# Simple as Possible Computer

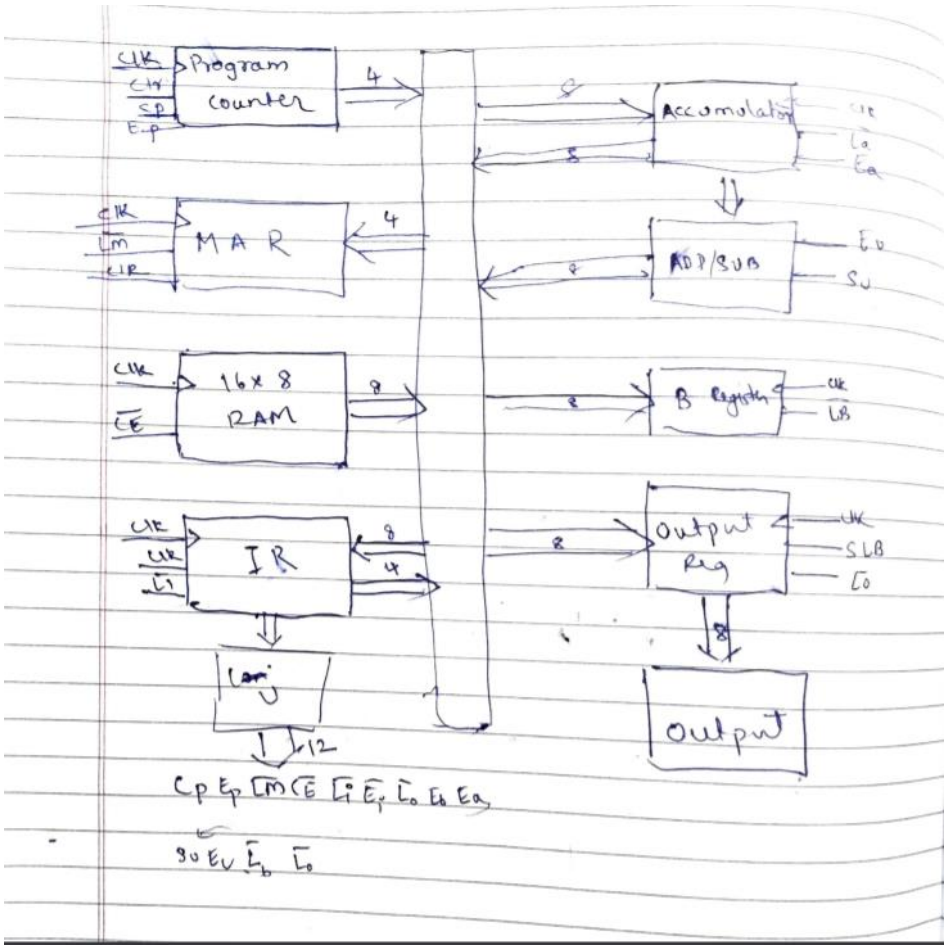
121901050

KUNAL KESHAV DAMAME | EE 2080

## Description

We have to develop a 8-bit micro computer using logisim which can perform basic addition, subtraction and display the output in binary form.

## Components required and Block Diagram



- **Accumulator**

Accumulator is a device which stores the result of add/sub operations. You can also load the data from RAM into the accumulator. It receives two control signals

1. ~LA

2. EA

> **Counter**

It is a device which produces a sequence of integers in lexographic order, in this case it produces numbers from 0-6 to point to locations in RAM. It has 2 control signal

1. EP- manages the connection b/w counter and bus

2. CP-The value is incremented only if cp is high

- **Memory Address register**

It stores and provides the RAM with the address of instruction and data to be worked on. It has 1 control signal

1. ~LM: connection b/w bus and MAR is enabled when this is low.

> **Instruction Register**

Instruction register receives a 8 bit data, in which first 4 MSB are about the type of operation and other 4 are the address of data from RAM on which the operation has to be performed. It sends this 4 bit back to the MAR, and instruction about operation is send to Control unit. The IS receives 2 control signals.

1. ~LI: The input connection with bus is enabled when low.
2. ~EI: The output connection with bus is enabled when low

- **ALU unit**
  This unit takes care of doing the addition and subtraction operations and also storing the data from this operations. It has 2 main units, B register and accumulator.

    - **Accumulator**
      It is a register which stores the value of operation carried out by our ALU.

- **B-register**
  It is a register which stores the value of data on which the operation is supposed to be performed

This ALU has 4 control signals

1. ~LA: The input connection b/w bus and Accumulator Is enabled when low

2. ~LB: The connection b/w b-register and bus is enabled

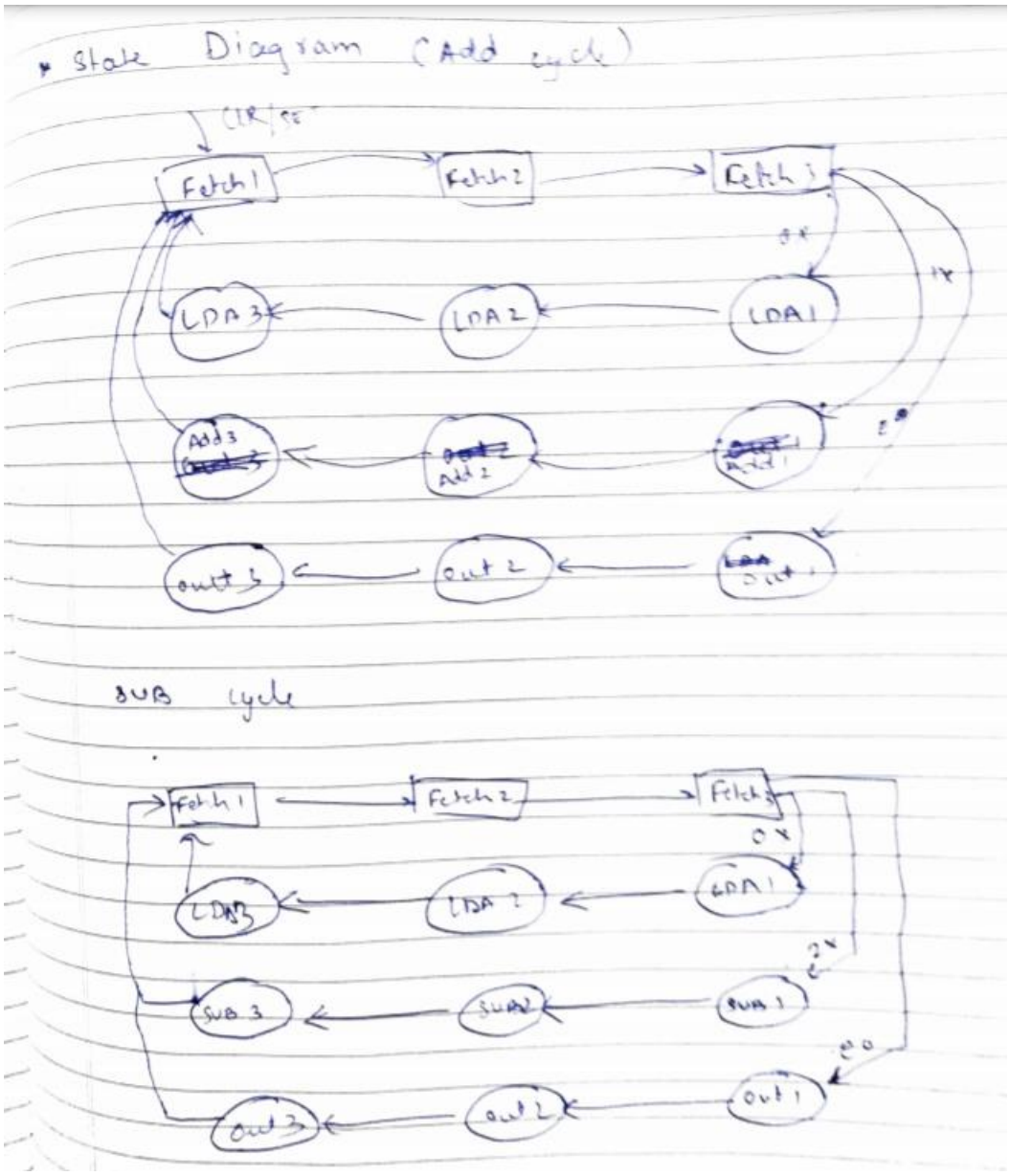
3. EA: The output connection of ALU with bus is enabled when high

4. SU: decides if data is to be added or subtracted, add when low and subtraction

When high.

- **Control Unit**
  This is the device that generates all the control signals in this computer. It has Two ROMS one is 16x12 and other is 8x8. First one maps with the output of each control signal while second maps with what output to select according to the instruction.

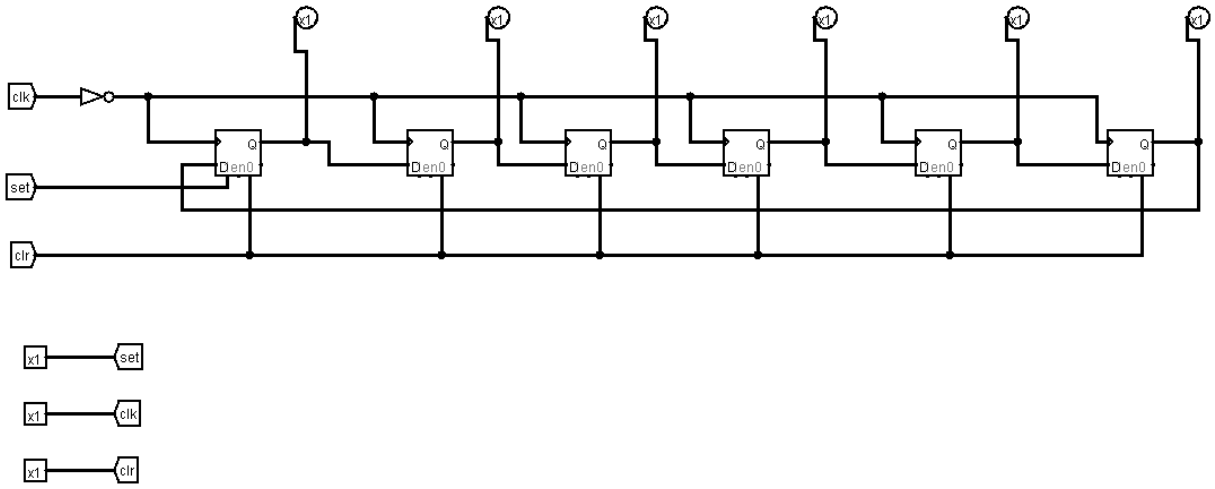
## State diagram and The Control values



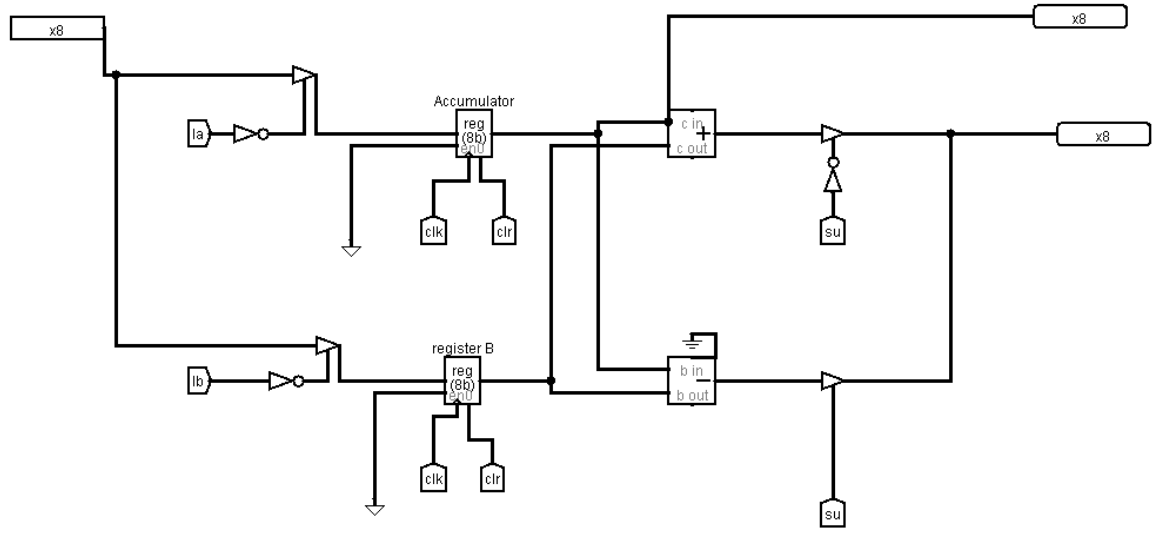
| State | Output |
| --- | --- |
| F 1 | 3E3 |
| F 2 | BE3 |
| F 3 | 263 |
| LDA1 | 1A3 |
| LDA2 | 2C3 |
| LDA 3 | 3E3 |
| ADD 1 | 1A3 |
| ADD 2 | 2E1 |
| ADD 3 | 3C7 |
| SUB 1 | 1A3 |
| SUB 2 | 2E1 |
| SUB 3 | 3CF |
| OUT 1 | 3P2 |
| OUT 2 | 3F3 |
| OUT 3 | 3E3 |

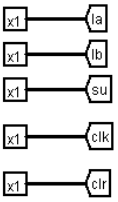
## The Various Components in Logisim

### 1.Ring Counter

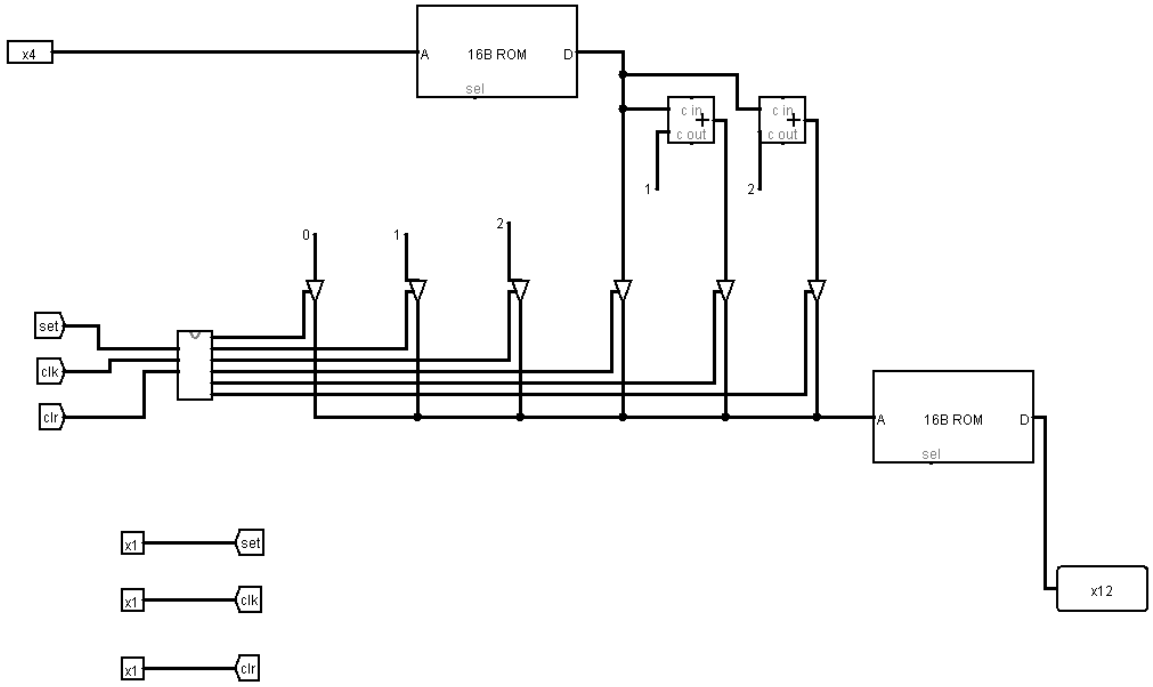


## 2.ALU unit

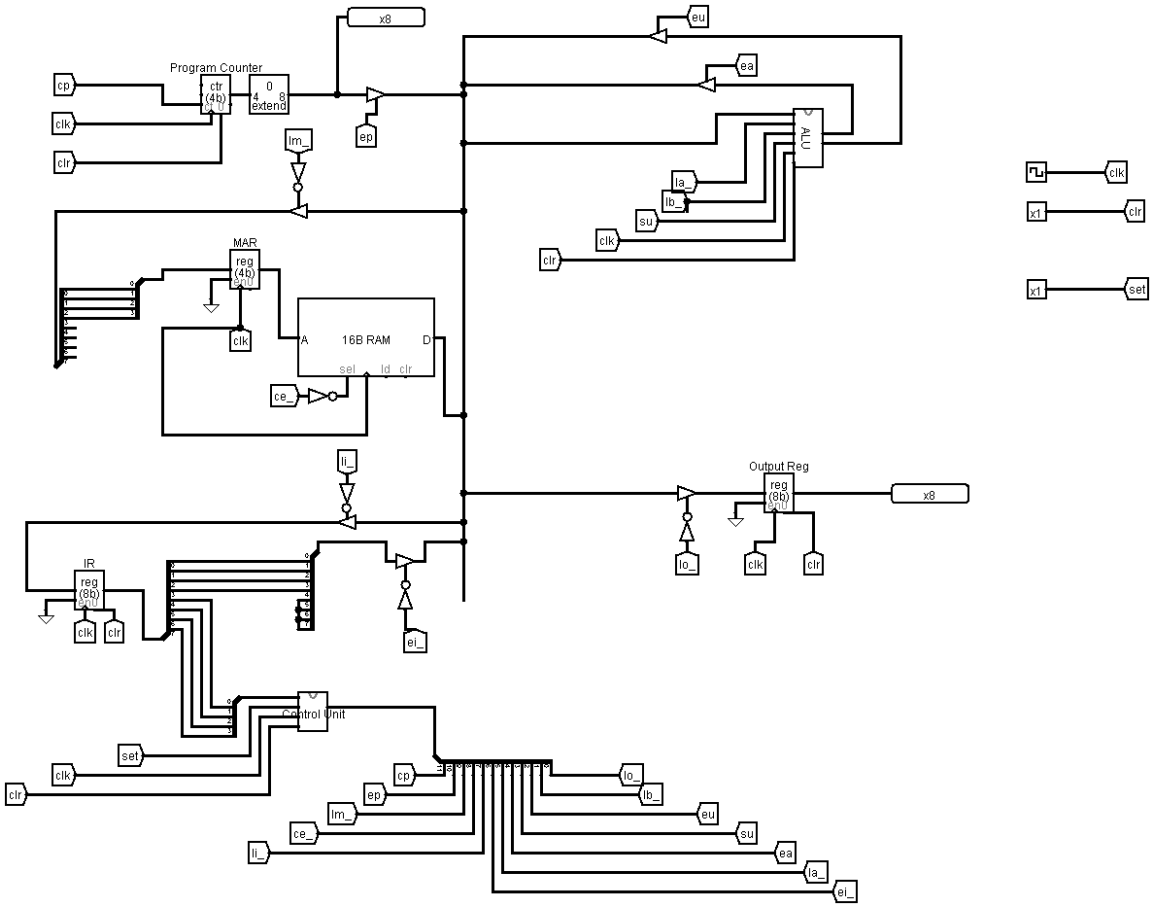

x8
Accumulator
reg
(8b)
la
clk
clr
c in
c out
x8
x8
su
register B
reg
(8b)
lb
clk
clr
b in
b out
su


x1
la
x1
lb
x1
su
x1
clk
x1
clr

# 3.Control Unit



# 4.Main Circuit




- Here bit extender is used as the bus carries 8 bit but counter outputs 4 bits
- The not operation on the control units of ALU is done inside the ALU.

## Working of The computer

The RAM has 2 cycles of instruction ,

### ***First one is a add cycle***

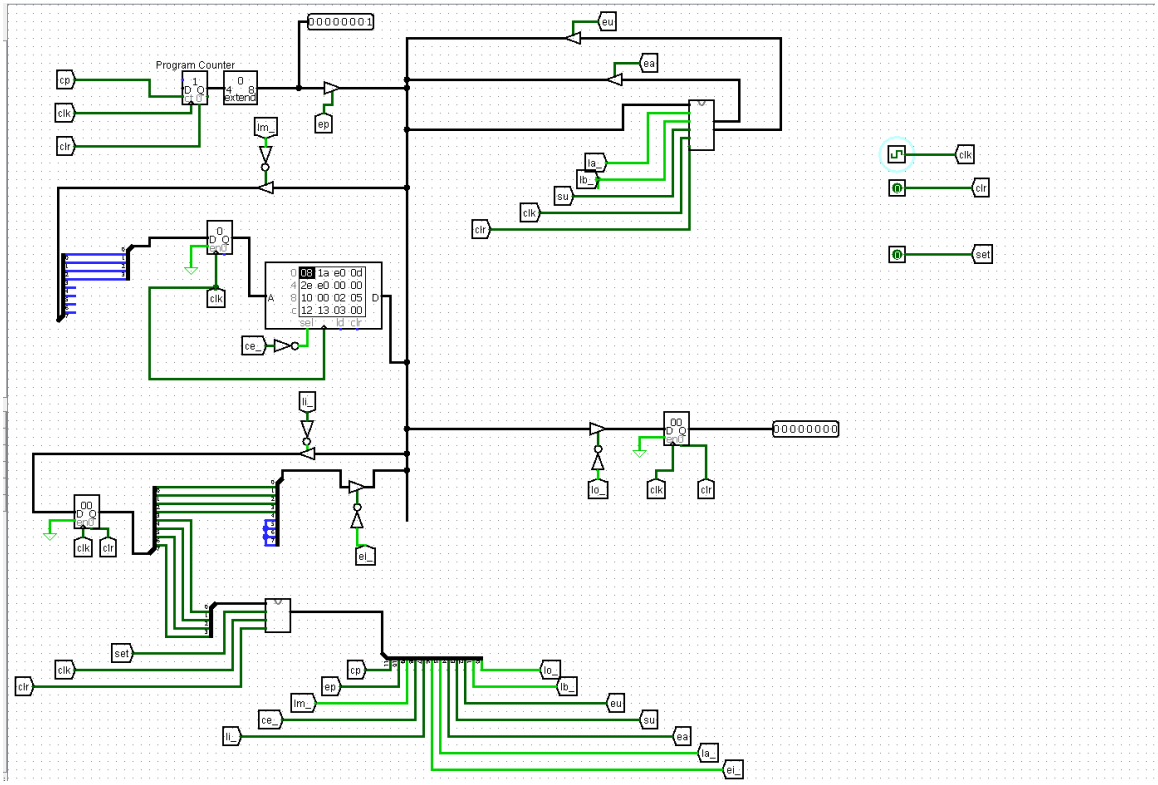
**08** – LDA 8

**1a** – ADD a

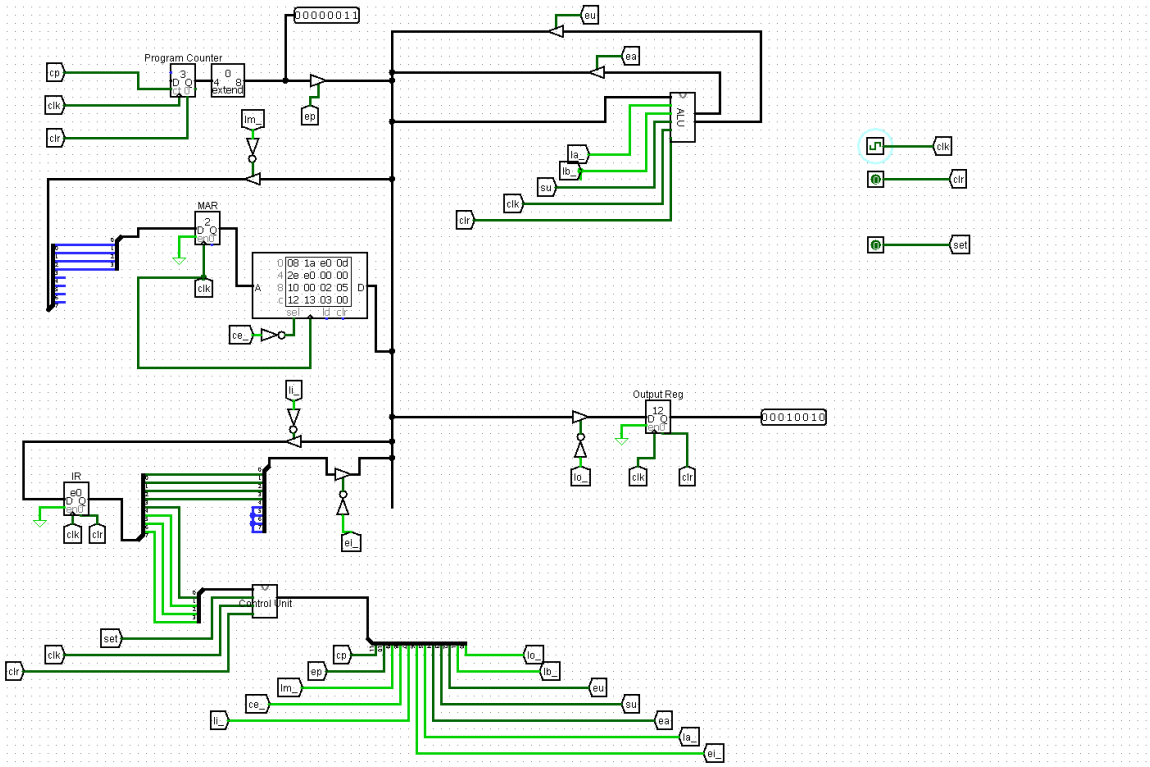
**E0** – output

Expected result is 1x16 + 2x1 = **18 or 00010010 in binary**

**Input-**



**Output**




## ***THE second cycle is subtract cycle***

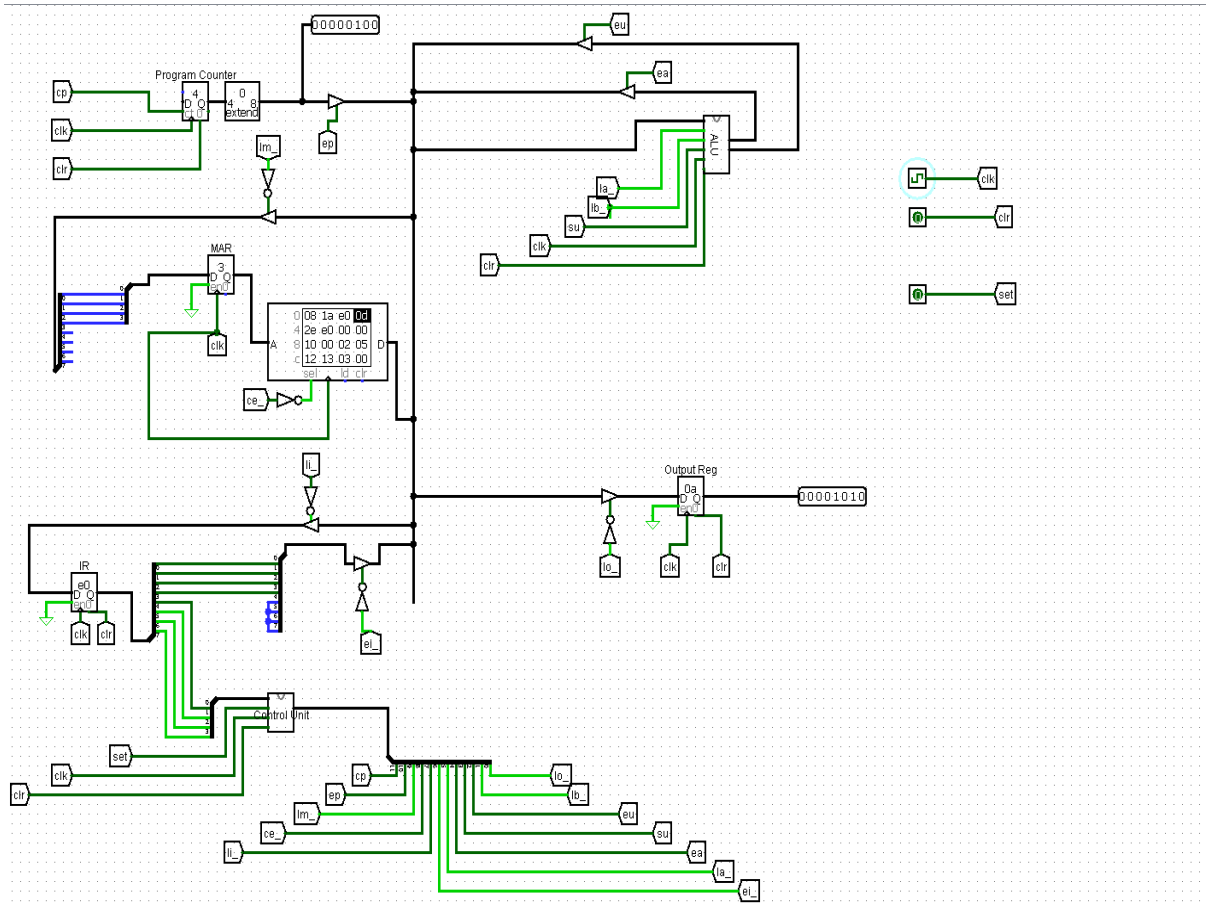
**od** – LDA

**2e** – Sub e

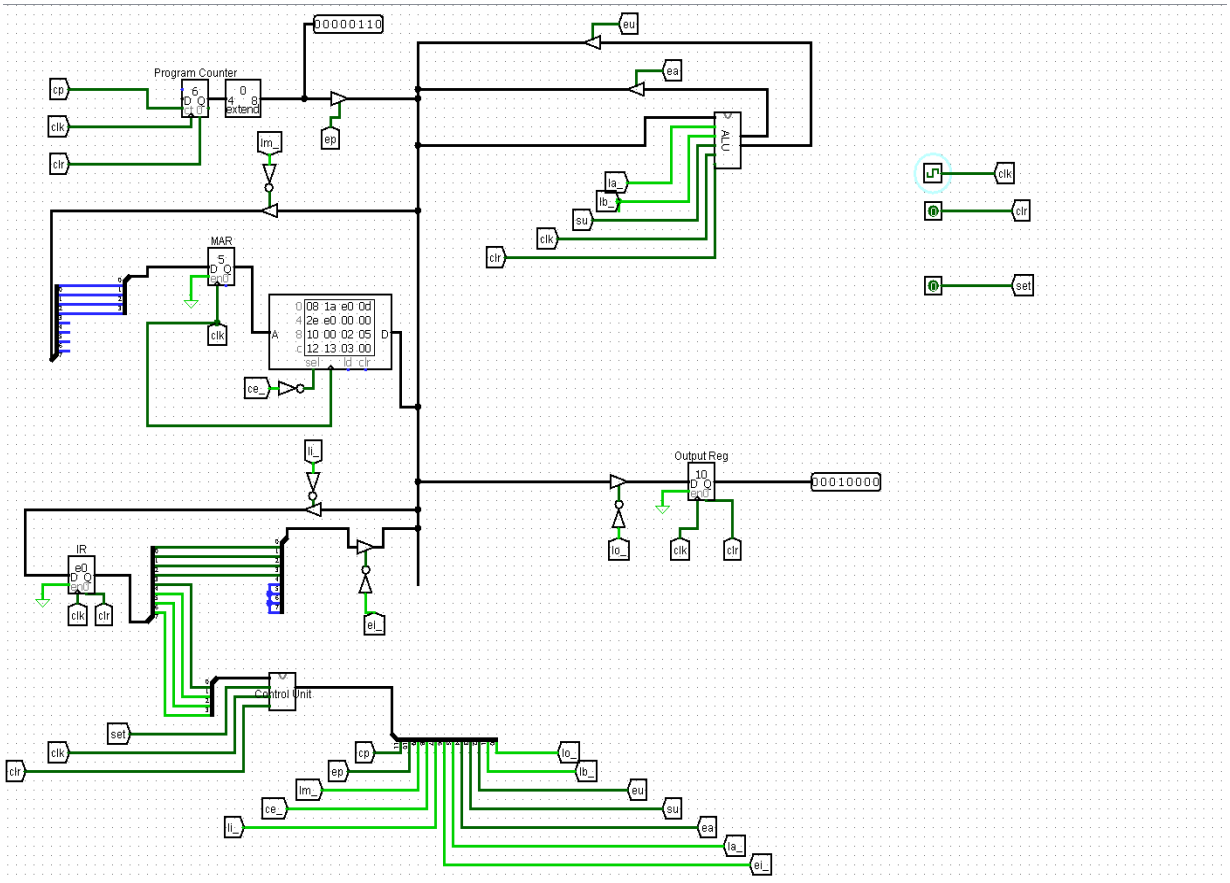
**Eo** – ouput

Expected output – 1x16 + 3- (0 + 3) = 16 = 10000 in binary

## INPUT


00000100
Program Counter
cp
clk
clr
extend
im_
ep
eu
ea
ALU
la_
lb_
su
MAR
ce_
li_
Output Reg
00001010
lo_
IR
ei_
Control Unit
set
cp
lm_
li_
lo_
lb_
eu
su
ea
la_
ei_

**OUTPUT**




## Result

All the addition and subtraction operations are performed as expected with output as expected.

## Conclusion

The experiment was concluded successfully. The computer was implemented as required and the arithmetic operations are performed accordingly.

- Learnings
  - The fetch cycle is common in all the operations which runs for 3 cycles. The other parts run for 3 more cycles. Each cycle is 6 clock cycles long.
  - Bit extender is used to make the output same length as bus .