

## COA

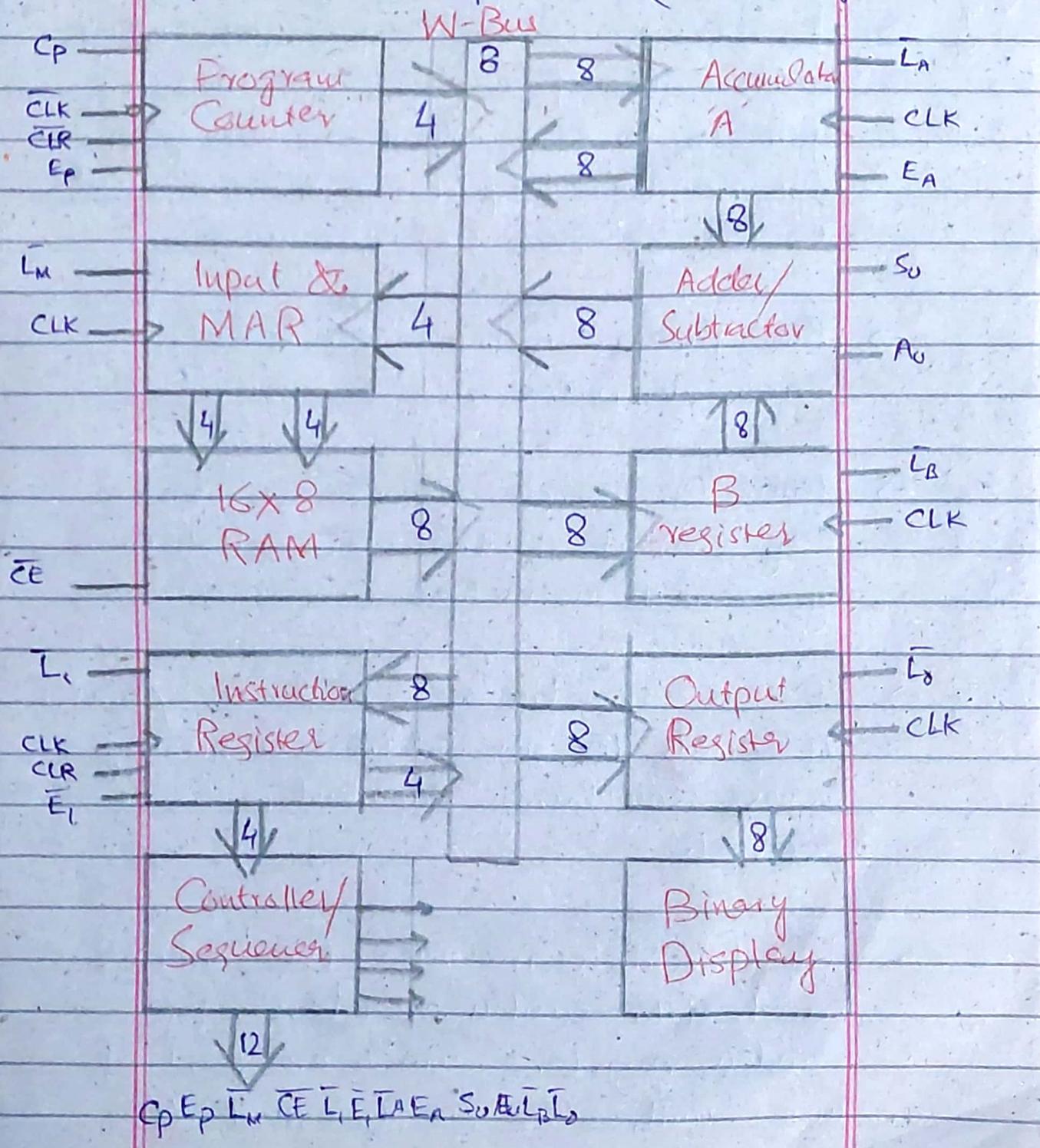
## Notes

Date-10/01/2024

Chapter 10: SAP - 1

## 10.1 Architecture:

→ Below figure shows the structure of Simple-as-possible (SAP-1) Computer.



②

→ All register output to the W Bus are three state; this allow orderly transfer of data. All other register outputs are two state.

→ Program Counter <sup>(PC)</sup> is part of control unit, counts from 0000 to 1111. Its job is to send to the memory the address of the next instruction to be fetched and executed.

→ The PC is reset to 0000 before each computer run.

→ When PC sends address to memory (say 0000) it is then incremented to get next instruction (say 0001).

→ PC is sometimes called a pointer; it points to an address in memory where something important is being stored.

→ Input & MAR includes the address and data switch registers. The switch registers which are part of input unit, allow us to send 4

address bits and 8 data bits to the RAM.

→ MAR is part of SAP-1 memory. The address in PC is latched into the MAR, which applies the <sup>4-bit</sup> address on RAM and a read operation is performed.

→ Input: 4 bits

Output: 8 bits

CLK: Clock cycles

$L_M$ :

-O: (MAR)  $\leftarrow$  bus W.

→ The RAM is a  $16 \times 8$  static TTL RAM. The RAM receives 4-bit addresses from the MAR and a read operation is performed, thus the instruction or data word stored in RAM is placed on W bus for use in some other part of a computer.

→ It stores both instruction and data.

→ Input: 8-bits from MAR

Output: 8-bits to bus W

$C_E$ :

-O: (RAM)  $\leftrightarrow$  bus W.

(4)

$$*\text{ nibble} = \frac{1}{2} \text{ byte} = 4\text{-bit}$$

- The Instruction Register (IR) is part of CU. It fetches instruction from memory.
- The content of IR are split into two nibbles.
  - The upper nibble is a two-state output that goes directly to the block Labeled "Control-sequencer".
  - The lower nibble is a three-state output that is read onto the W bus when needed.
- - Input: 8 bits from RAM
  - Output:
    - 4 bit to bus
    - 4 bit to CU
  - $L_I$ :
    - O:  $(IR) \leftarrow 8\text{-bit input}$
  - $E_I$ 
    - O:  $(IR) \xrightarrow{4\text{ bit}} \text{bus } W$ .
- The Controller / Sequencer control all execution flow.
- Before each computer run a CLR signal is sent to PC and a CLR to instruction Register (IR)

for resetting PC to 0000 and ~~wipe~~ write test instruction in IR.

- A clock signal CLK is set to all buffer registers for synchronizing the operations.
- The 12-bit of format GON =  $C_p E_p \bar{L}_M \bar{C}E \bar{L}_E \bar{L}_A E_A S_U E_U \bar{L}_B \bar{L}_S$   
 comes out of CU which form a word and control the rest of computer.  
 The control word is carried in Control bus.  
 The word determine how the register will react to the next positive clock edge.
- Input: 4-bits from IR  
 Output: 12-bits to distribute to all components. (micro-instructions).
- The Accumulator (A) is a buffer register that stores intermediate answers during a computer run.
- The two-state output goes directly to adder-subtractor, the three-state output goes to the W-bus.

(6)

- Input: 8-bit from bus-W  
Output: 8-bit to bus W and  
8-bit to ALU.  
 $L_A$ : -0: Acc  $\leftarrow$  bus W  
 $E_A$ : -1: Acc  $\rightarrow$  bus W.

→ SAP-1 uses a 2's complement Adder-Subtractor (ALU)

- When  $S_4$  is low ALU produces sum (i.e.  $S = A + B$ ) and when  $S_4$  is high ALU produces difference (i.e.  $S = A + B'$ ).
- The ALU is asynchronous (undelayed) meaning that the content of ALU changes as soon as inputs change.
- When  $E_U$  is high its content appears on W bus.
- Input: 8-bit from Acc and  
8-bit from B register  
Output: 8-bit to W bus.  
 $S_4$ : -0: add  
-1: subtract  
 $E_U$ : -1: ALU  $\rightarrow$  bus W.

- The B register is another buffer register used in arithmetic operations.
- A low  $L_B'$  and positive clock edge loads the word on the W bus into the B register.
- The two-state output of B-register drives the adder-subtractor, supplying the number to be added/subtracted from the contents of accumulator.
- Input: 8-bit from bus W.  
Output: 8-bit to ALU  
 $L_B' = 0 : B \leftarrow \text{bus } W$
- The Output register/part is used to transfer the processed data from accumulator to output world through this register.
- When  $E_A$  is high and  $E_o$  is low, the next positive clock edge loads the accumulator word into the output register.

8

- In microcomputers the output ports are interface circuits that drive peripheral devices.
- Input: 8-bit from ACC via bus W.  
Output: 8-bit to output device.  
 $L_0' : -0$ ; Output register  $\leftarrow$  bus W.
- The Binary Display is a row of 8 LEDs. This Display shows us the content of output port.
- In short; SAP-1 CU consists of PC, IR and controller/sequencer. SAP-1 ALU consists of ACC, adder-subtractor and B register. The SAP-1 memory has the MAR, and a  $16 \times 8$  RAM. The I/O unit includes programming switches, the output port and the binary display.

## 10.2 Instruction Set:

- Instruction Set is the basic operations that a computer can perform.

## SAP-1 Instruction Set.

→ #	Mnemonic	Operand Num	Opcode	Operation
1	LDA	1	0000	Load memory data to acc.
2	ADD $acc = acc + B$	1	0001	Add acc. with memory data
3	SUB $acc = acc - B$	1	0010	Sub acc. with memory data
4	OUT	0	1110	Move out the acc. data
5	HLT	0	1111	Stop program

- LDA stands for "Load the accumulator".
- LDA instruction include hexadecimal address of data to be loaded (eg: LDA 8H).
- ADD is another SAP-1 instruction.
- ADD instruction include the ~~instruction~~ <sup>address</sup> of the word to be added to accumulator content.  
eg ADD 9H means "add content of memory location 9H to the accumulator contents".
- During execution the B register is loaded with the content of address (9H) and then adder-subtractor

(10)

$$acc = acc + B$$

form the sum and the sum is stored to accumulator

- A complete **SUB** instruction includes the address of the word to be subtracted.

eg: SUB CH means "subtract the content of memory location C1t from content of accumulator".

- During execution, B is loaded with content of memory address (RJ); then adder-subtractor form the difference and store it in acc.

- The instruction **OUT** tells the SAP-1 computer to transfer the accumulator content to the output port

- OUT is complete by itself; i.e., we don't have to include an address when using OUT.

- **HLT** stands for "halt", which tells the computer to stop processing data.

- We must use HLT at the end of every SAP-1 program.
- HLT is complete by itself, we don't have to include a RAM word when using HLT.
- The instructions which uses data stored in the memory are called the memory-reference instructions (ie ADD, SUB, LDA).
- HLT and OUT are not memory-reference instructions.
- Abbreviated instructions like above are called mnemonics (memory aids).

### 10.3 Programming SAP-1 80

- The page 9 table shows the operation code (opcode) which tell the computer which operation to perform.
- When programming the data switches with an instruction, the op code

goes into the upper nibble and the operand into the lower nibble.

i.e.  $\overbrace{\text{LDA FH}}^{\text{opcode}} = \underbrace{0000}_{\text{operation}} \underbrace{1111}_{\text{operand}}$   
 $\overbrace{\text{ADD EH}}^{\text{opcode}} = \underbrace{0001}_{\text{operation}} \underbrace{1110}_{\text{operand}}$   
 $\overbrace{\text{HLT}}^{\text{opcode}} = \underbrace{1111}_{\text{operation}} \underbrace{\text{XXXX}}_{\text{operand}}$

→ Any program in machine language is called object program. The original program with mnemonics is called source program.

→ Instruction =  $\underbrace{\text{XXXX}}_{\text{operation}} \underbrace{\text{XXXX}}_{\text{opcode}}$   
 MSBs ←                    LSBs →  
 instruction field.      address field

#### 10.4 Fetch Cycle:

- The control unit is key to computer's automatic operation, which generates control words that fetch and execute each instruction.
- While each instruction is fetched and executed the computer passes through different timing states ( $T$  states),

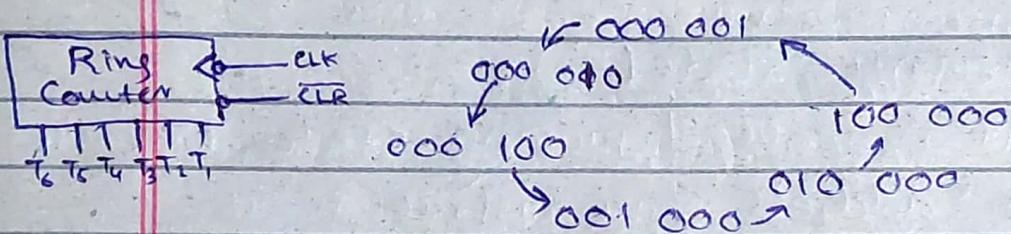
periods during which register contents change.

#### 10.4.1 Ring counter:

- Let we have a ring counter, which has an output of

$$T = T_6 T_5 T_4 T_3 T_2 T_1$$

At beginning of computer run the ring word is  $T: 000\ 001$



- Each ring word represent one  $T$  state.
- They are negative edge triggered.
- Each instruction is fetched and executed during these six  $T$ -states.

#### 10.4.2 Address State ( $T_1$ ):

- The  $T_1$  state is called address state because the address in PC is transferred to MAR during this state.

- The  $E_p$  and  $\bar{E}_m$  are active and all other control bits are inactive.

$$\text{CON} = CPE_p \bar{E}_m \bar{C}E \quad \bar{E}_1 \bar{E}_2 \bar{E}_A E_A \quad S_{UA} \bar{E}_B \bar{E}_D$$

0101 1110 0011

### 10.4.3 Increment State ( $T_2$ )

→ This is called Increment State because PC is incremented in this state

→ Only the Cp bit is active

$$\text{CON} = C_p E_p \bar{L}_M \bar{C_E} \bar{L}_1 \bar{E}_1 \bar{L}_A E_A S_U E_u \bar{L}_B \bar{L}_0$$

$$1 0 1 1 1 1 1 0 0 0 1 1$$

### 10.4.4 Memory State ( $T_3$ ):

→ It is called memory state because the address ram instruction is transferred from memory to IR.

→ Only  $\bar{L}_1$  and  $\bar{C_E}$  control bits are active in this state:

$$\text{CON} = C_p E_p \bar{L}_M \bar{C_E} \bar{L}_1 \bar{E}_1 \bar{L}_A E_A S_U E_u \bar{L}_B \bar{L}_0$$

$$= 0 0 1 0 0 1 1 0 0 0 1 1$$

### 10.4.5 Fetch Cycles:

→ The address, increment and memory states are called the fetch cycle of SAP-1.

## 10.5 Execution Cycles:

→ The next three state ( $T_4, T_5, T_6$ )

\* Nop (pronounced as no op)  
stands for no operation

(15)

are the execution cycle of SAP-1.

#### 10.5.1 LDA Routine:

- During  $T_4$  state, instruction field goes to control-sequence where it is decoded and the address field is loaded into MAR from IR.
- In  $T_4$  state,  $\overline{E}_1$  and  $\overline{E}_M$  are active.
- During  $T_5$  state, address data word in RAM is loaded into Accumulator on next tve clk edge.
- In  $T_5$ ,  $\overline{CE}$  and  $\overline{I}_A$  go low.
- $T_6$  is a no-operation state. (Nop) for LDA routine.

#### 10.5.2 ADD routine:

- During  $T_4$  state, instruction field goes to control-sequence and address field to MAR.
- In  $T_4$  only  $\overline{E}_1$  and  $\overline{E}_M$  are active
- Control bits  $\overline{CE}$  and  $\overline{E}_B$  are active during  $T_5$  state. which

allow addressed RAM word to set up the B register.

- During  $T_6$  state,  $E_U$  and  $E_A$  are active; therefore, the adder-subtractor set up the accumulator.

#### 10.5.3 SUB routine:

- The SUB routine is similar to ADD routine only the difference is that in  $T_6$  state a high  $S_U$  is sent to adder-subtractor.

#### 10.5.4 OUT routine:

- During  $T_4$  state, the instruction field goes to the control-sequencer which sends out the control word needed to load the accumulator content into the output register.
- $E_A$  and  $L_O$  are active control bits.
- $T_5$  and  $T_6$  are nops.

### 10.5.6 HLT.

- HLT doesn't require a control routine because no register are involved in execution of HLT instruction.
- When the IR contains "1111 XXXX" the instruction field signal the controller-sequencer to stop processing data by turning off the clk.

### 10.5.7 Machine and Instruction Cycles:

- SAP-1 has six T states (three fetch and three execute). These 6 states are called a machine cycle.
- The number of T states needed to fetch and execute an instruction is called instruction cycle.
- In SAP-1 instruction cycle == machine cycle.