# 1 Instructions set:

| Instruction | Opcode | Explanation |
|---|---|---|
| LDA | 0001 | It stores the data located in RAM into Accumulator register |
| LDB | 0110 | It stores the data located in RAM into register B |
| STR | 0101 | It stores output of arithmetic logic unit into RAM |
| ADD | 0010 | It adds the register A and register B and the added value is seen from ALU output. |
| SUB | 0011 | It subtracts register B from register A and the subtracted value is seen from ALU output. |
| AC_OUT | 0100 | It loads the value of register A into output register. |
| HLT | 1111 | Ih halts the program. |

# 2 Control Sequencer:

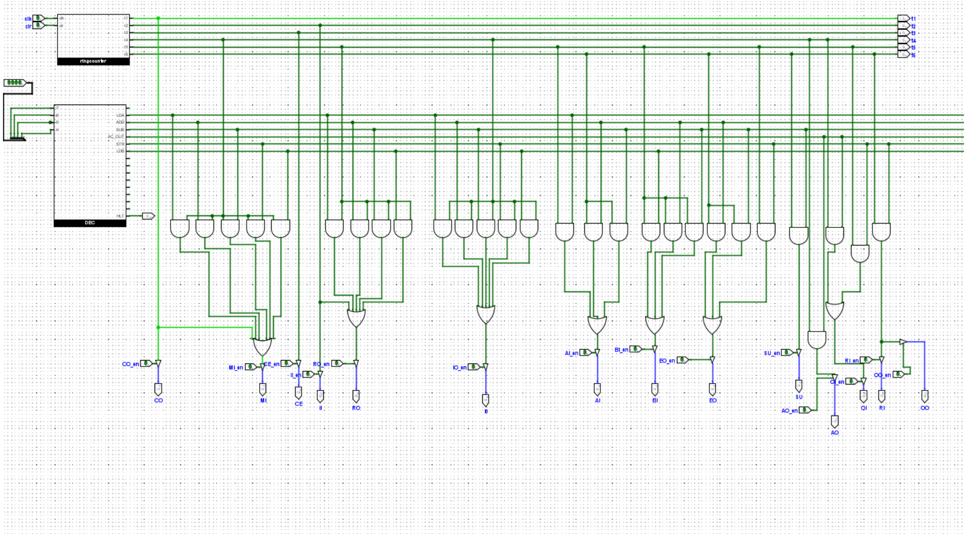

Figure 1: Final Layout of Control Sequencer

# 3 Pin Explanation:

| Pin | Explanation |
|---|---|
| IO | Instruction register output enable |
| II | Instruction register input enable |
| AI | Register A input enable |
| BI | Register B input enable |
| AO | Register A output enable |
| OO | Output Register output enable |
| OI | Output Register input enable |
| EO | Alu output enable |
| SU | Subtraction enable |
| RI | Ram input(sram wr) |
| RO | Ram output(sram rd) |
| CO | Program counter output enable |
| CE | Program counter enable |
| MI | Mar in enable |

# 4 Layout:

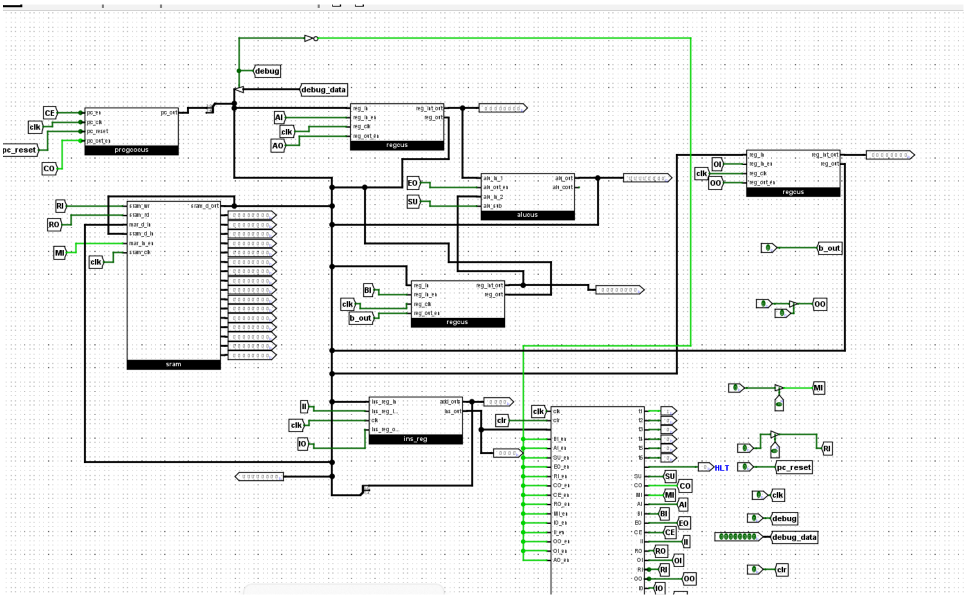

Figure 2: Final layout

### Data Load:LDA 2(DATA 3)

- Turing on debug pin and resetting the program counter the program was started.When debug data was given 0000 0000 bus loaded with the data .When mar_in_en was turned on and clock pulse were given, the data loaded on the memory address register then turned off mar_in_en.RAM knew the work would be done on 0000 0000 address.

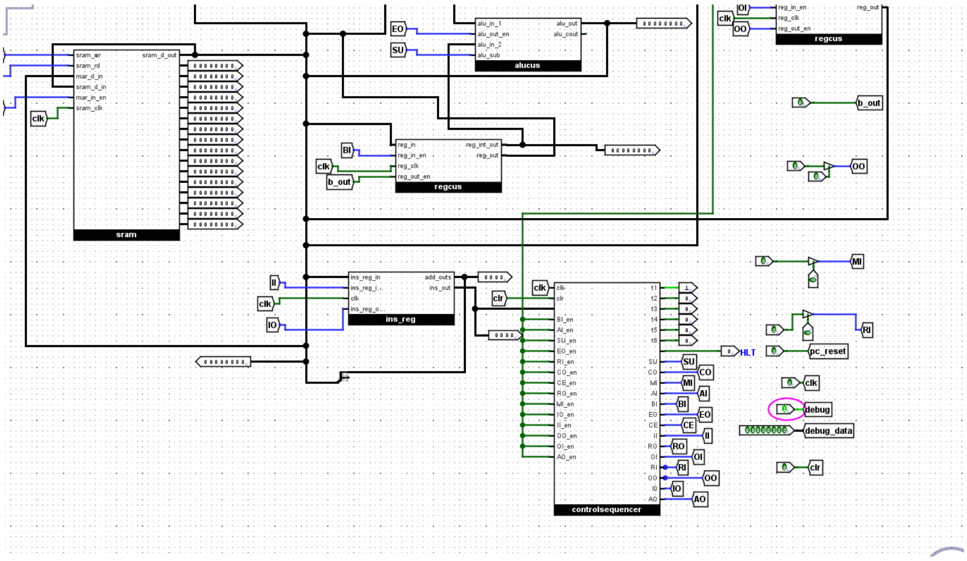

Figure 3: Debugging initialize

- Debug data was given 00010010(LDA 2) bus loaded with the data and sram_wr was turned on and clock pulse were given then the data 00010010 was saved on memory address 00000000 then sram_wr were turned off.That means at first which memory address would save the data were given then the saved data were given. Here 0001 was LDA and 0010 were memory address.

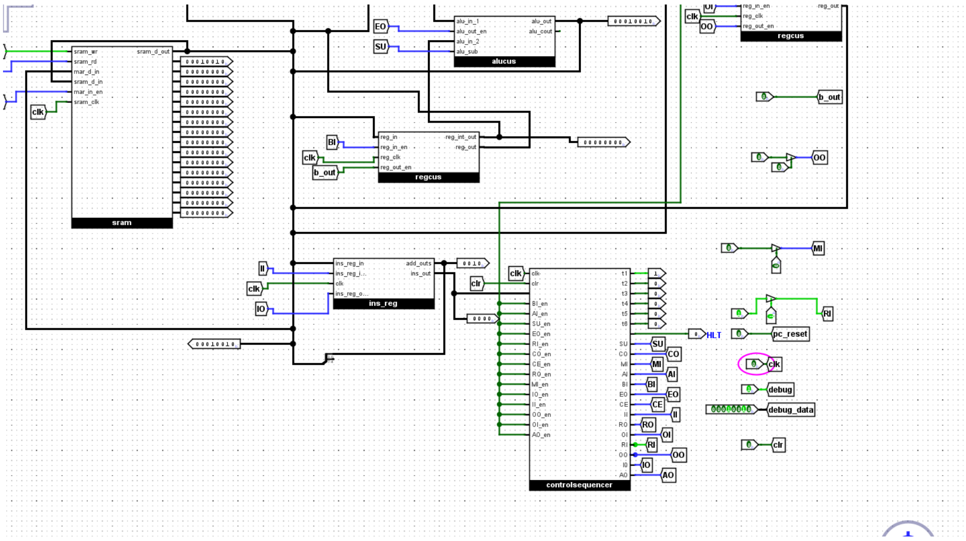

Figure 4: LDA 2 loaded

- When debug data was given 00000010 bus loaded with the data .When mar_in_en was turned on and clock pulse given,the data loaded on the memory address register then turned off mar_in_en.RAM knew the work would be done on 00000010 address.

- Debug data was given 00000011 bus loaded with the data and sram_wr was turned on and clock pulse were given then the data 00001111 was saved on memory address 00000010 then sram_wr were turned off.Then debug pin were turned off.

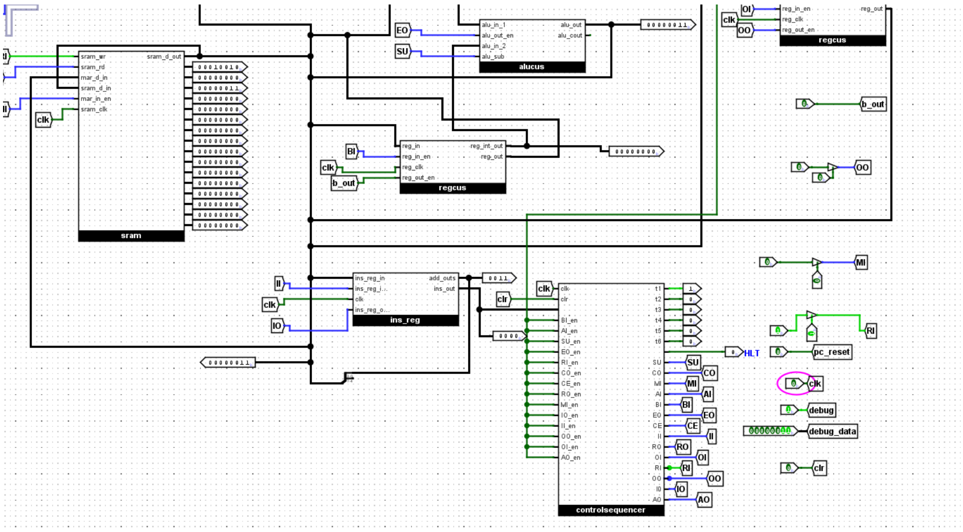

Figure 5: Data 3(00000011)loaded on address 2

**Fetch:**

- The program counter was reset to 0000.Clr pin were turned On.
- **T1:**Giving clock pulse,CO MI on
- **T2:**Giving clock pulse,RO II on
- **T3:**Giving clock pulse,CE on

**Decode:**

- This state was not sequential logic so no T states or clock pulse were required and here only combinational logic were executed.

**Execute:**

- **T4:**Giving clock pulse,IO MI on
- **T5:** Giving clock pulse,AI RO on.Accumulator register were loaded with 00000011.

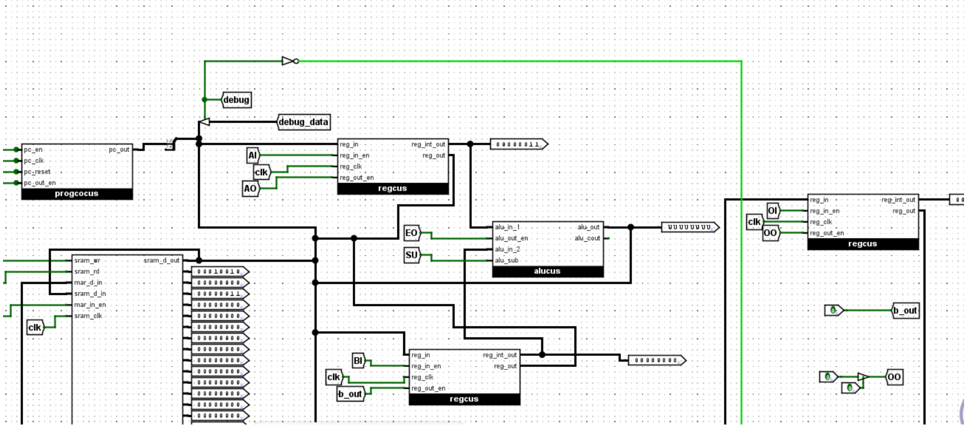

Figure 6: Accumulator Loaded with data

### Data Load:LDB 1(DATA 2)

- Same as LDA 2 .Here the instruction 0110 0001(0110 is opcode of LDB and 0001 memory address) were given to load data 0000 0010 (saved on memory address 0000 0001 )into register B. Here instead of AI ,BI were on at T5 state to save data on register B.At T6 state EO were ON so alu output were seen.

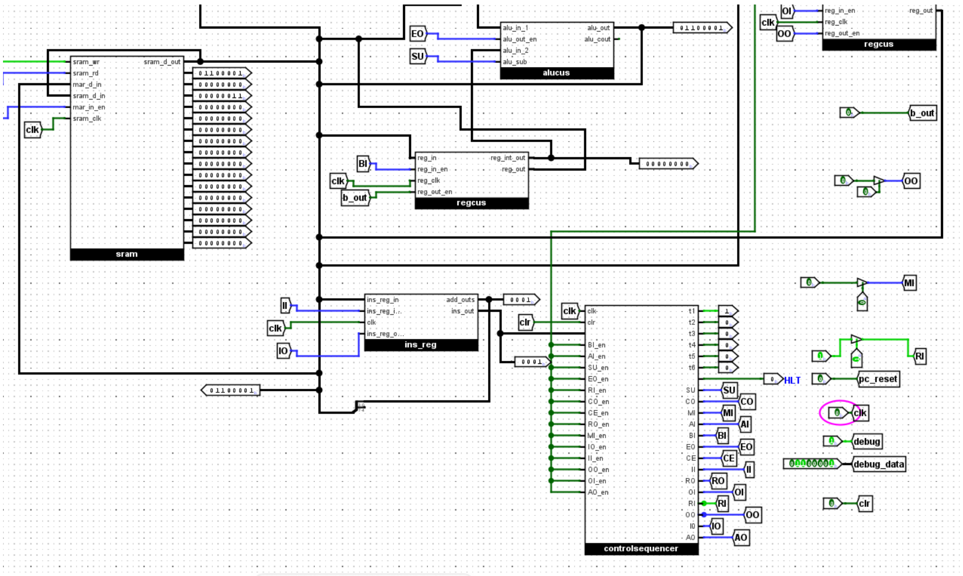

Figure 7: LDB 1 loaded

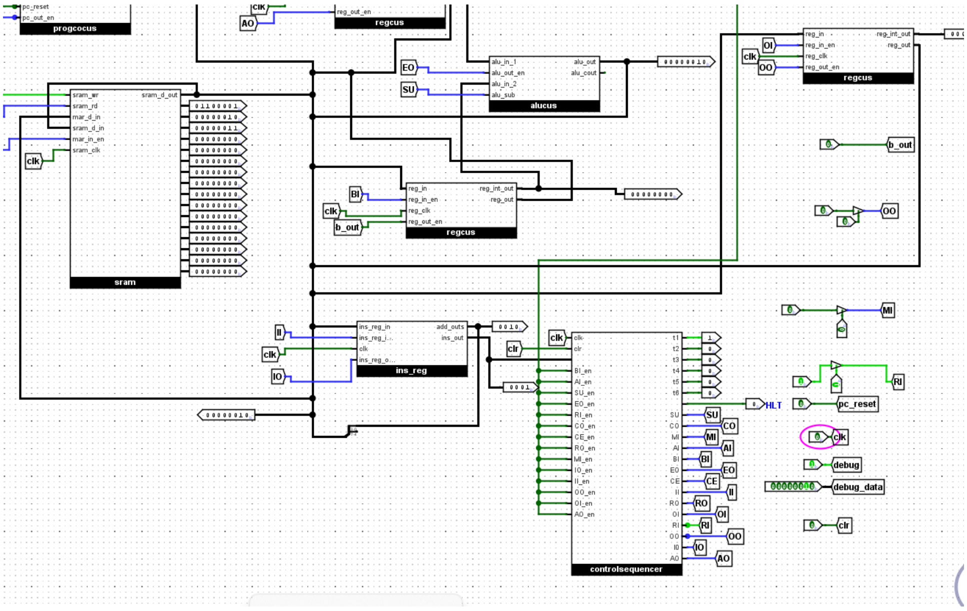

Figure 8: Data 2(00000010)loaded on address 1

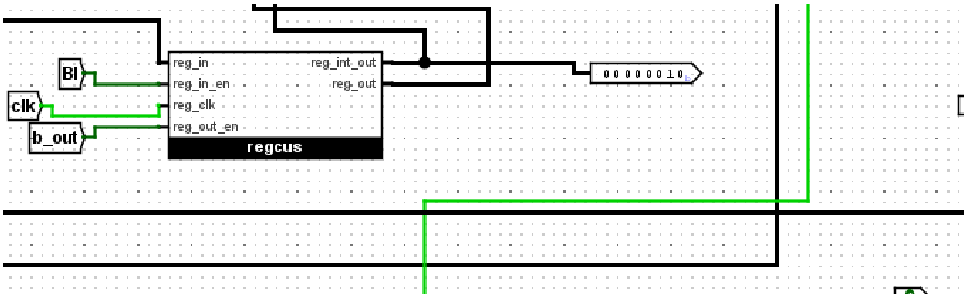

Figure 9: Register B Loaded with data

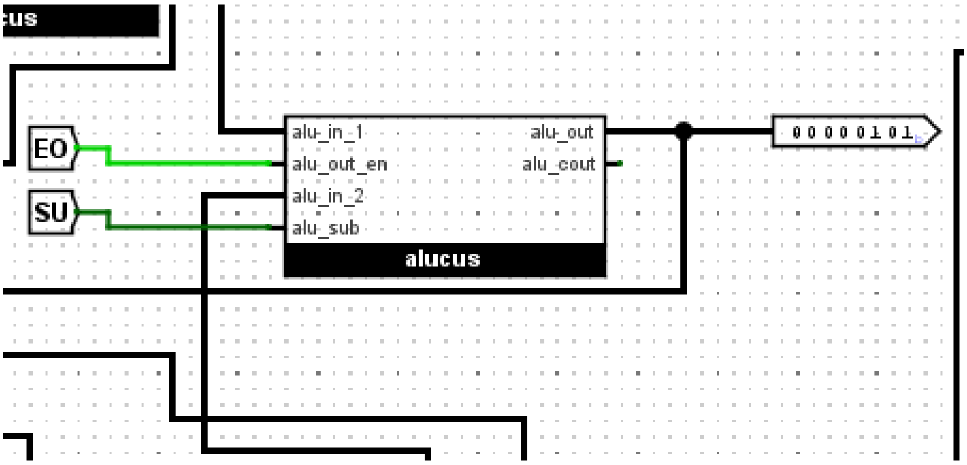

Figure 10: ALU ouput

**ALU DATA STORE:**

- Debug pin were truned on debug pin were set 0000 0000. When mar_in_en was turned on and clock pulse were given, the data loaded on the memory address register then turned off mar_in_en.RAM knew the work will be done on 0000 0000.
- Debug data was given 0101000 bus loaded with the data and sram_wr was turned on and clock pulse were given then the data 01010000 was saved on memory address 00000000 then sram_wr were turned off.Here 0101 was opcode of STR.

**Fetch:**

- The program counter was reset to 0000.Clr pin were turned On.
- **T1:**Giving clock pulse,CO MI on
- **T2:**Giving clock pulse,RO II on
- **T3:**Giving clock pulse,CE on

**Decode:**

- This state was not sequential logic so no T states or clock pulse were required and here only combinational logic were executed.

**Execute:**

- **T4:**Giving clock pulse,IO MI on
- **T5:** Giving clock pulse,EO OI on.Alu output were loaded on output register

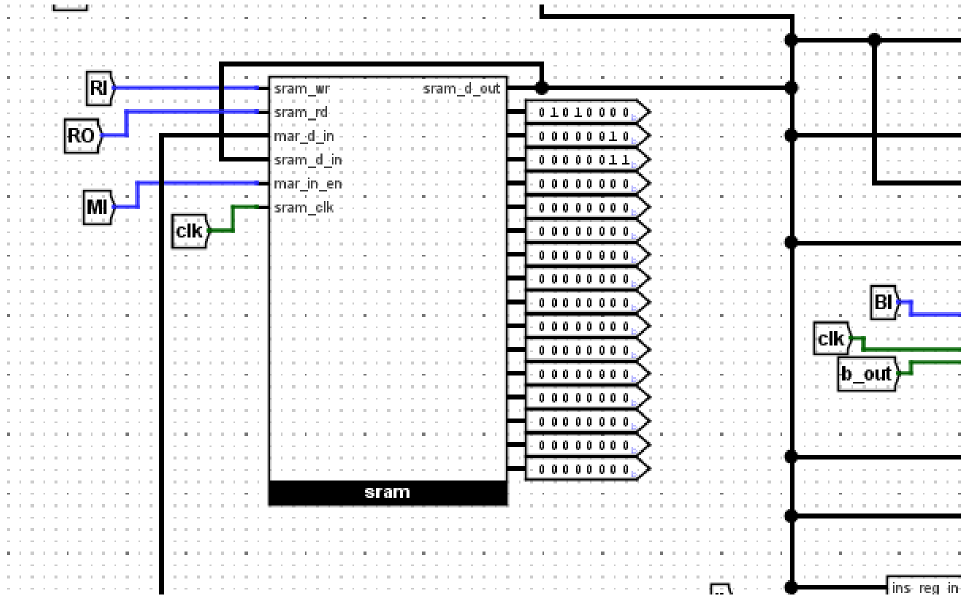

Figure 11: STR opcode loaded on 0 adress

- **T6:** Giving clock pulse,OO RI on so output register output enabled and data loaded on memory on 0000 0000 location

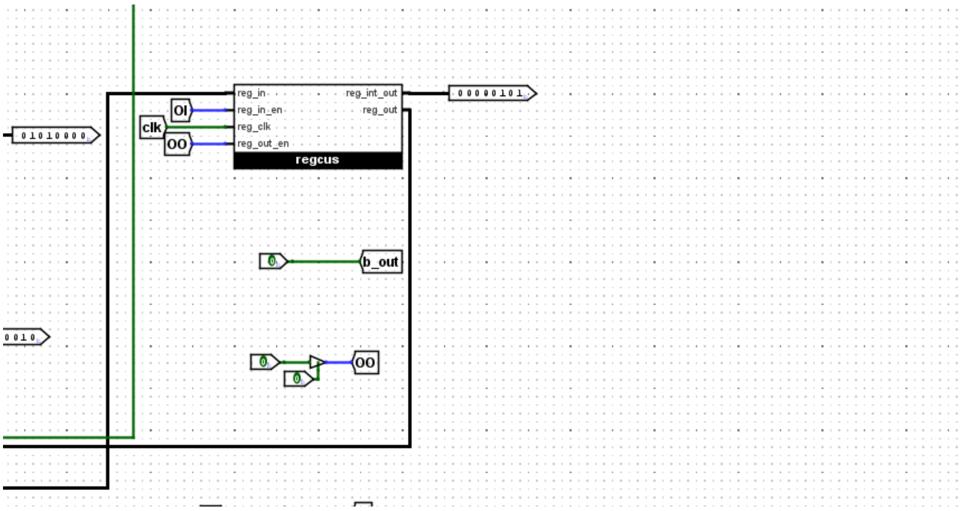

Figure 12: Output Register Loaded with ALU output

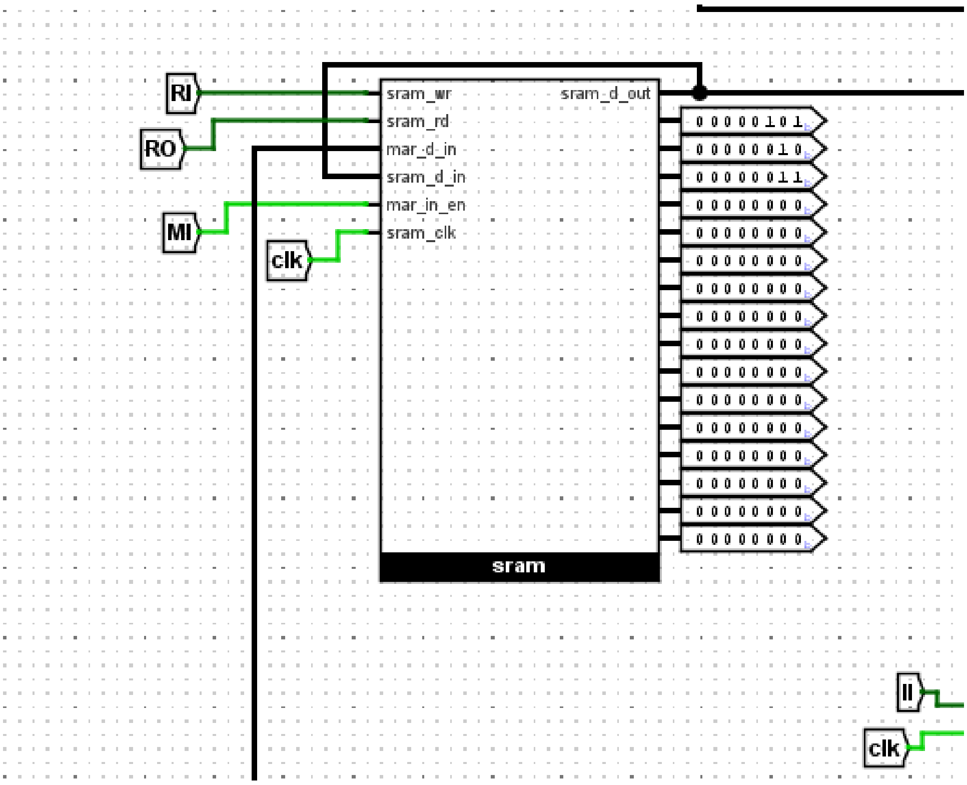

Figure 13: Alu output stored on Ram on address 0

**ADD:**

- Debug pin were truned on debug pin were set 0000 0000. When mar_in_en was turned on and clock pulse were given, the data loaded on the memory address register then turned off mar_in_en.RAM knew the work will be done on 0000 0000.
- Debug data was given 00100100 bus loaded with the data and sram_wr was turned on and clock pulse were given then the data 00100100 was saved on memory address 00000000 then sram_wr were turned off.Here 0101 was opcode of ADD.

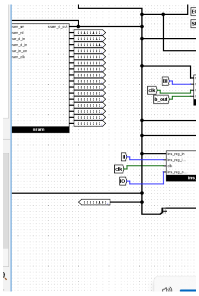

Figure 14: ADD opcode loaded on 0 adress

- Debug pin were truned on debug pin were set 0000 0100. When mar_in_en was turned on and clock pulse were given, the data loaded on the memory address register then turned off mar_in_en.RAM knew the work will be done on 0000 0100.
- Debug data was given 00000100 bus loaded with the data and sram_wr was turned on and clock pulse were given then the data 00000100 was saved on memory address 00000100 then sram_wr were turned off.

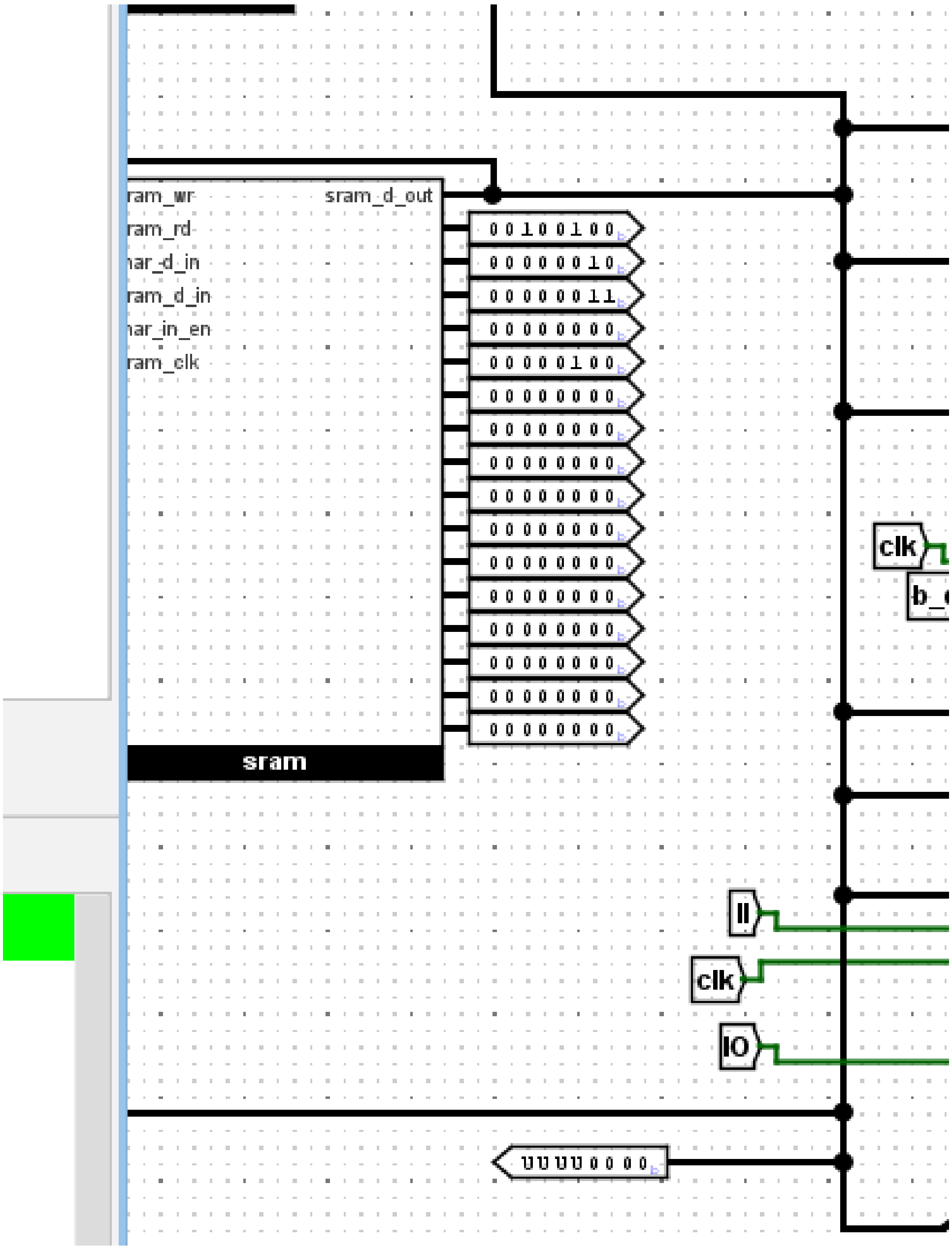

Figure 15: Data loaded on adress 4

**Fetch:**

- The program counter was reset to 0000.Clr pin were turned On.
- **T1:**Giving clock pulse,CO MI on
- **T2:**Giving clock pulse,RO II on
- **T3:**Giving clock pulse,CE on

**Decode:**

- This state was not sequential logic so no T states or clock pulse were required and here only combinational logic were executed.

**Execute:**

- **T4:**Giving clock pulse,IO MI on
- **T5:** Giving clock pulse,RO BI on.Data loaded on register B.
- **T6:** Giving clock pulse,EO AI on so ALU output saved on register A.

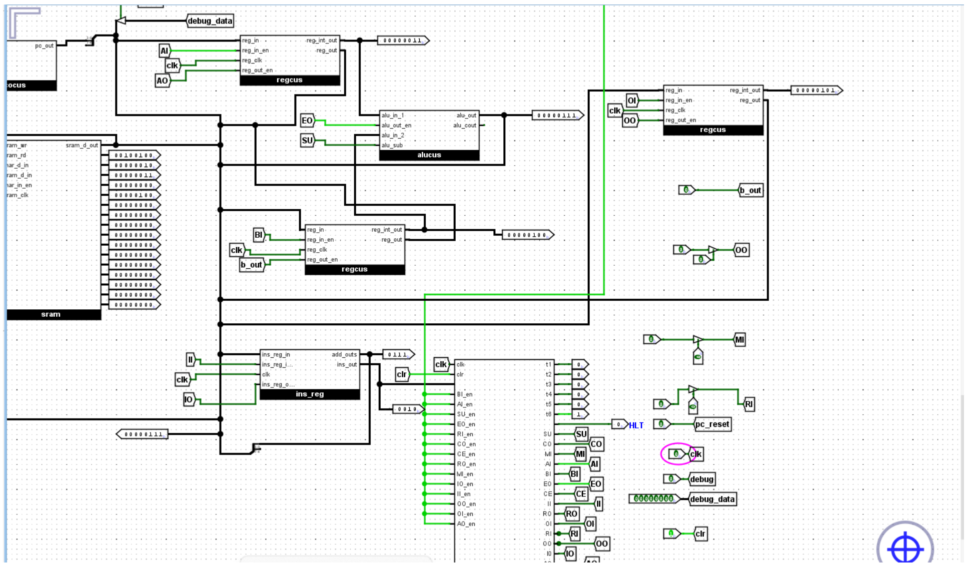

Figure 16: ALU output

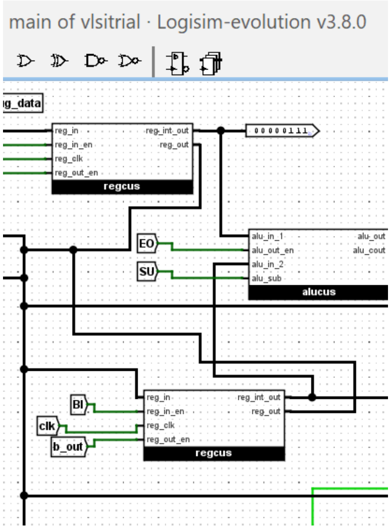
main of vlsitrial · Logisim-evolution v3.8.0

Figure 17: Register A loaded with Alu output