Part I) 2) resetn signal is synchronous because in the always block, it only looks for changes in clock and not resetn

It is active low since it resets when resetn == 1'b0. In order to reset the FSM, we need to set resetn to 0 and wait for the next positive edge (0 → 1) of the clock.

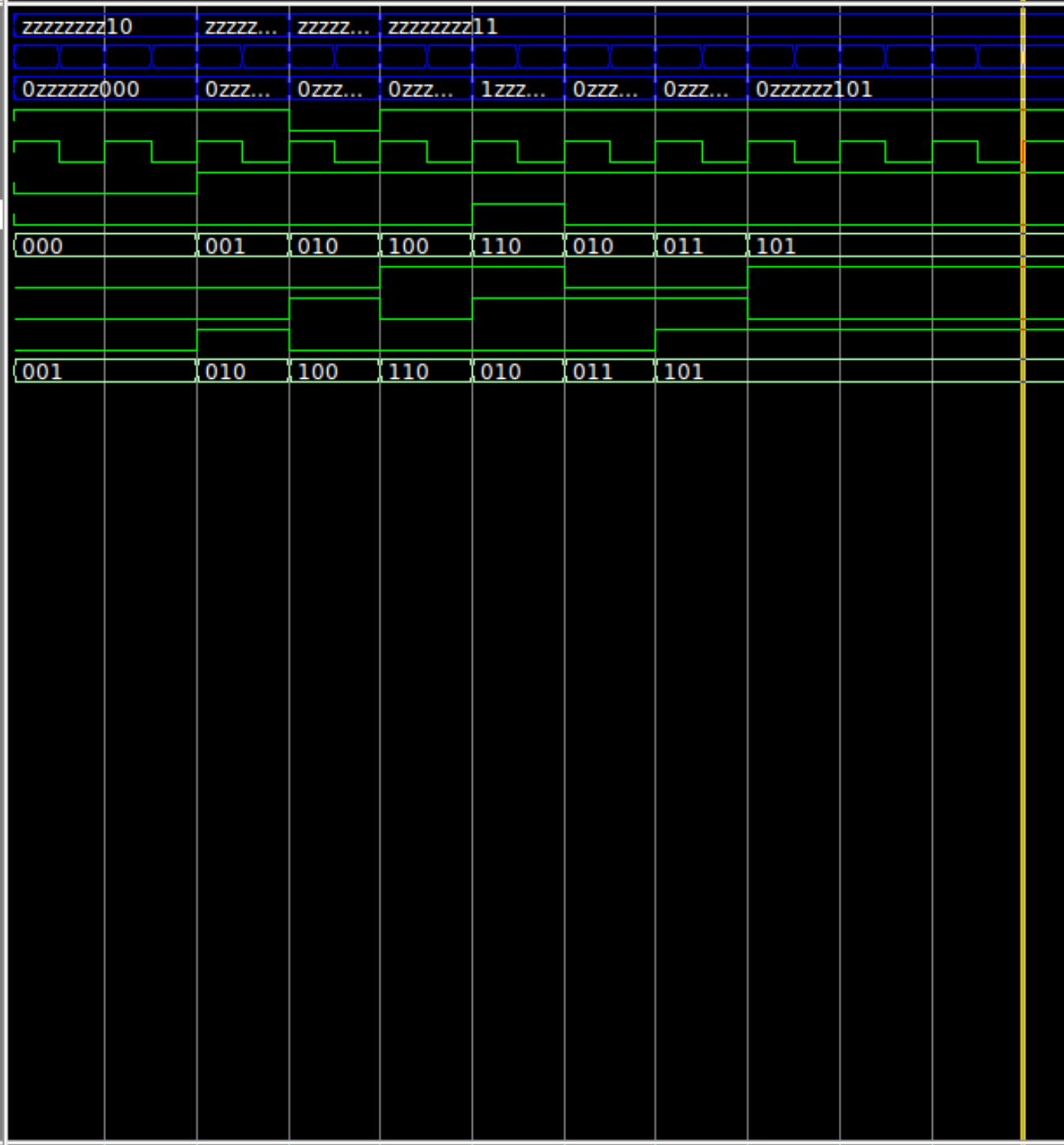
3) state table

		S_2	S_1	S_0	w	Next State
output is 0	A	0	0	0	0	000
		0	0	0	1	001
	B	0	0	1	0	000
		0	0	1	1	010
	C	0	1	0	0	100
		0	1	0	1	011
	D	0	1	1	0	100
		0	1	1	1	101
output is 1	E	1	0	0	0	000
		1	0	0	1	110
	F	1	0	1	0	100
		1	0	1	1	101
	G	1	1	0	0	000
		1	1	0	1	010

Curr State	w	Next State
A	0	A
A	1	B
B	0	A
B	1	C
C	0	E
C	1	D
D	0	E
D	1	F
E	0	A
E	1	G
F	0	E
F	1	F
G	0	A
G	1	C

Wave - Default

	Msgs
/sequence_detector/SW	zzzzzzzz11
/sequence_detector/KEY	zzz0
/sequence_detector/LEDR	0zzzzzz101
/sequence_detector/w	St1
/sequence_detector/clock	St1
/sequence_detector/resetn	St1
/sequence_detector/z	St0
/sequence_detector/y_Q	101
[2]	1
[1]	0
[0]	1
/sequence_detector/Y_D	101



Now	55 ns
Cursor 1	22 ns

ns 4 ns 8 ns 12 ns 16 ns 20 ns

22 ns

Part II) 2)

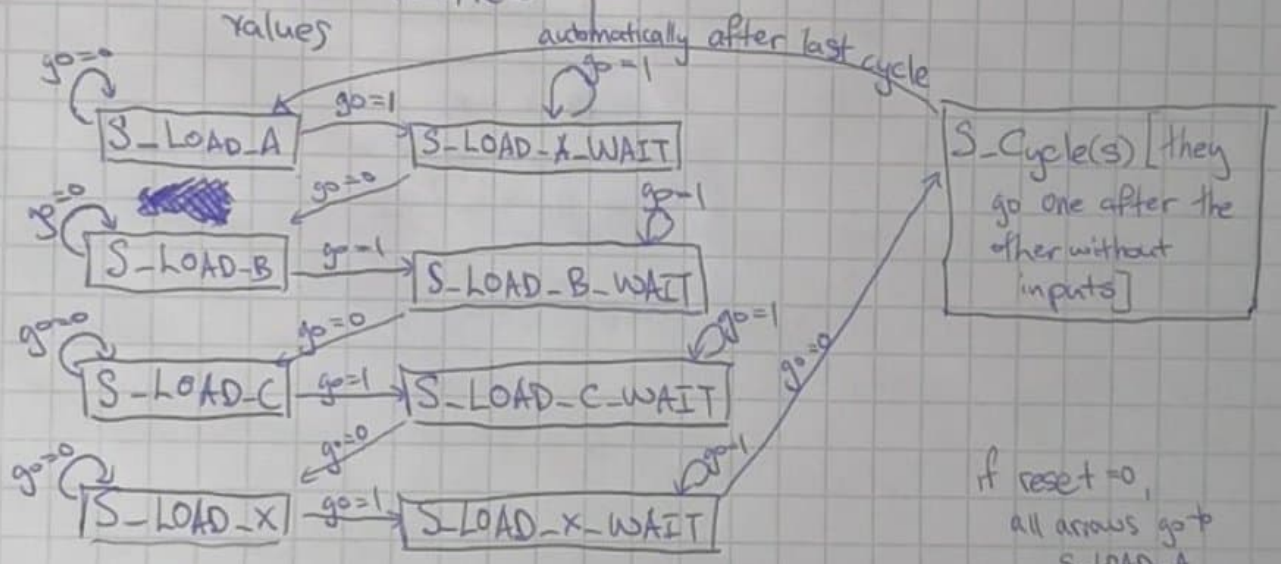
	Inputs State	Values stored
Initial	data_in = 0 reset = 0 go = 1	C = X = A = B = 0 R = 0
showing changes	data_in = A reset = 1 go = 0	C = X = B = 0 RA = A R = 0
	data_in = B reset = 1 go = 0	C = X = 0 RB = B RA = A R = 0
	data_in = C reset = 1 go = 0	C = C X = 0 RB = B RA = A R = 0
	data_in = X go = 0 reset = 1	C = C X = X RB = B RA = A R = 0
	data_in = 0 go = 0 reset = 1 (Cycle_0) → alu_select_a → x alu_select_b → RB alu_op → mult. ld_alu_out = 1, ld_b = 1	C = C X = X RA = A RB = B·x R = 0
	data_in = 0 go = 0 reset = 1 (Cycle_1) → alu_select_a → RA alu_select_b → RB alu_op → add ld_alu_out = 1 ld_b = 1	C = C X = X RA = A RB = B·x + A R = 0

go = 1 between each step so the states progress in Verilog Code

cont

Input State	Values Stored
data_in=0 go=0 reset=1 (Cycle 2) → alu_select_a → C alu_select_b → x alu_op → mult ld_alu_out = 1 ld_a = 1	C=c X=x RA=C·x RB=Bx+A R=0
data_in=0 go=0 reset= (Cycle 3) → alu_select_a → RA alu_select_b → x alu_op → mult ld_alu_out = 1 ld_a = 1	C=c X=x RA=C·x² RB=Bx+A R=0
data_in=0 go=0 reset=1 (Cycle 4) → alu_select_a = RA alu_select_b = RB alu_op → = add ld_r = 1 \# after, state returns to load_A so, turn go to 1 or data_in = A for the same value/new values	C=c X=x $R = Cx^2 + Bx + A$ RA=Cx² RB=Bx+A

3)



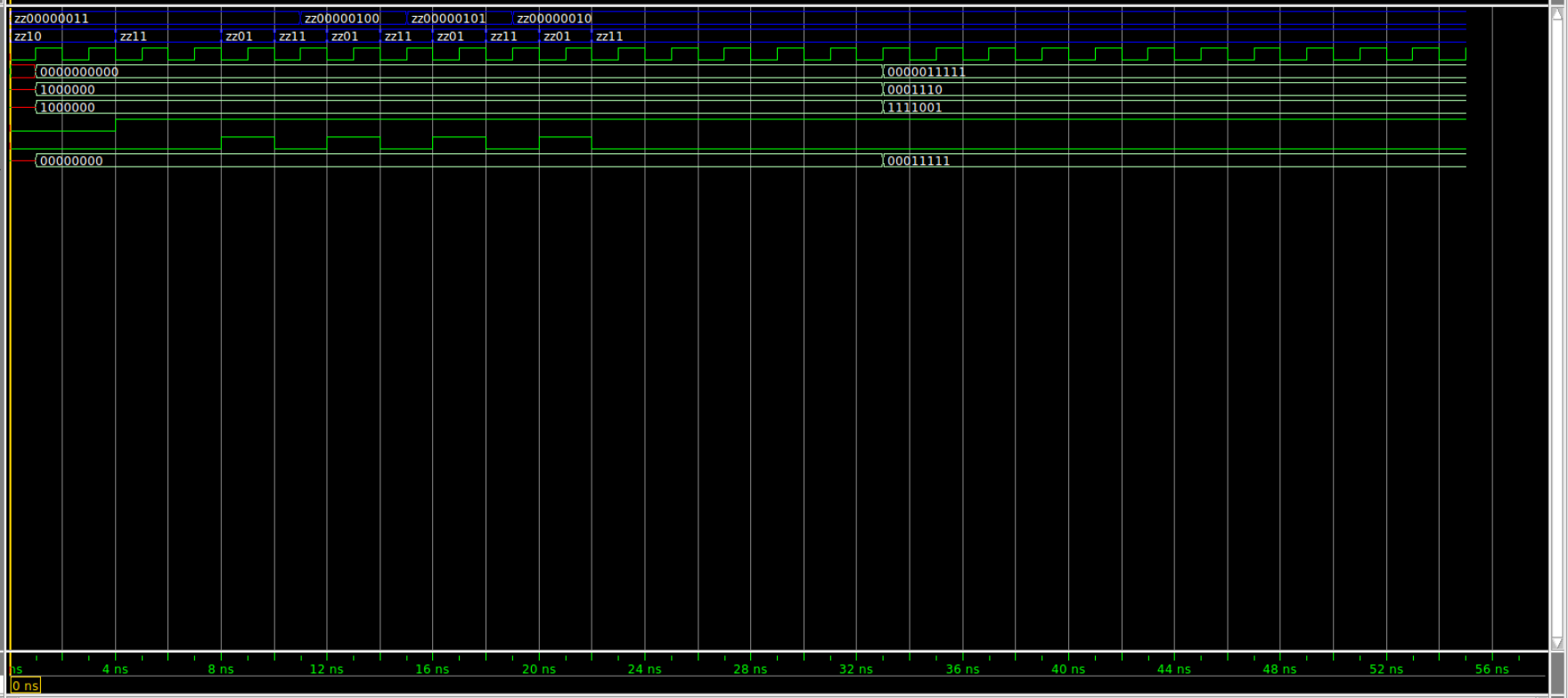
if reset=0, all arrows go to S_LOAD_A

if reset=1, then follow go values

File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help

Wave - Default

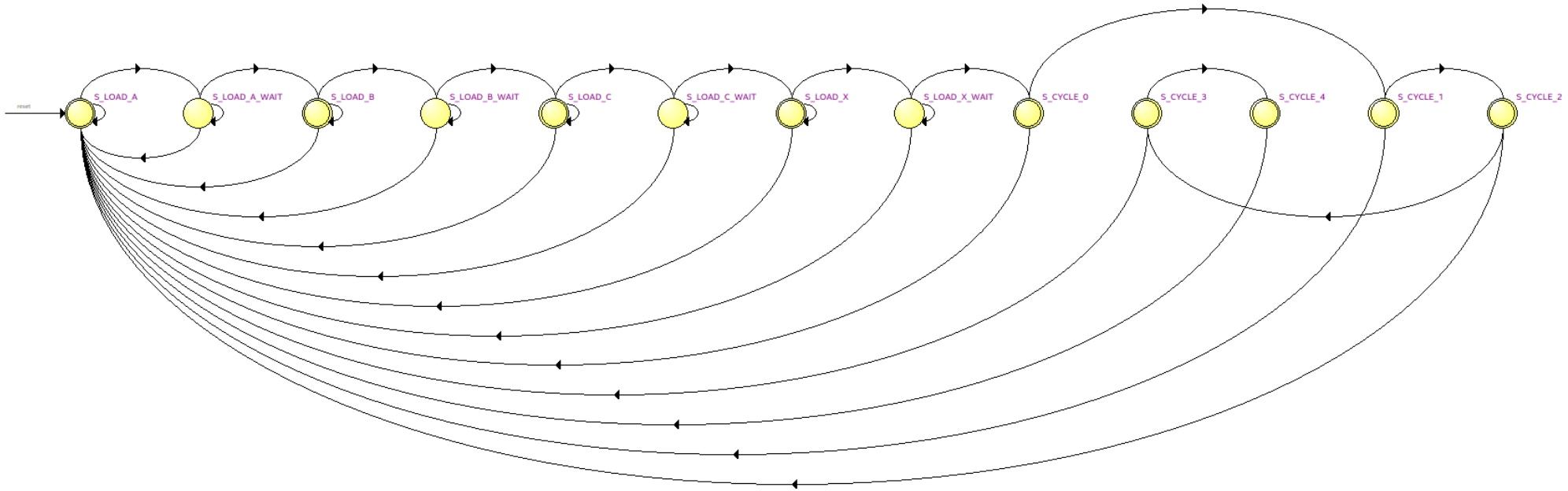
Msgs	
/fpga_top/SW	zz00000011
/fpga_top/KEY	zz10
/fpga_top/CLOC...	St0
/fpga_top/LEDR	00xxxxxxxx
/fpga_top/HEX0	xxxxxxx
/fpga_top/HEX1	xxxxxxx
/fpga_top/resetn	St0
/fpga_top/go	St0
/fpga_top/data_...	xxxxxxxx
Now	55 ns
Cursor 1	0 ns



Transcript

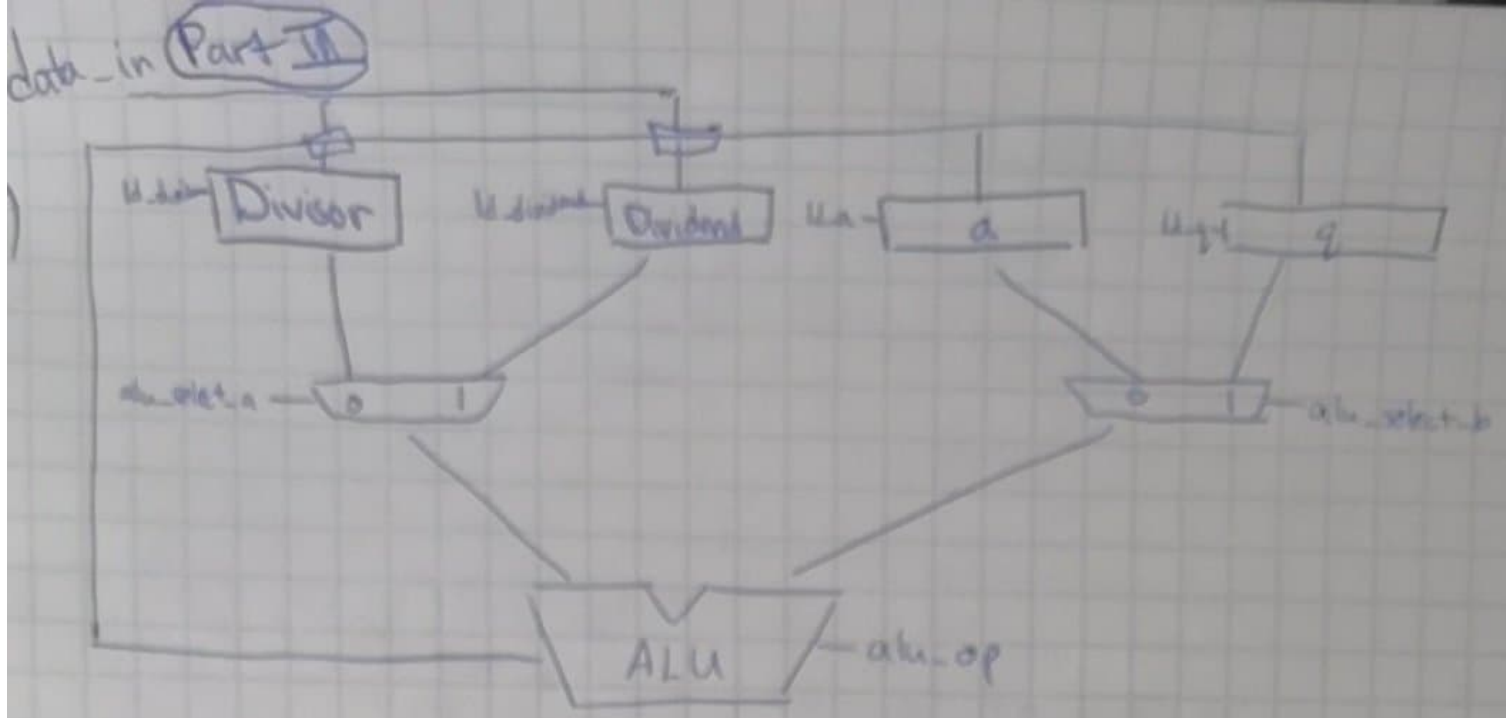
```
# Loading work.fpga_top
# Loading work.part2
# Loading work.control
# Loading work.datapath
# Loading work.hex_decoder
# ** Warning: (vsim-3116) Problem reading symbols from linux-gate.so.1 : can not open ELF file.

VSIM 14>
```

Now: 55 ns Delta: 1 | sim:/fpga_top | 0 ns to 58 ns

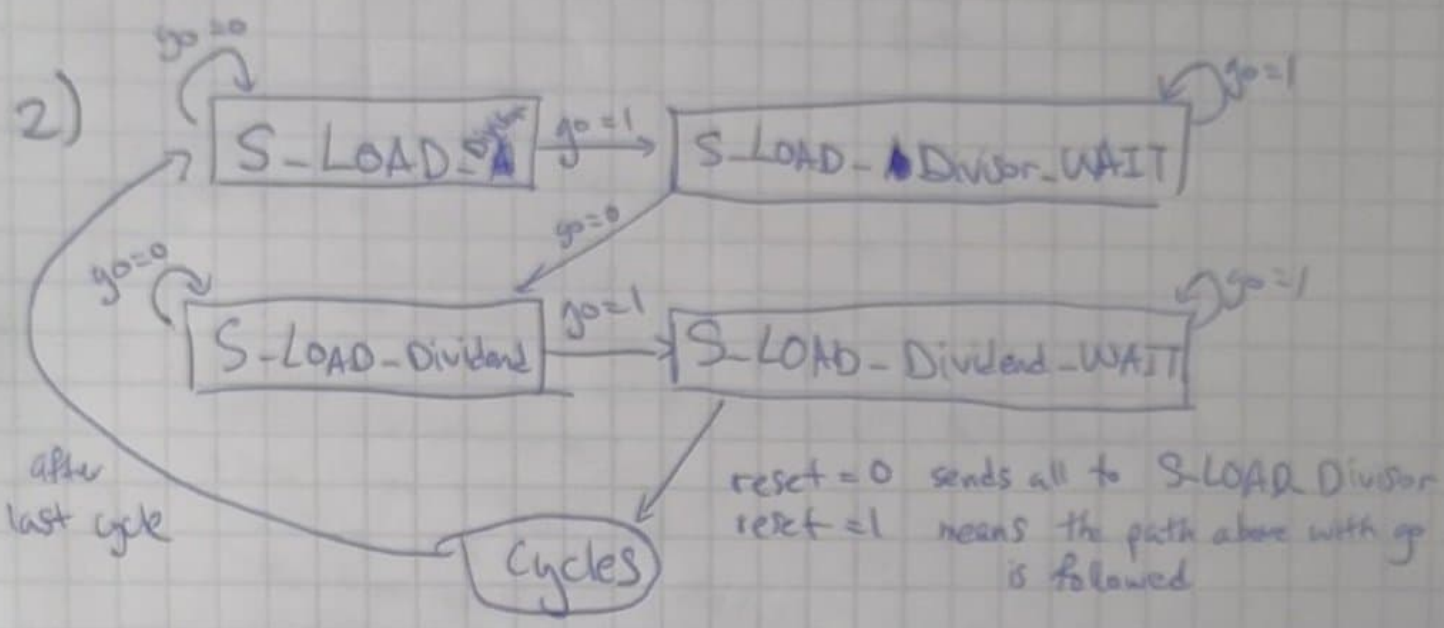

reset
S_LOAD_A
S_LOAD_A_WAIT
S_LOAD_B
S_LOAD_B_WAIT
S_LOAD_C
S_LOAD_C_WAIT
S_LOAD_X
S_LOAD_X_WAIT
S_CYCLE_0
S_CYCLE_3
S_CYCLE_4
S_CYCLE_1
S_CYCLE_2

	Name	current_state.S_CYCLE_4	current_state.S_CYCLE_3	current_state.S_CYCLE_2	current_state.S_CYCLE_1	current_state.S_CYCLE_0	current_state.S_LOAD_X_WAIT	current_state.S_LOAD_X	current_state.S_LOAD_C_WAIT	current_state.S_LOAD_C	current_state.S_LOAD_B_WAIT	current_state.S_LOAD_B	current_state.S_LOAD_A_WAIT	current_state.S_LOAD_A
1	current_state.S_LOAD_A	0	0	0	0	0	0	0	0	0	0	0	0	0
2	current_state.S_LOAD_A_WAIT	0	0	0	0	0	0	0	0	0	0	0	1	1
3	current_state.S_LOAD_B	0	0	0	0	0	0	0	0	0	0	1	0	1
4	current_state.S_LOAD_B_WAIT	0	0	0	0	0	0	0	0	0	1	0	0	1
5	current_state.S_LOAD_C	0	0	0	0	0	0	0	0	1	0	0	0	1
6	current_state.S_LOAD_C_WAIT	0	0	0	0	0	0	0	1	0	0	0	0	1
7	current_state.S_LOAD_X	0	0	0	0	0	0	1	0	0	0	0	0	1
8	current_state.S_LOAD_X_WAIT	0	0	0	0	0	1	0	0	0	0	0	0	1
9	current_s	0	0	0	0	1	0	0	0	0	0	0	0	1

	Name	current_state.S_CYCLE_4	current_state.S_CYCLE_3	current_state.S_CYCLE_2	current_state.S_CYCLE_1	current_state.S_CYCLE_0	current_state.S_LOAD_X_WAIT	current_state.S_LOAD_X	current_state.S_LOAD_C_WAIT	current_state.S_LOAD_C	current_state.S_LOAD_B_WAIT	current_state.S_LOAD_B	current_state.S_LOAD_A_WAIT	current_state.S_LOAD_A
	tate.S_CYCLE_0													
10	current_state.S_CYCLE_1	0	0	0	1	0	0	0	0	0	0	0	0	1
11	current_state.S_CYCLE_2	0	0	1	0	0	0	0	0	0	0	0	0	1
12	current_state.S_CYCLE_3	0	1	0	0	0	0	0	0	0	0	0	0	1
13	current_state.S_CYCLE_4	1	0	0	0	0	0	0	0	0	0	0	0	1

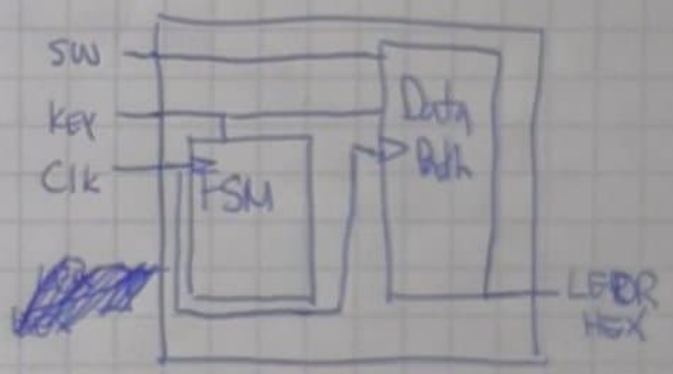


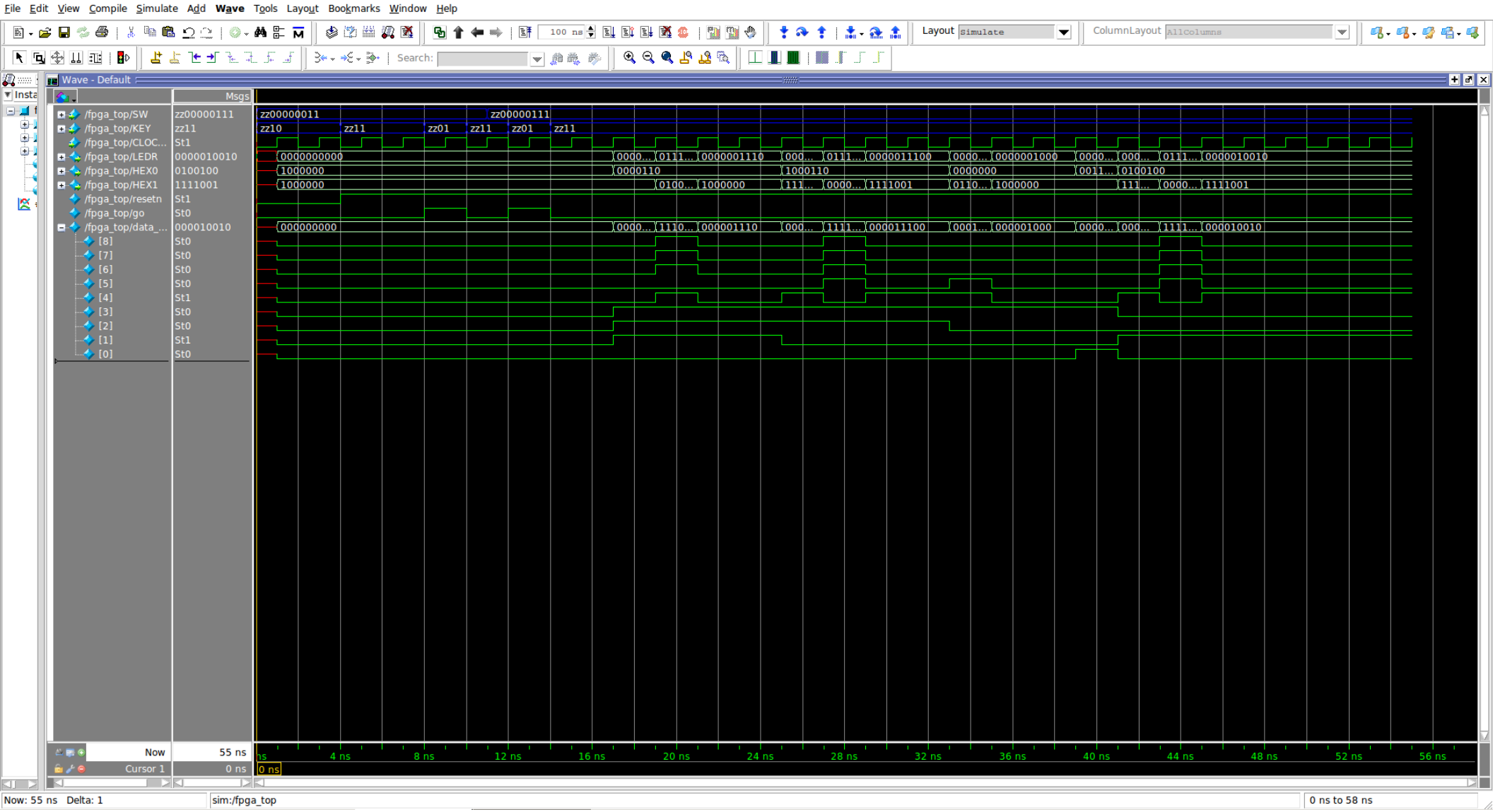
2)



reset = 0 sends all to S_LOAD_Divisor
reset = 1 means the path above with go is followed

3) It is just an FSM




File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Layout Simulate
ColumnLayout AllColumns
Search:
Wave - Default
Msgs
/fpga_top/SW zz00000111
/fpga_top/KEY zz11
/fpga_top/CLOC... St1
/fpga_top/LEDR 0000010010
/fpga_top/HEX0 0100100
/fpga_top/HEX1 1111001
/fpga_top/resetn St1
/fpga_top/go St0
/fpga_top/data_... 000010010
[8] St0
[7] St0
[6] St0
[5] St0
[4] St1
[3] St0
[2] St0
[1] St1
[0] St0
Now 55 ns
Cursor 1 0 ns
4 ns 8 ns 12 ns 16 ns 20 ns 24 ns 28 ns 32 ns 36 ns 40 ns 44 ns 48 ns 52 ns 56 ns
Now: 55 ns Delta: 1
sim:/fpga_top
0 ns to 58 ns