EE2080 – MICROPROCESSOR SYSTEMS LABORATORY EXPERIMENT 1

122001006 AMITH P. JOY

Aim: To develop an 8-bit microcomputer on Logisim using basic logic gates and perform basic functions.

Components of the 8-bit microcomputer:

The main components of the 8-bit microcomputer are program counter, memory access register (MAR), random access memory (RAM), instruction register, control unit, accumulator, register B, arithmetic and logic unit (ALU), output register, output interfacing.

Program Counter:

The program counter creates a 4-bit binary number which will correspond to an address in the RAM of the computer. There are 3 signals given to the program counter, they are Cp, clk and clr. Clk is the clock signal of the microcomputer which makes sure that the counter increments its value synchronously. Clr signal clears the count and resets it to '0000'. The counter will only increment if the Cp signal is 1. The output of the program counter is given to the bus as an 8-bit and this is controlled by the signal Ep. The program counter has been implemented by using 4 T flip-flops. The circuit is as follows:

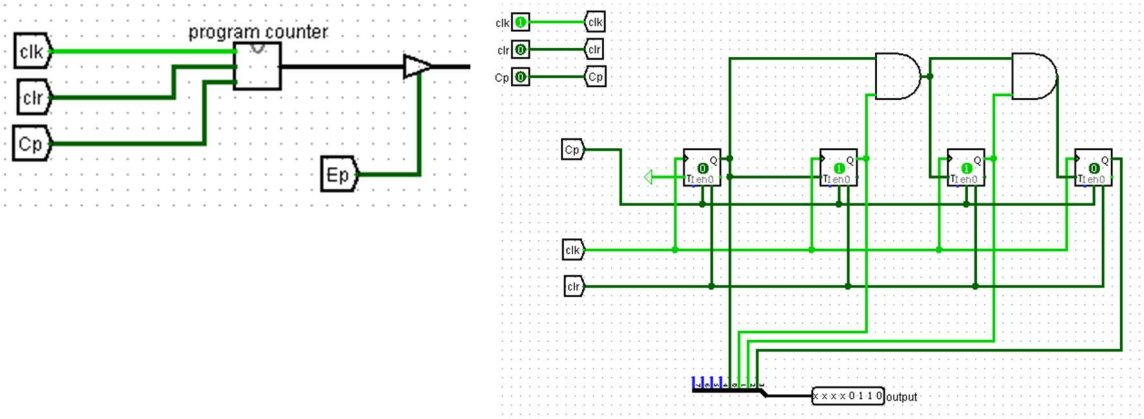


IMAGE: Program counter and its internal circuit

Memory Address Register:

Memory Address Register or MAR is a register which will have the address of the memory location in RAM which should be accessed. It is a 4-bit register with the control signal Lm' controlling its input data. The MAR is connected to the RAM. MAR also has clk as an input signal to make it synchronous.

RAM:

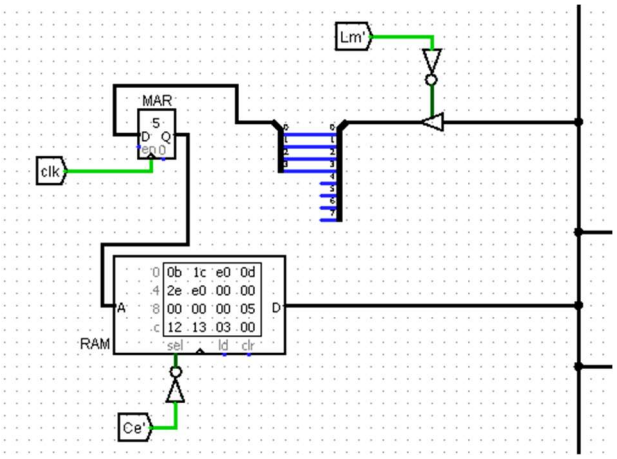


IMAGE: MAR and RAM

It has all the instructions to be performed and the data on which the instructions are to be performed stored in it. Its 4-bit address is given as an input from the MAR and its output, which will be the content of the RAM at that address is given to the bus under the control of the signal Ce'. It does not have clk input. The first half of the address locations will be used to store the instructions and the second half of the address locations will be used to store data in which these instructions will be executed.

Instruction Register:

It is the register which will receive the instructions from RAM and send it to the control unit. The 8-bit input to the instruction register from the bus is controlled by the signal Li'. These 8-bits are divided into 2. The first 4-bits (most significant bits, i.e., bits 7,6,5,4) are sent to the control unit and the rest of the 4-bits are sent back to the bus (these 4-bits are the RAM address at which the instruction will be carried out) under the control of the signal Ei'. Clk and clr are also inputs to the instruction register. Clr will reset the register by clearing the data in it and clk is the clock signal of the microcomputer to make it synchronous.

Control Unit:

The control unit gives out all the control signals of the microcomputer according to the instruction to be performed. The control unit gets the op code of the instruction from the instruction register and sends out the 12-bits of control signals. The control unit does this with the help of 2 ROMs (ROM 1 and ROM 2) and a ring counter. The circuit is as follows:

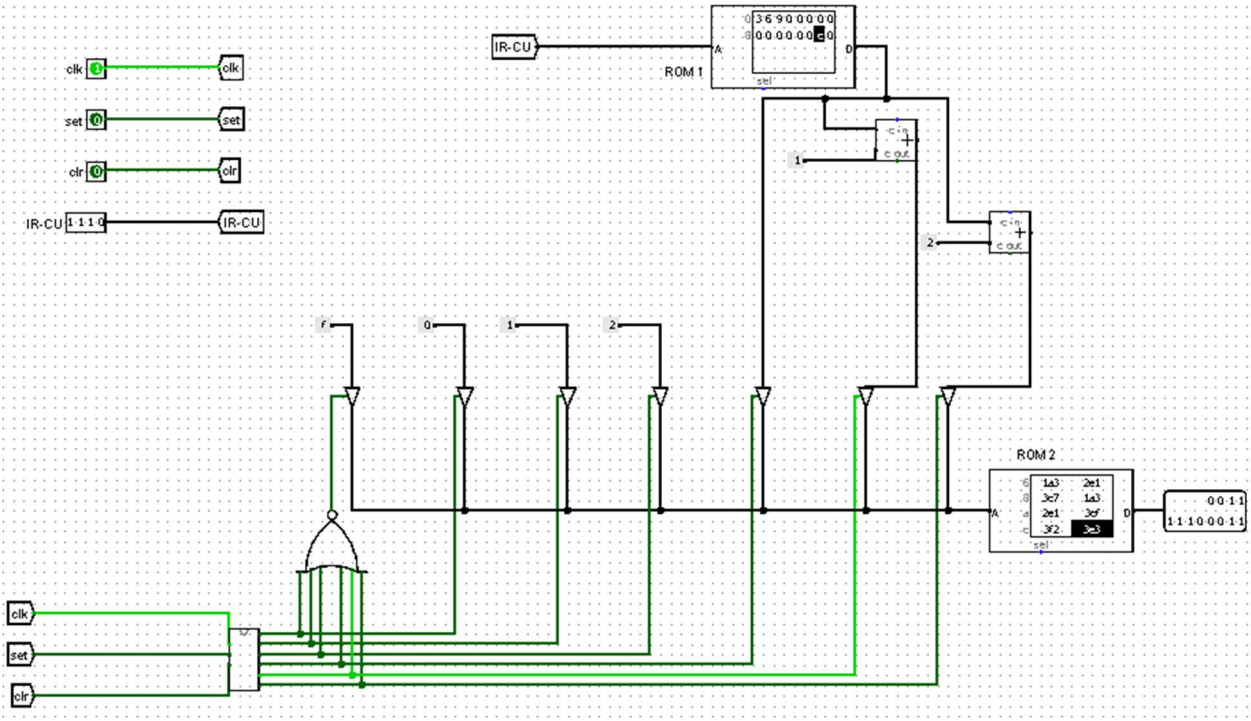


IMAGE: Control unit (we can see the 2 ROMs and also the ring counter)

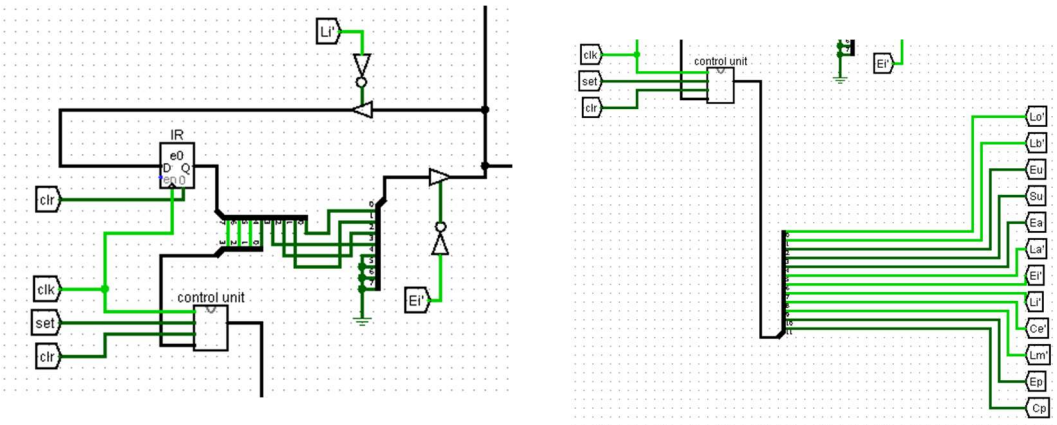


IMAGE: (1) Instruction register (IR) and the control unit and (2) 12 control signals as output from control unit

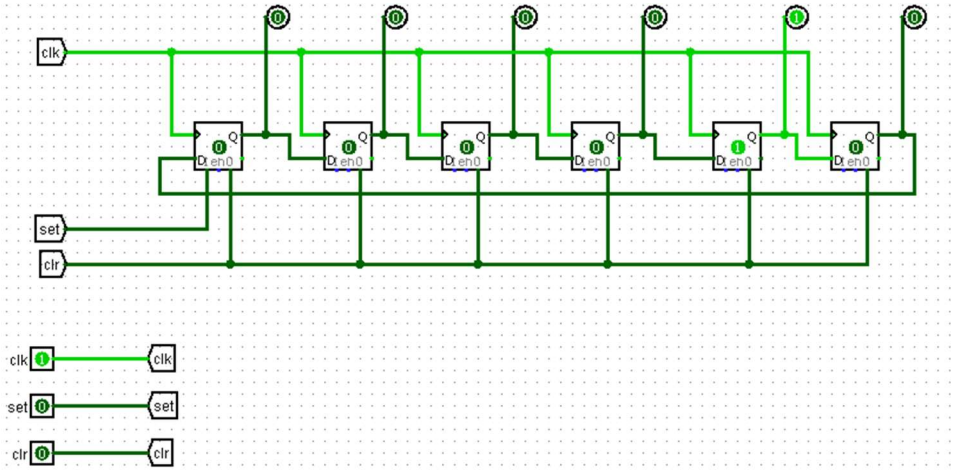


IMAGE: circuit of the ring counter

Clk and clr are also inputs to the instruction register. Clr will reset the register by clearing the data in it and clk is the clock signal of the microcomputer to make it synchronous. Set signal will set the ring counter to the first location in ROM 2. The microcomputer will work only if this happens.

(More details about the control unit and its working is explained under the title Working)

Accumulator:

The accumulator is a register which stores the data after LDA instruction and also the result of the arithmetic operation performed in the ALU. It has an 8-bit input from the bus and this input is controlled by the control signal La'. The 8-bit output of the accumulator can be sent to the bus or to the ALU. The output to the bus is controlled by the control signal Ea. The accumulator has clk as input to make it synchronous and clr as input to reset the register.

Register B:

It is a register which is used for performing the functions of the ALU. If the function performed by the ALU requires 2 sets of data simultaneously (e.g.: addition, subtraction, etc.), register B will give the 2nd data for the function. So, register B has its 8-bit output to the ALU. The input 8-bits of register is from the bus and is controlled by the signal Lb'. Register B also has clk as input to make it synchronous and clr as input to reset the register.

Arithmetic and Logic Unit (ALU):

The ALU is where all the arithmetic and logic functions of the microcomputer occur. In this microcomputer, the ALU performs addition and subtraction. The result of the operation is sent to the accumulator through the bus. The output to the bus is controlled by the signal Eu. The signal Su decides whether addition or subtraction has to

be performed. ALU gets the data stored in the accumulator and register B for performing its functions without using the bus. Clk and clr signals are not used for the ALU.

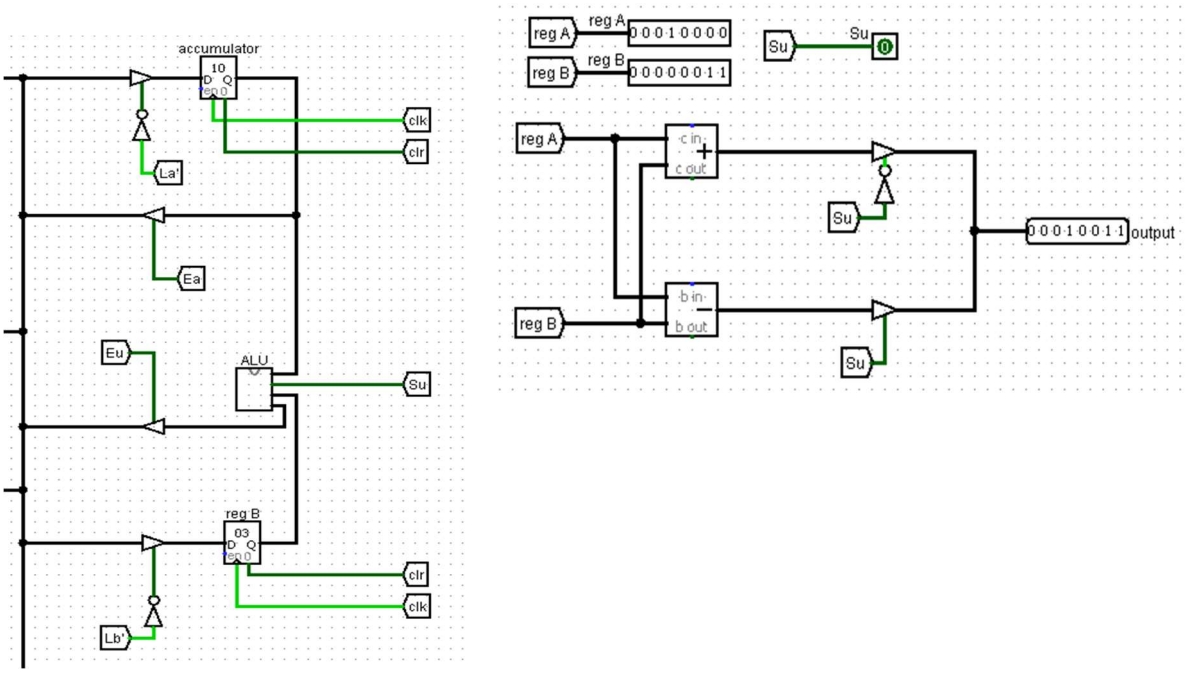


IMAGE: (1) Accumulator, ALU and Register B, (2) Internal circuit of ALU

Output Register:

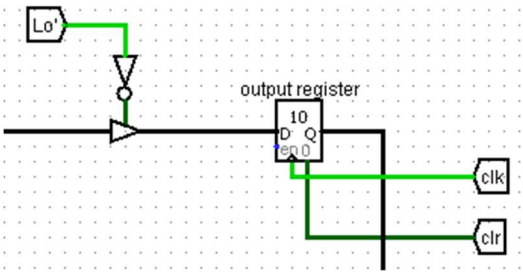


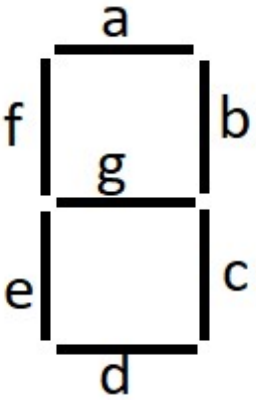
IMAGE: output register

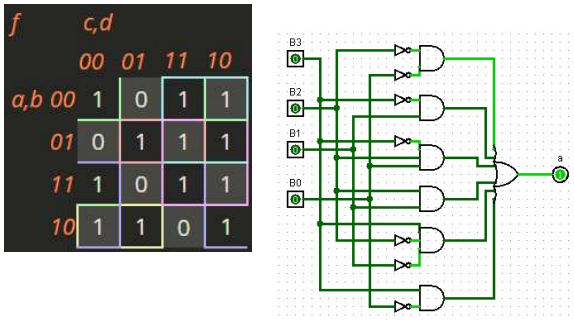
Output register is a register which receives the data to be given as output from the bus and sends it to the output interfacing. The 8-bit input to the register from the bus is controlled by the signal Lo'. Output register also has clk as input to make it synchronous and clr as input to reset the register.

Output Interfacing:

The output interfacing is used to display the output to a user. This microcomputer is using a 7-segment display to display the hexadecimal value of the output. The arrangement of segments and hex decoder truth table is:

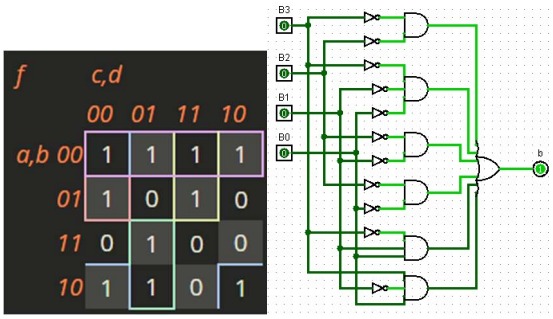
Inputs				Segments							
A	B	C	D	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	For display 0
0	0	0	1	0	1	1	0	0	0	0	For display 1
0	0	1	0	1	1	0	1	1	0	1	For display 2
0	0	1	1	1	1	1	1	0	0	1	For display 3
0	1	0	0	0	1	1	0	0	1	1	For display 4
0	1	0	1	1	0	1	1	0	1	1	For display 5
0	1	1	0	1	0	1	1	1	1	1	For display 6
0	1	1	1	1	1	1	0	0	0	0	For display 7
1	0	0	0	1	1	1	1	1	1	1	For display 8
1	0	0	1	1	1	1	1	0	1	1	For display 9
1	0	1	0	1	1	1	0	1	1	1	For display A
1	0	1	1	0	0	1	1	1	1	1	For display b
1	1	0	0	1	0	0	1	1	1	0	For display C
1	1	0	1	0	1	1	1	1	0	1	For display d
1	1	1	0	1	0	0	1	1	1	1	For display E
1	1	1	1	1	0	0	0	1	1	1	For display F





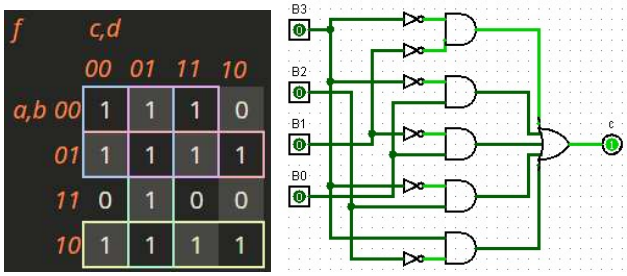
K-map and circuit for Segment A

f(a, b, c, d) = b'd' + a'c + a'bd + ab'c' + ad' + bc



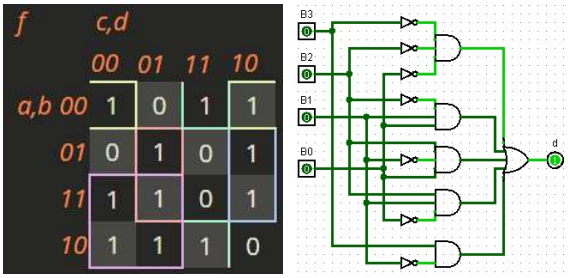
K-map and circuit for segment B

f(a, b, c, d) = a'c'd' + a'cd + b'd' + ac'd + a'b'



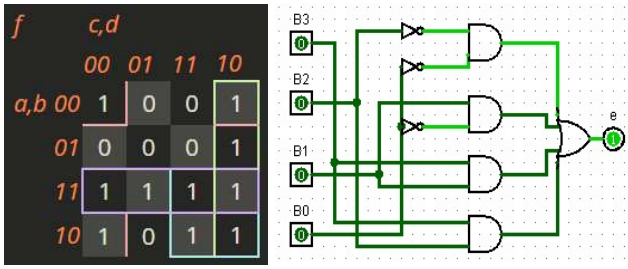
K-map and circuit for segment C

f(a, b, c, d) = a'b + ab' + c'd + a'c' + a'd



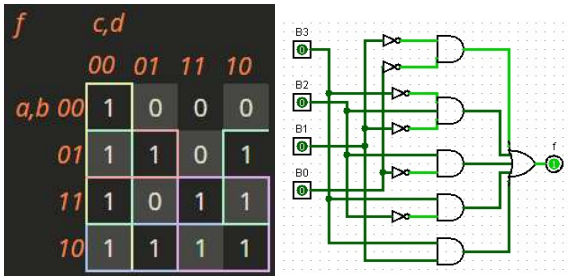
K-map and circuit for segment D

f(a, b, c, d) = bc'd + a'b'd' + b'cd + bcd' + ac'



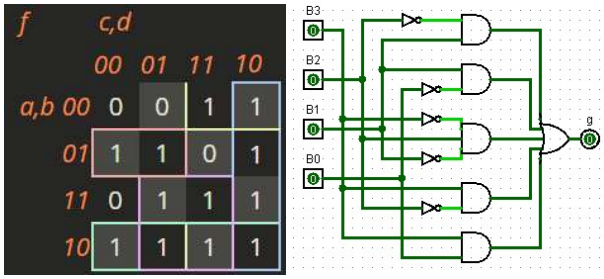
K-map and circuit for segment E

f(a, b, c, d) = b'd' + cd' + ac + ab

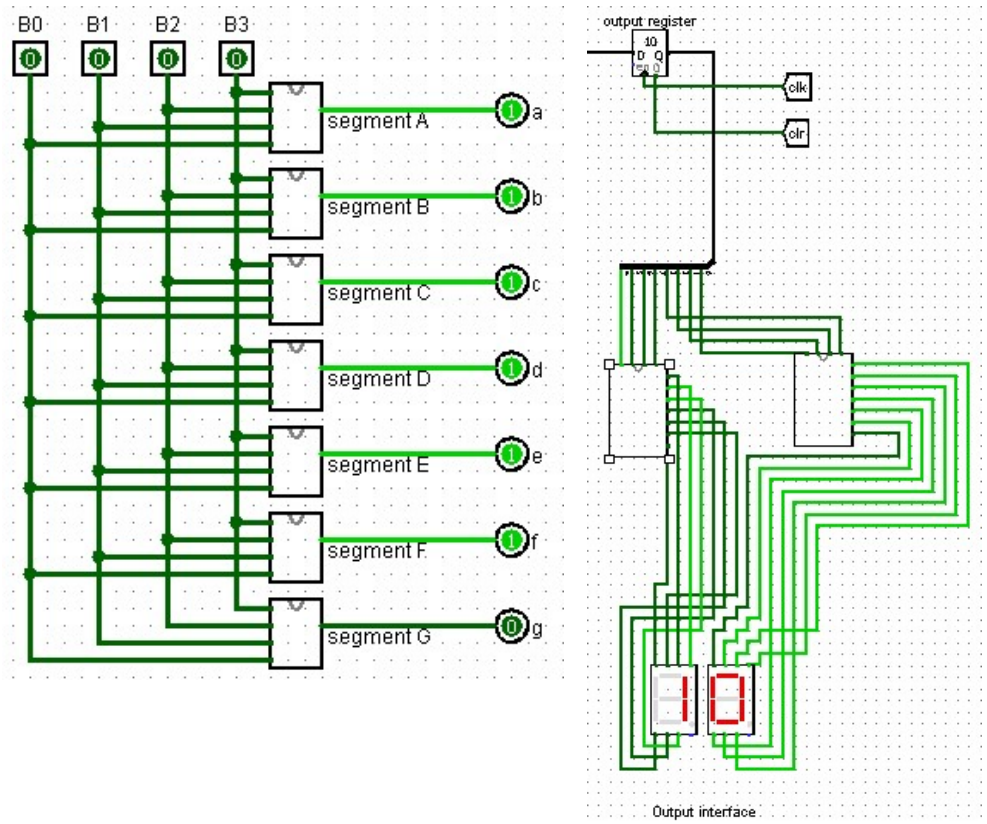


K-map and circuit for segment F

f(a, b, c, d) = c'd' + a'bc' + bd' + ab' + ac



K-map and circuit for segment G; f(a, b, c, d) = b'c + ab' + a'bc' + cd' + ad



Working:

The 8-bit microcomputer can perform the following operations:

Mnemonic	Operation
LDA	Load RAM data into accumulator
ADD	Add RAM data to accumulator
SUB	Subtract RAM data from accumulator
OUT	Load accumulator data into output register
HLT	Stop processing

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

LDA, ADD, SUB and OUT are executed in 6 clock cycles each. The first 3 clock cycles are the same for all of them and is called fetch cycle. The last 3 clock cycles are called the execution cycle.

Fetch cycle:
1st clock cycle (address state): Ep=0 and Lm'=1 so that program counter and MAR are connected to the bus. MAR receives a 4-bit address from the program counter.

2nd clock cycle (increment state): Program counter value is incremented by 1 and no other module is connected to the bus.

3rd clock cycle (memory state): Data stored in memory address in RAM pointed by MAR is sent to instruction register through the bus. RAM and instruction register are connected to the bus. Instruction register now has an 8-bit data. The 1st 4-bits are the op code of the operation and last 4-bits are the RAM address of the data on which the operation has to be done.

Execution cycle:

The last 3 cycles are called the execution cycle. It will be different for different operations.

Execution cycle for LDA:

1st cycle: Address of LDA process goes to MAR through the bus from instruction register.

2nd cycle: The data at the address in MAR is sent to the accumulator.

3rd cycle: In this cycle none of the components are connected to the bus. This cycle is redundant.

Execution cycle for ADD/SUB:

1st cycle: MAR receives the address of ADD/SUB from instruction register.

2nd cycle: Data at the address pointed by MAR is sent to register B. Addition/subtraction happens in the ALU.

3rd cycle: The result of addition/subtraction is now loaded in the accumulator.

Execution cycle for OUT:

1st cycle: Data in the accumulator is now sent to the output register and the output can be seen at the output interface.

2nd cycle and 3rd cycle: Redundant cycles. None of the components are connected to the bus.

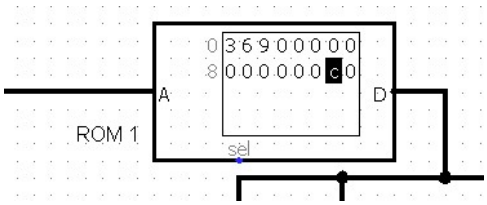
Using this information, the value of each control signal in each cycle could be decided.

Instruction	State	C_P	E_P	$\overline{L_M}$	$\overline{CE}$	$\overline{L_I}$	$\overline{E_I}$	$\overline{L_A}$	E_A	S_U	E_U	$\overline{L_B}$	$\overline{L_O}$	Hex Code
Fetch	T_1	0	1	0	1	1	1	1	0	0	0	1	1	5E3
	T_2	1	0	1	1	1	1	1	0	0	0	1	1	BE3
	T_3	0	0	1	0	0	1	1	0	0	0	1	1	263
LDA (0000)	T_4	0	0	0	1	1	0	1	0	0	0	1	1	1A3
	T_5	0	0	1	0	1	1	0	0	0	0	1	1	2C3
	T_6	0	0	1	1	1	1	1	0	0	0	1	1	3E3
ADD/SUB (0001/0010)	T_4	0	0	0	1	1	0	1	0	0	0	1	1	1A3
	T_5	0	0	1	0	1	1	1	0	0	0	0	1	2E1
	T_6	0	0	1	1	1	1	0	0	0/1	1	1	1	3C7/3CF
OUT (1110)	T_4	0	0	1	1	1	1	1	1	0	0	1	0	3F2
	T_5	0	0	1	1	1	1	1	0	0	0	1	1	3E3
	T_6	0	0	1	1	1	1	1	0	0	0	1	1	3E3

The control unit has to be designed in such a way that each operation is done in 6 cycles. So, the ring counter will have 6 D flip-flops with only one of them having 1 stored at a time. The 1st 3 cycles will be same for all operations (fetch cycle). ROM 2 will have hex code of each cycle of the operations. ROM 1 will have the address where hex code of the 1st cycle of the execution cycle stored in it and the address of it will be the op code of the corresponding function.

The ring counter makes the control unit choose the hex codes for the fetch cycle and also for the 2nd and 3rd cycle of execution cycle when the address of only the 1st cycle is given. The ring counter has been made using 6 D flip-flops connected like a ring such that when the set signal is applied, one of the flip-flops will have 1 stored in it asynchronously and that '1' will be passed to the next ones in the subsequent cycles. In the 1st, 2nd and 3rd cycles, the ring counter will activate the 0th, 1st and the 2nd locations of ROM 2 completing the fetch cycle. In the 4th cycle, ring counter will activate the address location with hex code for the 1st cycle of the operation. In the 5th and 6th cycles, 1 and 2 will be added to this location so that the hex code for completing the operation is selected. These hex codes are 12-bit binary numbers of which each bit is a control signal of the microcomputer. After completion of 6 clock cycles, ring counter will again activate the 0th location of ROM 2 because the D flip-flops are connected like a ring. If the set signal is not applied, then none of the flip-flops will have '1' stored and the address '1111' of ROM 2 will be activated, where none of the operations happen.

Address	Data	Operation
0000	5E3	Fetch
0001	BE3	
0010	263	
0011	1A3	LDA
0100	2C3	
0101	3E3	
0110	1A3	ADD
0111	2E1	
1000	3C7	
1001	1A3	SUB
1010	2E1	
1011	3CF	
1100	3F2	OUT
1101	3E3	
1110	3E3	
1111	FFF	NOP



Example: Let the first three 8-bit data in the RAM be 0b 1c e0 (the same RAM of which the picture is given). In the 1st 6 clock cycles, 0b, i.e., loading the data at location b of RAM to the accumulator happens. The next 6 clock cycles complete the operation 1c, i.e., loading data at location c to reg B, adding it with the number in accumulator and storing the sum in the accumulator. The next 6 clock cycles will complete e0, i.e., value in the accumulator is given to the output interface.

Conclusion: The 8-bit microcomputer capable of performing addition and subtraction on numbers and then displaying the output has been developed on Logisim.

Extra contributions: 7-segment display for as output interface for displaying hexadecimal value of the output. Designed hexadecimal to 7-segment decoder.

Link for project folder:
https://drive.google.com/drive/folders/1hO5EKPW5RA9zHaVv4nwQrRvDurLxt2t2?usp=sharing