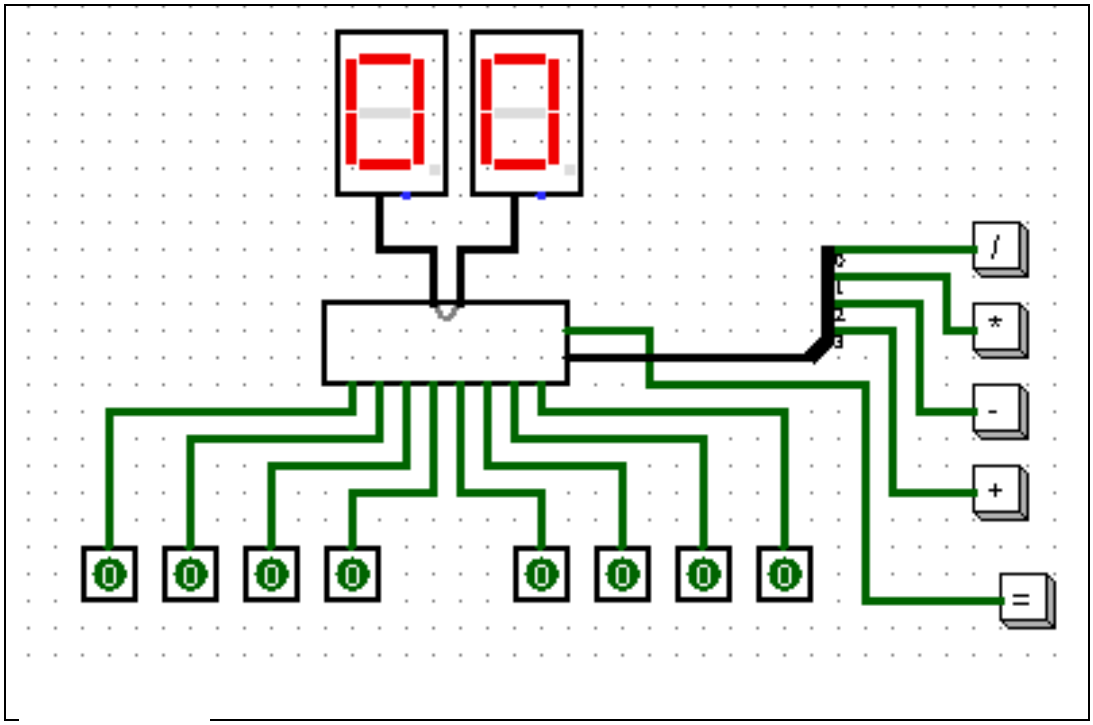
Logisim Calculator I Project Documentation

September 2018

The project modules

1- Main



It includes:

a) Two 7-segments to display the

   i. First operand (8-bit) before pressing the operator

   ii. Result after writing second operand (9-bit) and pressing the equal button.

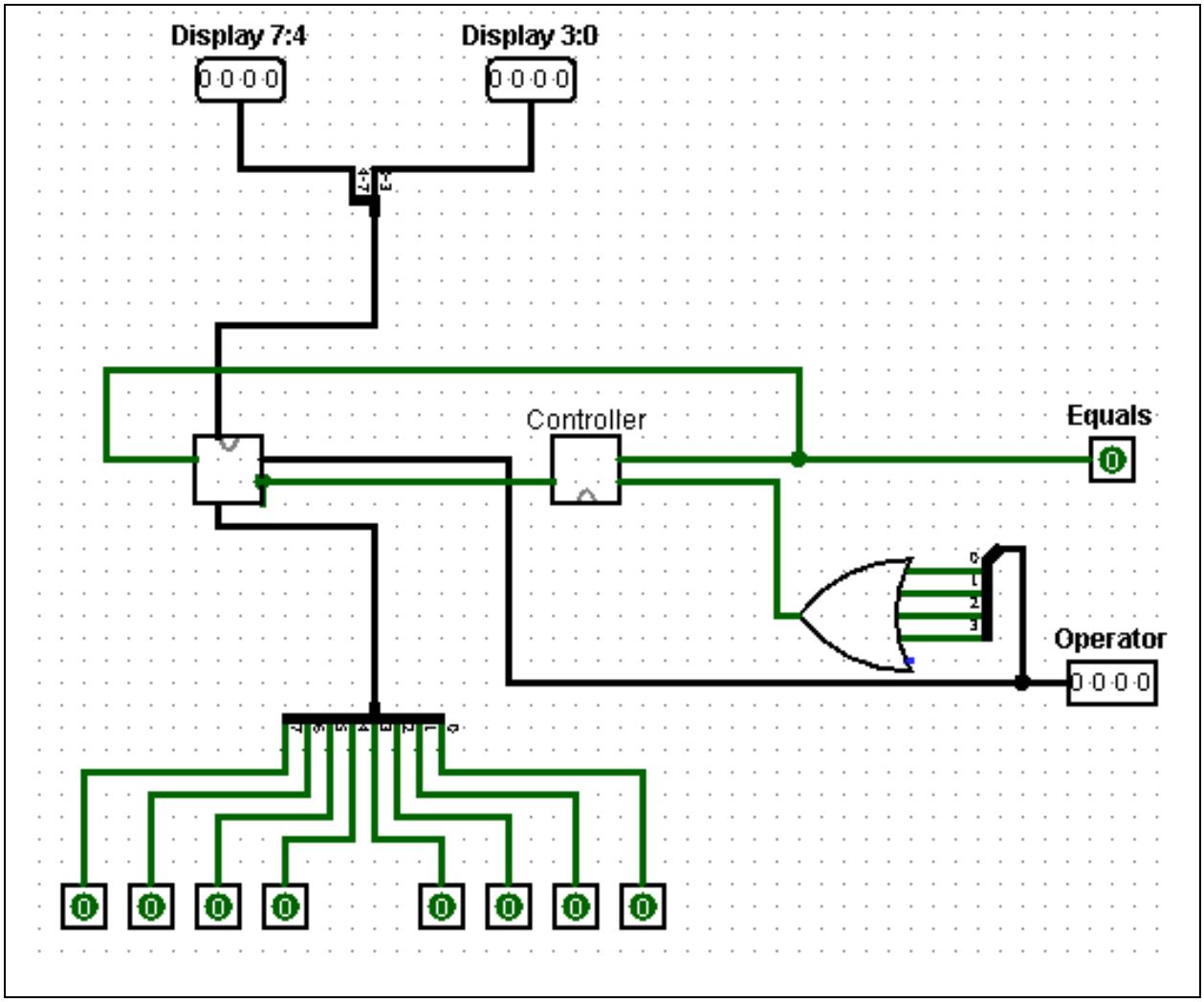
b) 8-block calculator

c) 4 buttons : ‘+’ , ‘-‘, ‘*’, ‘/’

d) ‘=’ button

e) 8 input ports for the data

2- 8-bit Calculator



It includes:

a) Controller sub-circuit

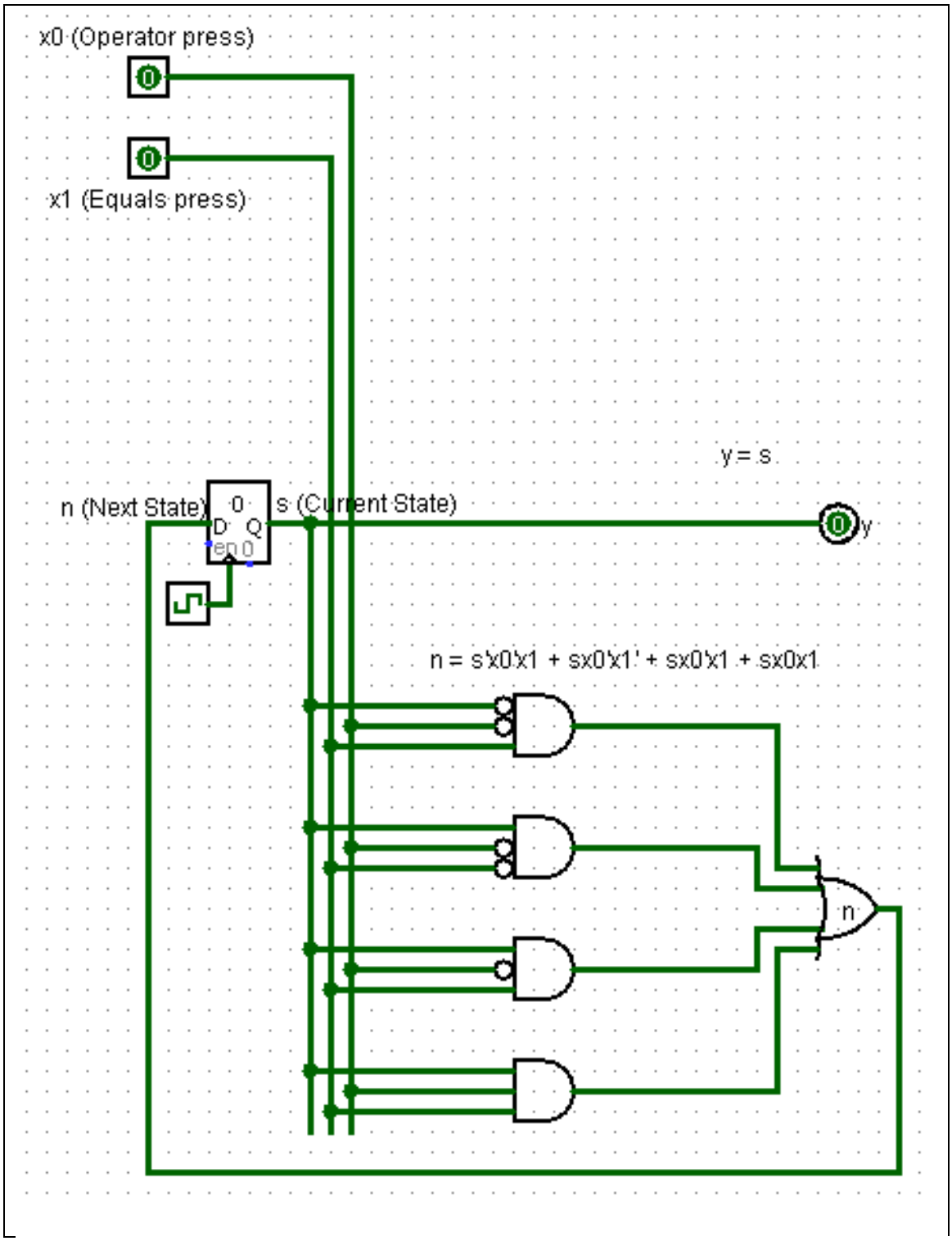
b) Datapath sub-circuit

c) 4-bit Operator input port

d) 8 –bit Data input port

e) Two 4-bit display output ports.

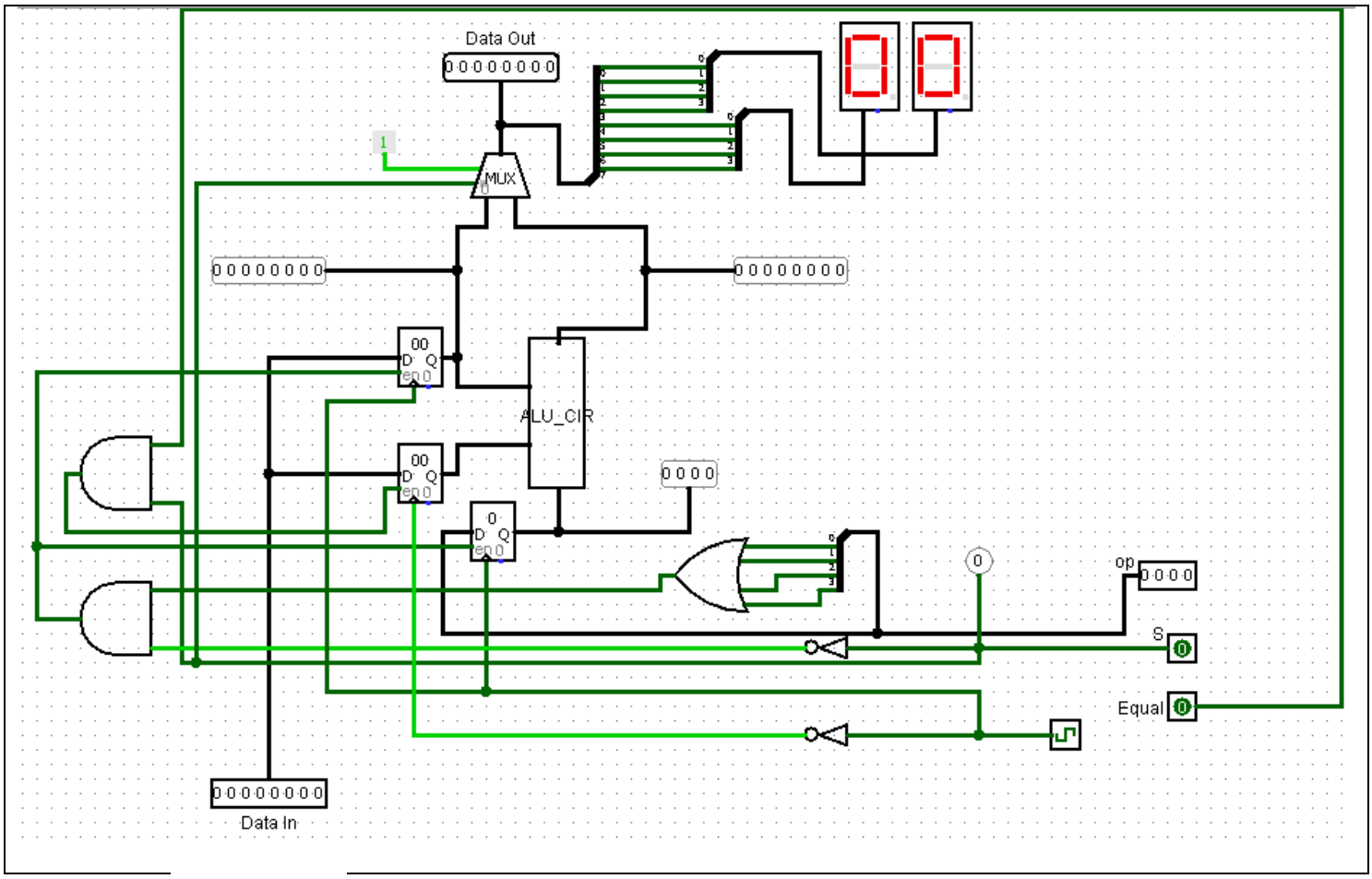
3- Controller (FP Part 1)



It is a finite state machine stores ‘1’ when press equal and clears the stored value when any operator is pressed.

$S_{t+1} = S_t \wedge ((X_1 \wedge X_0) \vee (X_1 \overline{\wedge X0}) \vee \overline{(X1 \wedge X0)}) \vee (\overline{St} \wedge X_1 \wedge \overline{X0})$

4- Datapath (FP Part 2)



It includes:

a) Three register to store the ALUA,ALUB,Operator:

   i. ALUA register (8-bit) enabled when operator is pressed only

   ii. ALUB register (8-bit) enabled when equal is pressed only.

   iii. Operator register (4-bit) enabled when operator is pressed only.

b) ALU_CIR which is ALUCIR sub-circuit for ALU capable for the four operations ‘+’, ‘-‘, ‘*’, and ‘/’.

c) 8-bit 2x1 MUX to choose which of the ALUA or ALUZ to be transferred to the output display. (MUX selection is the state of the controller), if 0 then transfer ALUA, and if 1 then transfer ALUZ to the display output.

d) S 1-bit input port from the state of the controller sub-ciorcuit

e) Equal 1-bit input port from the user equal button

f) Operator 4-bit input port from the user operator.

g) Input data 8-bit port

h) Output data 8-bit port.

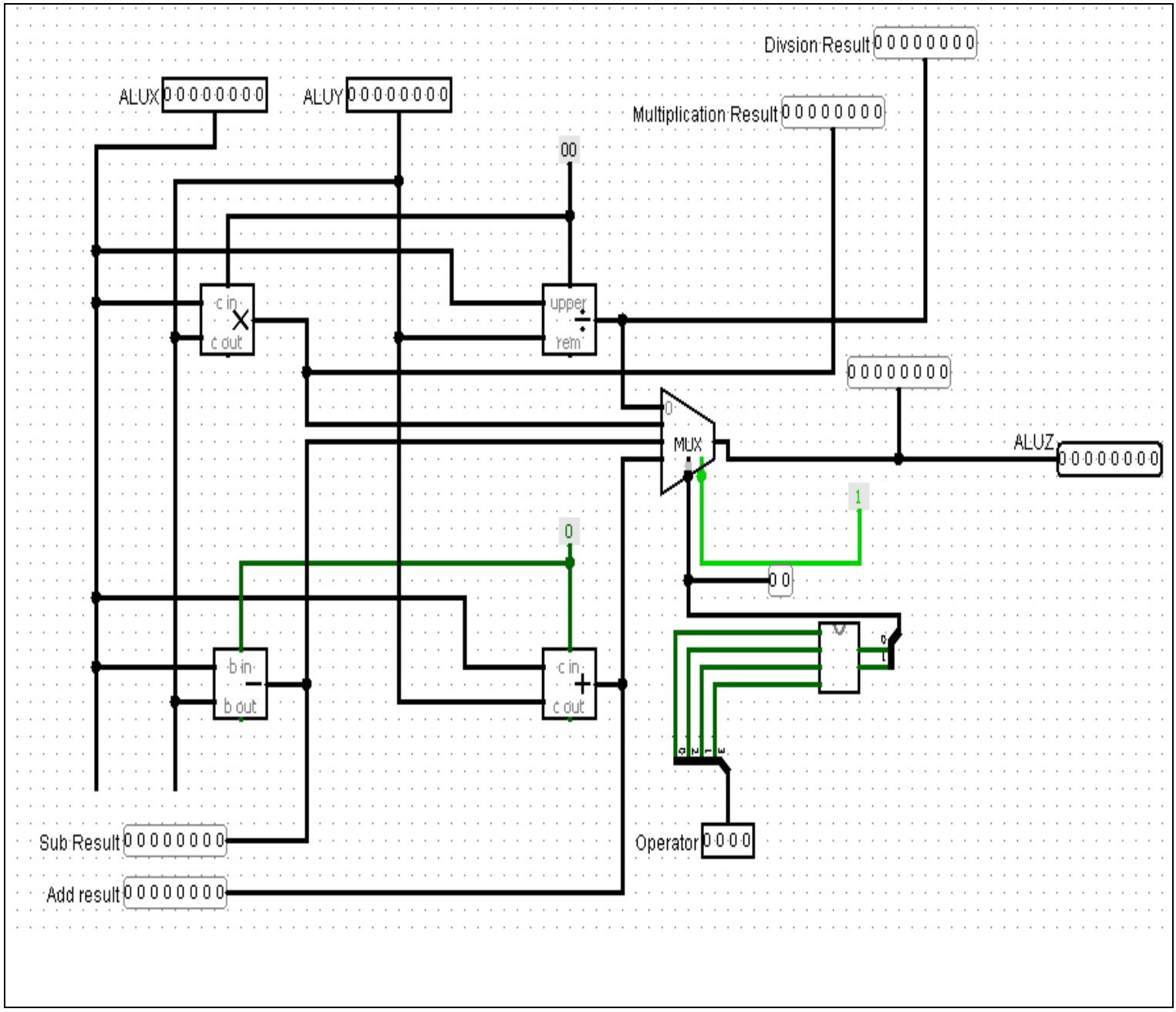
i) Enable for the ALUA and operator registers

$\bar{S} \wedge (Operator_0 \vee Operator_1 \vee Operator_2 \vee Operator_3)$

j) Enable for the ALUB register

$S \wedge Equal$

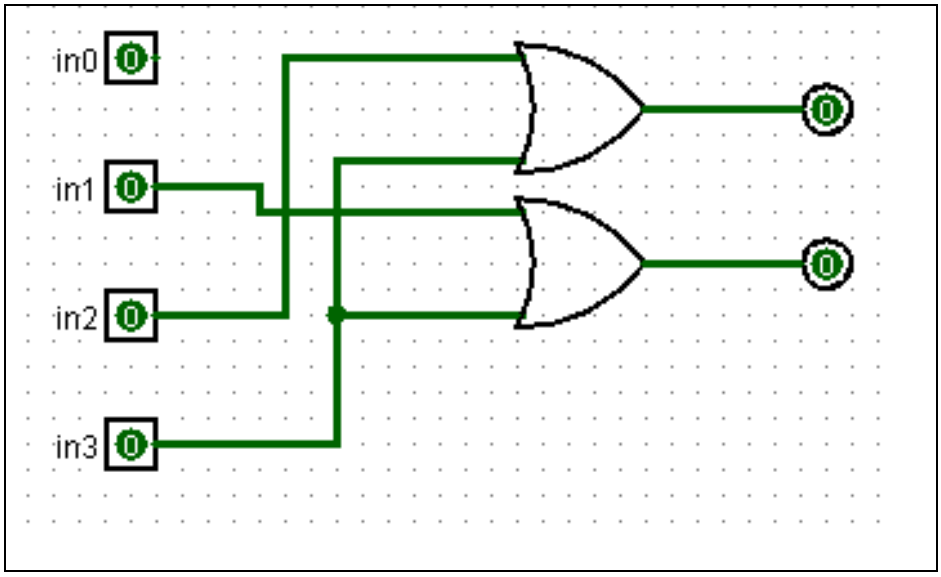
5- ALUCIR


Divsion Result
00000000
ALUX
00000000
ALUY
00000000
Multiplication Result
00000000
00
c in
X
c out
upper
÷
rem
00000000
0
MUX
ALUZ
00000000
1
0
00
b in
−
b out
c in
+
c out
Sub Result
00000000
Operator
0000
Add result
00000000

It includes:

a) Four arithmetic units of 8-bit binary inputsfor addition, subtraction, multiplication and division.
b) 8-bit 4x1 MUX to select certain output from any of the arithmetic unit based on the selection, where the MUX selection is the output of the encoder circuit of operator input.
c) Operator 4-bit input port from the user operator.
d) Input1 data 8-bit port
e) Input2 data 8-bit port
f) Output data 8-bit port.

6- Encoder



It includes:

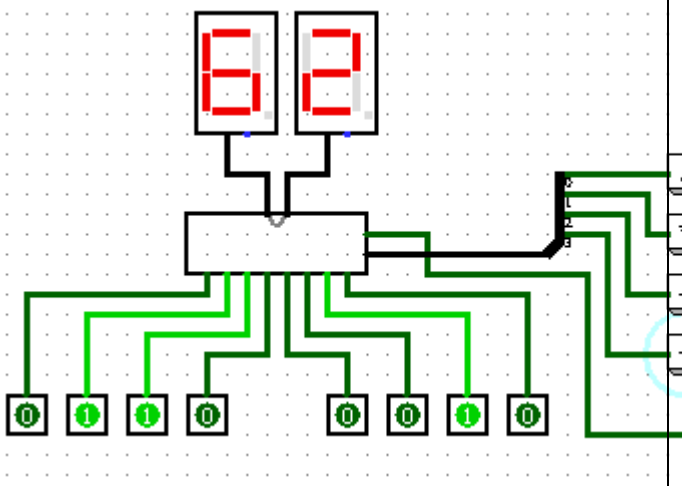
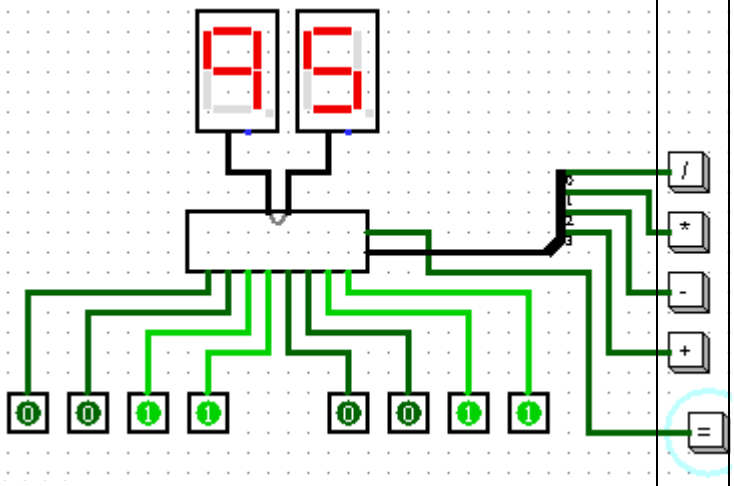
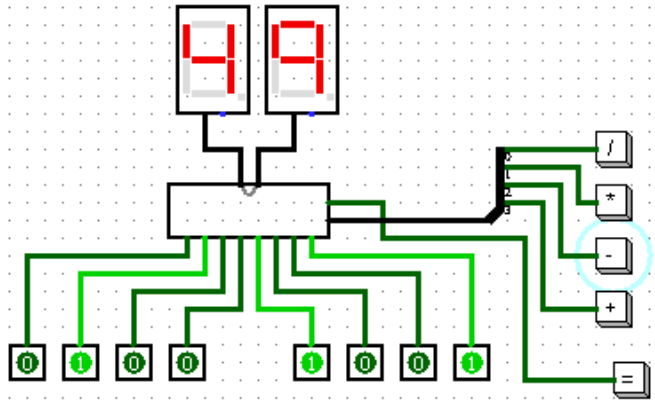
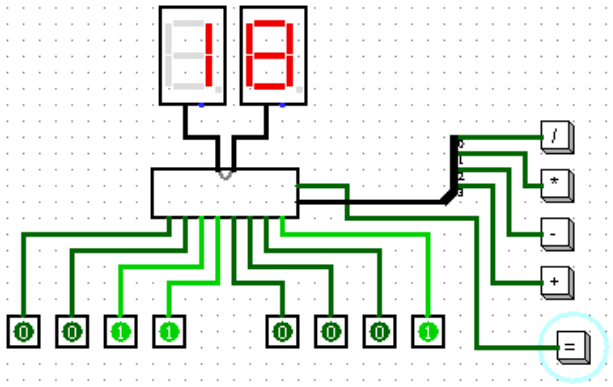
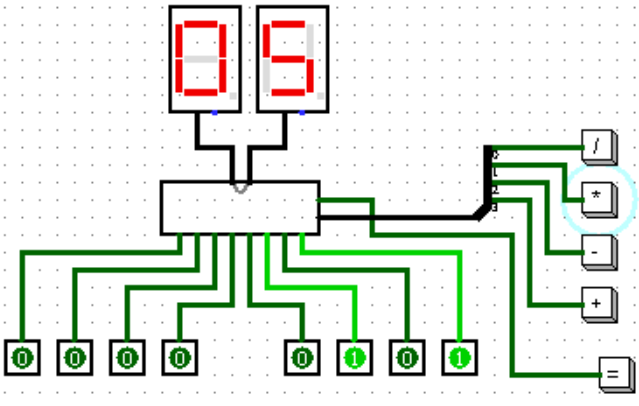
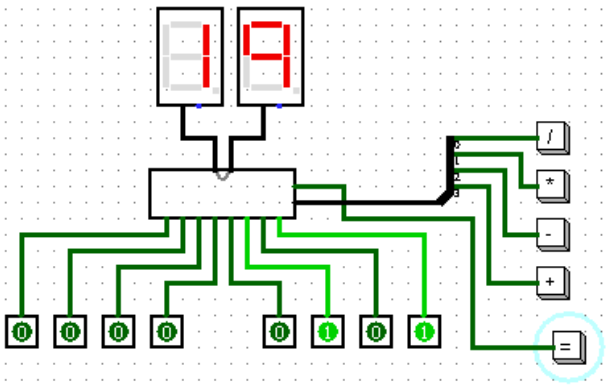
a) $Out_0 = In_1 \vee In_3$.

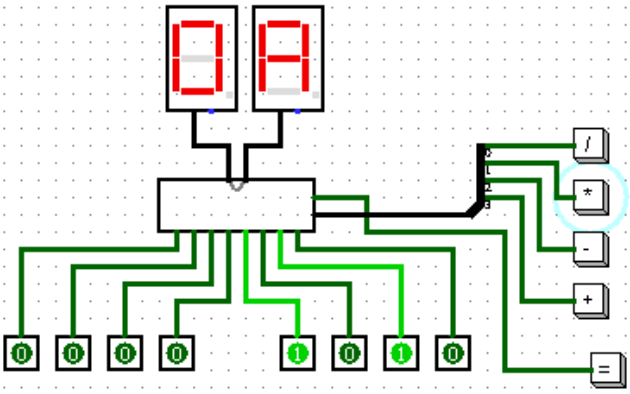
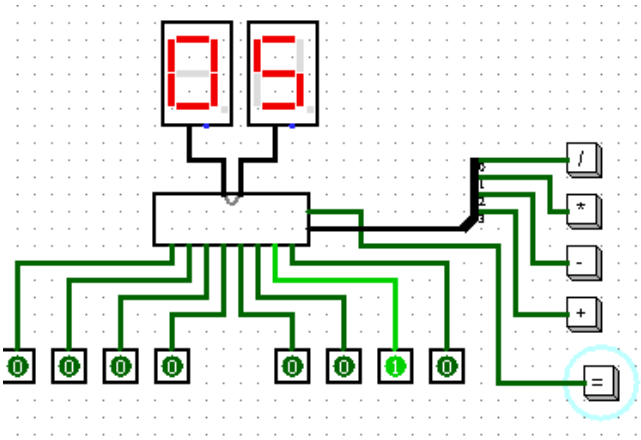
b) $Out_1 = In_2 \vee In_3$.

In_3	In_2	In_1	In_0	Out_1	Out_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

No need to consider other cases because only one button can be pressed at a time.

Simulation results

Data1	Data2	op	O/P
0x62 	0x33 	+	0x95
0x49 	0x31 	-	0x18
0x05 	0x05 	*	0x19

0x0A 	0x02 	/	0x14