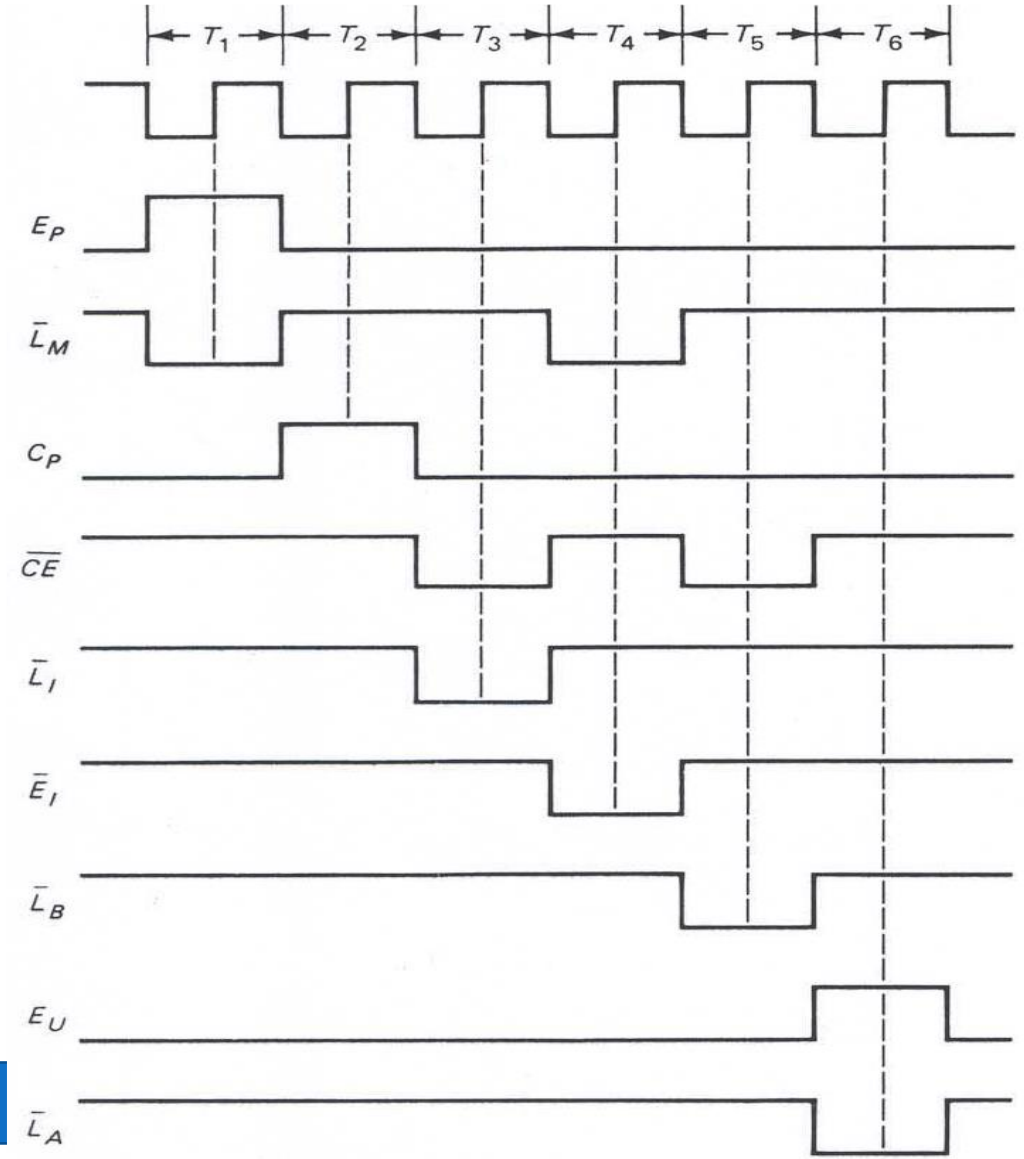
ADD Routine

- Suppose at the end of the fetch cycle the instruction register contains ADD BH: IR= 0001 1011
- During the T4 state, the instruction field goes to the controller sequencer and address field to the MAR. $\overline{E_I}$ and $\overline{L_M}$ are active.
- During T5 state, $\overline{CE}$ and $\overline{L_B}$ are active. This allows the addressed RAM word to set up B register.
- During the T6 state, E_U and $\overline{L_A}$ are active. Therefore, the adder-subtractor sets up the accumulator.



Timing Diagram of fetch & ADD Routine

- During T4 state, $\overline{E_I}$ and $\overline{L_M}$ are active; on the next positive clock edge, the address field in the instruction register goes to MAR.
- During T5 state, $\overline{CE}$ and $\overline{L_B}$ are active; therefore, the addressed RAM word is loaded into the B register at the +ve clock edge.
- During T6 state, E_U and $\overline{L_A}$ are active. When the +ve clock edge hits, the sum out of adder-subtractor is stored in the accumulator.

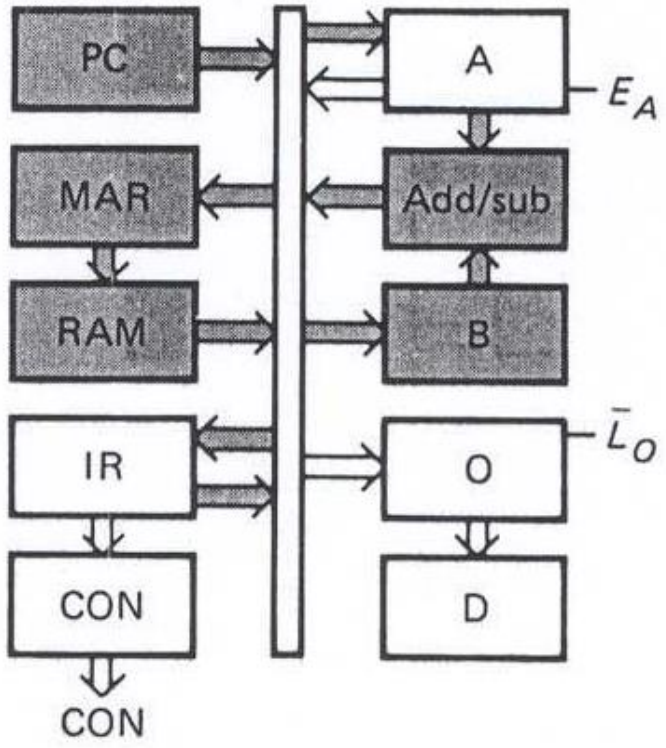


SUB Routine

- Similar to the ADD routine.
- Only differences is during T6 state, a high S_U is sent to the adder-subtractor.
- Timing diagram is almost identical except T6 state where S_U is active.

OUT Routine

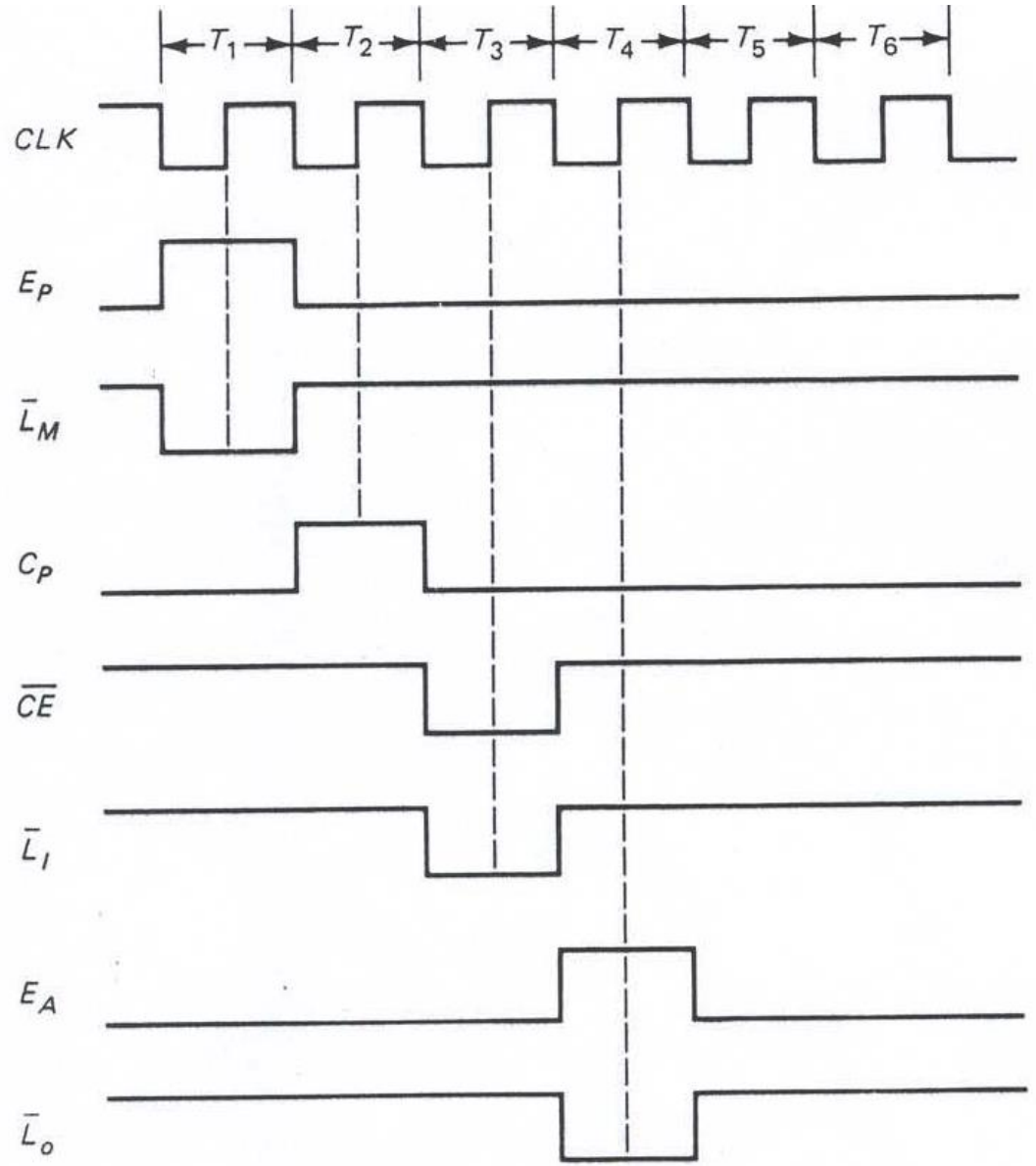
- Suppose the instruction register contains the OUT instruction at the end of the fetch cycle. Then, IR= 1110 XXXX.
- The instruction field goes to the controller sequencer for decoding. Then the controller-sequencer sends out the control word needed to load the accumulator contents into the output register.
- During T4 state, E_A and $\overline{L_0}$ are active. The next +ve edge load the accumulator contents into the output register during T4 state.
- T5 and T6 states are NOP.



AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Fetch and Out Timing Diagram

- Fetch cycle is same for all routine (T1,T2 and T3)
- During T4 state, E_A and $\overline{L_0}$ are active; this transfers the accumulator word to the output register when the +ve clock edge occurs.



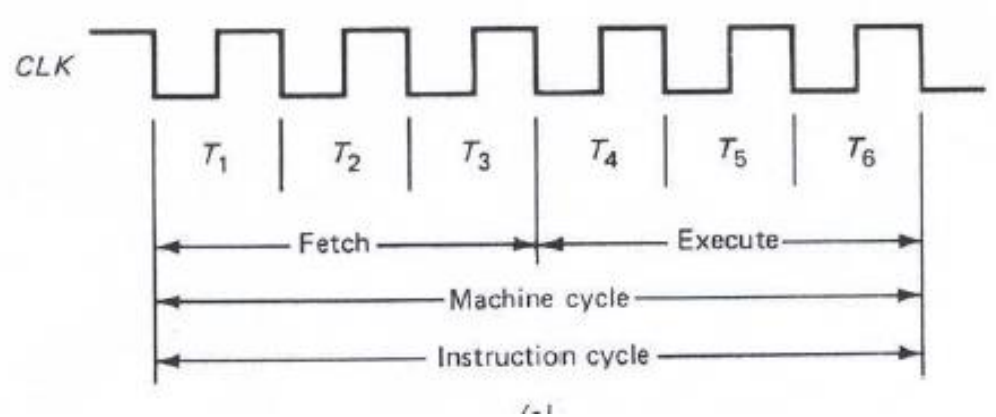
HLT Routine

- HLT does not require a control routine because no registers are involved in the execution of an HLT instruction. When the IR contains

  IR= 1111 XXXX

- The instruction field 1111 signals the controller sequencer to stop processing data.
- The controller sequencer stops the computer by turning off the clock.

Machine Cycle & Instruction Cycle



6 T states are called **machine cycle**

The Number of T states needed to fetch and execute an instruction is called **instruction cycle.**

SAP-1 control Signals

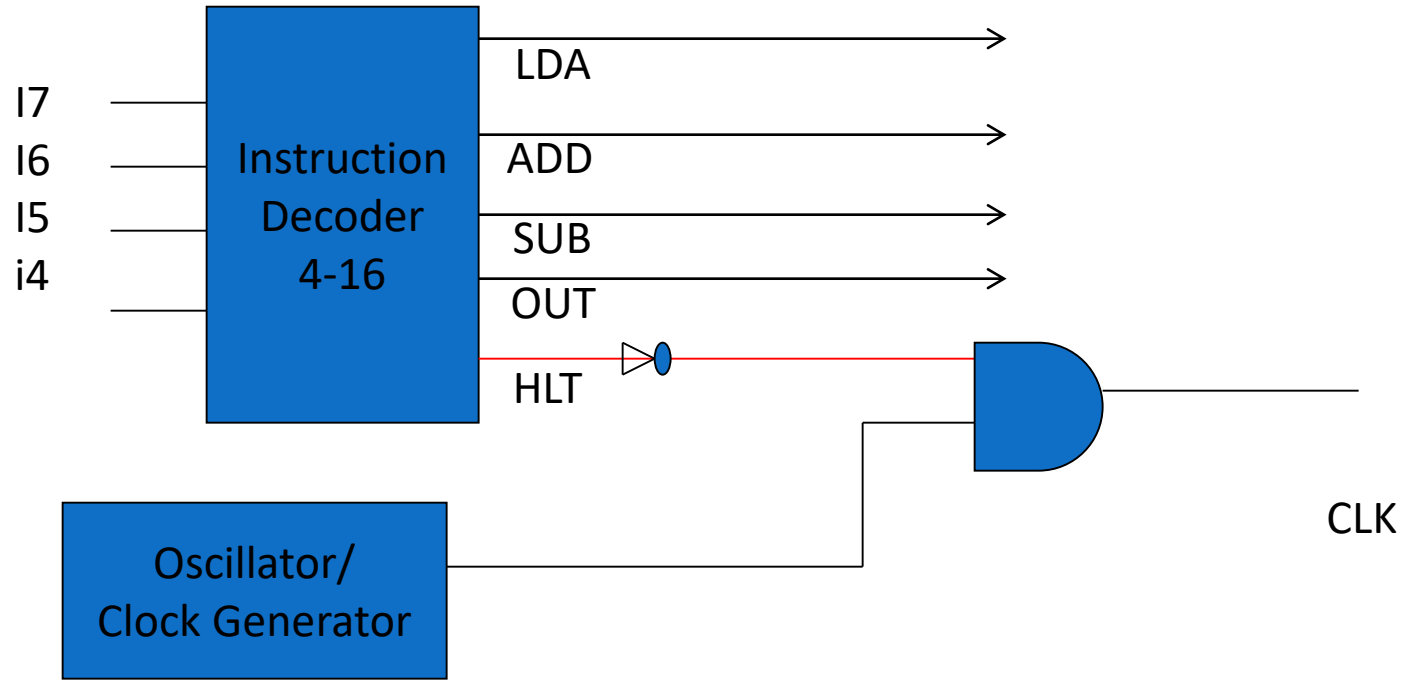
Fetch Cycle- T1,T2,T3
Execution Cycle- T4,T5,T6

Instruction	T1	T2	T3	T4	T5	T6
LDA	Ep Lm'	Cp	Ce' Li'	Ei' Lm'	Ce' La'	X
ADD	Ep Lm'	Cp	**Ce' Li'**	Ei' Lm'	Ce' Lb'	Eu La'
SUB	Ep Lm'	Cp	**Ce' Li'**	Ei' Lm'	Ce' Lb'	Su La'
OUT	Ep Lm'	Cp	**Ce' Li'**	Ea Lo'	X	X
HLT	Ep Lm'	Cp	**Ce' Li'**	X	X	X

State Equations

- Ep= T1
- Cp= T2
- Lm’= T1+ T4.LDA+T4.ADD+T4.SUB
- Ce’= T3+ T5.LDA+T5.ADD+T5.SUB
- Li’= T3
- Ei’= T4.LDA+T4.ADD+T4.SUB
- La’= T5.LDA+T6.ADD+T6.SUB
- Ea= T4.OUT
- Su= T6.SUB
- Eu= T6.ADD
- Lb’= T5.ADD+T5.SUB
- Lo’= T4.OUT

Implementation of HLT



Microprogramming

- Reduces the complexity of control circuit.
- With large instructions control matrix become very complicated and requires hundreds even thousands of gates.
- In Microprogramming, microinstructions are stored in a ROM instead of producing them with hardwired control matrix.

Address State (T = 00 0001 = 1 = T_1)

- The T_1 state is called the *address state* because the address in the program counter (PC) is transferred to the memory address register (MAR) during this state.
- During the address state, E_P and L'_M are active; all other control bits are inactive. This means that the controller-sequencer is sending out a control word of 5E3H during this state

$$CON = C_P \cdot E_P \cdot \overline{L}_M \cdot \overline{CE} \cdot \overline{L}_I \cdot \overline{E}_I \cdot \overline{L}_A \cdot E_A \cdot S_U \cdot E_U \cdot \overline{L}_B \cdot \overline{L}_O$$

$$= \quad 0 \quad 1 \quad 0 \quad 1 \qquad 1 \quad 1 \quad 1 \; 0 \qquad 0 \quad 0 \quad 1 \quad 1$$

$$= \qquad 5 \qquad\qquad E \qquad\qquad 3$$

Increment State (T = 00 0010 = 2 = T_2)

- The T_2 state is called the *increment state* because the program counter is incremented.
- During the increment state, the controller-sequencer is producing a control word of BE3H
- Only the C_P bit is active in this state.

$$CON = C_P \cdot E_P \cdot \overline{L}_M \cdot \overline{CE} \cdot \overline{L}_I \cdot \overline{E}_I \cdot \overline{L}_A \cdot E_A \cdot S_U \cdot E_U \cdot \overline{L}_B \cdot \overline{L}_O$$

$$= \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1$$

$$= \quad \text{B} \qquad\qquad \text{E} \qquad\qquad \text{3}$$

Memory State (T = 00 0100 = 4 = T_3)

- The T_3 state is called the *memory state* because the addressed RAM instruction is transferred from the memory to the instruction register.
- The only active control bits during this state are CE' and L_I, and the word out of the controller-sequencer is 263H

$$CON = C_P \cdot E_P \cdot \overline{L}_M \cdot \overline{CE} \cdot \overline{L}_I \cdot \overline{E}_I \cdot \overline{L}_A \cdot E_A \cdot S_U \cdot E_U \cdot \overline{L}_B \cdot \overline{L}_O$$

$$= \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1$$

$$= \quad 2 \quad\quad\quad 6 \quad\quad\quad 3$$

Fetch and Execute Cycle of SAP-1

Macro Inst.	T State	Micro Operation	Active	CON
All Instructions	T_1	MAR ← PC	L'_M, E_P	5E3H
	T_2	PC← PC+1	C_P	BE3H
	T_3	IR ← RAM[MAR]	CE', L'_I	263H
LDA	T_4	MAR ← IR(3…0)	L'_M, E'_I	1A3H
	T_5	ACC ← RAM[MAR]	CE', L'_A	2C3H
	T_6	None	None	3E3H
ADD	T_4	MAR ← IR(3…0)	L'_M, E'_I	1A3H
	T_5	B ← RAM[MAR]	CE', L'_B	2E1H
	T_6	ACC ← ACC+B	L'_A, E_U	3C7H
SUB	T_4	MAR ← IR(3…0)	L'_M, E'_I	1A3H
	T_5	B ← RAM[MAR]	CE', L'_B	2E1H
	T_6	ACC ← ACC – B	L'_A, S_U, E_U	3CFH
OUT	T_4	OUT ← ACC	E_A, L'_O	3F2H
	T_5	None	None	3E3H
	T_6	None	None	3E3H
HLT	T_3	None	HLT '	263H

Storing the microprogram

- By assigning addresses and including the fetch routine, come up with SAP-1 microinstructions.
- These microinstructions can be stored in **Control ROM** with the fetch routine at addresses shown in Table.
- To access any routine, we need to apply correct addresses. For instance to get ADD, apply 6H,7H and 8H.
- Three steps of accessing routine
  - Knowing the starting address
  - Stepping through the routine address
  - Apply the address to the Control ROM

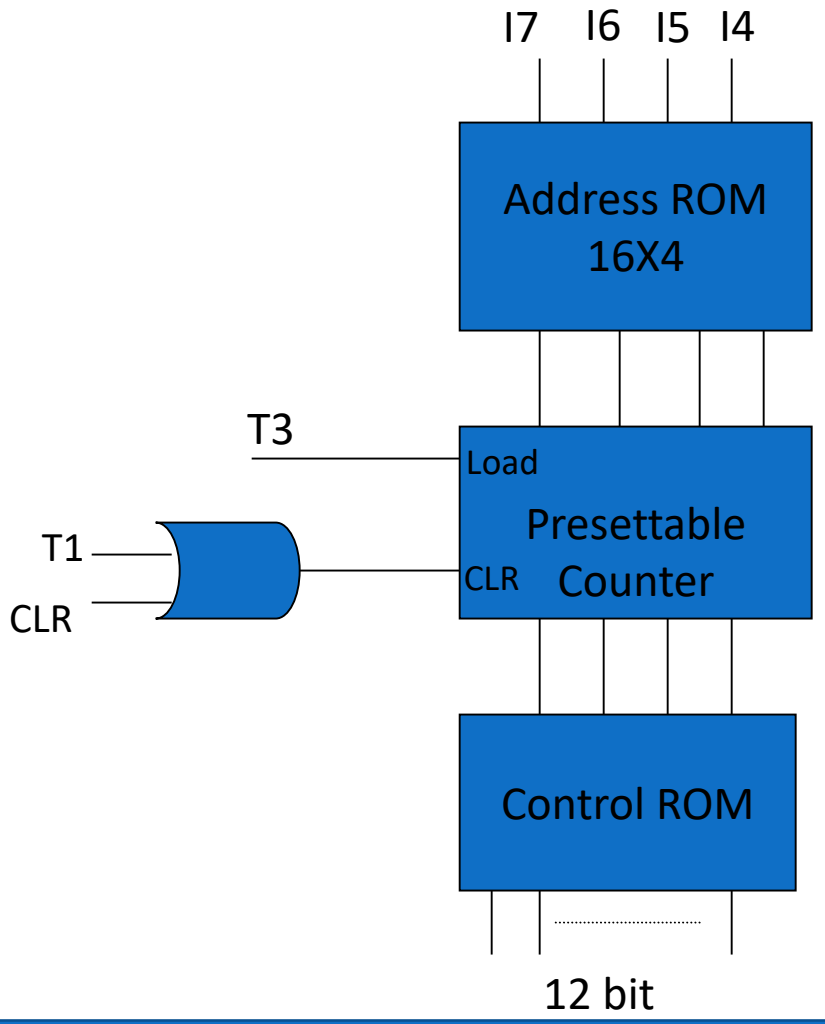
Address	CON	Routine	Active
0H	5E3H	Fetch	Ep,Lm'
1H	BE3H		Cp
2H	263H		CE',Li'
3H	1A3H	LDA	Lm',EI'
4H	2C3H		CE',La'
5H	3E3H		None
6H	1A3H	ADD	Lm',EI'
7H	2E1H		CE',Lb'
8H	3C7H		La',Eu
9H	1A3H	SUB	Lm',EI'
AH	2E1H		CE',Lb'
BH	3CFH		La',Su,Eu
CH	3F2H	OUT	Ea,Lo'
DH	3E3H		None
EH	3E3H		None
FH	X	X	NOT USED

SAP-1 Address ROM

- Microprogrammed Control Circuit of SAP-1 has an address ROM, a pre-settable counter and a control ROM.
- The Address ROM contains the staring addresses of each routine. For instance the starting address of LDA is 3H means 0011.
- When the opcode bits $I_7 I_6\ I_5\ I_4$ drive the address ROM, the staring address is generate.

Address	Contents	Routine
0000	0011	LDA
0001	0110	ADD
0010	1001	SUB
0011	XXXX	NONE
0100	XXXX	NONE
0101	XXXX	NONE
0110	XXXX	NONE
0111	XXXX	NONE
1000	XXXX	NONE
1001	XXXX	NONE
1010	XXXX	NONE
1011	XXXX	NONE
1100	XXXX	NONE
1101	XXXX	NONE
1110	1100	OUT
1111	XXXX	NONE

Microprogrammed Controller circuit



AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Microprogram SAP-1

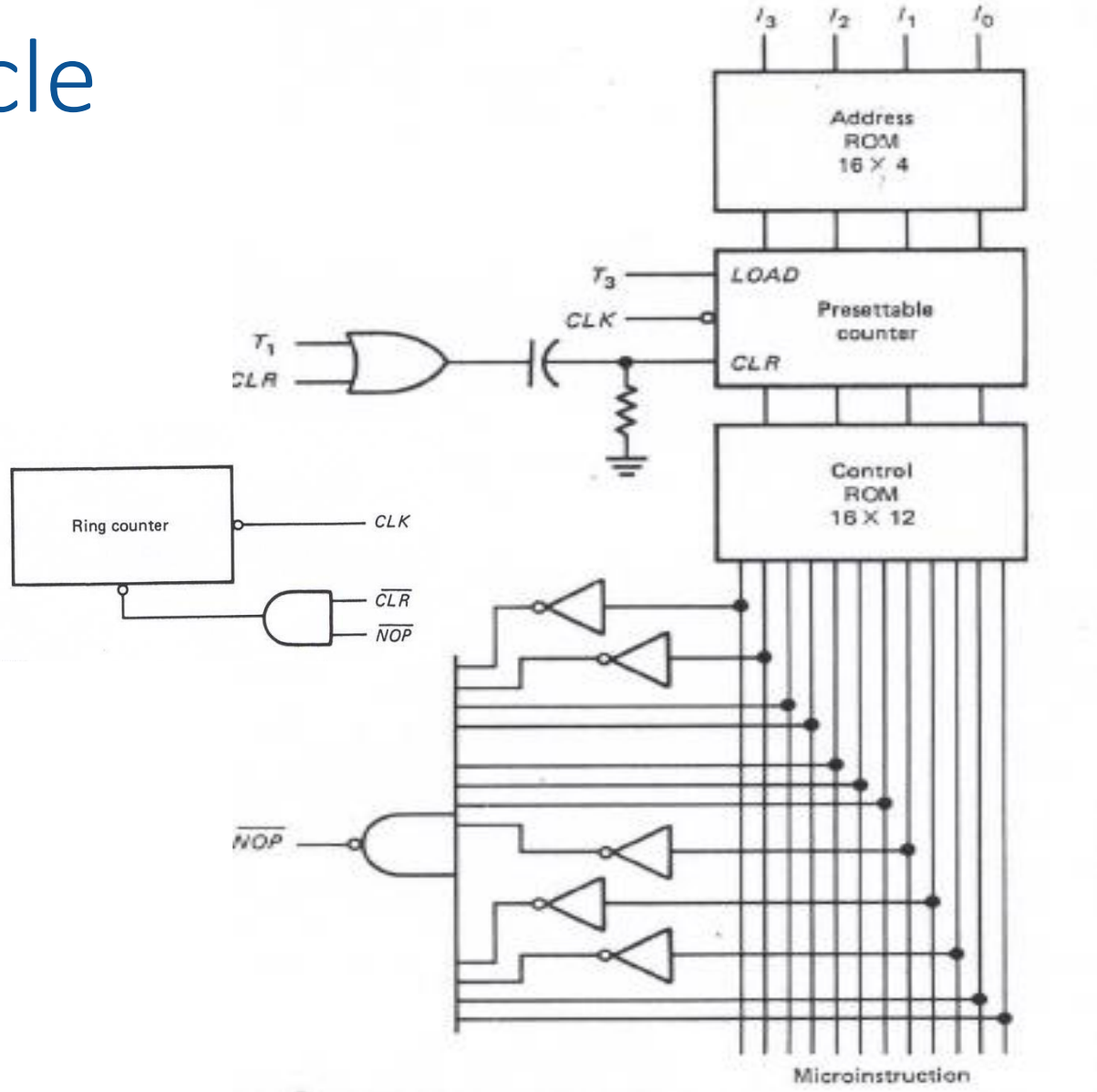
- Presettable Counter
- When T3 is high, the load input of the presettable counter is high and the counter loads the starting address form the address ROM. During the other T states the counter counts.
- Control ROM
  - The Control ROM store the SAP-1 microinstructions. During fetch instructions it receives 0000, 0001 and 0010 therefore, the outputs are 5E3H,BE3H,263H

Variable Machine Cycle

- There is A microinstruction called NOP. It occurs once in LDA and twice in the OUT routine. These NOPs are used in SAP-1 to get a fixed machine cycle for all instructions. In other words each machine cycle takes exactly 6 T states. No matter what the instruction is.
- In some computers fixed machine cycle is advantage.
- But when speed is important, the NOPs are a waste of time and can be eliminated.
- One way to speed up machine is to skip any T state with NOP. This will shorten LDA instructions ot 5 states and OUT instructions to 4 states. This is called variable machine cycle.

Variable Machine Cycle

- Redesigned to speed up machine cycle.
- With an LDA instruction, the actions is same as before during the T1 to T5 states.
- When T6 begins the control ROM sends output 3E3H. The NAND gates detect the NOP instantly and produce a low output signal, which is feedback to a ring counter through an and gate.
- This resets the ring counter to T1 states, and a new machine cycle begins.



INTERNATIONAL UNIVERSITY-BANGLADESH

Question

- What is the motivation behind Microprogramming? With complete block diagram briefly explain the controller/sequencer design using microprogramming technique. Control word of the controller/sequencer is given.
- Specify SAP-1 Control ROM and Address ROM contents.