EE2080 MICROPROCESSOR SYSTEMS LABORATORY

EXPERIMENT 1 – REPORT

8-BIT MICROCOMPUTER

Arun Kumar V - 122001049

AIM To develop an 8-bit microcomputer on Logisim using basic logic gates.

INTRODUCTION

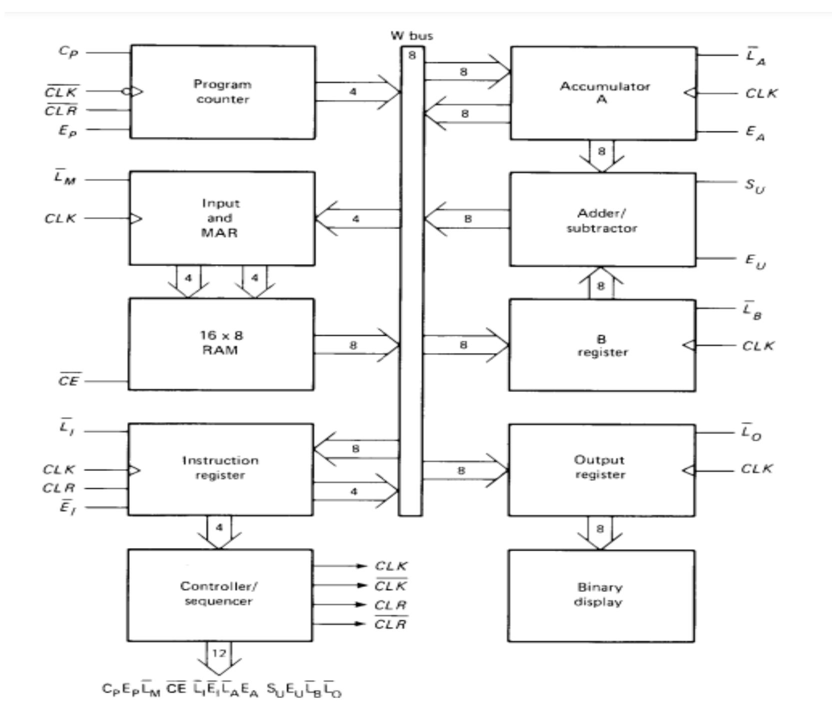
The aim of this project is to build an 8 Bit Microcomputer using basic logic gates. The microcomputer uses a Simple As Possible (SAP-1) architecture as explained by Albert Paul Malvino.

A microcomputer can perform basic arithmetic operations like addition, subtraction based on some data stored in RAM. It can be easily analysed by splitting it into separate blocks based on their functioning.

The individual subcircuits required are first built and then combined with other necessary elements to obtain the final circuit. The components used are Program Counter, Memory Address Register, RAMs, ROMs, Instruction Register, Control Unit, ALU, Hexadecimal Display, LEDs etc. Some of these components are first designed as a subcircuit and then integrated into the main circuit. The microcomputer uses an 8 Bit bus for the transfer of data and instruction in accordance with Von Neumann architecture where a single bus is used for both purposes.

Given below is a model of a microprocessor with basic functionalities.

BLOCK DIAGRAM OF MICROCOMPUTER:



INSTRUCTIONS AND OPCODE VALUES

Mnemonic	Operation	Op code
LDA	Loads RAM data into accumulator	0000
ADD	Adds RAM data into accumulator	0001
SUB	Subtracts Ram data from accumulator	0010
OUT	Loads accumulator data into output register	1110
HLT	Stop processing	1111

CYCLE AND STATES

Each operation in a microprocessor consists of 6 states.

The **first three** states are for the **fetch operation** in which the instruction is decoded and the **next three** states consist of an execution cycle. In each operation the first three states are the same. The difference lies in the states during the execution cycle. The states are achieved by using a ring counter with 6 bits in which only a single bit is high at a time.

CONTROL LOGIC:

Instruction	State	C_P	E_P	LM_	CE_	LI_	EI_	LA_	EA	Su	Eu	LB_	Lo_	HEX
FETCH	T1	0	1	0	1	1	1	1	0	0	0	1	1	**5E3**
	T2	1	0	1	1	1	1	1	0	0	0	1	1	**BE3**
	T3	0	0	1	0	0	1	1	0	0	0	1	1	**263**
LDA	T4	0	0	0	1	1	0	1	0	0	0	1	1	**1A3**
	T5	0	0	1	0	1	1	0	0	0	0	1	1	**2C3**
	T6	0	0	1	1	1	1	1	0	0	0	1	1	**3E3**
ADD	T4	0	0	0	1	1	0	1	0	0	0	1	1	**1A3**
	T5	0	0	1	0	1	1	1	0	0	0	0	1	**2EA**
	T6	0	0	1	1	1	1	0	0	0	1	1	1	**3C7**
SUB	T4	0	0	0	1	1	0	1	0	0	0	1	1	**1A3**
	T5	0	0	1	0	1	1	1	0	0	0	0	1	**2EA**
	T6	0	0	1	1	1	1	0	0	1	1	1	1	**3CF**
OUT	T4	0	0	1	1	1	1	1	1	0	0	1	0	**3F2**
	T5	0	0	1	1	1	1	1	0	0	0	1	1	**3E3**
	T6	0	0	1	1	1	1	1	0	0	0	1	1	**3E3**
HLT	T4	1	1	1	1	1	1	1	1	1	1	1	1	**FFF**

ROM1

ROM1

OPERATION	DATA	ADDRESS
FETCH	5E3	0000
	BE3	0001
	263	0010
LDA	1A3	0011
	2C3	0100
	3E3	0101
ADD	1A3	0110
	2EA	0111
	3C7	1000
SUB	1A3	1001
	2EA	1010
	3CF	1011
OUT	3F2	1100
	3E3	1101
	3E3	1110
HLT	FFF	1111

ROM2

ROM2

Address	Data
0000(LDA)	0011
0001(ADD)	0110
0010(SUB)	1001
1110(OUT)	1110
1111(HLT)	1111

IMPLEMENTATION AND RESULTS

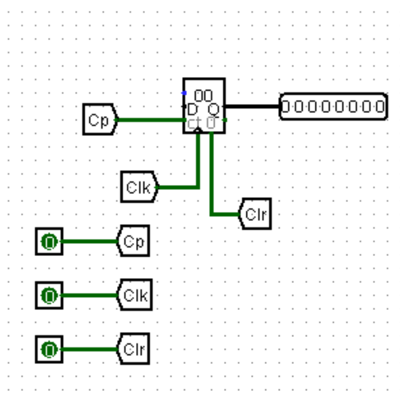
The 8 Bit microcomputer is designed to perform Addition, Subtraction, XOR, Bit-Shifting, loading data, and displaying output.

There are subcircuits used in the main circuit, to perform each of these operations. We are also using 16x8 RAM to store the instruction and data beforehand.

Each instruction has a unique code which gets decoded to give the necessary signals for carrying out the operation. The control unit performs this function. The output is stored in an output register before displaying it.

The individual subcircuits used are listed below.

PROGRAM COUNTER



The program counter gives the RAM address of the operation to be performed to the Memory address register. It counts till 7 since the first 8 bits in the RAMare instructions.
The program counter is incremented by 1 in each fetch cycle after sending the address

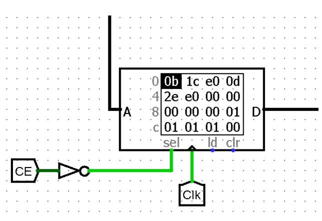
MAR (MEMORY ADDRESS REGISTER)

The MAR gets the address of the data/instruction in the RAM from the program counter or the IR. In the next state MAR applies this 4-bit address to the RAM.

RAM (RANDOM ACCESS MEMORY)

A 16x8 RAM is used to store data as well as instruction before a computer runs. First 8 bits signify the instruction to be carried out while the next 8 bits contain the data that can be used for various operations.

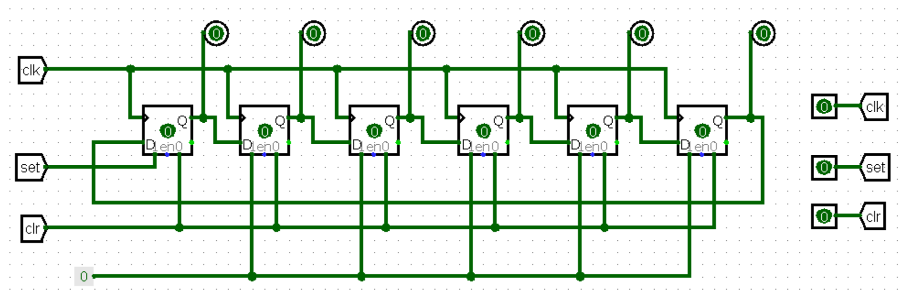
During a computer run the RAM receives 4-bit addresses from the MAR and thus the content in RAM is placed on the bus for use.



In the fetch cycle, instructions are read and transferred to the instruction register and in the execution cycle, it reads the data values at the given address locations and sends them to 8 bit bus, which are taken into accumulators or registers.

RING COUNTER

It is a part of the control unit. It is made using 6 d flip-flops, one for each state. Set is given to the first one to start the ring counter.

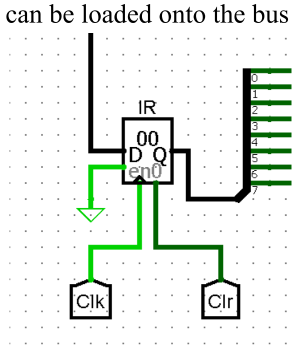


Output of the first flip-flop is set to be the input of the second flip-flop and so on for every clock tick, that is, it shifts values by 1 bit in every clock cycle.

INSTRUCTION REGISTER (IR)

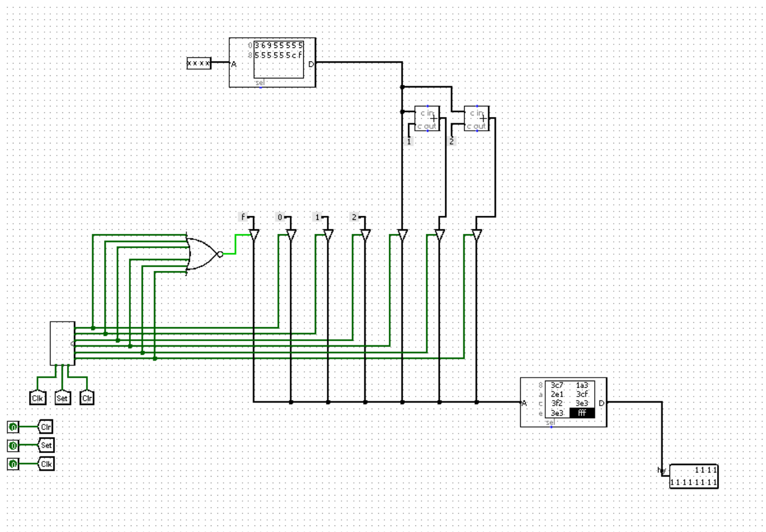
It is a 8 bit register used to latch onto the instruction passed by the RAM for execution.Just like MAR, the instruction register is also an 8 bit register, so the same 8 bit register made for MAR is also used here.

Control signal LI (negative logic) is used for loading the instruction from RAM.then the 4 most significant byte containing the op code is sent to the control unit for execution.last 4 least significant having the address of the data can be loaded onto the bus using EI connected to the bus using tristate buffer.



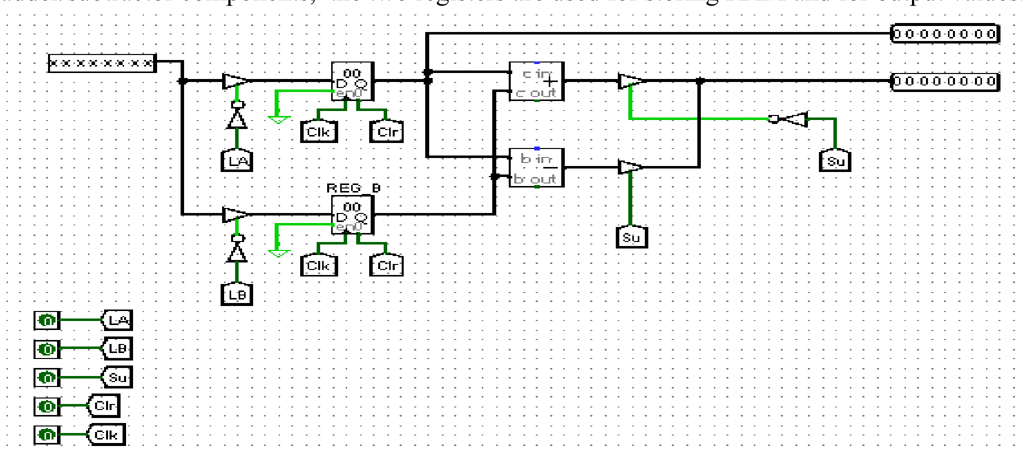
CONTROL UNIT (CU)

It is like the brain of the microcomputer and is responsible for controlling all the other components in the computer. It consists of a ring counter responsible for the 6 states to fetching and executing any instruction.the ring counter is implemented using 6 D flip flops in series fashion, one for each state.



ALU

ALU is used for arithmetic operations like addition, subtraction etc. It consists of accumulator, register B and adder/subtractor components, the two registers are used for storing RAM and for output values.



ADDER/SUBTRACTOR

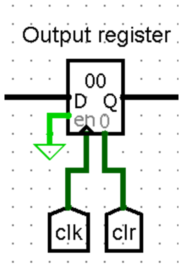
It calculates the arithmetic result using data in accumulator and register B, according to the instruction and sends the output to the accumulator.

ACCUMULATOR (REGISTER A)

Accumulator stores the intermediate 8-bit data which is displayed as well used for operations. The Output is displayed after the data from the accumulator goes to the output register. The data obtained after arithmetic and logical operations are also stored in the accumulator

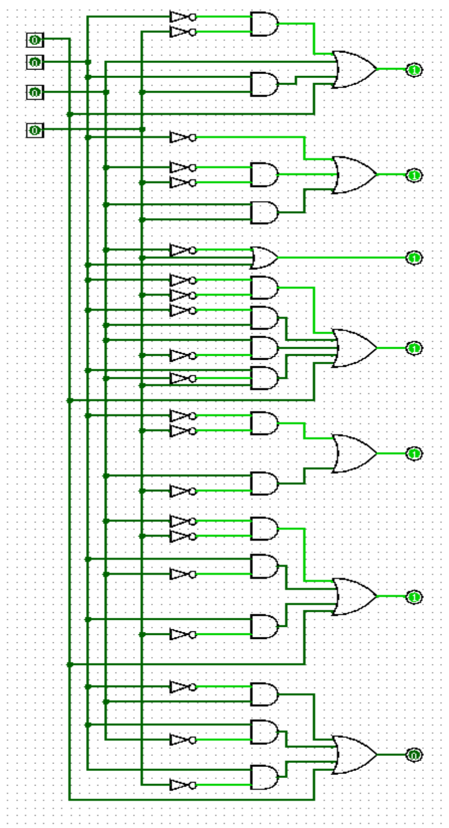
OUTPUT REGISTER

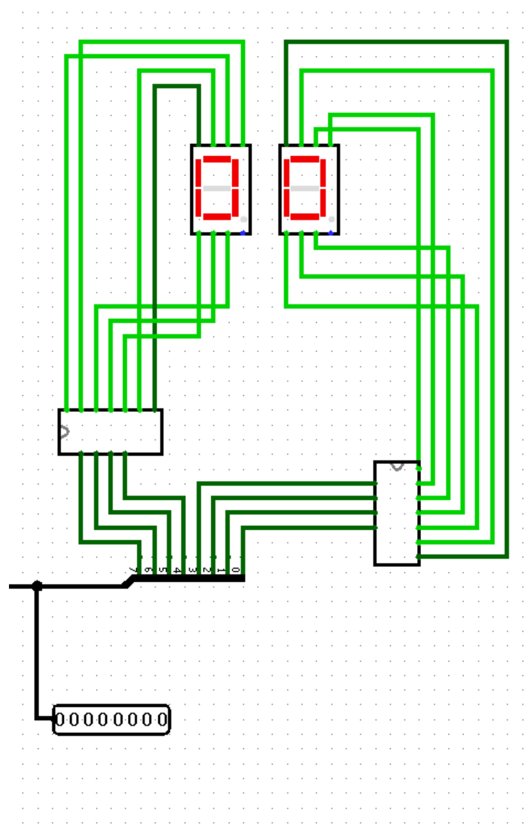
It is a register used to store the output from the accumulator.



OUTPUT DISPLAY: SEVEN SEGMENT DISPLAY

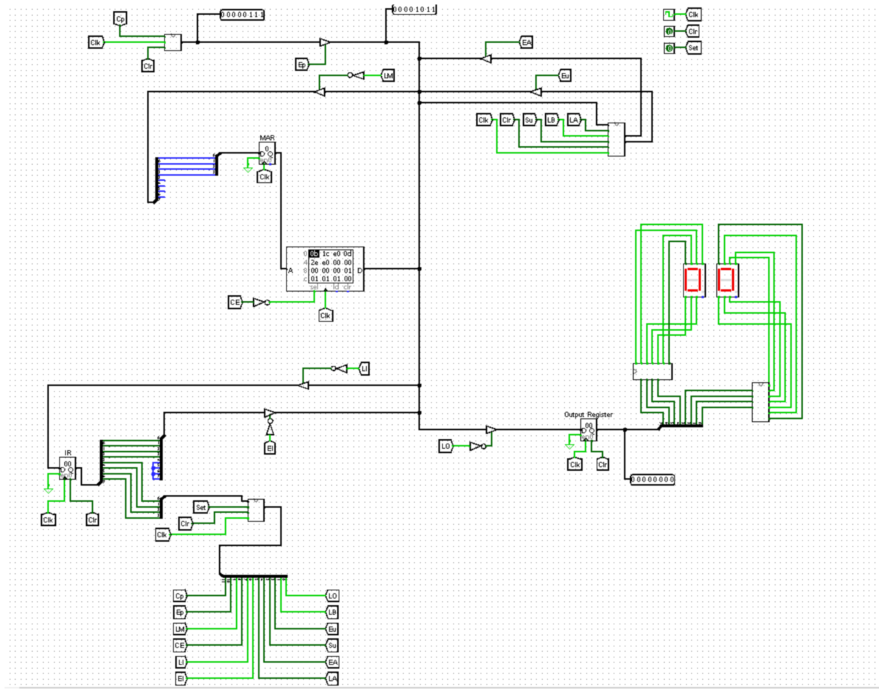
It displays the binary output from the output register as decimal using 7 segments for each digit.
BCD to 7 segment display circuit diagram:





It is a two digit 7-segment display unit which displays the data in the output register. The highest possible value from the output is 99.

COMPLETE CIRCUIT



RESULTS OF EXECUTION CYCLE

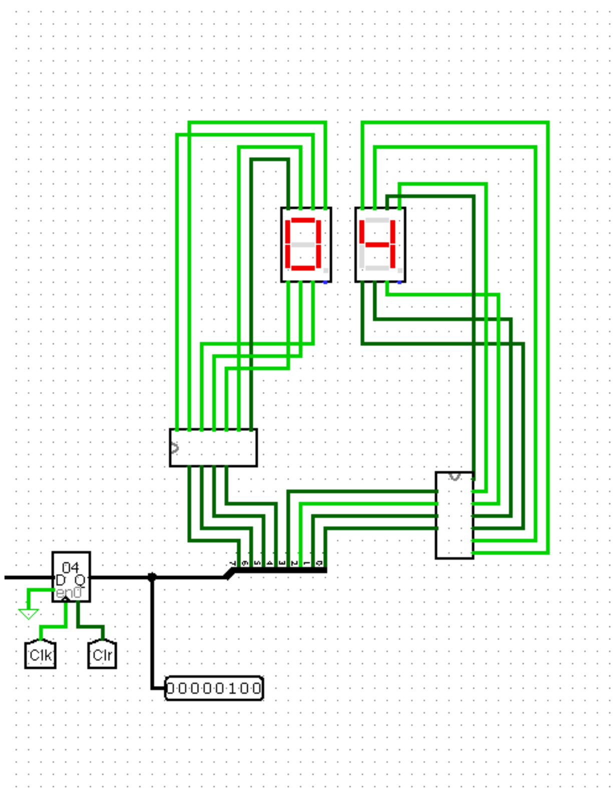
Initially LDA is called which loads value 1 to the accumulator , then the next value 3 is stored into register B which gets added to the value in accumulator resulting in the output 4 , which now gets stored to the accumulator.Now the output is displayed on the 7 - segment display.

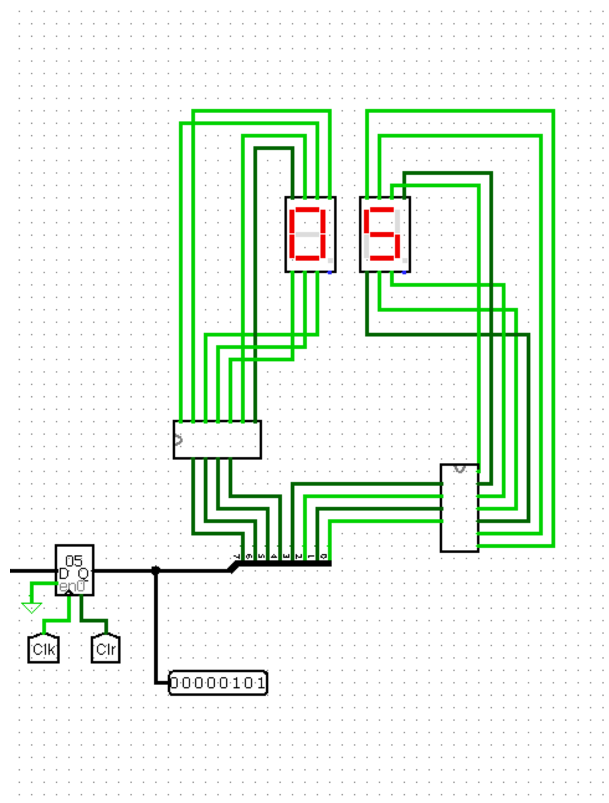
For subtraction also LDA is called which stores the value 8 to the accumulator and our next value 3 is stored into register B which gets subtracted from the value in accumulator resulting in the output 5, which is stored again to the accumulator.Now the output is displayed on the 7 - segment display.

Addition : 01 + 03 = 04

Subtraction : 08 - 03 = 05

ADDITION **SUBTRACTION**





Link of simulation video:

https://drive.google.com/drive/folders/144ieA-3Yi4cwIyWTaXvTP0VBFlCJK5Ei?usp=sharing

CONCLUSION

An 8-bit microcomputer that can perform basic operations like addition and subtraction was made using Logisim, which can also be modified to perform more operations like multiplication and division etc.

Complexity of this circuit increases with increase in bit size and number of control states.