Bangladesh University of Engineering and Technology

Department of Electrical and Electronics Engineering

Course Number: EEE 304

Course Title: Digital Electronics Laboratory

Experiment Number: 06

Name of the Experiment(s):

Introduction to SAP-1 (Simple as Possible) Computer

Prepared by: Tasmin Khan

Student ID: 1906044

Level:3 Term:2

Task 1

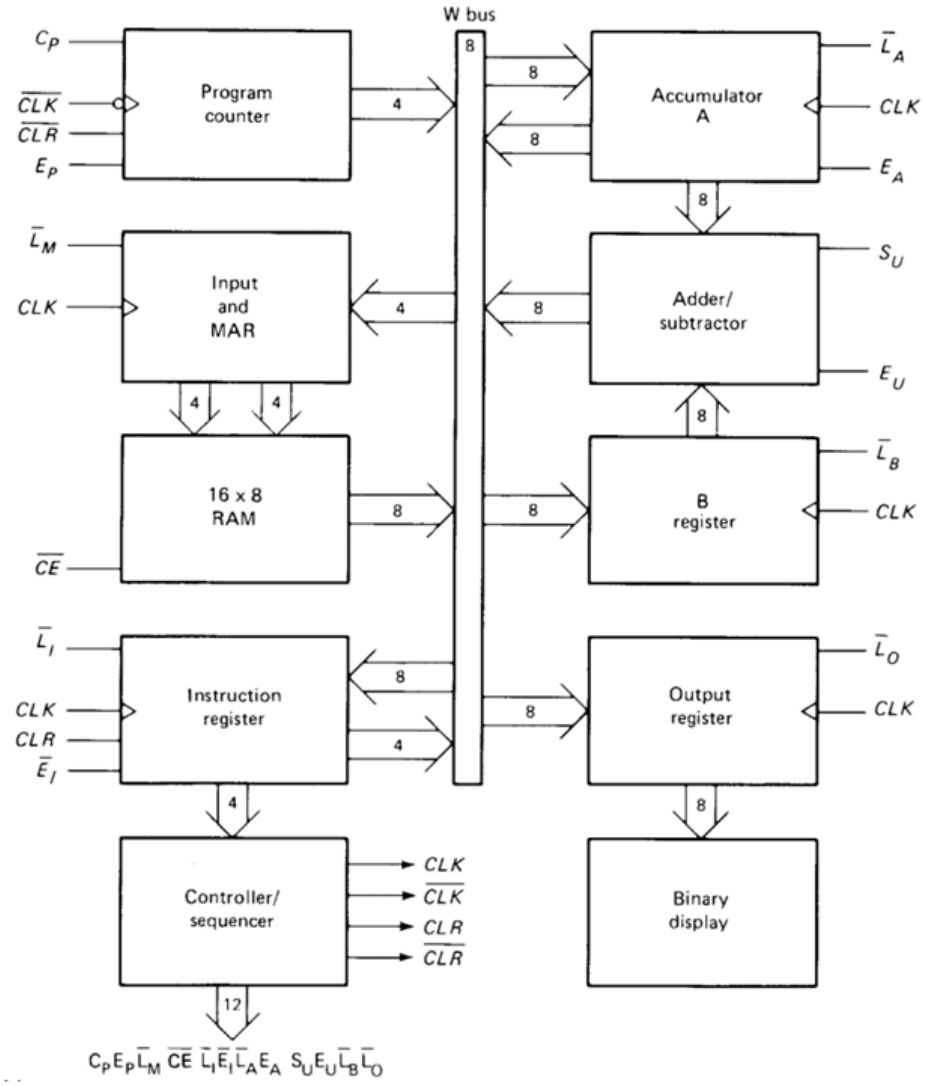


Figure: SAP I Architecture

Question: Disassemble the following instructions, and identify the control signals (12-bit control word) for each of the T4, T5 and T6 state.

1. LDA 9H
2. SUB 10H
3. OUT
4. HLT

Answer: The first three clock cycles are the same irrespective of target operation. Together they are called the Fetch cycle comprising of T1, T2 and T3

Address state T1: In this cycle, address from Program Counter (PC) is transferred to Address Register (MAR) via W bus. Hence EP and LM' are active.

Increment state T2: PC is incremented for the next instruction and so only CP is active.

Memory state T3: Here, the addressed RAM instruction is transferred from memory to the instruction register (IR). Hence CE' and LI' are active.

Now for the last three clock cycles, the control signal depends on the instruction.

1. **LDA 9H**

   T4: Here first 4 bits for LDA goes to controller-sequencer and last 4 bits for address 9H goes to MAR. So, LM' and EI' are active.

   T5: The 8 data bits stored at 9H in RAM are loaded into A register via w bus. So, CE' and LA' are active.

   T6: Since data is already loaded, it's an idle cycle with no active controller sequence.

Controller sequence	CP	EP	LM'	CE'	LI'	EI'	LA'	EA	SU	EU	LB'	LO'
T4	0	0	0	1	1	0	1	0	0	0	1	1
T5	0	0	1	0	1	1	0	0	0	0	1	1
T6	0	0	1	1	1	1	1	0	0	0	1	1

2. **SUB 10H**

T4: First 4 bits for SUB goes to the controller and last 4 bits for address 10H goes to MAR. LM' and EI' are active.

T5: The 8 data bits stored at 10H are loaded into B register via w bus. So CE' and LB' are active. The ALU unit asynchronously subtracts these two numbers and the result is ready to be loaded into the Accumulator in the next cycle.

T6: Result is loaded into A register. LA', SU and EU are active. SU is active to be able to perform subtraction.

Controller sequence	CP	EP	LM'	CE'	LI'	EI'	LA'	EA	SU	EU	LB'	LO'
T4	0	0	0	1	1	0	1	0	0	0	1	1
T5	0	0	1	0	1	1	1	0	0	0	0	1
T6	0	0	1	1	1	1	0	0	1	1	1	1

3. OUT XXX

T4: The data from the accumulator is transferred to the output register for binary display via the w bus. So, EA and LO' are active.
T5 and T6: Idle cycle

Controller sequence	CP	EP	LM'	CE'	LI'	EI'	LA'	EA	SU	EU	LB'	LO'
T4	0	0	1	1	1	1	1	1	0	0	1	0
T5	0	0	1	1	1	1	1	0	0	0	1	1
T6	0	0	1	1	1	1	1	0	0	0	1	1

4. HLT XXX

Halt means to stop the data processing and so the state remains at T3 for all the next clock cycles and never reaches T4, T5 or T6.

Task 2

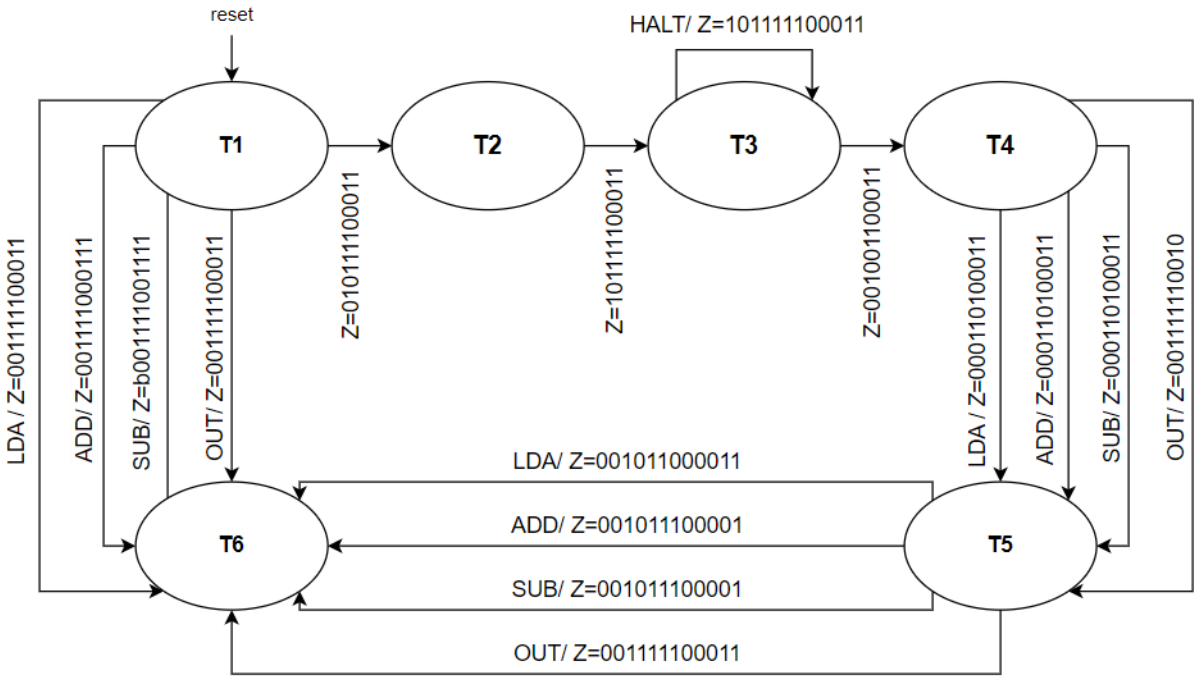
Question: Draw a state diagram that will implement the controller unit FSM for the SAP-1 architecture.

1. States: T1, T2, T3, T4, T5
2. Inputs: Op code
3. Outputs: 12-bit Control Word

Here the opcodes for the operations are:

Mnemonic	Op code
LDA	0000
ADD	0001
SUB	0010
OUT	1110
HLT	1111

Answer: The state diagram for SAP1 operation is as follows:



Here HALT has the highest priority in the step from T3 to T4. Also the output signal Z is the controller sequence output

Report Task

Question: Write a Verilog code for the controller/ sequencer unit and generate waveforms. Your module will have a clock and Reset input, and a 4-bit input opcode. It must have all the outputs shown in figure 4.1. In your vector wave file, set all possible opcodes and show that the appropriate control signals are being generated in each cycle. You must take account of the fact that each instruction takes 6 clock cycles.

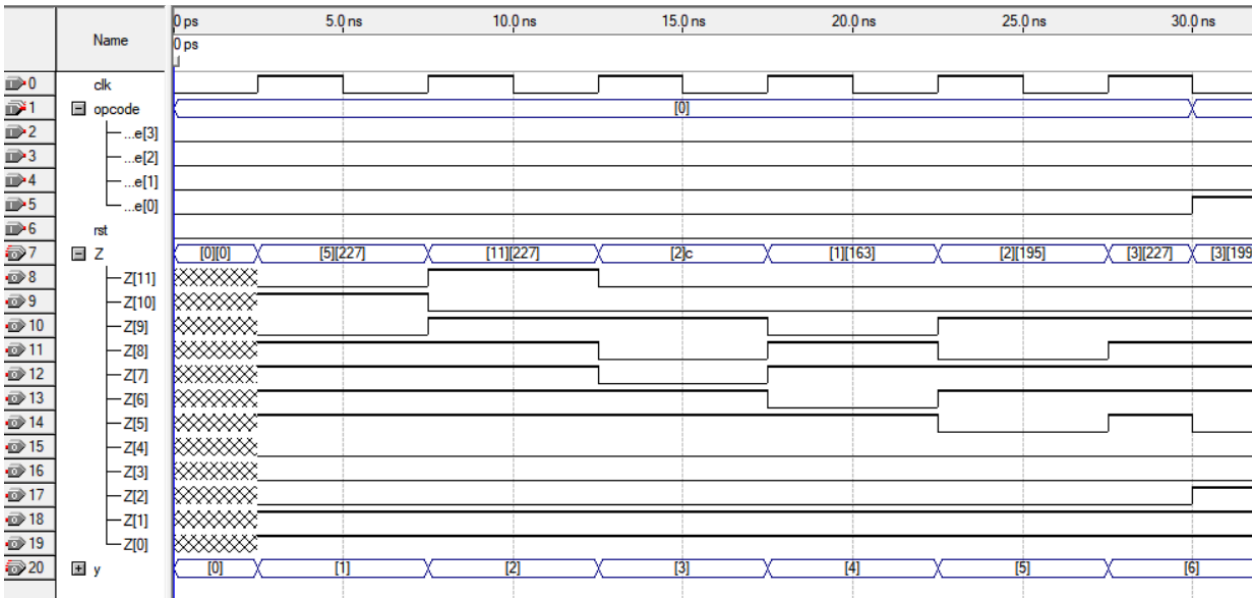
Verilog code:

```
module Sap(opcode,Z,clk,y,rst);
    input [3:0]opcode;
    output reg [3:0]y;
    output reg [11:0]Z;
    reg [3:0]Y;
    input clk, rst;
    parameter[3:0] T1=1,T2=2,T3=3,T4=4,T5=5,T6=6;
    parameter LDA=4'b0000, ADD=4'b0001, SUB=4'b0010, OUT=4'b1110, HLT=4'b1111;
    always @(opcode,y)
    begin
       case(y)
           T1: begin
               Z<=12'b010111100011;
               Y<=T2;
               end
           T2: begin
               Z<=12'b101111100011;
               Y<=T3;
               end
           T3: begin
               Z<=12'b001001100011;
               if (opcode==HLT) Y=T3;
               else Y=T4;
               end

           T4: begin
               Y=T5;
               if(opcode==LDA)
               Z=12'b000110100011;
               else if(opcode==ADD)
               Z=12'b000110100011;
               else if(opcode==SUB)
               Z=12'b000110100011;
               else if(opcode==OUT)
               Z=12'b001111110010;
               end

           T5: begin
               Y=T6;
               if(opcode==LDA)
               Z=12'b001011000011;
               else if(opcode==ADD)
               Z=12'b001011100001;
               else if(opcode==SUB)
               Z=12'b001011100001;
               else if(opcode==OUT)
               Z=12'b001111100011;
               end
```

```
            T6: begin
                Y=T1;
                if(opcode==LDA)
                Z=12'b001111100011;
                else if(opcode==ADD)
                Z=12'b001111000111;
                else if(opcode==SUB)
                Z=12'b001111001111;
                else if(opcode==OUT)
                Z=12'b001111100011;
                end

              default: Y=T1;
              endcase
          end

          always @(posedge clk, posedge rst)
             if(rst) y<=T1;
             else    y<=Y;

endmodule
```

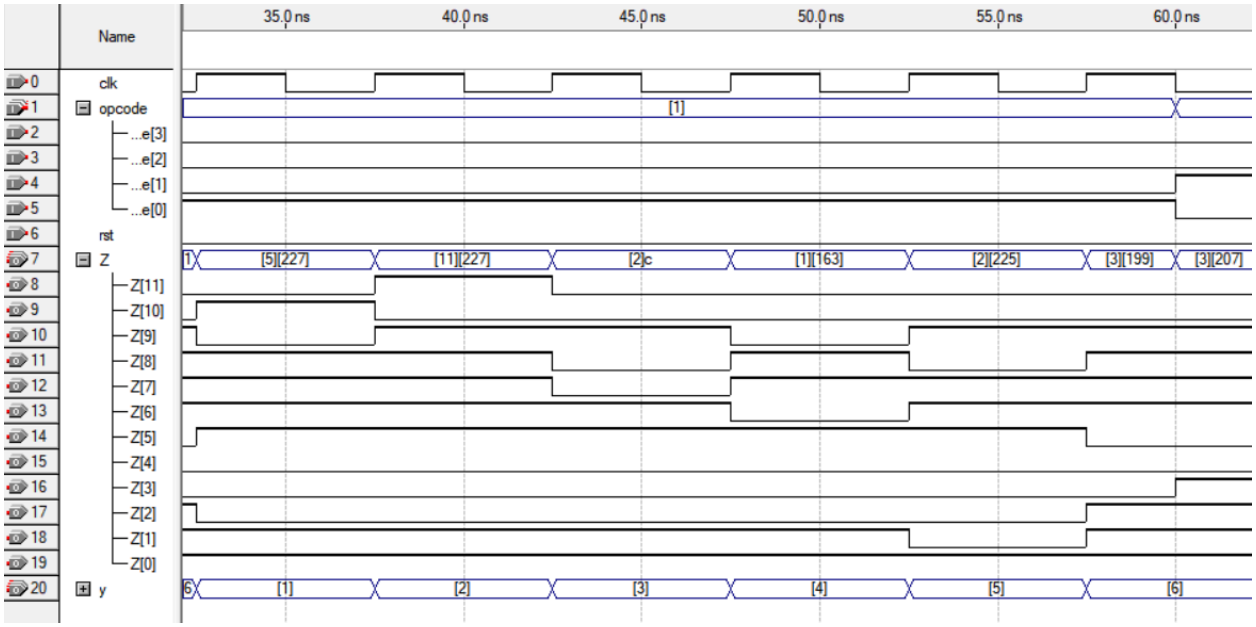
Waveform: (y=current state)

LDA:

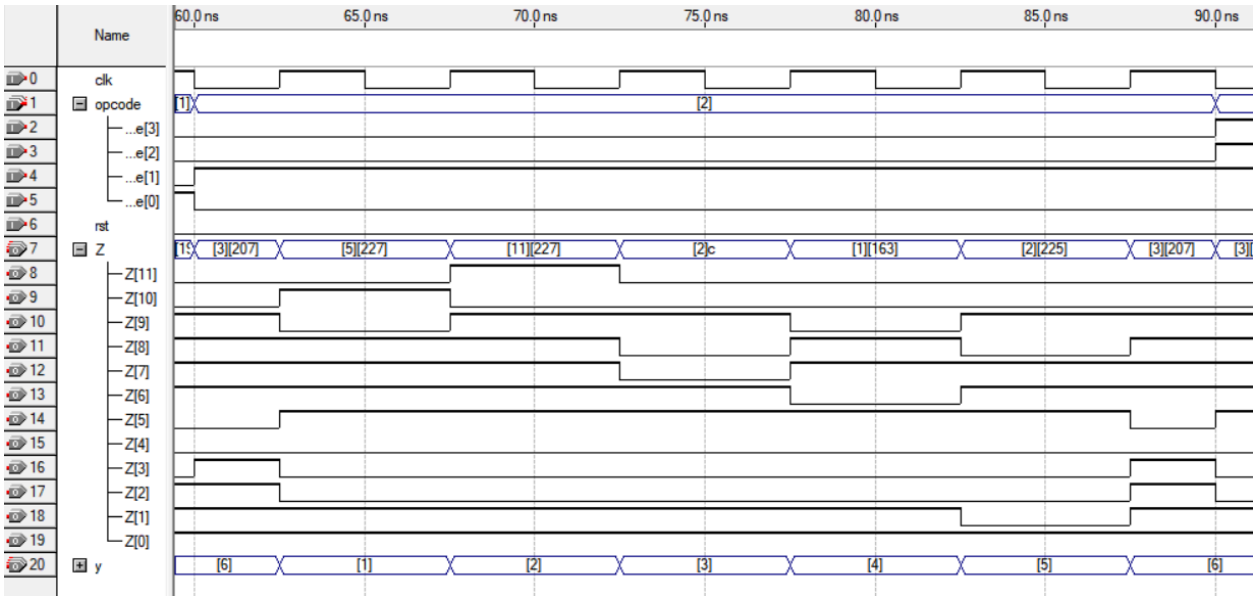


Here, the output sequence is undefined before the first clock posedge. We could have avoided this by starting the system with a reset. WIth the first clock, T1 state is activated and with each clock edge, states change and so does the controller output sequence .

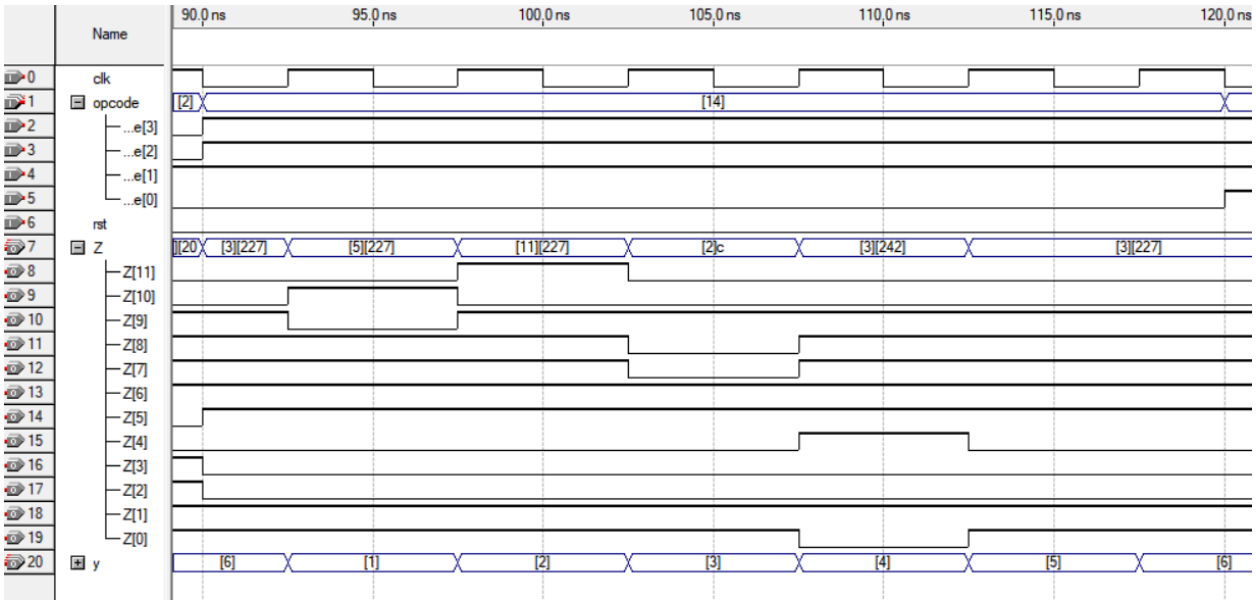
ADD:



SUB

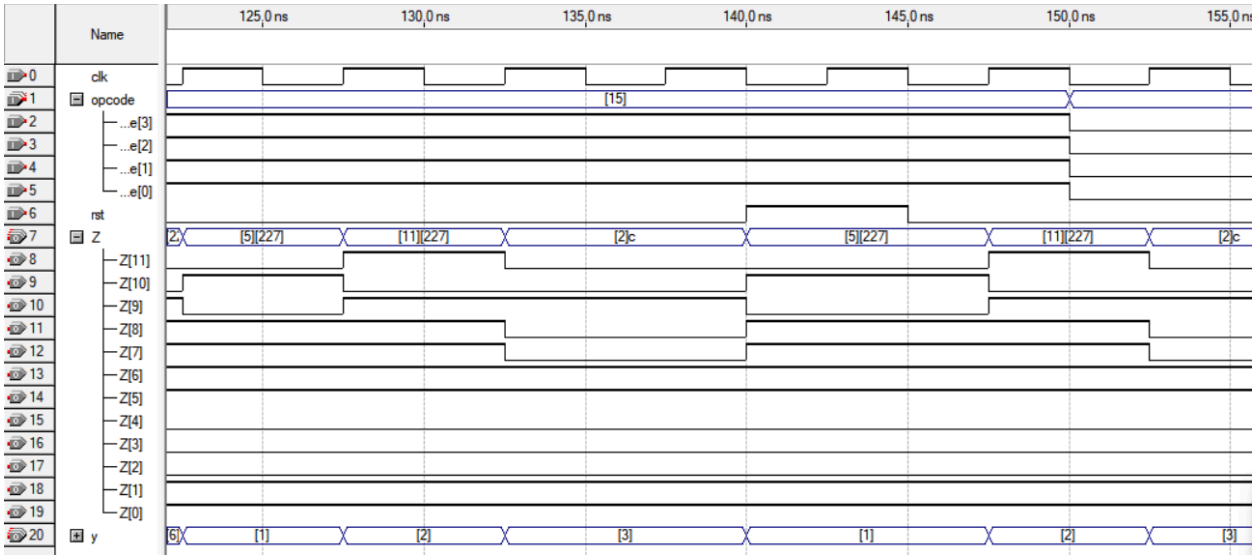


OUT:



Here, output is done by the T4 state and T5 and T6 states are idle. Hence, output sequence Z does not change for the last two clock pulses and all the control signal remains inactive.

HLT:



Here, Halt Opcode freezes the state at T3 and hence there is no change at 4th clock pulse posedge. It shows the same output as the 3rd clock pulse posedge. We activated reset afterwards and since reset was set to be asynchronous, the state goes immediately to T1. Reset is high for the 5th clock pulse too and so no system remains in T1 state. Afterwards as reset goes to zero, the cycle continues and state goes to T2 and then T3 in the consecutive clock pulses.The output sequence for T1, T2, and T3 are the same for all the opcodes as evident from the hex codes of Z from the waveform.