Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Department of CSE

Programme: B.E
Term: Jan to May 2019
Course: Computer Organization
Course Code: CS45

Activity V: Designing an ALU to perform arithmetic and logical functions using Logisim simulator.

Name: Mohit Raj Soni	**Marks: /10**	**Date:24-05-2020**
USN:1MS18CS074	**Signature of the Faculty:**	

Objective: To simulate the working of Arithmetic and Logical Unit using simulator.

Simulator Description: Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to build larger circuits from smaller sub circuits, and to draw bundles of wires with a single mouse drag, Logisim can be used (and is used) to design and simulate entire CPUs for educational purposes.

Activity to be performed by students:

List out the steps in designing ALU

1 Add the two i/P pins, Name them A and B

2 Add OR, AND, EX-OR, NOR gates and a 1-bit adder.

3 Connect the A's and B's of all the gates to their respective pins

4 Add an output pin and name it result

5 Add a 1-bit multiplexer with 3 select bits

6 Connect the outputs of all gates to the MUX

7 Connect 3-bit input pin to Mux

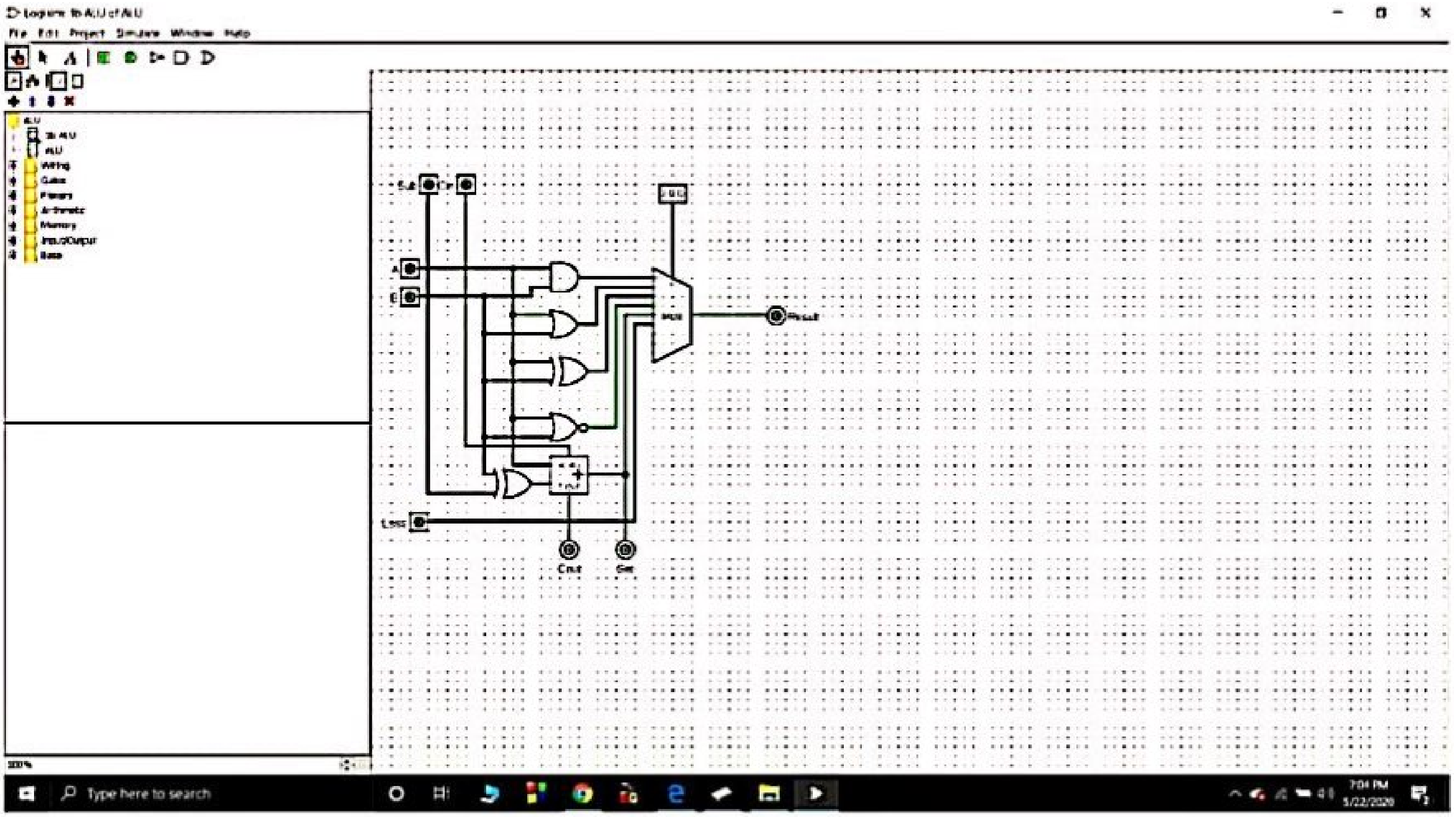
8 Add i/P pin to Cin & output pin to Cout

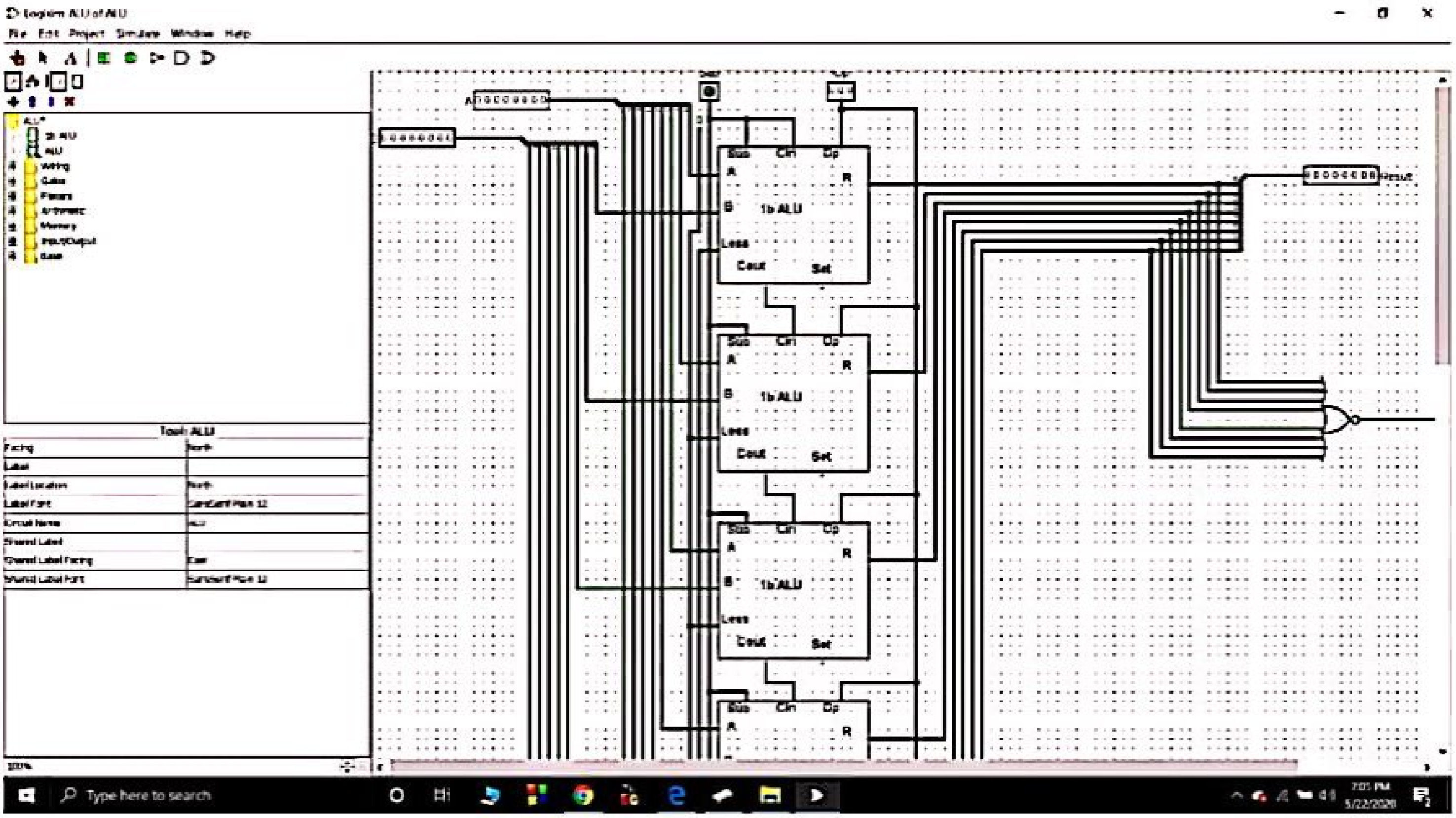
9 Add an EX-OR gate & connect its O/P to Cout.
The first i/P must be connected to B and the second to another
i/P pin SUB.

10 Add another i/P and name it less connect it to Mux

11 Add an output pin and name it slt connect it to the multiplexer
O/P of adder unit

Snapshots:


Logisim: 1b ALU of ALU
File Edit Project Simulate Window Help
ALU
1b ALU
ALU
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Sub
Cin
A
B
MUX
Result
Less
Cout
Set
100%
Type here to search
7:04 PM
5/22/2020



Tool: ALU

Facing	North
Label	
Label Location	North
Label Font	SansSerif Plain 12
Circuit Name	ALU
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Plain 12

100%

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Department of CSE

Programme: B.E
Term: Jan to May 2019
Course: Computer Organization
Course Code: CS45

Activity VI: Designing memory system using Logisim simulator.

Name: Mohit Raj Soni	**Marks: /10**	**Date:24-05-2020**
USN:1MS18CS074	**Signature of the Faculty:**	

Objective: To simulate the writing operation on memory.

Simulator Description: Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to build larger circuits from smaller sub circuits, and to draw bundles of wires with a single mouse drag, Logisim can be used (and is used) to design and simulate entire CPUs for educational purposes.

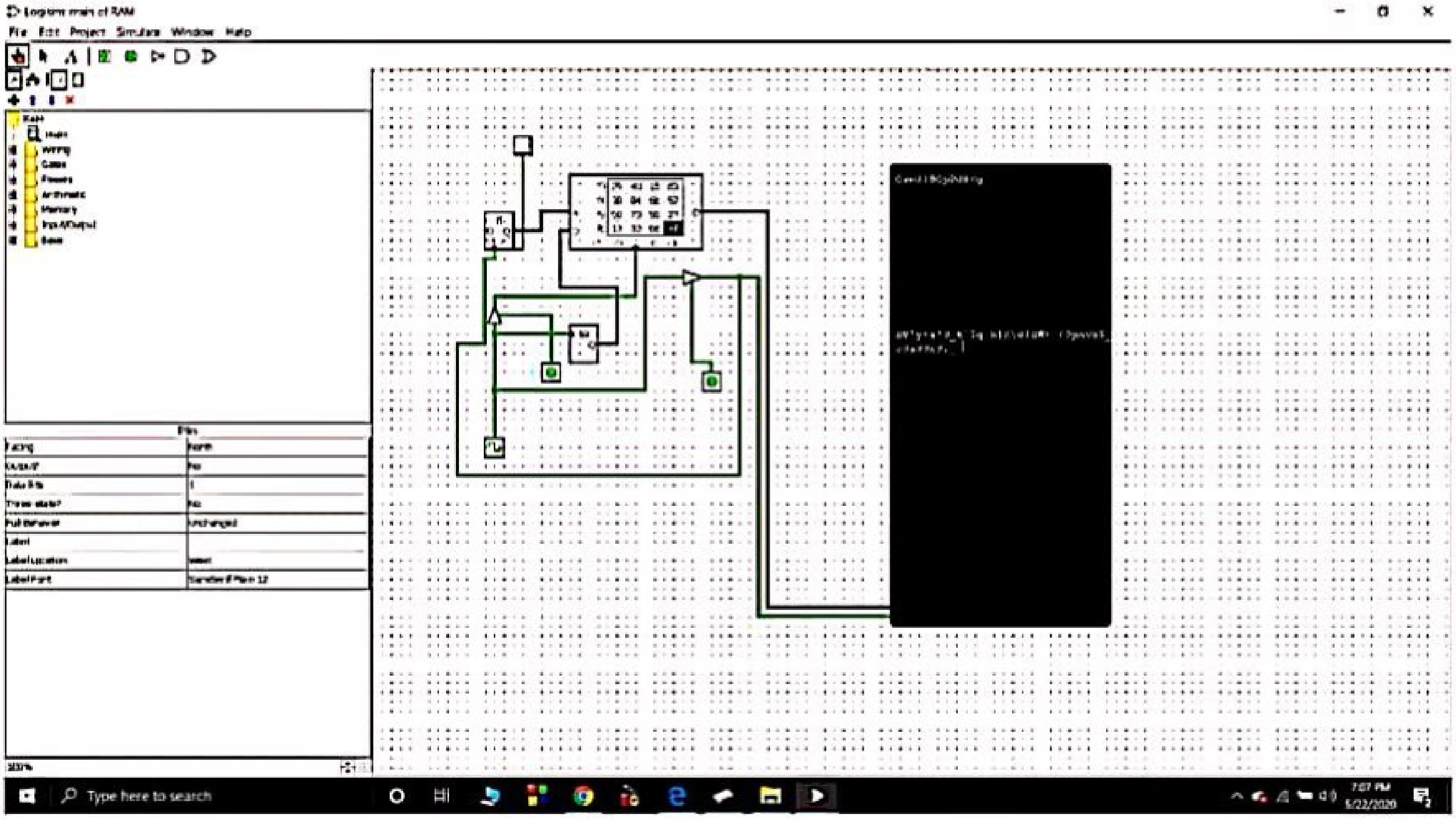
Activity to be performed by students:

List out the steps in designing memory system

Observations and Snapshots:

1 Add a RAM with seperate load and store selected
2 Add a counter and convert Q to A of the RAM
3 Add a Controller buffer and connect its O/P to RAM
4 Add a clock and connect to the I/P of the buffer
5 Add a TTY unit with 32 rows and columns.
Make the connections with RAM
6 Add a 7 bit random number generator, connect Q to D
7 Add another controlled buffer connect it to TTY Also add an I/P pin to the buffer
8 connect the O/P of the second buffer to the counter
9 connect a button to the counter

Snapshot :



Pin

Facing	North
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12

Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)

Department of CSE

Programme: B.E **Term:**
Jan to May 2019
Course: Computer Organization
Course Code: CS45

Activity VII: To simulate advantages of using pipeline technique in executing a program.

Name: Mohit Raj Soni	**Marks: /10**	**Date:24-05-2020**
USN:1MS18CS074	**Signature of the Faculty:**	

Objective: To learn and analyze the performance of the CPU by overlapping of instructions using CPUOS-SIM simulator.

Simulator Used: CPUOS-SIM is a software development environment for the simulation of simple computers. It was developed by Dale Skrien to help users to understand computer architectures.

Modern CPU's contain several semi-independent circuits involved in decoding and executing each machine instruction. Separate circuit elements perform each of these typical steps:

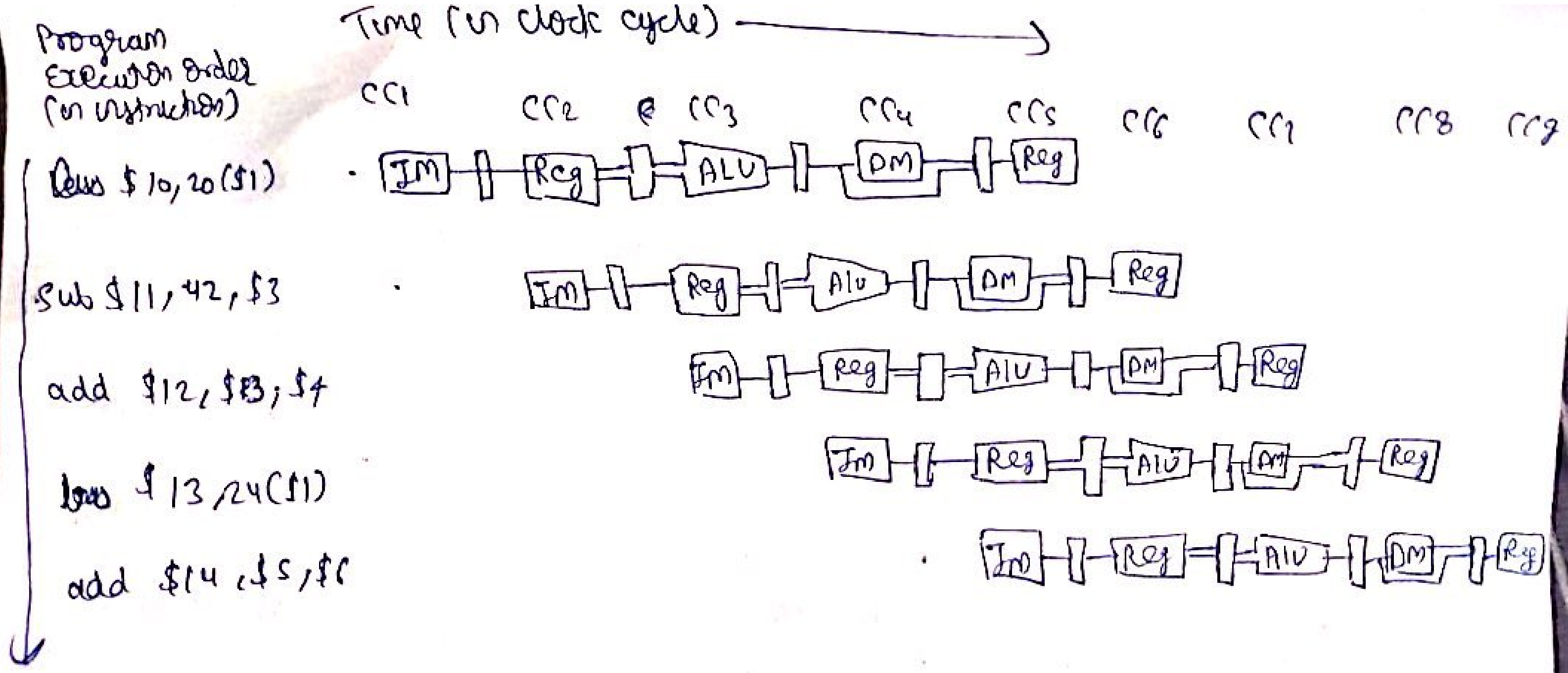
- Fetch the next instruction from memory into an internal CPU register.
- Decode the instruction to determine which function sub-circuits it requires.
- Read any input operands required from high-speed registers or directly from memory.
- Execute the operation using the selected sub-circuits.
- Write any output results to high-speed registers or directly to memory.

Separate sections of the CPU circuitry are used for each of these steps. This allows these circuit sections to be arranged into a sequential pipeline, with the output of one step feeding into the next step.

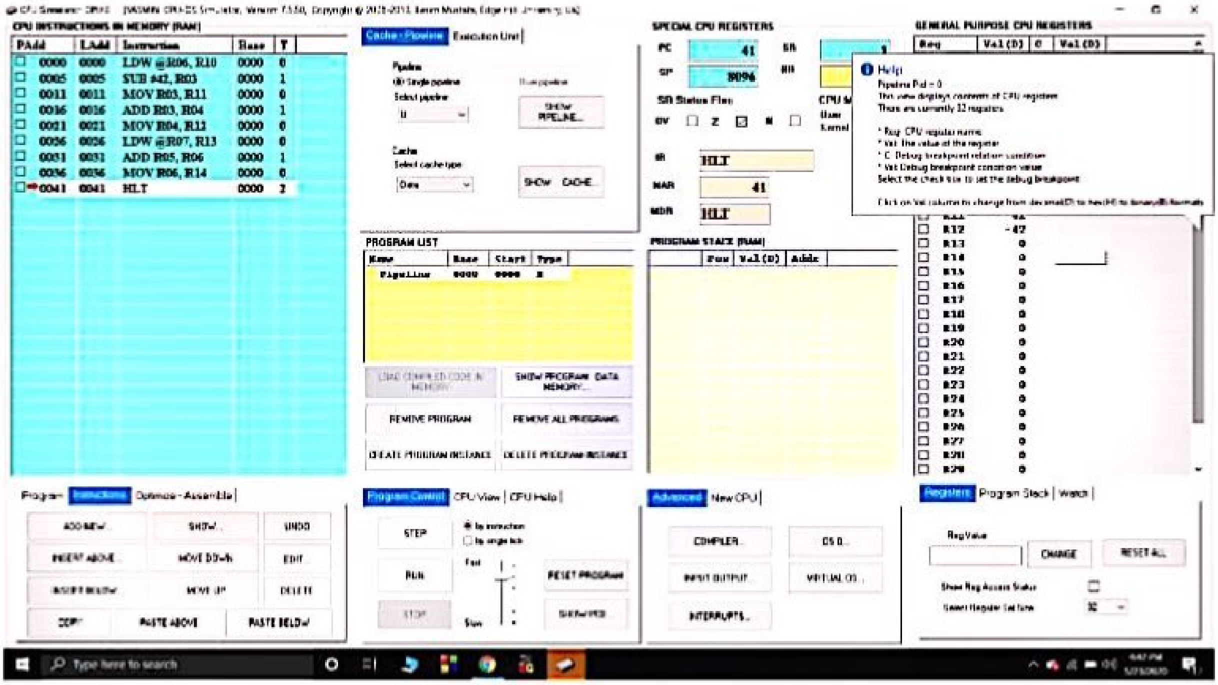
Program Execution order (in instructions)

Time (in clock cycles) →

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9
less $10, 20($1)	Instruction fetch	Instruction decode	Execution	Data access	write back				
sub $11 $2, $3		Instruction fetch	Instruction decode	Execution	Data access	write back			
add $12, $3, $4			Instruction fetch	Instruction decode	Execution	Data access	write back		
less $13, 24($1)				Instruction fetch	Instruction decode	Execution	Data access	write back	
add $14, $5, $6					Instruction fetch	Instruction decode	Execution	Data access	write back


Program Execution order (in instructions)
Time (in clock cycle)
CC1
CC2
CC3
CC4
CC5
CC6
CC7
CC8
CC9
lw $10, 20($1)
sub $11, $2, $3
add $12, $3, $4
lw $13, 24($1)
add $14, $5, $6
IM
Reg
ALU
DM
Reg

Observations and Snapshots: Take the snap shot of CPU statistics and pipeline design.



CPU INSTRUCTIONS IN MEMORY (RAM)

PAdd	LAdd	Instruction	Base	T
0000	0000	LDW @R06, R10	0000	0
0005	0005	SUB #42, R03	0000	1
0011	0011	MOV R03, R11	0000	0
0016	0016	ADD R03, R04	0000	1
0021	0021	MOV R04, R12	0000	0
0026	0026	LDW @R07, R13	0000	0
0031	0031	ADD R05, R06	0000	1
0036	0036	MOV R06, R14	0000	0
0041	0041	HLT	0000	2

PROGRAM LIST

Name	Base	Start	Type
Pipeline	0000	0000	2

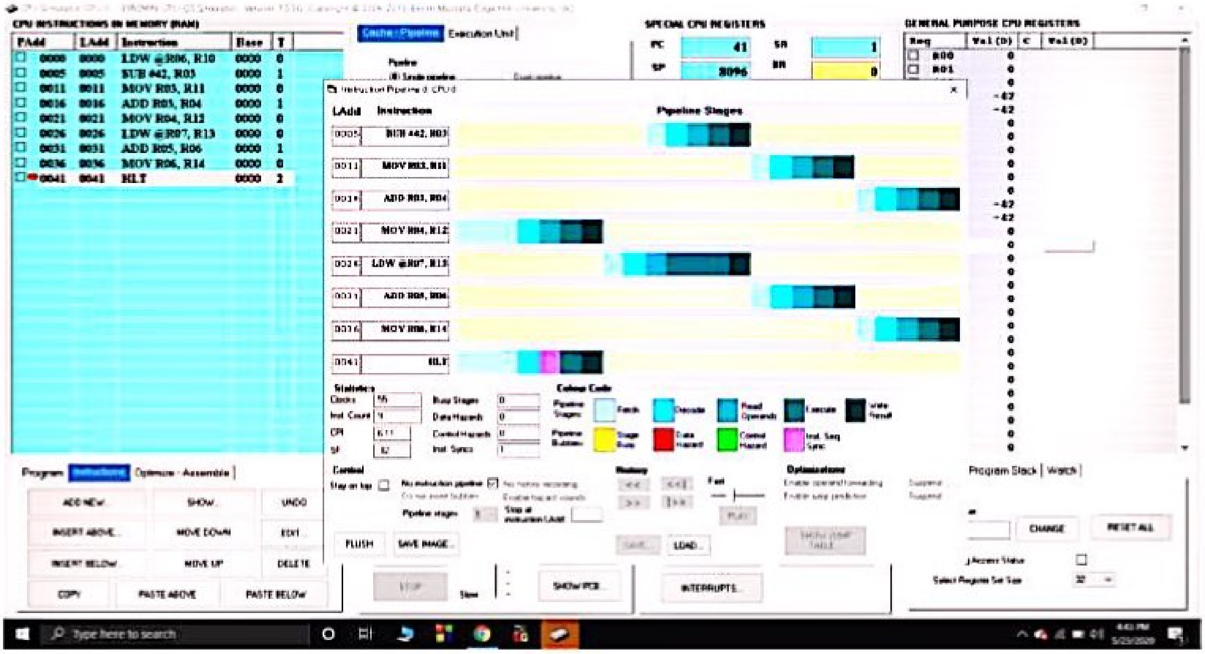
SPECIAL CPU REGISTERS

PC 41 SP 8096

IR HLT

MAR 41

MDR HLT



Instruction Pipeline d CPU0

LAdd	Instruction
0005	SUB #42, R03
0011	MOV R03, R11
0016	ADD R03, R04
0021	MOV R04, R12
0026	LDW @R07, R13
0031	ADD R05, R06
0036	MOV R06, R14
0041	HLT

Statistics

Clocks 55

Inst Count 9

CPI 6.11

Busy Stages 0

Data Hazards 0

Control Hazards 0

Inst. Syncs 1

Scanned with CamScanner