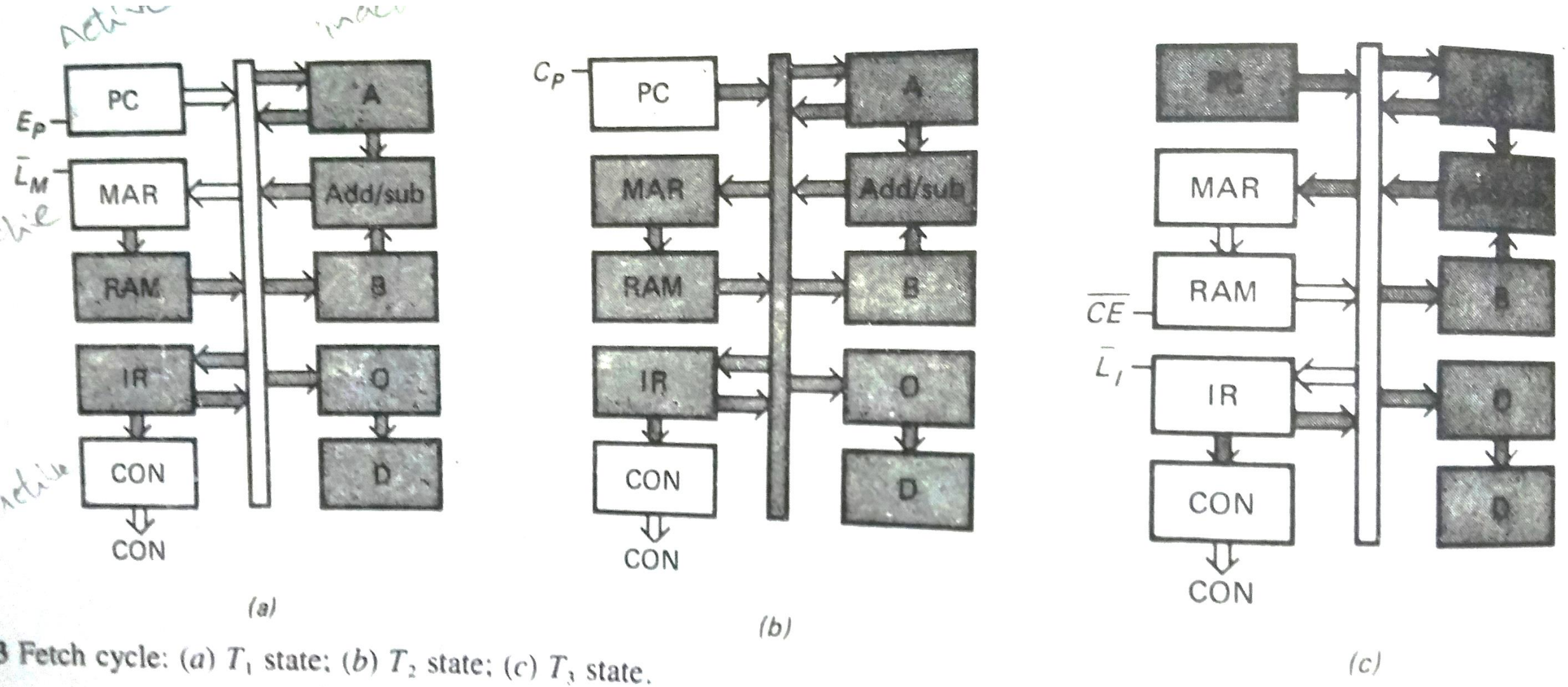
FETCH CYCLE

Fetch cycle : T1, T2, T3



3 Fetch cycle: (*a*) T_1 state; (*b*) T_2 state; (*c*) T_3 state.

T1: Address state

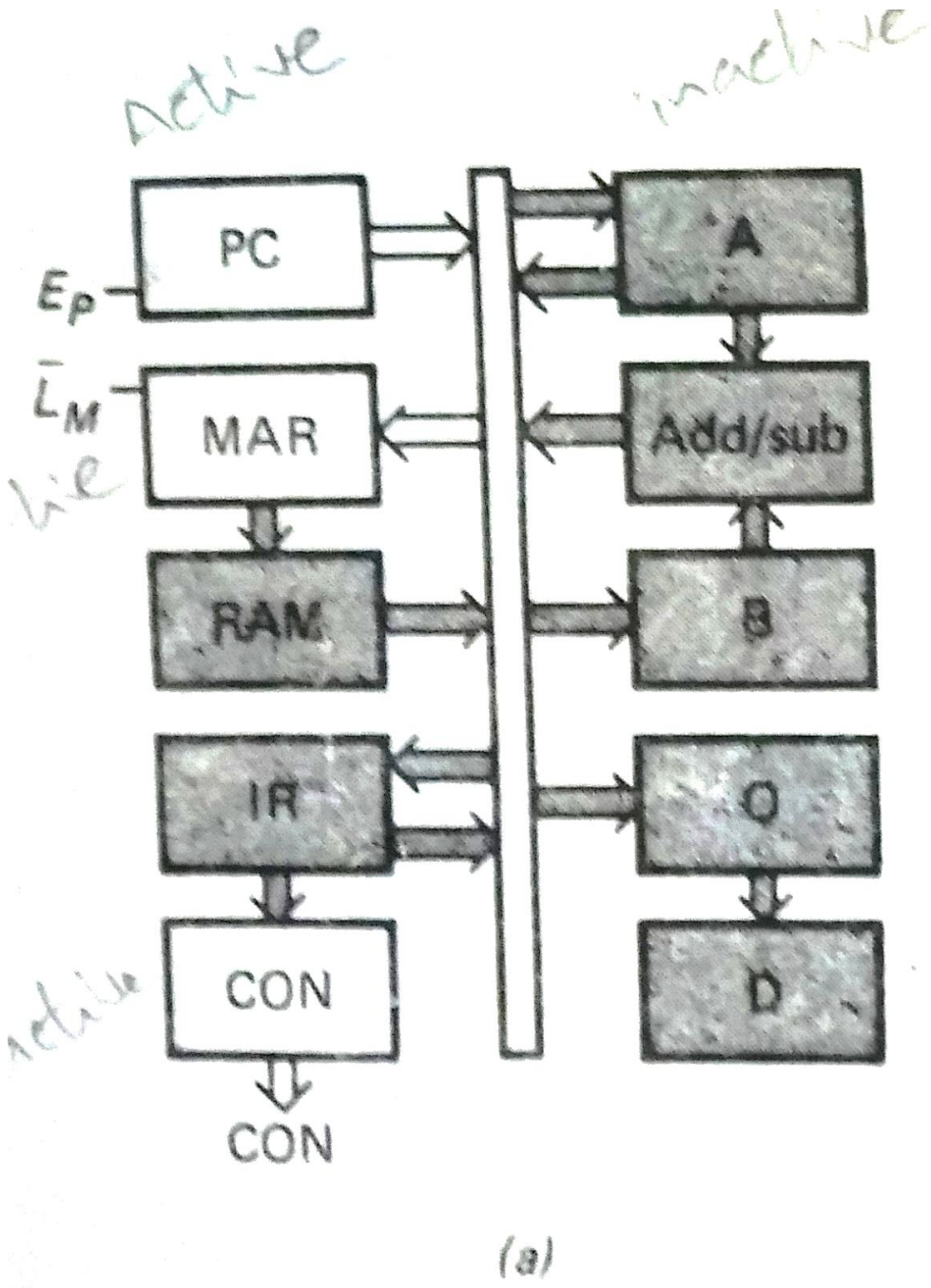
Address State

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. Figure 10-3*a* shows the computer sections that are active during this state (active parts are light; inactive parts are dark).

During the address state, E_P and $\overline{L}_M$ are active; all other control bits are inactive. This means that the controller-sequencer is sending out a control word of

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

during this state.



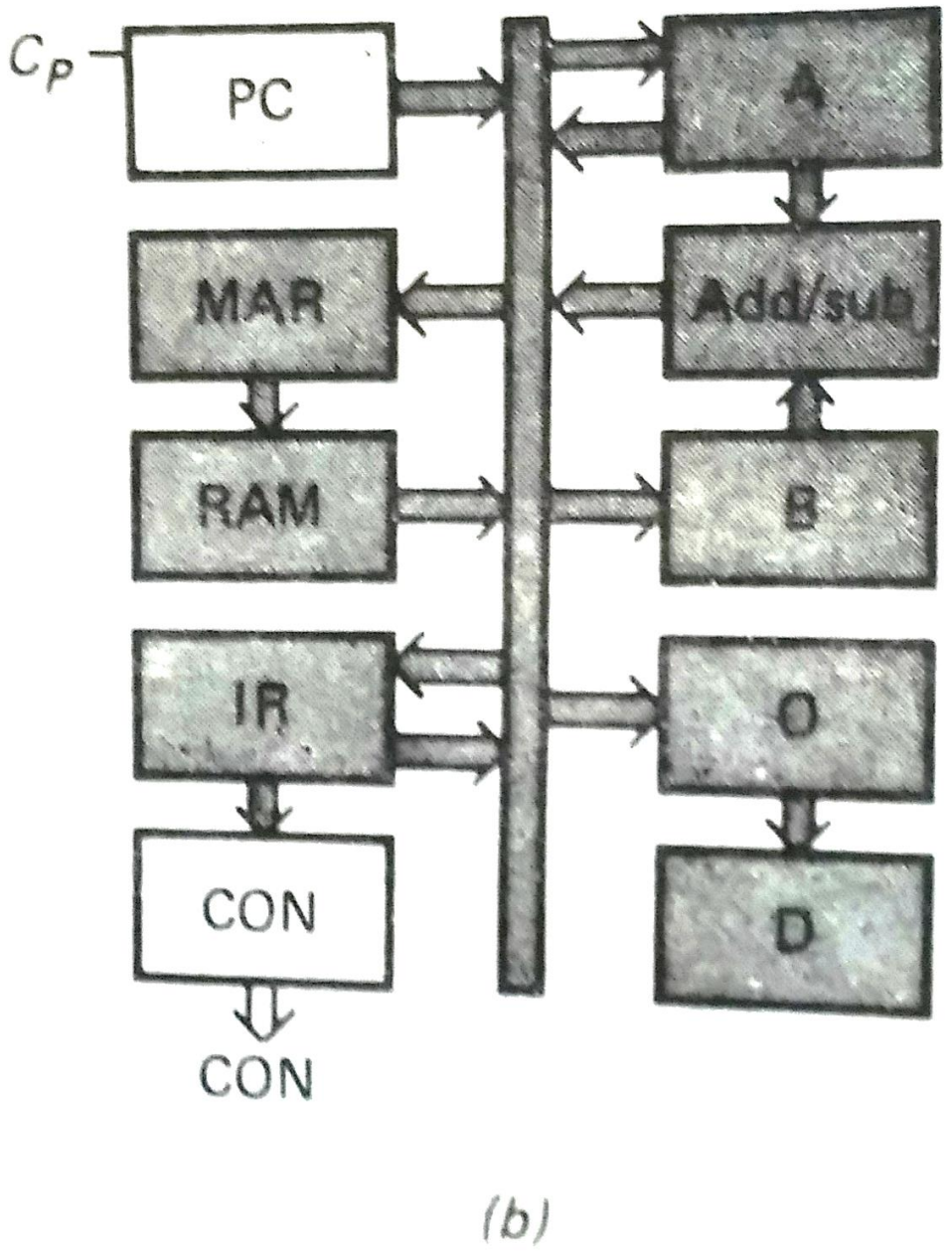
T2: Increment state

Increment State

Figure 10-3*b* shows the active parts of SAP-1 during the T_2 state. This 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

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

As you see, the C_P bit is active.



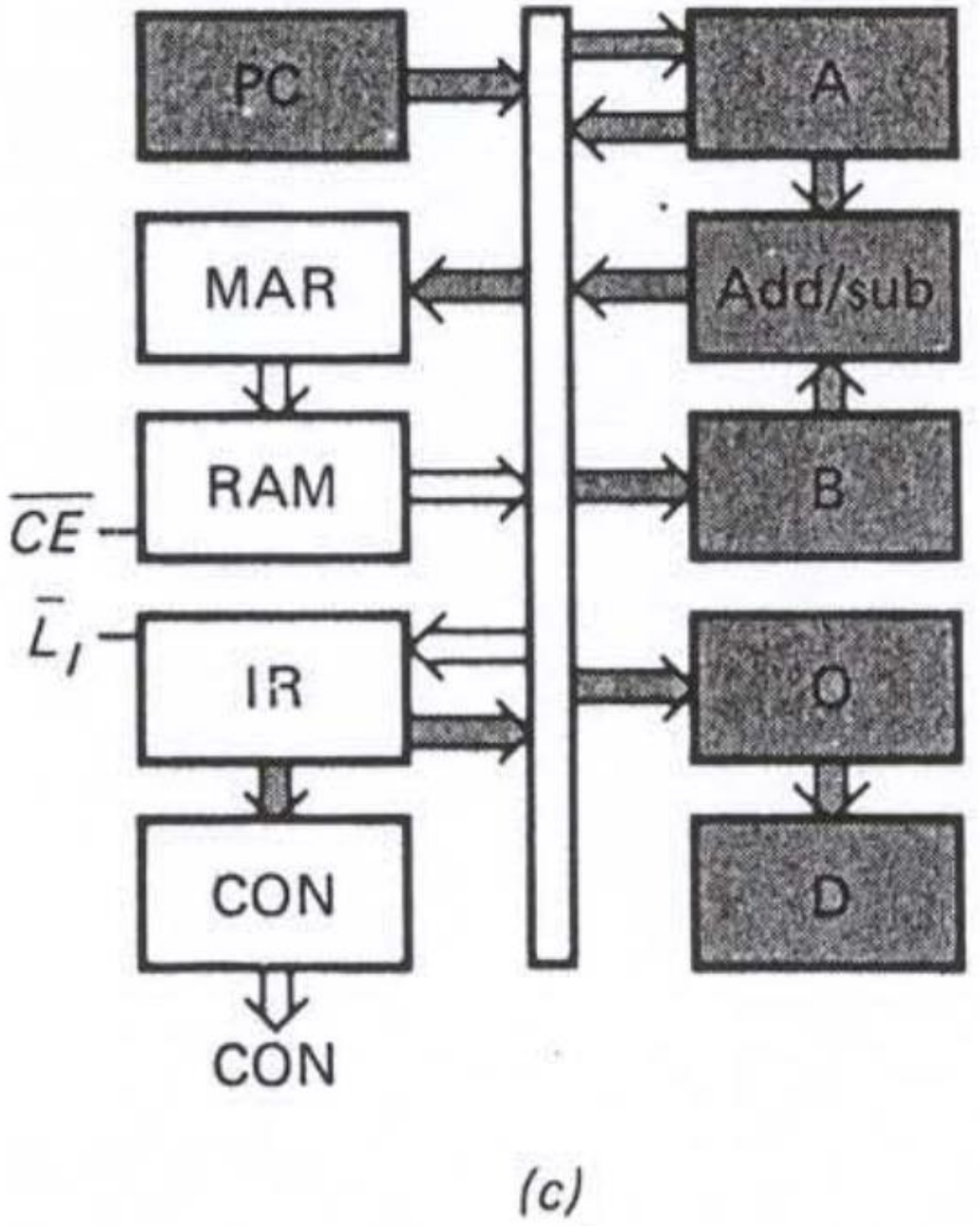
(b)

T3: memory state

Memory State

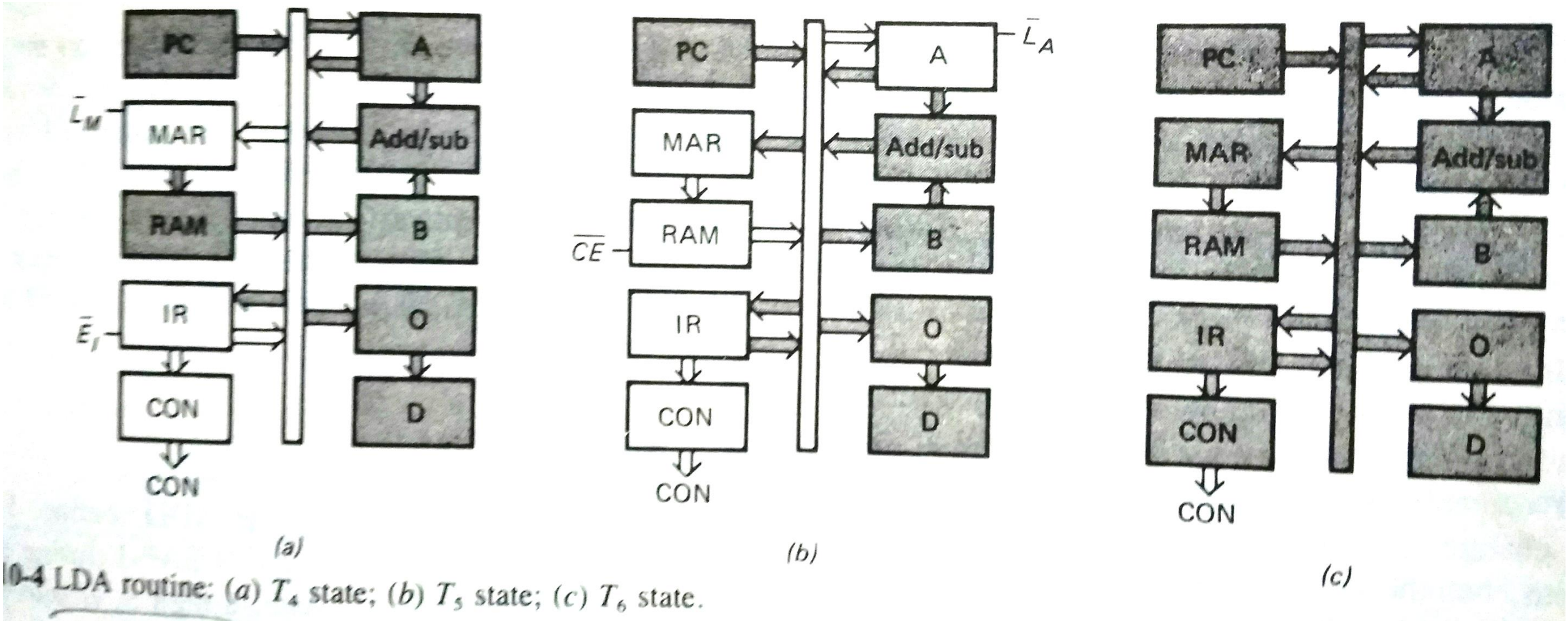
The T_3 state is called the *memory state* because the addressed RAM instruction is transferred from the memory to the instruction register. Figure 10-3*c* shows the active parts of SAP-1 during the memory state. The only active control bits during this state are $\overline{CE}$ and $\overline{L}_I$, and the word out of the controller-sequencer is

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



EXECUTE CYCLE

LDA routine (a) T4 state (b) T5 state (c) T6 state



0-4 LDA routine: (*a*) T_4 state; (*b*) T_5 state; (*c*) T_6 state.

LDA routine (a) T4 state

Let's assume that the instruction register has been loaded with LDA 9H:

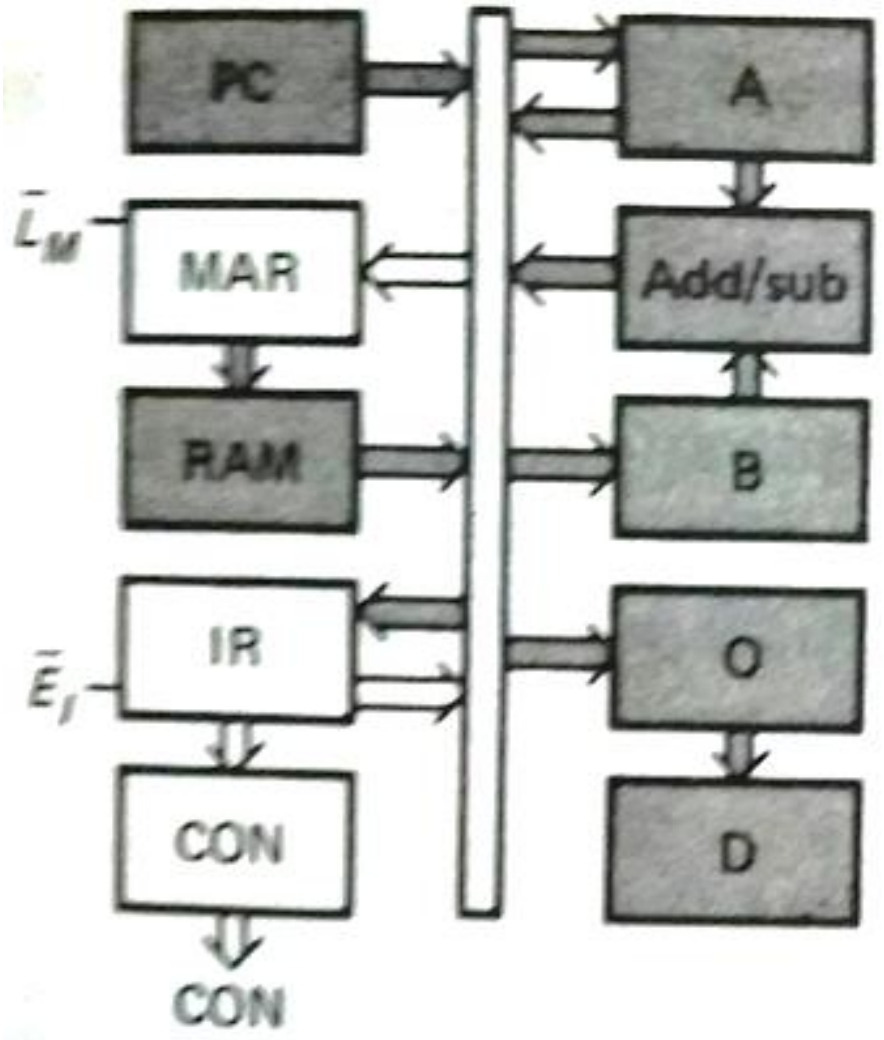
IR = 0000 1001

T4 state: The instruction field 0000 goes to the controller-sequencer, where it is decoded. The address field 1001 is loaded into the MAR. At this state, $\overline{E_I}$ and $\overline{L_M}$ are active.

The controller sequence word is:

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

0 0 0 1 1 0 1 0 0 0 1 1



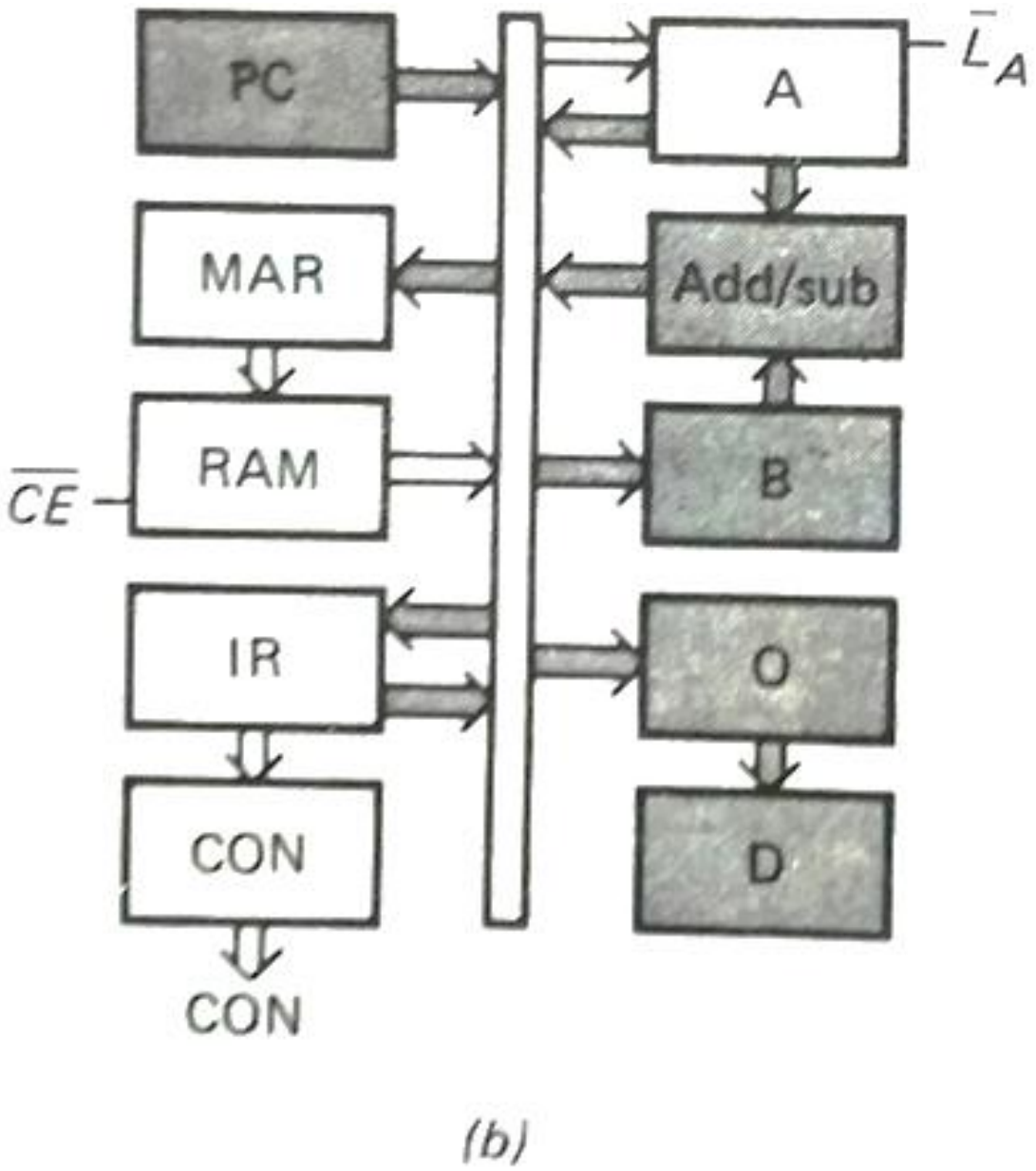
LDA routine (b) T5 state

T5 state: During T5 state, $\overline{CE}$ and $\overline{L_A}$ go low (active). The addressed data word in the RAM will loaded into the accumulator.

The controller sequence word is:

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

0 0 1 0 1 1 0 0 0 0 1 1



(b)

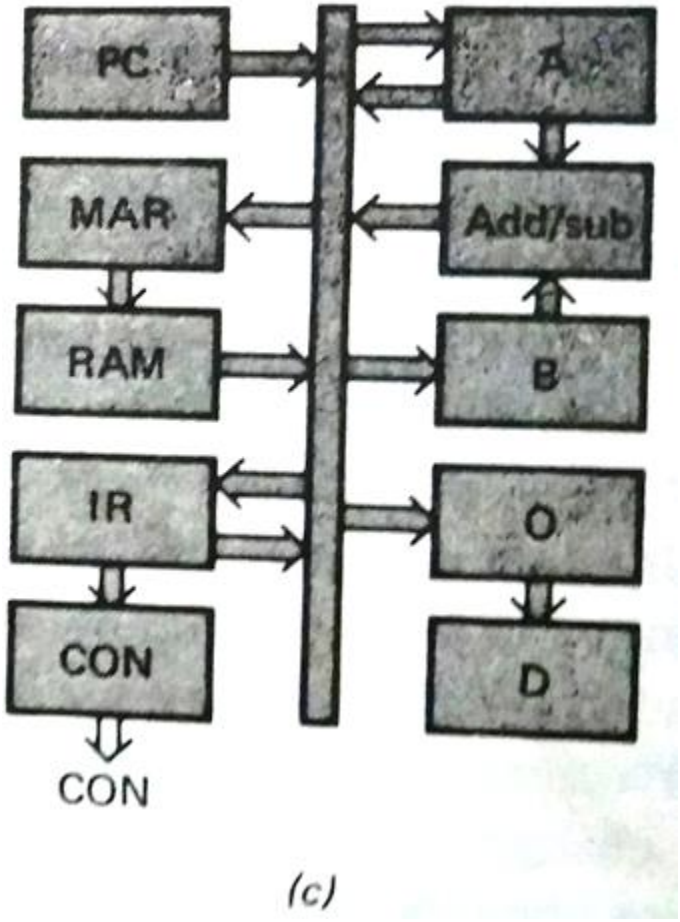
LDA routine (c) T6 state

T6 state: T6 state is non-operational for LDA routine, all registers all inactive. This means that the controller-sequencer is sending out a word whose bits all inactive. The T6 state of LDA routine is inactive.

The controller sequence word is:

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

0 0 1 1 1 1 1 0 0 0 1 1



Timing diagram showing LDA routine (Fetch cycle and Execute cycle)

Figure 10-5 shows the timing diagram for the fetch and LDA routines. During the T_1 state, E_P and $\overline{L}_M$ are active; the positive clock edge midway through this state will transfer the address in the program counter to the MAR. During the T_2 state, C_P is active and the program counter is incremented on the positive clock edge. During the T_3 state, $\overline{CE}$ and $\overline{L}_I$ are active; when the positive clock edge occurs, the addressed RAM word is transferred to the instruction register. The LDA execution starts with the T_4 state, where $\overline{L}_M$ and $\overline{E}_I$ are active; on the positive clock edge the address field in the instruction register is transferred to the MAR. During the T_5 state, $\overline{CE}$ and $\overline{L}_A$ are active; this means that the addressed RAM data word is transferred to the accumulator on the positive clock edge. As you know, the T_6 state of the LDA routine is a nop.

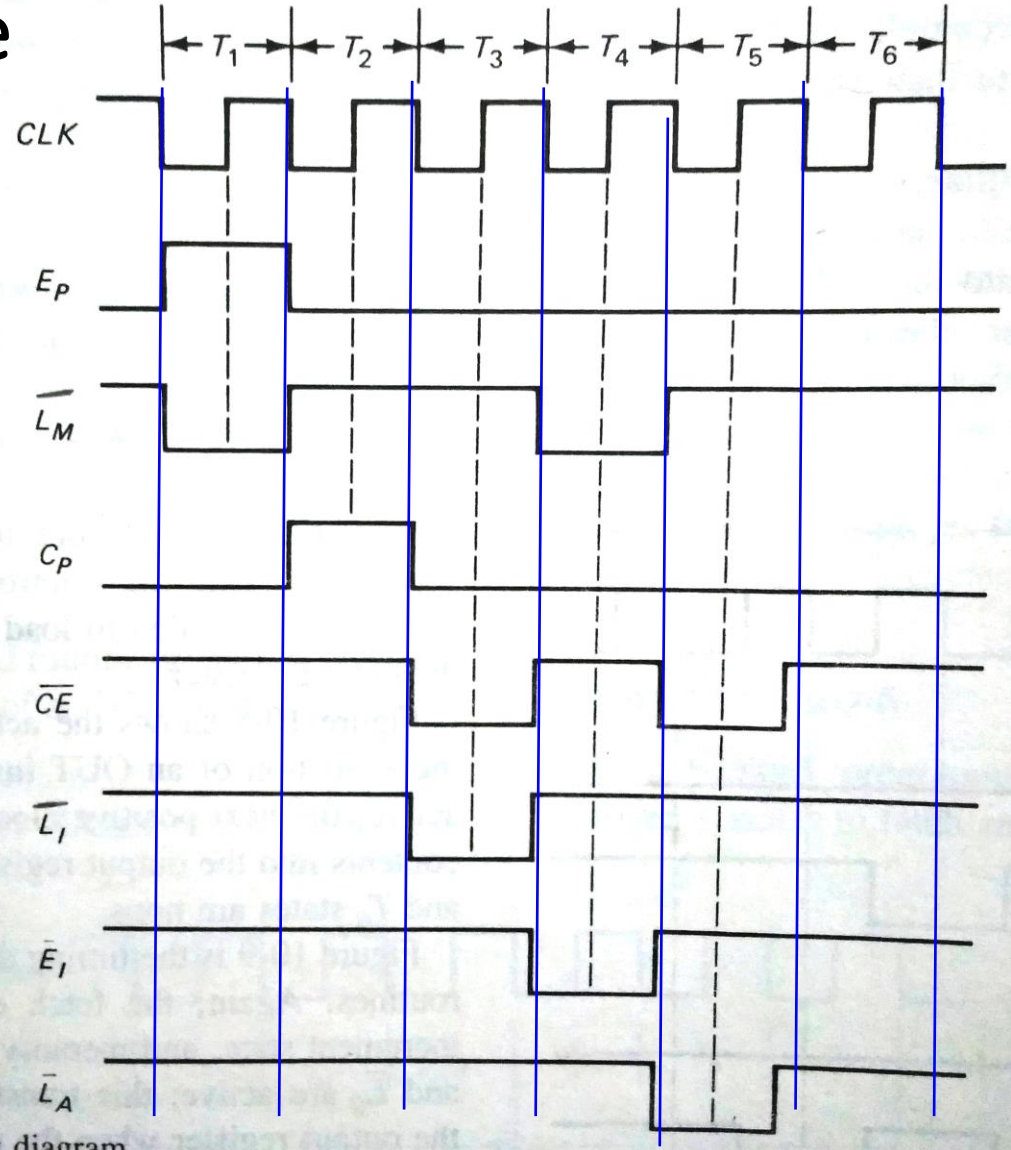


Fig. 10-5 Fetch and LDA timing diagram.

ADD routine (a) T4 state (b) T5 state (c) T6 state

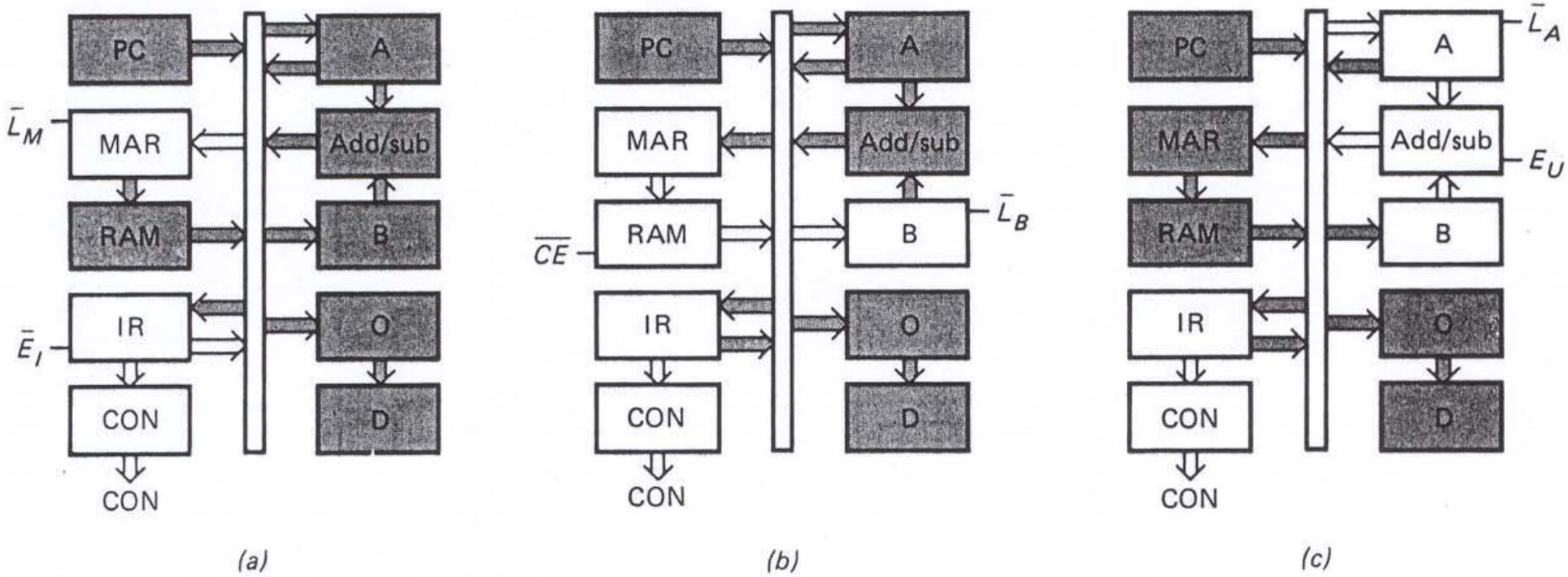


Fig. 10-6 ADD and SUB routines: (*a*) T_4 state; (*b*) T_5 state; (*c*) T_6 state.

ADD routine (a) T4 state

Let's assume that at the end of the fetch cycle the instruction register contains ADD BH:

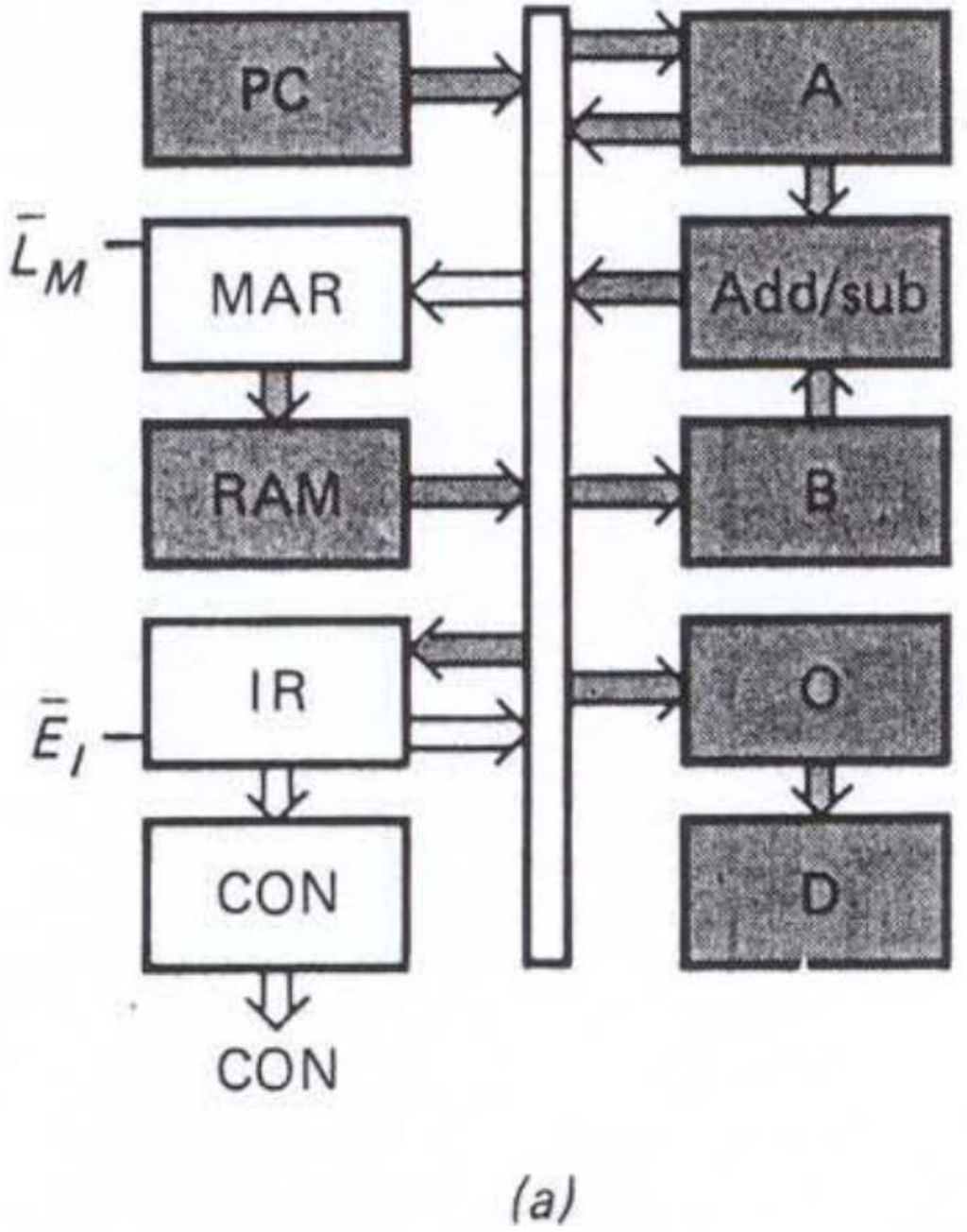
IR = 0001 1011

T4 state: The instruction field 0001 goes to the controller-sequencer, where it is decoded. The address field 1011 is loaded into the MAR. At this state, $\overline{E_I}$ and $\overline{L_M}$ are active.

The controller sequence word is:

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

0 0 0 1 1 0 1 0 0 0 1 1



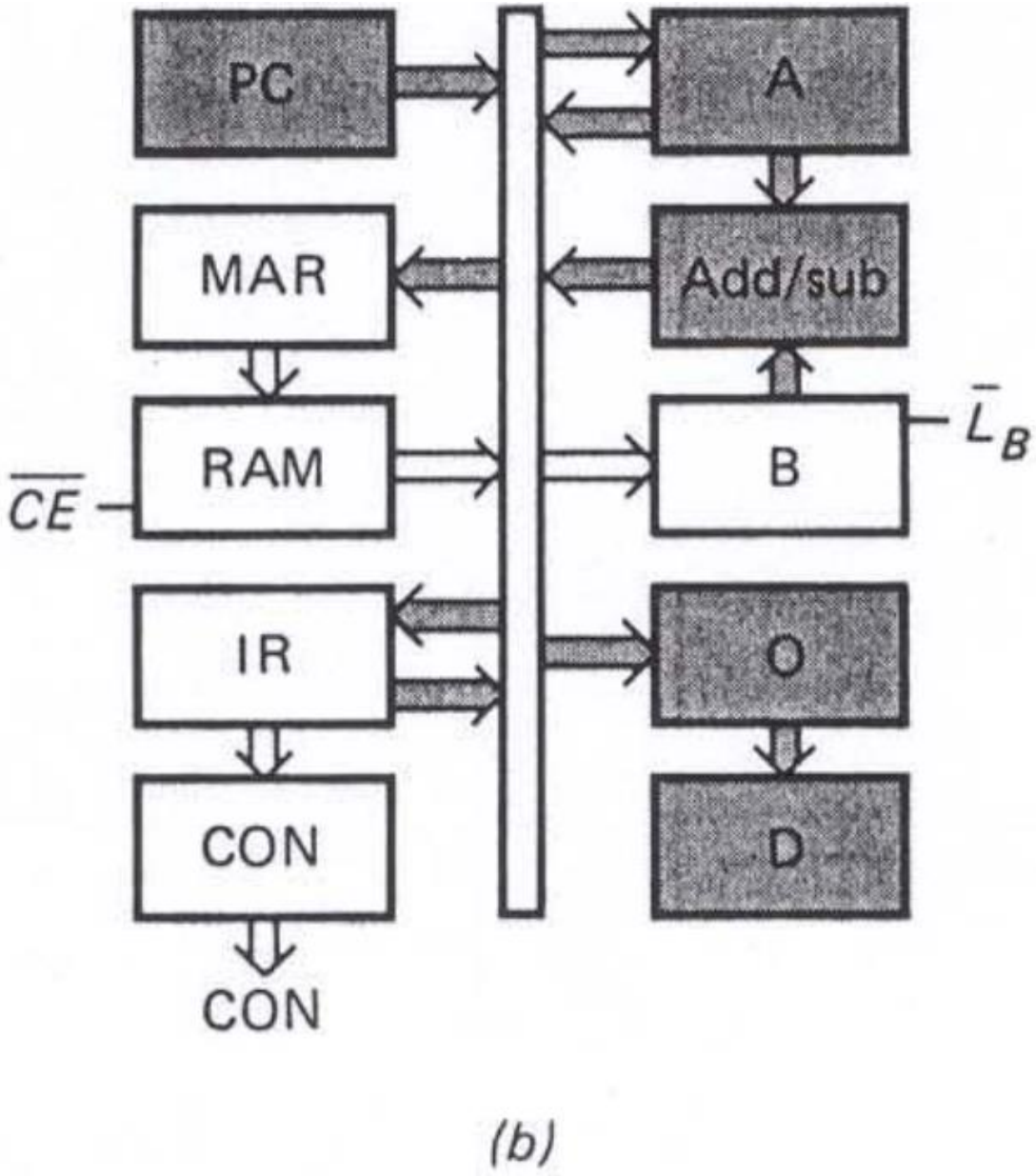
ADD routine (b) T5 state

T5 state: During T5 state, $\overline{CE}$ and $\overline{L_B}$ go low (active). The addressed data word in the RAM will loaded into the Register B.

The controller sequence word is:

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

0 0 1 0 1 1 1 0 0 0 0 1



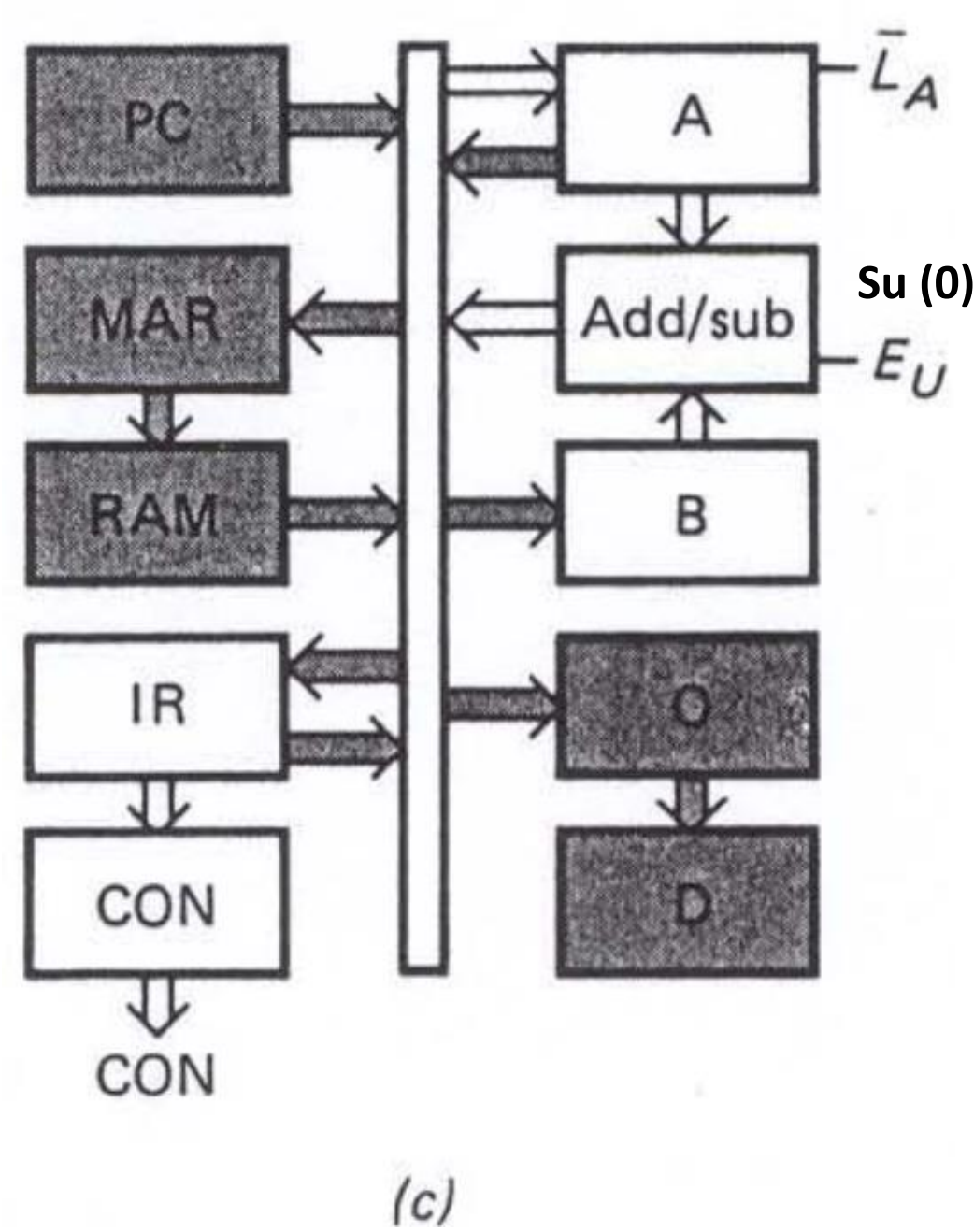
ADD routine (c) T6 state

T6 state: During T6 state, E_U and $\overline{L_A}$ go low (active). The adder/subtractor is activated.

The controller sequence word is:

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

0 0 1 1 1 1 0 0 0 1 1 1



Timing diagram showing ADD routine (Fetch cycle and Execute cycle)

Figure 10-7 shows the timing diagram for the fetch and ADD routines. The fetch routine is the same as before: the T_1 state loads the PC address into the MAR; the T_2 state increments the program counter; the T_3 state sends the addressed instruction to the instruction register.

During the T_4 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 the MAR. During the T_5 state, $\overline{CE}$ and $\overline{L}_B$ are active; therefore, the addressed RAM word is loaded into the B register midway through the state. During the T_6 state, E_U and $\overline{L}_A$ are active; when the positive clock edge hits, the sum out of the adder-subtracter is stored in the accumulator.

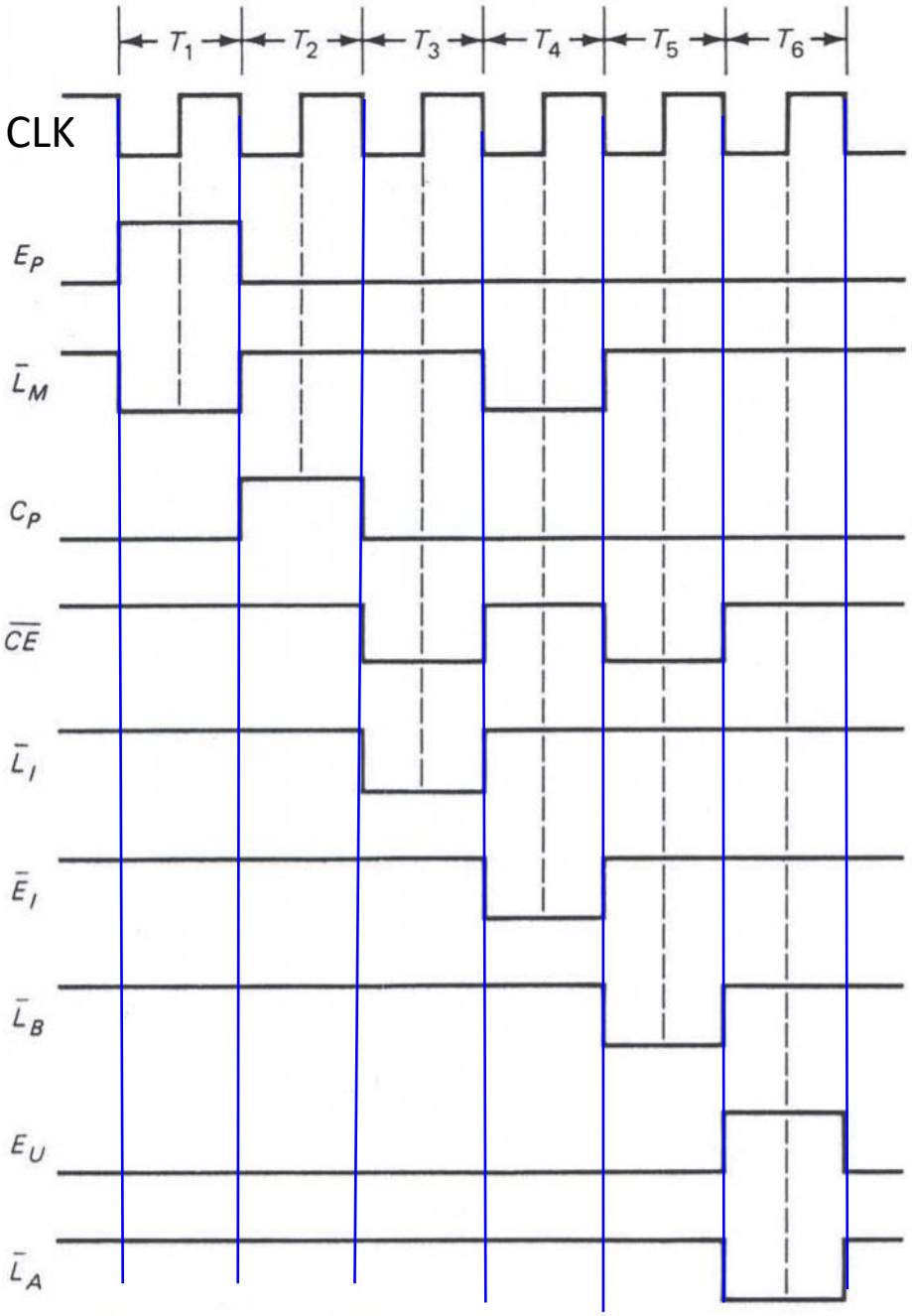


Fig. 10-7 Fetch and ADD timing diagram.

SUB routine (a) T4 state (b) T5 state (c) T6 state

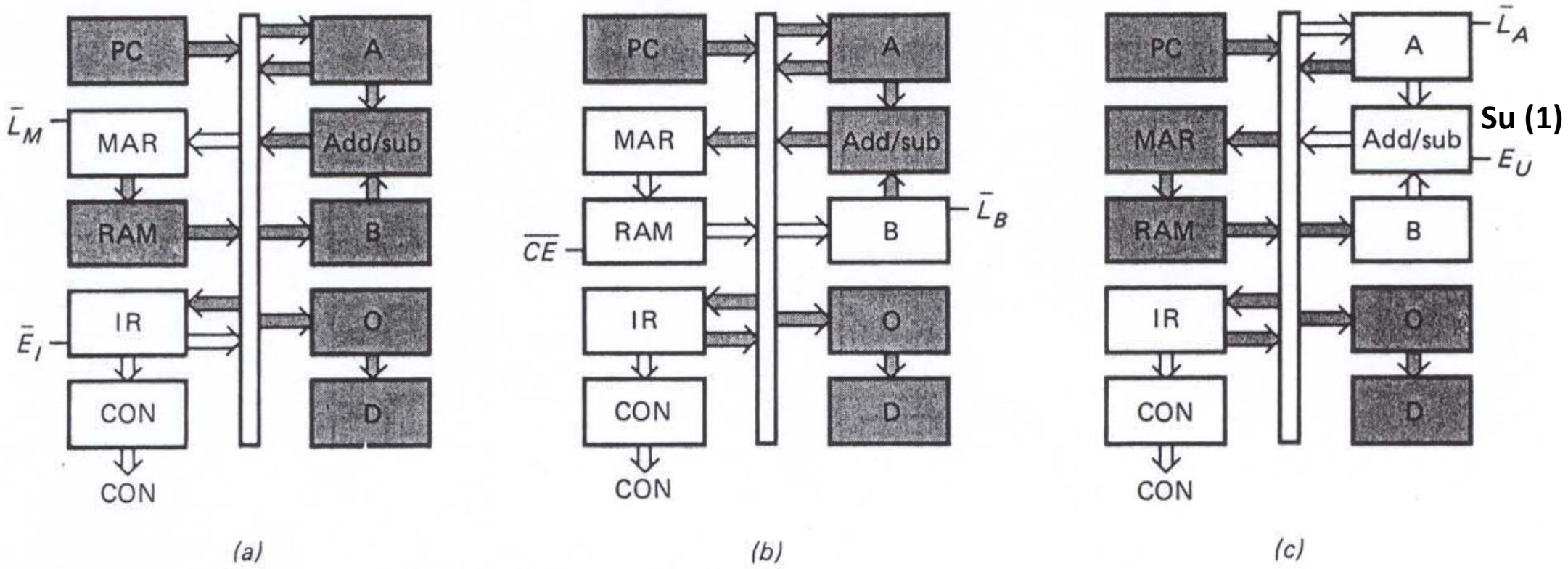


Fig. 10-6 ADD and SUB routines: (*a*) T_4 state; (*b*) T_5 state; (*c*) T_6 state.

The SUB routine is similar to the ADD routine. Figure 10-6*a* and *b* show the active parts of SAP-1 during the T_4 and T_5 states. During the T_6 state, a high S_U is sent to the adder-subtracter of Fig. 10-6*c*. The timing diagram is almost identical to Fig. 10-7. Visualize S_U low during the T_1 to T_5 states and S_U high during the T_6 state.

OUT routine (a) T4 state

Suppose the instruction register contains the OUT instruction at the end of a fetch cycle. Then

$$\mathbf{IR} = 1110\ XXXX$$

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.

Figure 10-8 shows the active sections of SAP-1 during the execution of an OUT instruction. Since E_A and $\overline{L}_O$ are active, the next positive clock edge loads the accumulator contents into the output register during the T_4 state. The T_5 and T_6 states are nops.

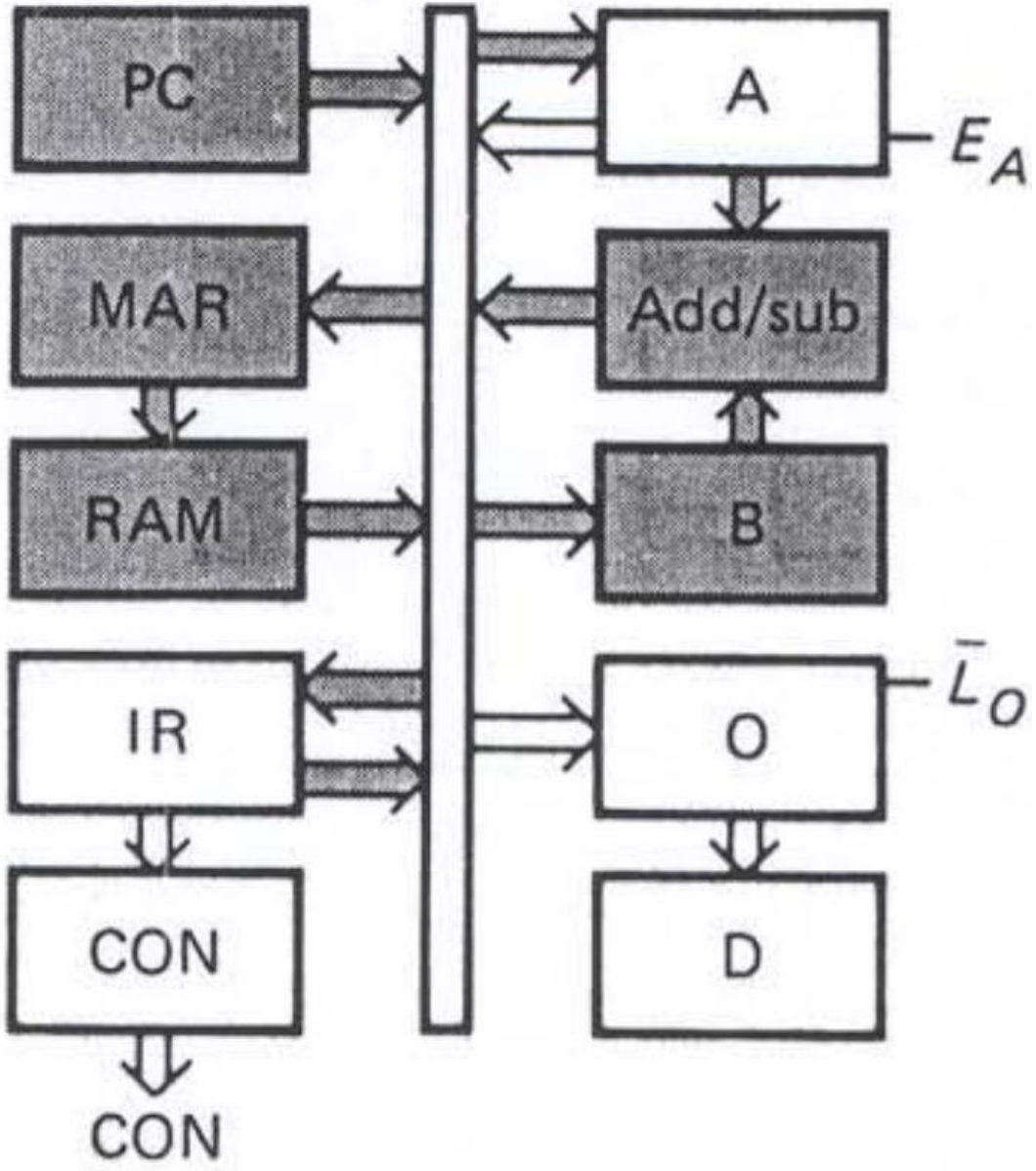


Fig. 10-8 T_4 state of OUT instruction.

OUT routine (a) T4 state

Figure 10-9 is the timing diagram for the fetch and OUT routines. Again, the fetch cycle is same: address state, increment state, and memory state. During the T_4 state, E_A and $\overline{L}_O$ are active; this transfers the accumulator word to the output register when the positive clock edge occurs.

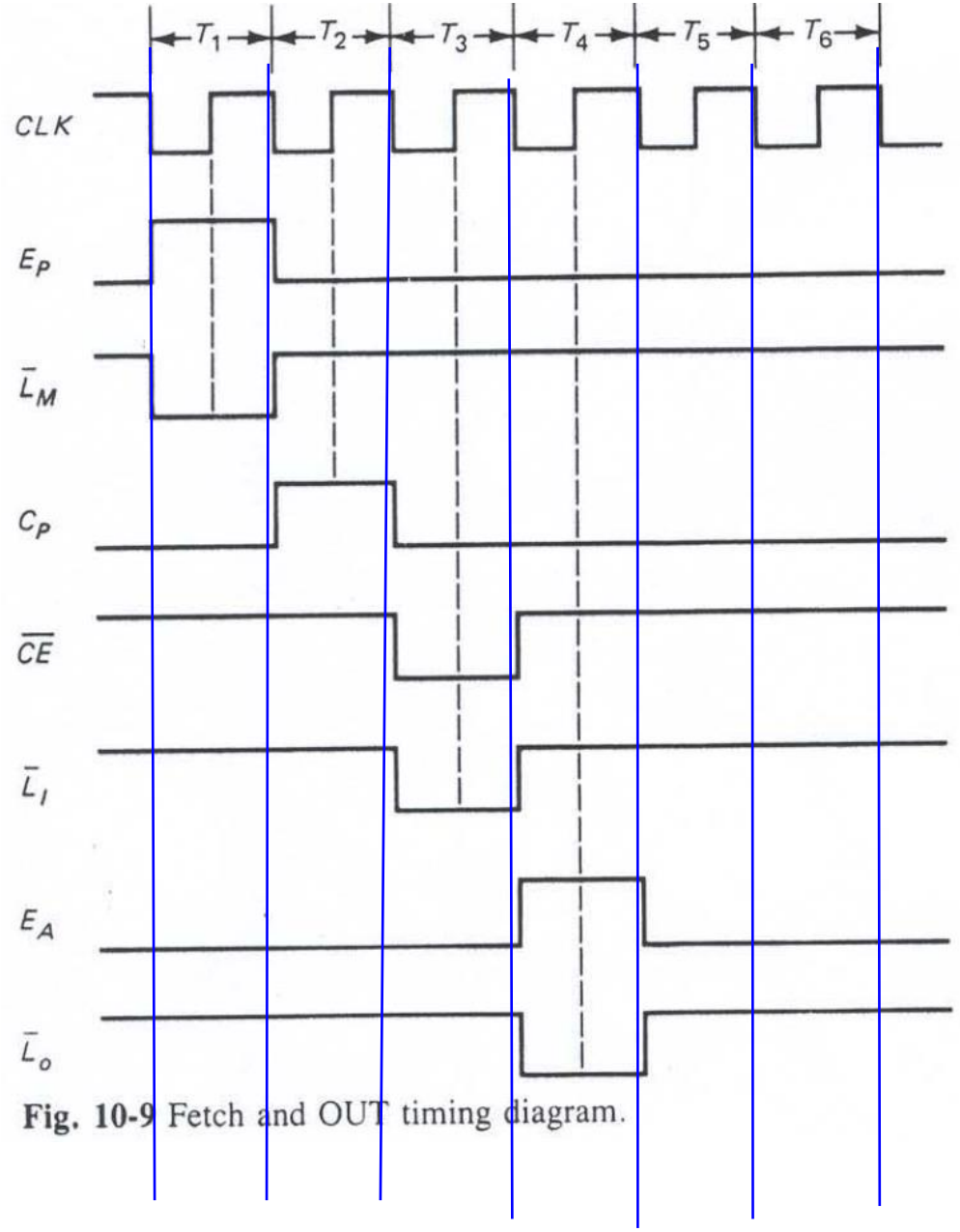


Fig. 10-9 Fetch and OUT timing diagram.

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

$$\mathbf{IR} = 1111\ XXXX$$

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