Roll: 1607045
1607023

Objectives :-

The purposes of this project are given below :-

(i) To learn about Sap-1 computer.

(ii) To learn about different components of Sap-1 computer.

(iii) To design a Sap-1 computer using logisim.

Introduction :-

The simple-As-possible (SAP)-1 computer is a very basic model of a micro-processor explained by Albert Paul Malvino. The SAP-1 design contains the basic necessities for a functional Micro-processor. Its primary purpose is to

develop basic understanding of how a microprocessor works, interacts with memory and other parts of the system like input and output. The instruction set is very limited and simple. SAP-1 is the first state in the evaluation towards modern computers.

Fig-1 shows the architecture of SAP-1, a bus organised computer. All registers outputs to the W bus are three-state, this allows orderly transfer of data. All other register outputs are two state, these outputs continuously drive the boxes they are connected to.

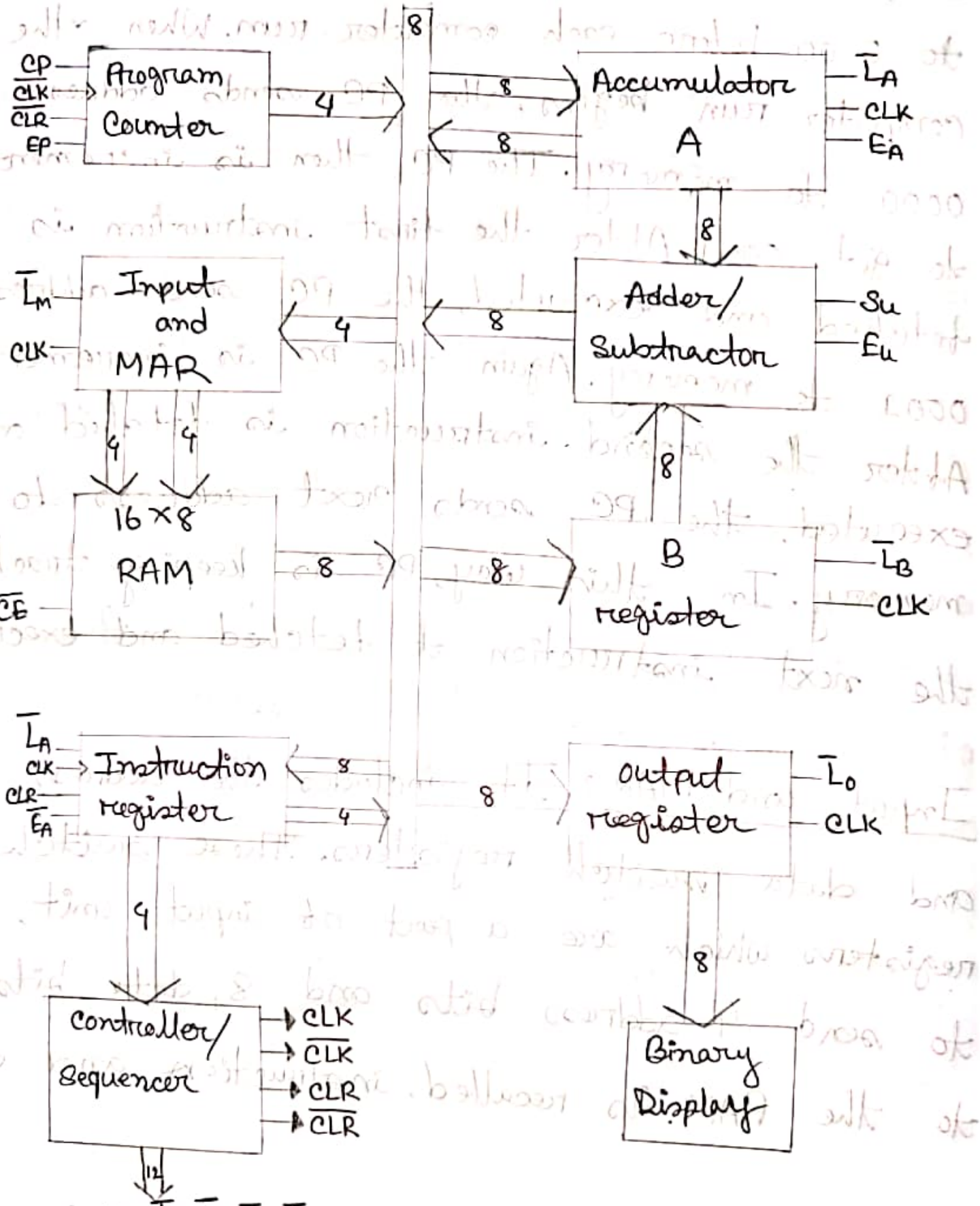


Fig 1 :- SAP-1 architecture

Program Counter :- The program counter is reset to 0000 before each computer run. When the computer run begins, the PC sends address 0000 to memory. The PC then is incremented to get 0001. After the first instruction is fetched and executed, the PC sends address 0001 to memory. Again the PC is incremented. After the second instruction is fetched and executed, the PC sends next address to memory. In this way, PC is keeping track of the next instruction to fetched and executed.

Input and MAR :- It includes the address and data switch registers. These switch registers which are a part of input unit, allow to send 4 address bits and 8 data bits to the RAM. As recalled, instructions and data

words are written into RAM before a computer run.

The MAR is part of SAP-1 memory. During a computer run, the address in PC is ~~last~~ latched into MAR. A bit later, the MAR applies this 4-bit address to RAM where a read operation is performed.

The RAM :- The RAM is a 16x8 static TTL RAM. This allows one to store a program and data in the memory before computer run. During a computer run, RAM receives 4 bit address from MAR and a read operation is performed. In this way, the data or instruction word in RAM is placed on the W bus for use in some other part of the computer.

Instruction Register (IR) :- IR is a part of control unit. To fetch an instruction, the computer does a memory operation. This places the contents of addressed memory location on W bus. At the same time, IR is set up for loading on the next positive clock edge. The contents of the IR are split into the nibbles. The upper nibble is a two-state output that goes directly to the block labeled, "Controller-sequencer". The other lower nibble is a three-state output that is read into W bus when needed.

Controller-sequencer :- Before each computer run a $\overline{CLR}$ is sent to PC and CLR signal to the IR. This resets the PC to 0000 and wipes out the last instruction to the IR.

A clock signal CLK is sent to all bubber register, this syncronizes the operation of the computer, ensuring that all register transfer occur on the positive edge of common CLK signal.

The 12 bits that come out of controller from a coord controlling the ~~read~~ rest of the computer. The 12 wires carrying the control word are called the control bus.

The control word has the format of:

$$CON = C_P\ E_P\ \bar{L}_M\ \bar{C}\bar{E}\ \bar{L}_1\ \bar{E}_1\ \bar{L}_A\ E_A\ S_U\ E_U\ \bar{L}_B\ \bar{L}_O$$

This word determines how to registers will react to the next positive clock edge.

Accumulator :- A accumulator (A) is a buffer register that stores intermediate answer during a computer run. It has two outputs. The two-state output goes directly to adder-subtractor. The three-state output goes to the W bus. Therefore, the 8-bit accumulator word continuously drives the adder-subtractor, the same word appears on W bus when E_A is high.

The Adder-Subtractor :- SAP-1 uses a 2's complement adder-subtractor. When Su is low the sum output of adder-subtracter is

$$S = A + B$$

When Su is high, the difference appears:

$$A = A + B'$$

It is asynchronous. When Eu is high, the contents apear on the W bus.

B register :- The B register is another buffer register. It is used in arithmetic operations. A low $\overline{L}_B$ and positive clock edge load the word on W bus into B register. The two-state output of B register drives the adder-subtractor supplying the number to be adder or subtracted from the contents of the accumulator.

Output register :- At the end of computer run, the accumulator contains the answer to the problem being solved. When E_A is high and $\overline{L}_O$ is low, the next positive clock edge loads the accumulator word into the output register. The output register is often called an output port because processed data can leave the computer through this register.

Binary display :- The binary display is a row of eight LEDs. Because each LED connects to one flip-flop of the output port, the binary display shows the contents of the output port.

Instruction Set of SAP-1 :-

The SAP-1 instruction set below:

(1) LDA :- LDA stands for "Load the Accumulator." A computer LDA instruction includes the hexadecimal address of the data to be loaded.

(2) ADD :- A complete ADD instruction includes the address of the word to be added. For instance, ADD 9H means "add the contents of memory location 9H to accumulator, replace the original contents of accumulator.

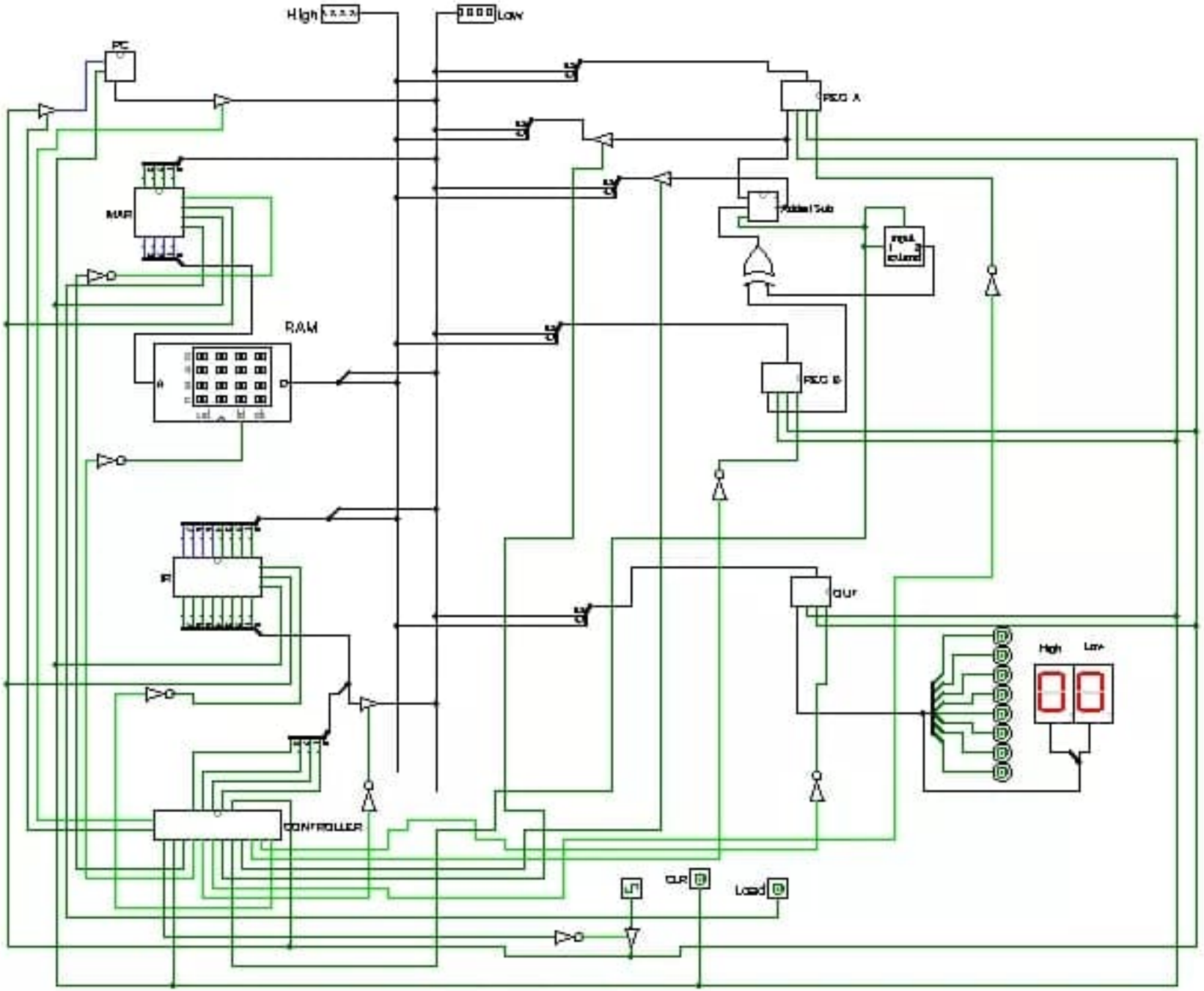
(3) SUB :- A complete SUB instruction includes the address of word to be subtracted. For example, SUB CH means "subtract the contents of memory locations CH from the contents of the accumulator.

(4) OUT :- The ~~Out~~ OUT instruction tells the SAP-1 computer to transfer the accumulator contents to the output port. OUT is complete by itself. you do not have to include an address when using OUT because the instruction doesn't involve data in the memory.

(5) HLT :- HLT stands for halt. It tells the computer to stop processing data. Hlt makes the end of a program. You must use a HLT instruction at the end of every SAP-1 program.

1607023

1607045



Conclusion :-

In this project, we learnt about SAP-1 computer ~~design~~ design process. We also learnt how to design every components of SAP-1 computer in logisim. We implemented SAP-1 computer in logisim using those components. Finally, we ran a program in logisim and checked the output using LEDs and display module.