Mehmet Alp Şarkışla

# EE 540 Assignment3

1. **Shifter**

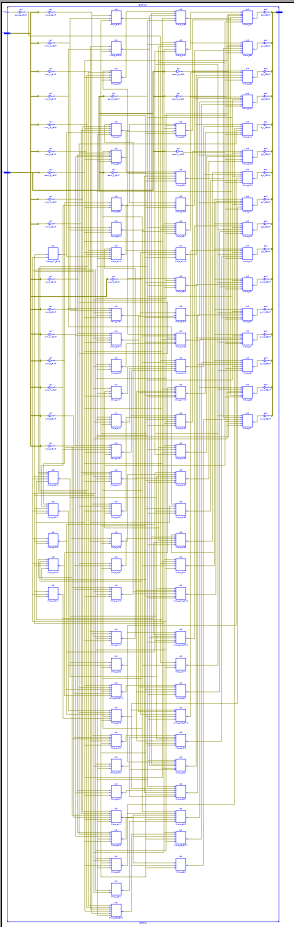
I code my Shifter as behavioral. It consists of if checks and a for loop.

My shifter has two modes of operations

op type:

1: logical shift (shift and pad with zeros)

0: arithmetic shift (left: shift and pad with first value, right: shift and pad with zeros)

I wrote the code in behavioral since Verilog has built in shifters, namely << (for left shift) or >>> (for arithmetic right shift).

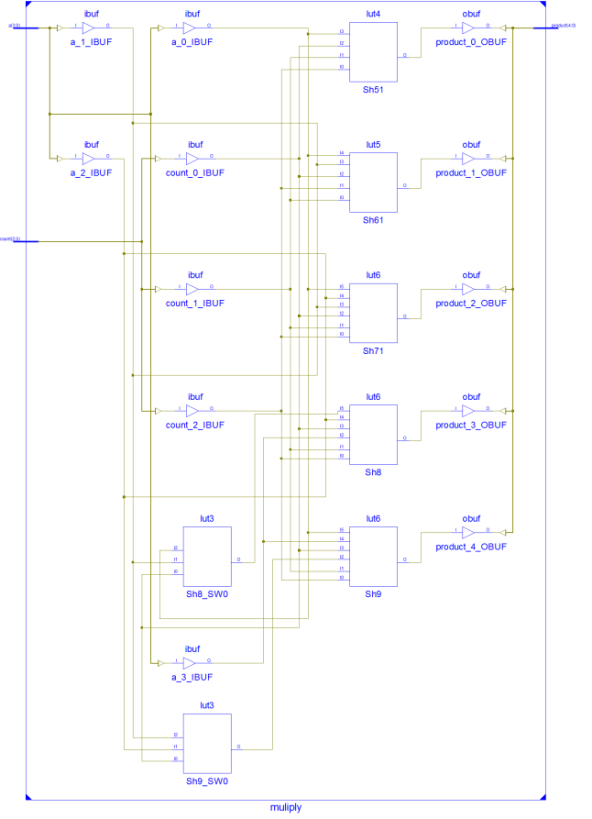
Shifters basically mean multiply or divide by 2s, Other than shifters I used and built in adder to find 2’s complement value. And if checks decide between right-left shift, and logical-arithmetic shift.

If checks are synthesized as 1 bit select multiplexers, adder synthesized (Fig1) as 5 bit full adder, shifters are consist of multiplexers and buffers. Example for one left shift [buf(out[1],0) buf(out[2],in[1]) buf(out[3],in[2]) etc..]

You can see the buffers adder and multiplexers in the left. I also write the one bit left shifter for testing. Result is shown below (Fig 2). If ground connected to product\_4 instead of one, this would be right shift and if ground would be 1 instead of ground (0), this would be arithmetic right shift. Circuit is static so it is easier to implement.

I also coded an dynamic 4bit left shifter, as you can see (Fig 3) synthesize added 4 multiplexers to choose which output bits will be zero and which will be buffed as other input bits.

Final circuit in Fig1 has too many multiplexers to choose outputs correctly.



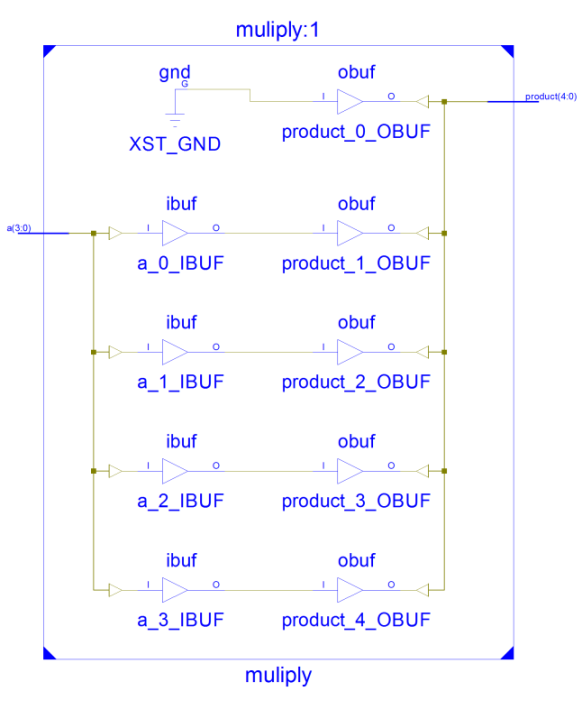


Fig1 Final Shifter Fig2 Static one bit left Shifter Fig3 Dynamic x bit left Shifter

**Test Bench monitor results**

# run all

Simulator is doing circuit initialization process.

Finished circuit initialization process.

#### Arithmatic Shifter

100 num= 0000000000000000 count= 0 o= 0000000000000000 optype =0

300 num= 1101001010110110 count= -16 o= 1111111111111111 optype =0

500 num= 1101001010110110 count= -15 o= 1111111111111111 optype =0

700 num= 1101001010110110 count= -14 o= 1111111111111111 optype =0

900 num= 1101001010110110 count= -13 o= 1111111111111110 optype =0

1100 num= 1101001010110110 count= -12 o= 1111111111111101 optype =0

1300 num= 1101001010110110 count= -11 o= 1111111111111010 optype =0

1500 num= 1101001010110110 count= -10 o= 1111111111110100 optype =0

1700 num= 1101001010110110 count= -9 o= 1111111111101001 optype =0

1900 num= 1101001010110110 count= -8 o= 1111111111010010 optype =0

2100 num= 1101001010110110 count= -7 o= 1111111110100101 optype =0

2300 num= 1101001010110110 count= -6 o= 1111111101001010 optype =0

2500 num= 1101001010110110 count= -5 o= 1111111010010101 optype =0

2700 num= 1101001010110110 count= -4 o= 1111110100101011 optype =0

2900 num= 1101001010110110 count= -3 o= 1111101001010110 optype =0

3100 num= 1101001010110110 count= -2 o= 1111010010101101 optype =0

3300 num= 1101001010110110 count= -1 o= 1110100101011011 optype =0

3500 num= 1101001010110110 count= 0 o= 1101001010110110 optype =0

3700 num= 1101001010110110 count= 1 o= 1010010101101100 optype =0

3900 num= 1101001010110110 count= 2 o= 0100101011011000 optype =0

4100 num= 1101001010110110 count= 3 o= 1001010110110000 optype =0

4300 num= 1101001010110110 count= 4 o= 0010101101100000 optype =0

4500 num= 1101001010110110 count= 5 o= 0101011011000000 optype =0

4700 num= 1101001010110110 count= 6 o= 1010110110000000 optype =0

4900 num= 1101001010110110 count= 7 o= 0101101100000000 optype =0

5100 num= 1101001010110110 count= 8 o= 1011011000000000 optype =0

5300 num= 1101001010110110 count= 9 o= 0110110000000000 optype =0

5500 num= 1101001010110110 count= 10 o= 1101100000000000 optype =0

5700 num= 1101001010110110 count= 11 o= 1011000000000000 optype =0

5900 num= 1101001010110110 count= 12 o= 0110000000000000 optype =0

6100 num= 1101001010110110 count= 13 o= 1100000000000000 optype =0

6300 num= 1101001010110110 count= 14 o= 1000000000000000 optype =0

6500 num= 1101001010110110 count= 15 o= 0000000000000000 optype =0

#### Logical Shifter

6700 num= 1101001010110110 count= -16 o= 0000000000000000 optype =1

6900 num= 1101001010110110 count= -15 o= 0000000000000001 optype =1

7100 num= 1101001010110110 count= -14 o= 0000000000000011 optype =1

7300 num= 1101001010110110 count= -13 o= 0000000000000110 optype =1

7500 num= 1101001010110110 count= -12 o= 0000000000001101 optype =1

7700 num= 1101001010110110 count= -11 o= 0000000000011010 optype =1

7900 num= 1101001010110110 count= -10 o= 0000000000110100 optype =1

8100 num= 1101001010110110 count= -9 o= 0000000001101001 optype =1

8300 num= 1101001010110110 count= -8 o= 0000000011010010 optype =1

8500 num= 1101001010110110 count= -7 o= 0000000110100101 optype =1

8700 num= 1101001010110110 count= -6 o= 0000001101001010 optype =1

8900 num= 1101001010110110 count= -5 o= 0000011010010101 optype =1

9100 num= 1101001010110110 count= -4 o= 0000110100101011 optype =1

9300 num= 1101001010110110 count= -3 o= 0001101001010110 optype =1

9500 num= 1101001010110110 count= -2 o= 0011010010101101 optype =1

9700 num= 1101001010110110 count= -1 o= 0110100101011011 optype =1

9900 num= 1101001010110110 count= 0 o= 1101001010110110 optype =1

10100 num= 1101001010110110 count= 1 o= 1010010101101100 optype =1

10300 num= 1101001010110110 count= 2 o= 0100101011011000 optype =1

10500 num= 1101001010110110 count= 3 o= 1001010110110000 optype =1

10700 num= 1101001010110110 count= 4 o= 0010101101100000 optype =1

10900 num= 1101001010110110 count= 5 o= 0101011011000000 optype =1

11100 num= 1101001010110110 count= 6 o= 1010110110000000 optype =1

11300 num= 1101001010110110 count= 7 o= 0101101100000000 optype =1

11500 num= 1101001010110110 count= 8 o= 1011011000000000 optype =1

11700 num= 1101001010110110 count= 9 o= 0110110000000000 optype =1

11900 num= 1101001010110110 count= 10 o= 1101100000000000 optype =1

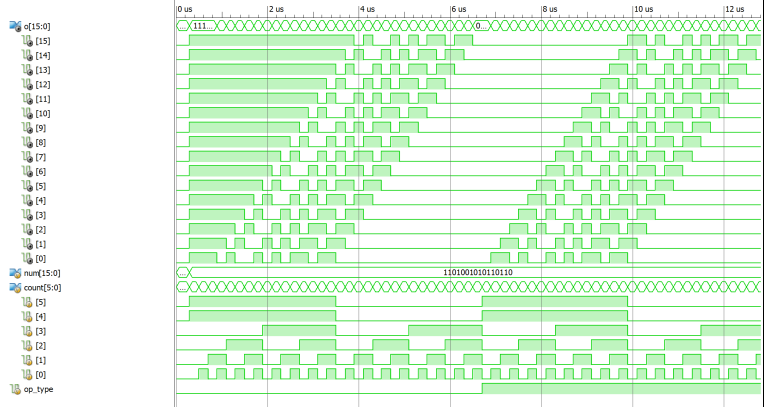
12100 num= 1101001010110110 count= 11 o= 1011000000000000 optype =1

12300 num= 1101001010110110 count= 12 o= 0110000000000000 optype =1

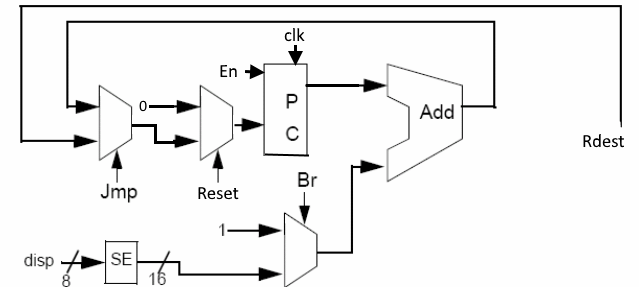
12500 num= 1101001010110110 count= 13 o= 1100000000000000 optype =1

12700 num= 1101001010110110 count= 14 o= 1000000000000000 optype =1

12900 num= 1101001010110110 count= 15 o= 0000000000000000 optype =1

**Test Bench Wave results**

1. **Program Counter (PC)**

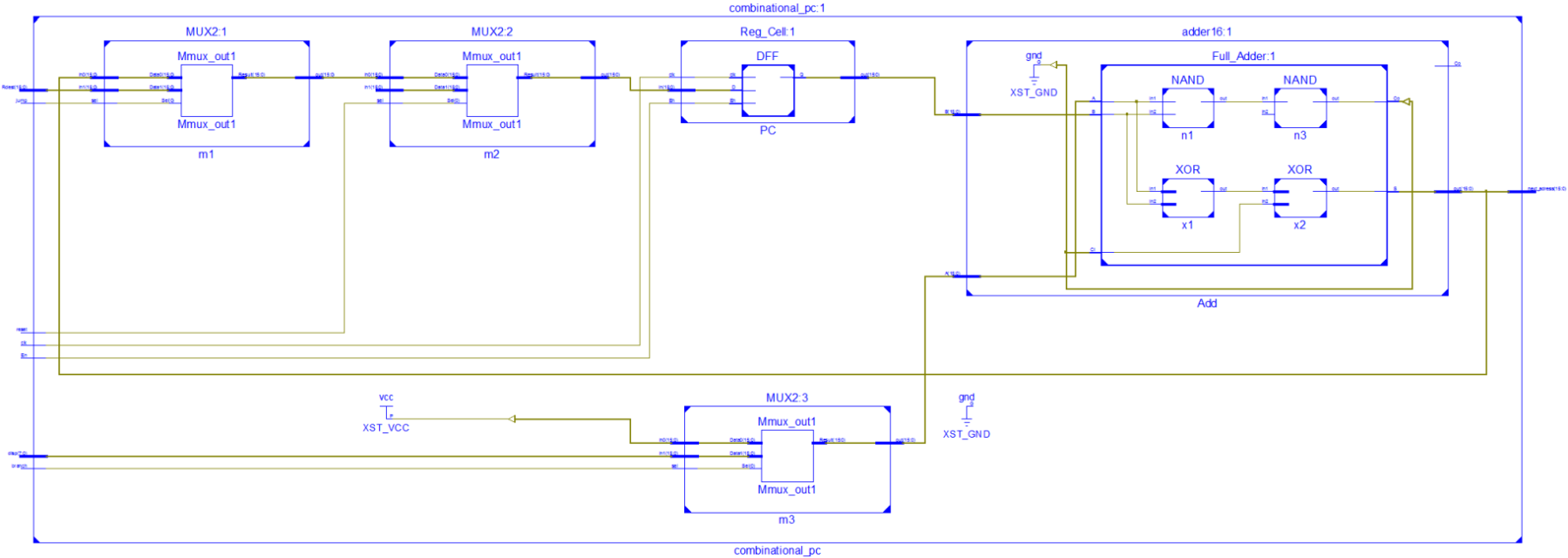
Circuit I used for my combinational pc is given below (Fig4). It has a synchronous reset (address is 0 if reset is 1), enable inputs as extra. Normally PC doesn’t have enable but I need it to make asynchronous load and reset for Fpga implementation part. Since we it must do parallel load (jump) and reset even when it is stopping.

I code circuit with structural style except I combined wires with dataflow.

We have 3 2x16 bit bus multiplexers, 1 16bit full adder without carry out and 116 bit register cell.

Fig4 PC schematic

I used register cell in previous assignment.

You can see the RTL schematic in Fig5.

**Test Bench monitor results (I didn’t wave results since it won’t be readable)**

Simulator is doing circuit initialization process.

Enable is always one

Finished circuit initialization process.

0 jump= 0 reset= 0 clk= 0 Rdest = 0000 branch = 0 disp = 00

next\_adress = xxxx

1 jump= 0 reset= 0 clk= 1 Rdest = 0000 branch = 0 disp = 00

First reset

1 jump= 0 reset= 1 clk= 0 Rdest = 0000 branch = 0 disp = 00

next\_adress = xxxx

2 jump= 0 reset= 1 clk= 1 Rdest = 0000 branch = 0 disp = 00

2 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 0001

3 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

3 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 0002

4 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

4 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 0003

5 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

5 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 0004

6 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

6 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 0005

7 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

7 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 0006

8 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

8 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 0007

9 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

9 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 0008

10 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

10 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 0009

11 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

11 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 000a

12 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

Try jump

12 jump= 1 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 00

next\_adress = 000b

13 jump= 1 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 00

13 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f02b

14 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

14 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f02c

15 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

15 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f02d

16 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

16 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f02e

17 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

17 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f02f

18 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

18 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f030

19 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

19 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f031

20 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

20 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f032

21 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

21 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f033

22 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

22 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f034

23 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

Try branch

23 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 1 disp = 05

next\_adress = f039

24 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 1 disp = 05

24 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 1 disp = 05

next\_adress = f03e

25 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 1 disp = 05

25 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 1 disp = 05

next\_adress = f043

26 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 1 disp = 05

26 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 1 disp = 05

next\_adress = f048

27 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 1 disp = 05

27 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 1 disp = 05

next\_adress = f04d

28 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 1 disp = 05

28 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f04e

29 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

29 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f04f

30 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

30 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f050

31 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

31 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f051

32 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

32 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f052

33 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

33 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f053

34 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

34 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f054

35 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

35 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f055

36 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

36 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f056

37 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

37 jump= 0 reset= 0 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f057

38 jump= 0 reset= 0 clk= 1 Rdest = f02a branch = 0 disp = 05

Try reset

38 jump= 0 reset= 1 clk= 0 Rdest = f02a branch = 0 disp = 05

next\_adress = f058

39 jump= 0 reset= 1 clk= 1 Rdest = f02a branch = 0 disp = 05

39 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 0001

40 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

40 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 0002

41 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

41 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 0003

42 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

42 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 0004

43 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

43 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 0005

44 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

44 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 0006

45 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

45 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 0007

46 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

46 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 0008

47 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

47 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 0009

48 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

48 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 000a

49 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

Try jump again

49 jump= 1 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = 000b

50 jump= 1 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

50 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aaab

51 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

51 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aaac

52 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

52 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aaad

53 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

53 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aaae

54 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

54 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aaaf

55 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

55 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aab0

56 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

56 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aab1

57 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

57 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aab2

58 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

58 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aab3

59 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

59 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aab4

60 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05

60 jump= 0 reset= 0 clk= 0 Rdest = aaaa branch = 0 disp = 05

next\_adress = aab5

61 jump= 0 reset= 0 clk= 1 Rdest = aaaa branch = 0 disp = 05