**International Islamic University, Islamabad**

**Faculty of Engineering and Technology**

**Department of Electrical and Computer Engineering**



**COMPUTER ARCHITECTURE AND ORGANIZATION**

**Project Report: SAP-1**

Project Members: Muhammad Yaseen

Riyan Rafi

Ahmed ali siddique

Roll No: 21-FET/BSCE/F-22

15-FET/BSCE/F-22

36-FET/BSCE/F-22

Submitted To :  **ENGR. RASHID FAREED CHICHTI**

**Introduction**:

The Simple-As-Possible (SAP-1) computer is a foundational learning tool designed to demonstrate basic computer architecture and operations. This project involved implementing the SAP-1 using Logisim, a digital logic simulator, to understand the core principles of computing.

**SAP-1 Features:**

* **Output Device**: An 8-LED display for output.
* **Memory:** 16 bytes of read-only memory (ROM).
* **Instructions:** 5 instructions:
* 3 instructions with one operand: LDA, ADD, STA.
* 2 instructions with implicit operands: OUT, HLT.
* **Architecture:**

**Accumulator**: 8-bit register for arithmetic and logic operations

**Output Register:** 8-bit register for output.

**B Register:** 8-bit register for temporary data storage.

**Memory Address Register (MAR):** 4-bit register for memory addresses.

**Instruction Register (IR):** 8-bit register for the current instruction.

**Bus:** 8-bit wide "W" bus for data transfer.

**Program Counter (PC):** 4-bit counter from 0 to 15.

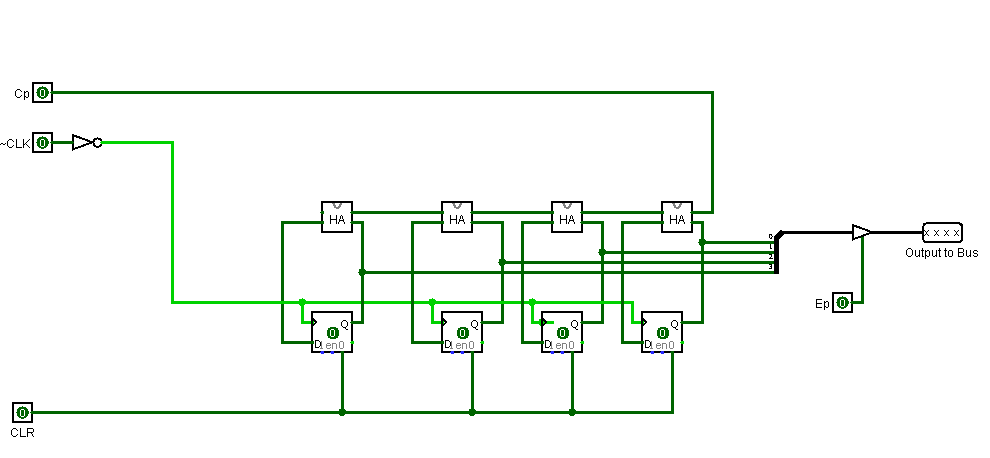
**Controller:** 6-cycle controller with a 12-bit microinstruction word.

**Adder/Subtractor:** 8-bit unit for arithmetic operations.

**SAP-1 Modules:**

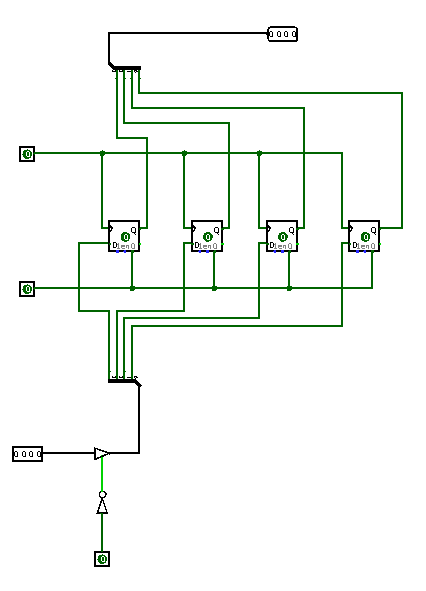
**1. Program Counter**

The program counter’s job is to store and send out the memory address of the next instruction to be fetched and executed. The program counter, which is part of the control unit, counts from 0000 to 1111 as the program is stored at the beginning of the memory with the first instruction at binary address 0000, the second instruction at address 0001, the third at address 0010, and so on. At the start of each computer run, the program counter is reset to 0000. When the computer run starts, the program counter sends out the address 0000 to the memory and is then incremented by 1. After the first instruction is fetched and executed, the program counter sends the next address 0001 to the memory and again, after that, the program counter is incremented. In this way, the program counter keeps track of the next instruction to be fetched and executed.



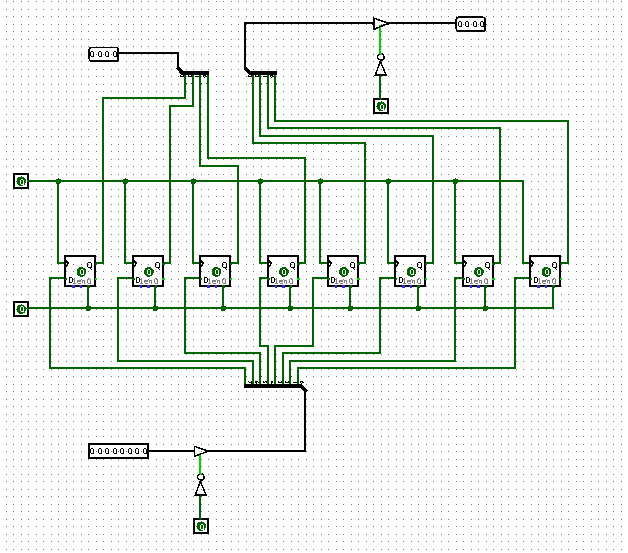
**2. Memory Address Register (MAR)**

The MAR stores the 4-bit address of data or instruction which are placed in memory. When the SAP-1 is running, the 4-bit address is gotten from the Program Counter through the W-bus and then stored. This stored address is sent to the RAM where data or instructions are read from.



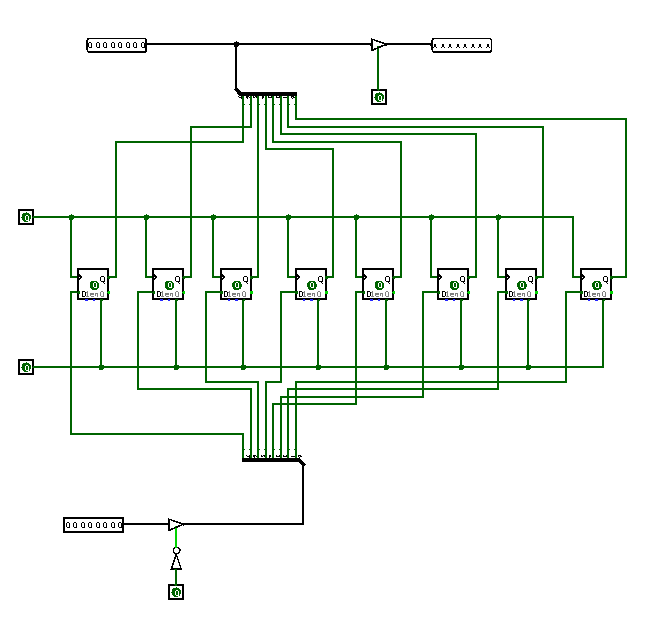
**3. Instruction Register (IR)**

The instruction receives and stores the instruction placed on the bus from the RAM. The content of the instruction register are then split into two nibbles. The upper nibble is a two-state output that goes into the Controller-sequencer while the lower nibble is a three-state output that is read from the bus when needed.



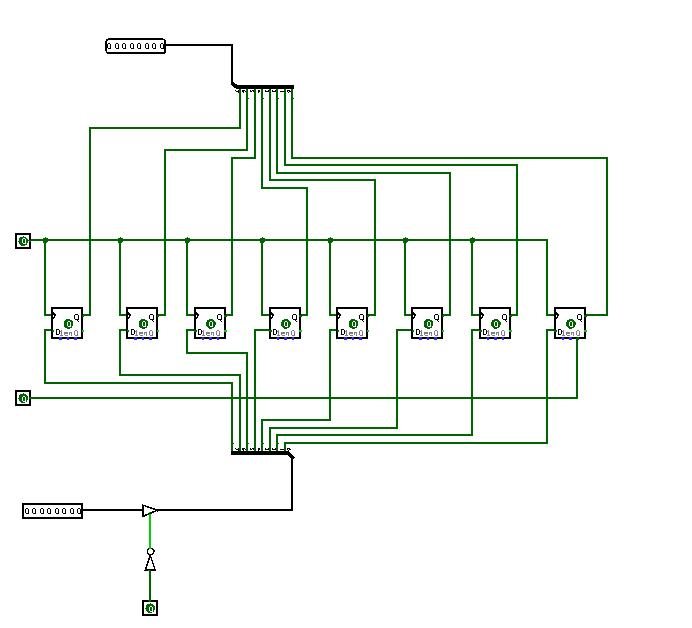
**4. Accumulator:**

The accumulator is an 8-bit buffer register that stores intermediate answers during a computer run. The accumulator has two outputs. The two-state output goes directly to the adder-subtractor and the three-state output goes to the bus. This implies that the 8-bit accumulator word continuously drives the adder- subtractor but only appears on the W bus when Ea is high.



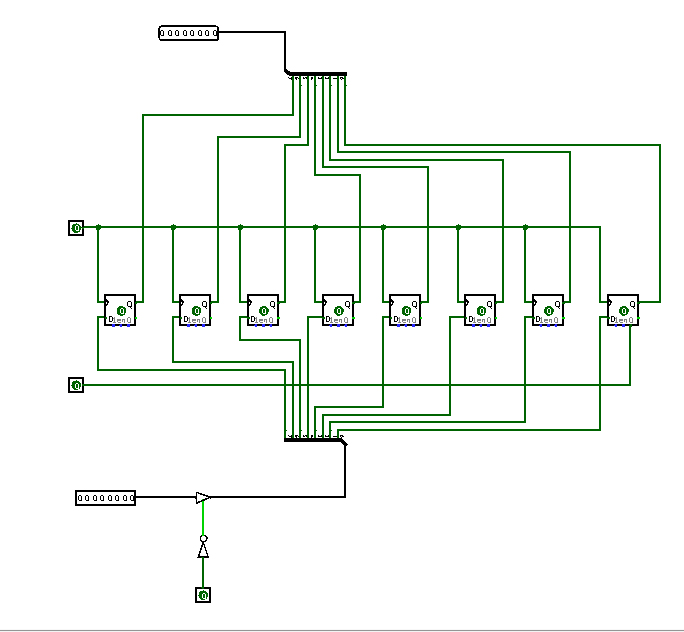
**5. B Register**

The B-register is a buffer register used in performing arithmetic operations. It supplies the number to be added or subtracted from the contents of the accumulator to the adder/subtractor. When data is available at the bus and Lb is low, at the positive clock edge, B register gets and stores the data.



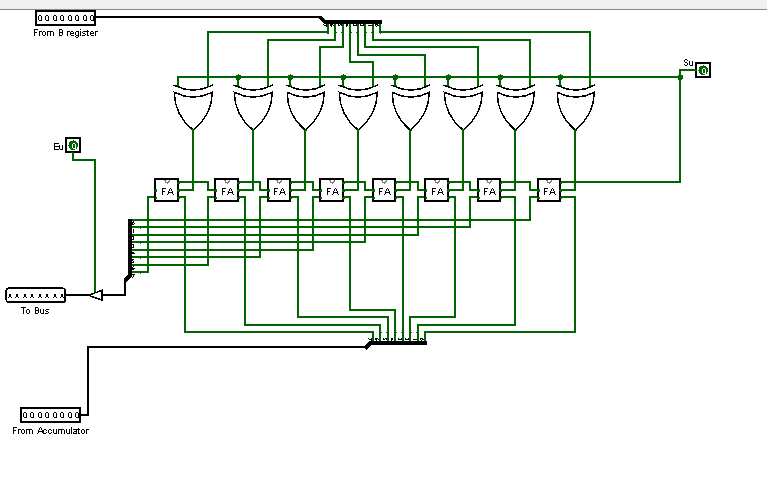
6. Output Register

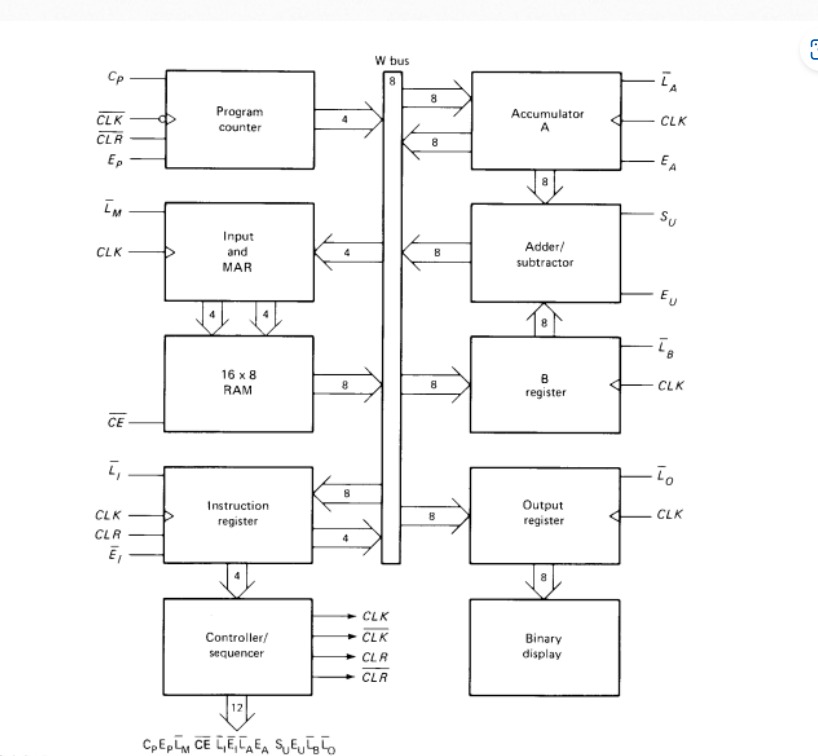
The output register gets and stores the value stored in the accumulator usually after the performance of an arithmetic operation. The answer that is stored in the accumulator is loaded into the output register through the W bus. This is done in the next positive clock edge when Ea is high and Lo is low. The processed data can now be displayed to the outside world.



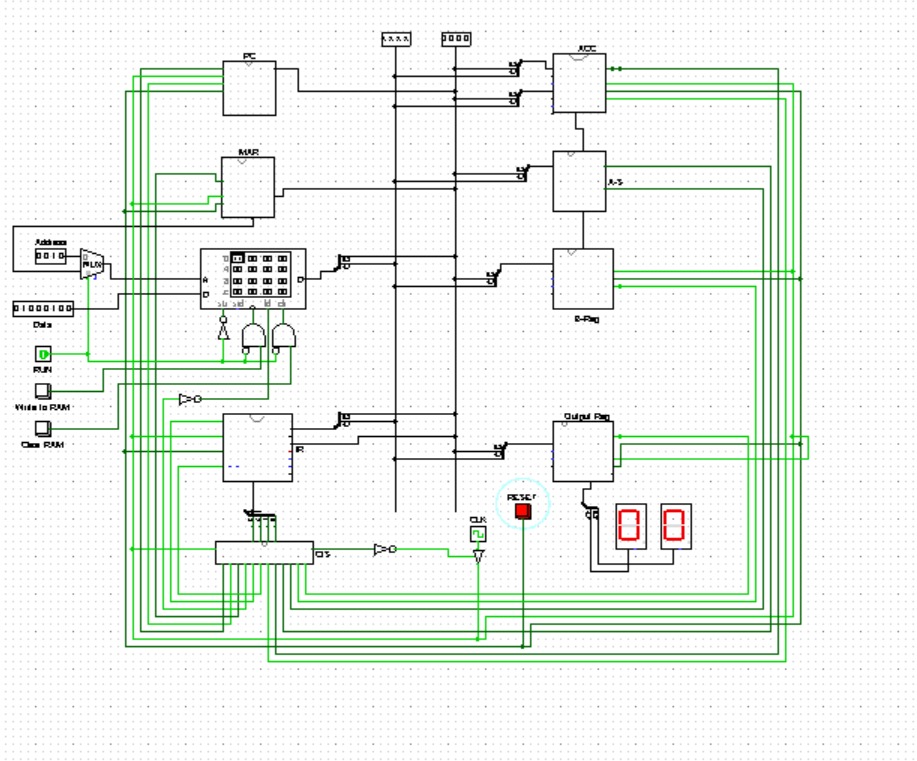
**7. 8-bit Adder/Subtractor**

The adder-subtractor asynchronously adds to or subtracts a value from the the accumulator depending on the value of Su. It makes use of 2’s complement to achieve this When Su is low the output of the adder-subtractor is the sum of the values in the accumulator and in the B-register (O/P = A + B). When Su is high, the output is the difference between them (O/P = A + B’).



**8. 8-bit "W" Bus**

**9. Final Circuit:**



* **Assembly Program and Machine Code**:

Example Assembly Program:

assembly

LDA 05H ; Load the value from memory address 05H into the accumulator

ADD 06H ; Add the value from memory address 06H to the accumulator

STA 07H ; Store the value of the accumulator into memory address 07H

OUT ; Output the value of the accumulator

HLT ; Halt the program

* **Corresponding Machine Code:**

Opcode Operand

0001 0101 ; LDA 05H

0010 0110 ; ADD 06H

0011 0111 ; STA 07H

1110 0000 ; OUT

1111 0000 ; HLT

**Conclusion:**

Our project successfully demonstrated the implementation of the SAP-1 computer architecture using Logisim. Each module was carefully designed and tested, resulting in a functional 8-bit computer capable of performing basic operations. This foundational project provided invaluable insights into computer architecture and the functioning of a simple processor.