Chittagong University of Engineering & Technology

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

NAME OF THE EXPERIMENT / REPORT

Schematic Modeling and Implementation of Modified SAP Architecture in Logisim - 2

COURSE NO	**:**	ETE 404
COURSE TITLE	**:**	VLSI Technology Sessional
EXPERIMENT NO.	**:**	02
DATE OF EXP.	**:**	17.05.2023
DATE OF SUBMISSION	**:**	22.05.2023

R E M A R K S

NAME	**:**	Touhidul Islam Khan
ROLL NO.	**:**	1808034
LEVEL	**:**	4
TERM	**:**	I
SECTION	**:**	
GROUP	**:**	G-2

Objectives:

- To familiarize with the BUS in a microprocessor system.
- To familiarize with addressing and RAM.
- To familiarize with program execution cycle.

Required Software:

1. Logisim

Design:

Decoder

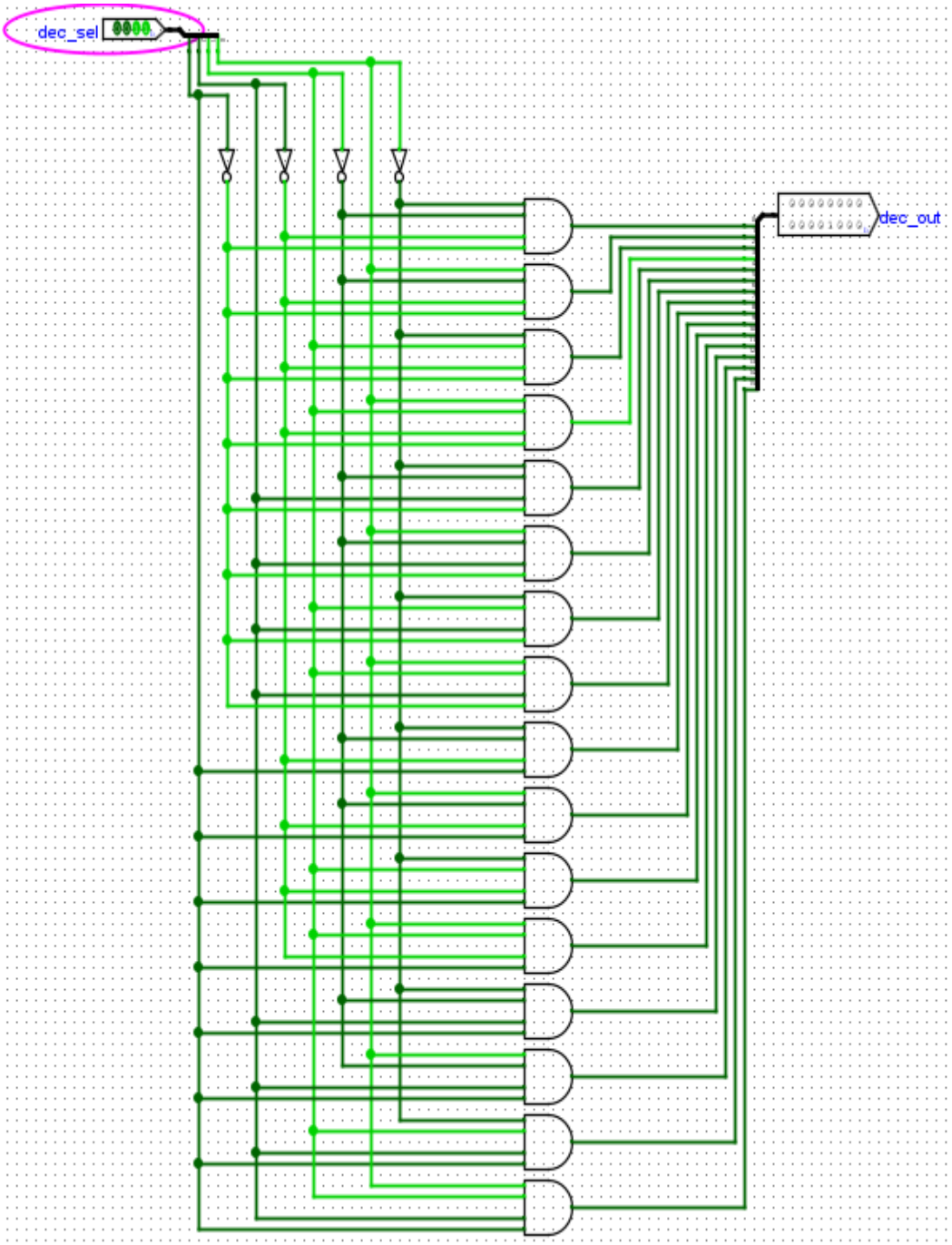

Figure No.01: Schematic of 4 to 16 decoder

A combinational circuit that has *n* number of select bits and a maximum of 2^n number off output lines is known as decoder. Depending on the combination of the input bits, only one of the outputs of the decoder was active. In out design a decoder with 4 selection lines and 16 output lines, i.e., 4 to 16 decoder was used. The 16 outputs were used to access 16 different memory address in the RAM.

Random Access Memory (RAM):

A combination of general-purpose register and AND gate can form a single SRAM cell, which can be used to design our RAM. The 'cs' or chip select pin controls the read and write enable. That is, without 'cs' enabled no read or write operation can happen even if 'wr_en' and 'rd_en' is enabled.

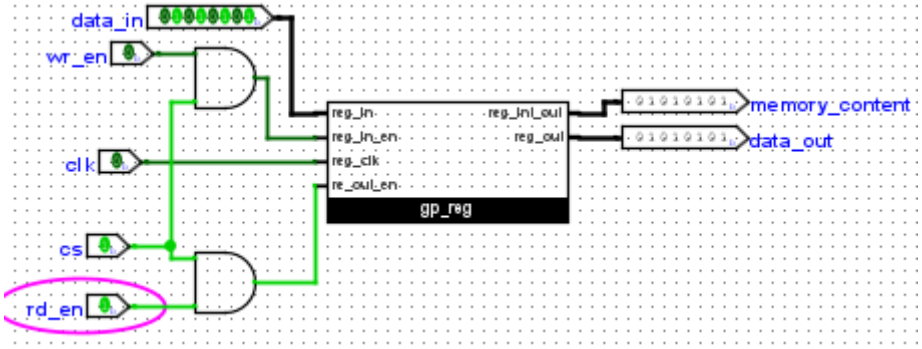

Figure No.02: Schematic of single SRAM cell

RAM is a bank of memory available to the processor to be used while executing a program. Any program that needs to be executed has to be stored in the RAM first.

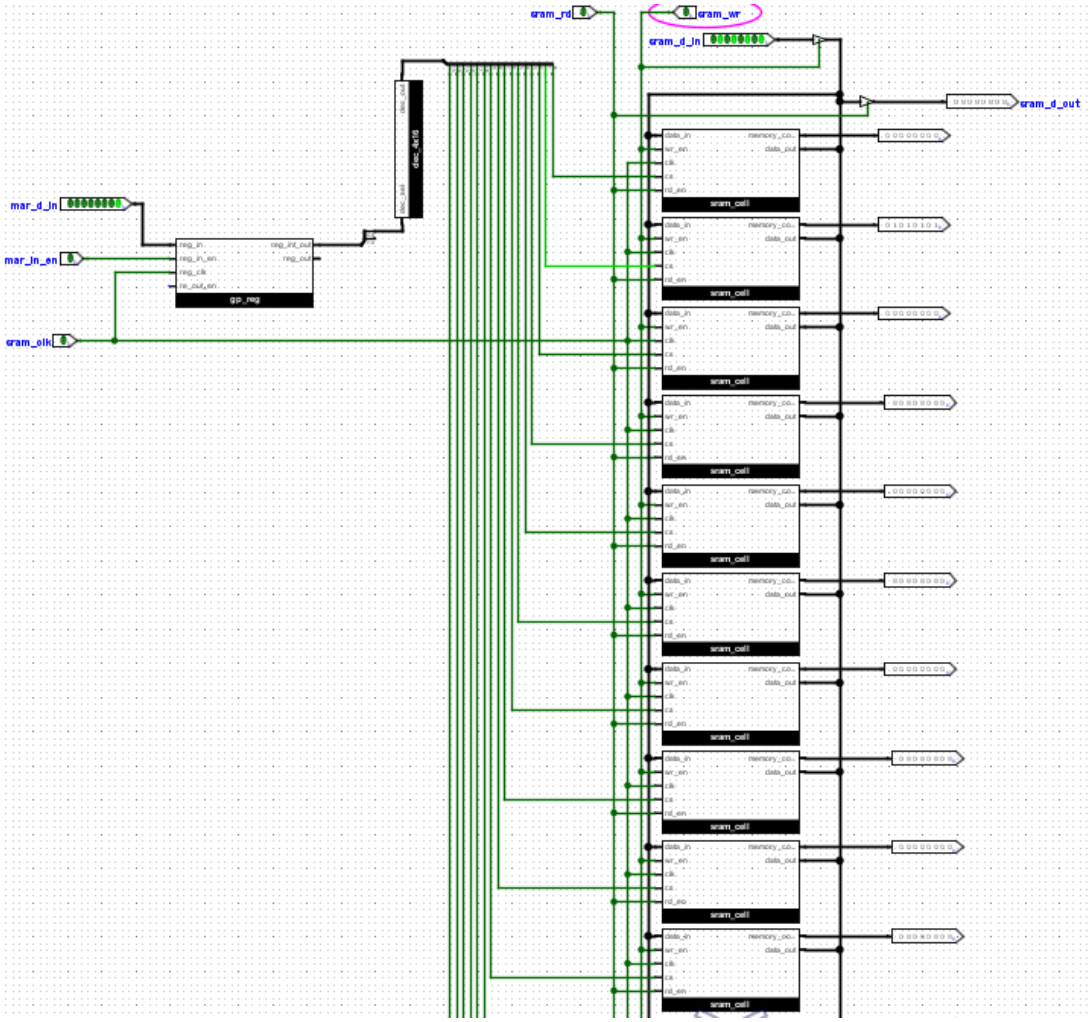

Figure No.03: Schematic of RAM part-1.

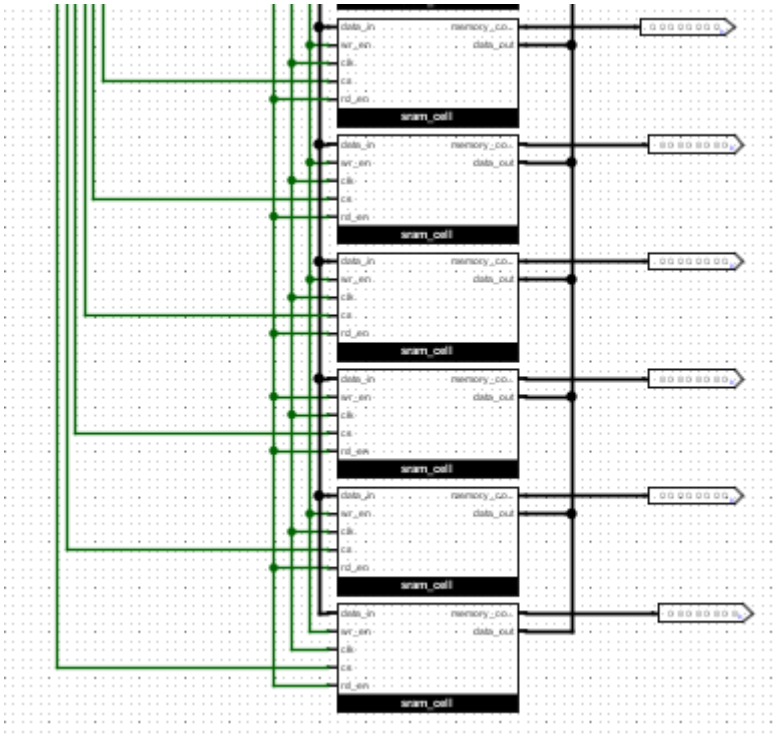

Figure No.04: Schematic of RAM part-2.

Instruction Register:

The instruction register holds the instruction that is to be executed. This separates the BUS information into Opcode, the higher 4bits and the Operands or address, the lower 4bits. It is simply built using a general-purpose register. The 'ins_reg_out_en' controls the address or operand to be outputted.

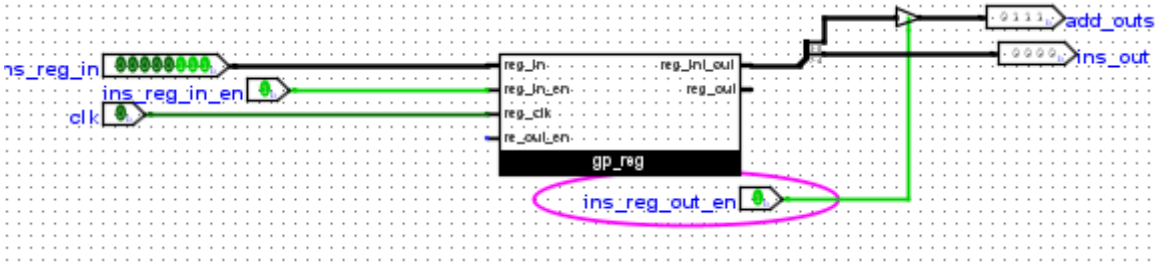

Figure No.05: Schematic of Instruction Register.

Central Processing Unit (CPU):

The CPU is the heart of any programmable device. In our design we have built a simple CPU with RAM, registers, counter and ALU. The ALU only can perform add and subtract. The program counter keeps track of the next instruction. The RAM stores the holds the address or data that is to be used. The instruction register holds opcode and operand and address of the instruction to be executed.

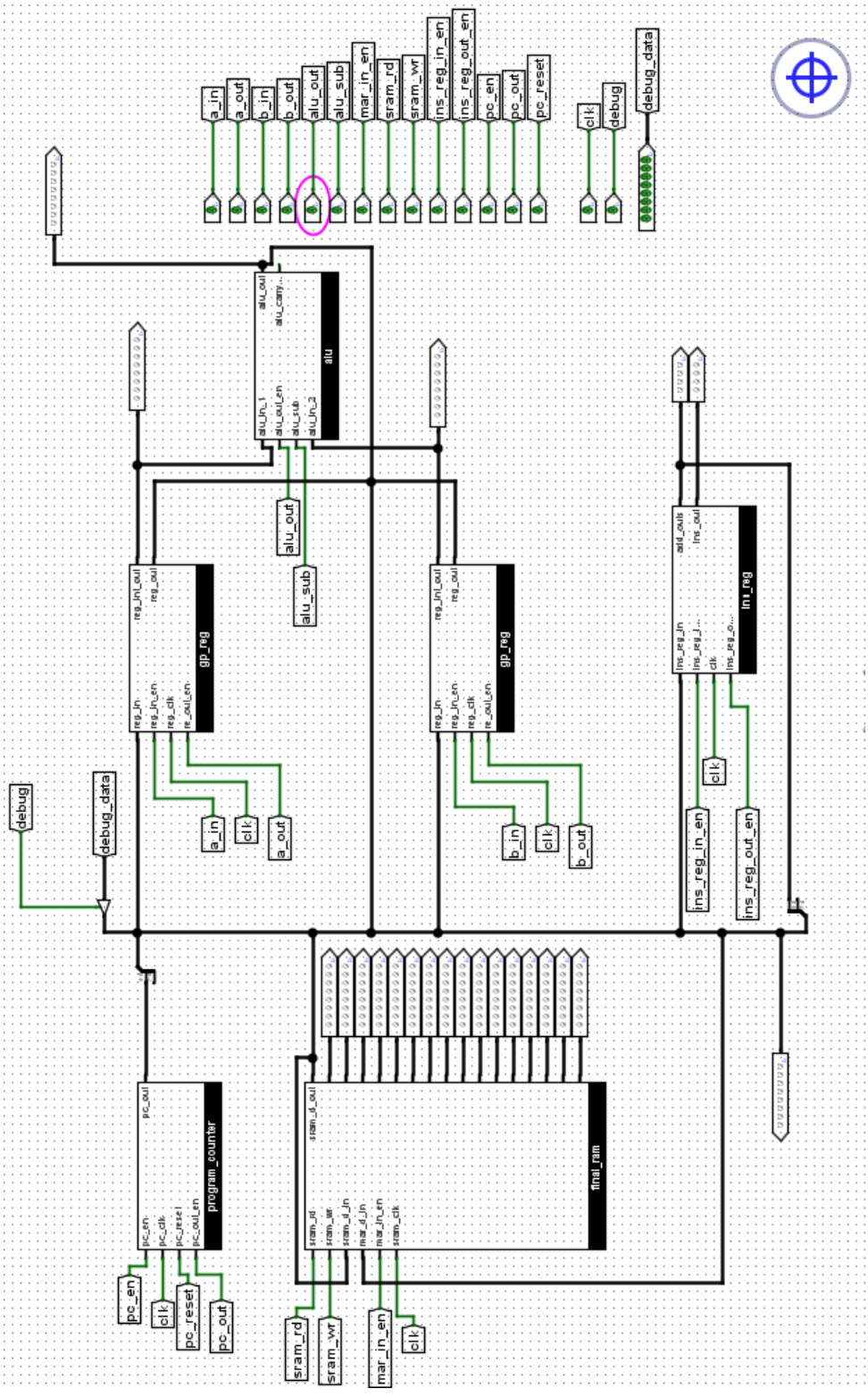

Figure No.06: Schematic of Simple CPU

Home Task:

In the lab we performed the execution of LDA or load A instruction following the complete instruction cycle. The instruction cycle is also known as fetch-decode-execute cycle.

At first the memory address was stored in the MAR (memory address register) and this address was used to specify the RAM address for the data to be stored. The data was stored by enabling the 'sram_wr'. This address in the MAR was also used during the Load A operation. At the memory 0000 0000 of RAM 0001 1010 is stored. This works as the address of the RAM slot where our data 0000 0111 is stored. This data is then loaded to register A.

For performing load B operation, the following steps are followed:

Fetch:

- T1: Toggle pc_out & mar_in_en and give a clock pulse. Address of next instruction to be executed is sent to MAR from PC. Turn off the control pins.
- T2: Toggle sram_rd & ins_reg_in_en and give a clock pulse. Check the IR loaded with the Opcode and operand. Turn off the control pins.
- T3: Toggle pc_en and give a clock pulse. This increments the counters value to be 0010 or location of the next line of code. Turn of the control pin.

Execute:

- T1: Toggle ins_reg_out_en & mar_in_en and give a clock pulse. This saves the address to be fetched into the MAR. Turn off the control pins.
- T2: Toggle sram_rd & b_in and give a clock pulse. This reads the memory contents of address 0000 1011 which is 0000 1001 and saves it in register B. Turn off the control pins.
- • T3: This state is unused by LDA.

Now to store the output of the ALU we have to perform the following steps:

- Toggle pc_out & mar_in_en and give a clock pulse. Address of next instruction to be executed is sent to MAR from PC. Turn off the control pins.
- Toggle sram_rd & ins_reg_in_en and give a clock pulse. Check the IR loaded with the Opcode and operand. Turn off the control pins.
- Toggle ins_reg_out_en & mar_in_en and give a clock pulse. Turn off the control pins.
- Toggle sram_wr & alu_out and give a clock pulse. This writes the contents of ALU output which is 00010000 and saves it in RAM memory 0000 1100. Turn off the control pins.

Similarly, the register B was loaded with data 0000 1001 from memory address 0000 1011. The ALU then performed addition and the result 0001 0000 was stored at memory location 0000 1100.

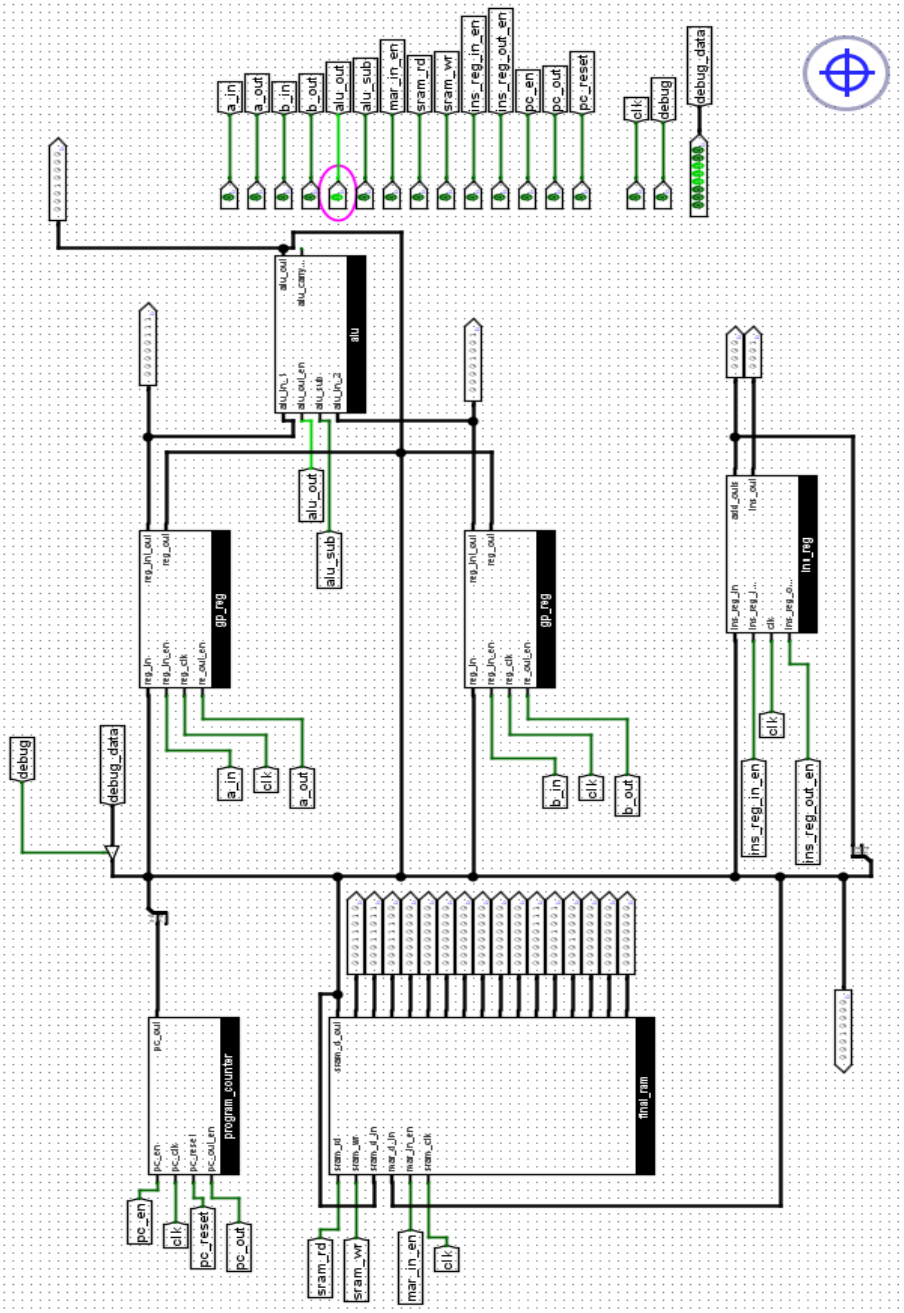

Figure No.07: Output after Load A, Load B and ALU operation

Discussion:

1. Design of decoder, RAM, instruction register and CPU was learnt and implemented in this experiment.
2. The decoder was designed to convert 4 inputs to 16 output lines so that memory address of the ram could be selected.
3. Connection of the wires were made carefully so that they do not touch each other and form cross connection.
4. The instruction register held the address of the next memory address where the data to be stored.
5. One execution cycle was performed to check if the CPU built works fine or not.
6. Addition between values stored in two different registers and the sum was again stored at a memory location of the RAM.
7. The instruction set can be designed differently depending on the tasks to be performed.