Giao Tran

Barrel_shifter:

1. SystemVerilog design

```
1   // Barrel Shifter RTL Model
2   `include "mux_2x1_behavioral.sv"
3   module barrel_shifter (
4     input logic select,  // select=0 shift operation, select=1 rotate operation
5     input logic direction, // direction=0 right move, direction=1 left move
6     input logic[1:0] shift_value, // number of bits to be shifted (0, 1, 2 or 3)
7     input logic[3:0] din,
8     output logic[3:0] dout
9   );
10
11  logic [3:0] w; //wires between each mux
12  logic [3:0] t_din;
13  logic [3:0] t_dout;
14
15  always @ (*) begin
16      if (direction==0) //right direction
17          begin
18              t_din=din;
19          end
20      else if (direction==1) //left direction
21          begin
22              t_din[0]=din[3];
23              t_din[1]=din[2];
24              t_din[2]=din[1];
25              t_din[3]=din[0];
26          end
27  end
28
29  always @(*) begin
30  if (direction==0) //right direction
31      begin
32          dout=t_dout;
33      end
34  else if (direction==1) //left direction
35      begin
36          dout[0]=t_dout[3];
37          dout[1]=t_dout[2];
38          dout[2]=t_dout[1];
39          dout[3]=t_dout[0];
40      end
41  end
42
43  logic a,b, c; //// these are the values that will in plug in .in1 (either 0 or t_din)
44
45  always @ (*) begin
46      if (select==0)//shift
47          begin
48              a=0; b=0; c=0;
49          end
50      else if (select==1)//rotate
51          begin
52              a=t_din[0];
53              b=t_din[1];
54              c=w[0];
55          end
56  end
57
58  mux_2x1 inst0 (.in0(t_din[0]),.in1(t_din[2]),.sel(shift_value[1]),.out(w[0]));
59  mux_2x1 inst1 (.in0(t_din[1]),.in1(t_din[3]),.sel(shift_value[1]),.out(w[1]));
60  mux_2x1 inst2 (.in0(t_din[2]),.in1(a),.sel(shift_value[1]),.out(w[2]));
61  mux_2x1 inst3 (.in0(t_din[3]),.in1(b),.sel(shift_value[1]),.out(w[3]));
62
63  mux_2x1 inst4 (.in0(w[0]),.in1(w[1]),.sel(shift_value[0]),.out(t_dout[0]));
64  mux_2x1 inst5 (.in0(w[1]),.in1(w[2]),.sel(shift_value[0]),.out(t_dout[1]));
65  mux_2x1 inst6 (.in0(w[2]),.in1(w[3]),.sel(shift_value[0]),.out(t_dout[2]));
66  mux_2x1 inst7 (.in0(w[3]),.in1(c),.sel(shift_value[0]),.out(t_dout[3]));
67
68  endmodule: barrel_shifter
69
```

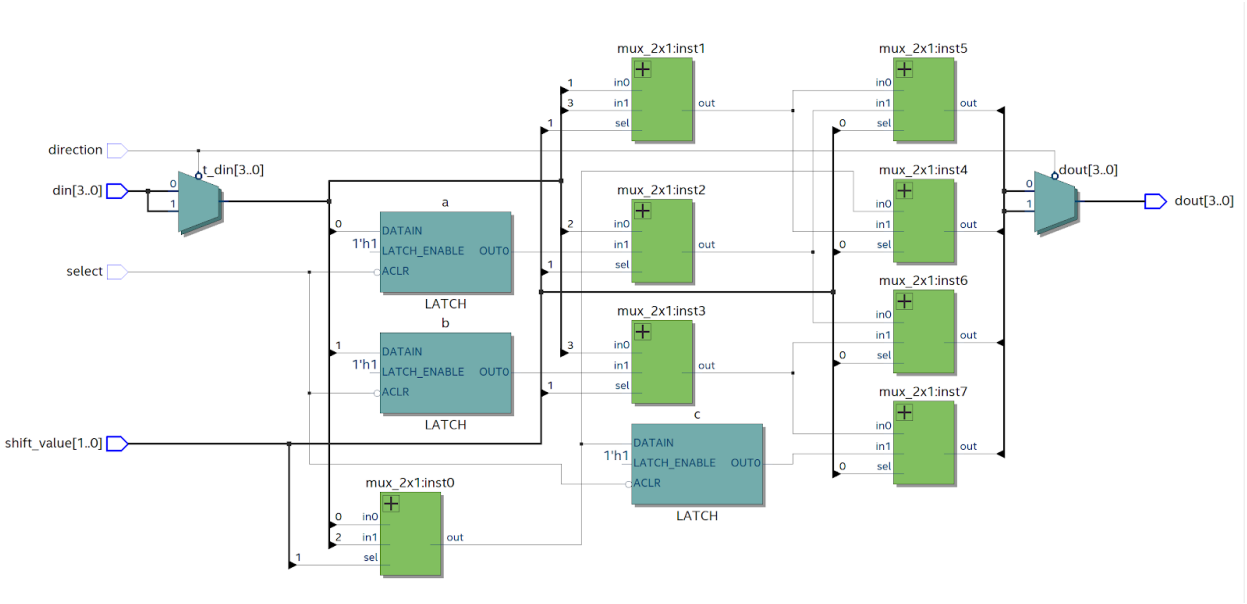
2. Resource Usage Table

	Resource	Usage
1	Estimated ALUTs Used	8
1	-- Combinational ALUTs	8
2	-- Memory ALUTs	0
3	-- LUT_REGs	0
2	Dedicated logic registers	0
3		
4	Estimated ALUTs Unavailable	6
1	-- Due to unpartnered combinational logic	6
2	-- Due to Memory ALUTs	0
5		
6	Total combinational functions	8
7	Combinational ALUT usage by number of inputs	
1	-- 7 input functions	2
2	-- 6 input functions	4
3	-- 5 input functions	2
4	-- 4 input functions	0
5	-- <=3 input functions	0
8		
9	Combinational ALUTs by mode	
1	-- normal mode	6
2	-- extended LUT mode	2
3	-- arithmetic mode	0
4	-- shared arithmetic mode	0
10		
11	Estimated ALUT/register pairs used	14
12		
13	Total registers	0
1	-- Dedicated logic registers	0
2	-- I/O registers	0
3	-- LUT_REGs	0
14		
15		
16	I/O pins	12
17		
18	DSP block 18-bit elements	0
19		
20	Maximum fan-out node	direction~input
21	Maximum fan-out	8
22	Total fan-out	64
23	Average fan-out	2.00

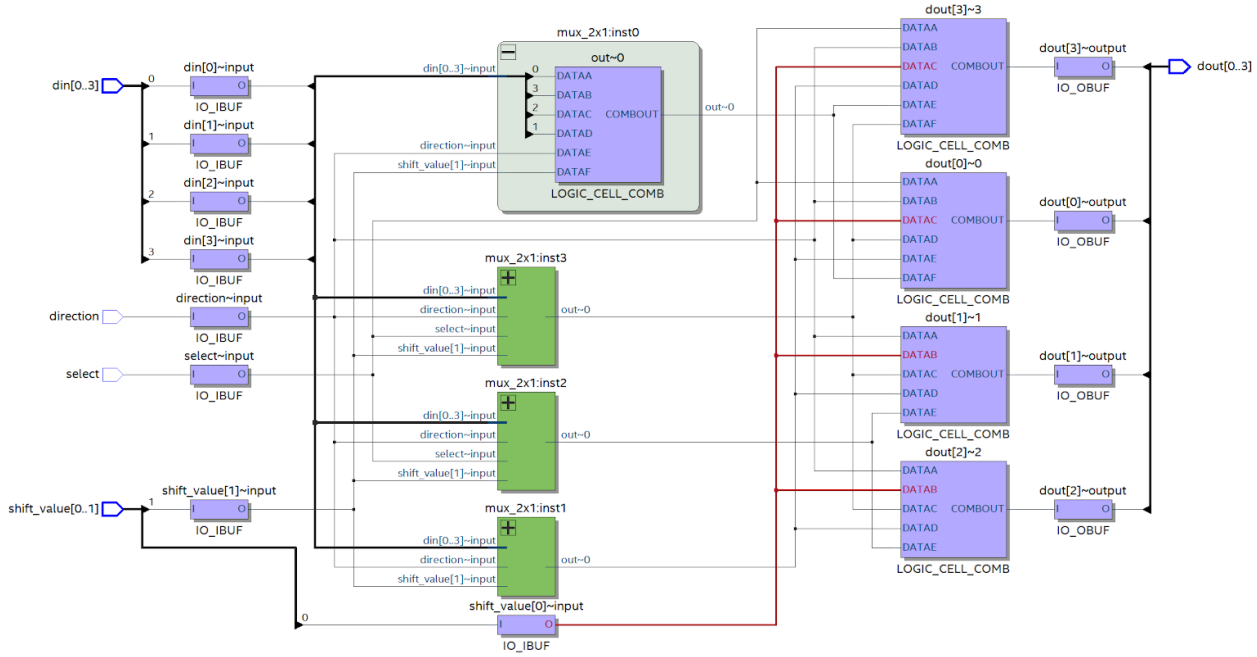
Number of ALUT: 8 (12 I/O pins)
Number of Functions: 8

- 2 7 input functions
- 4 6 input functions
- 2 5 input functions

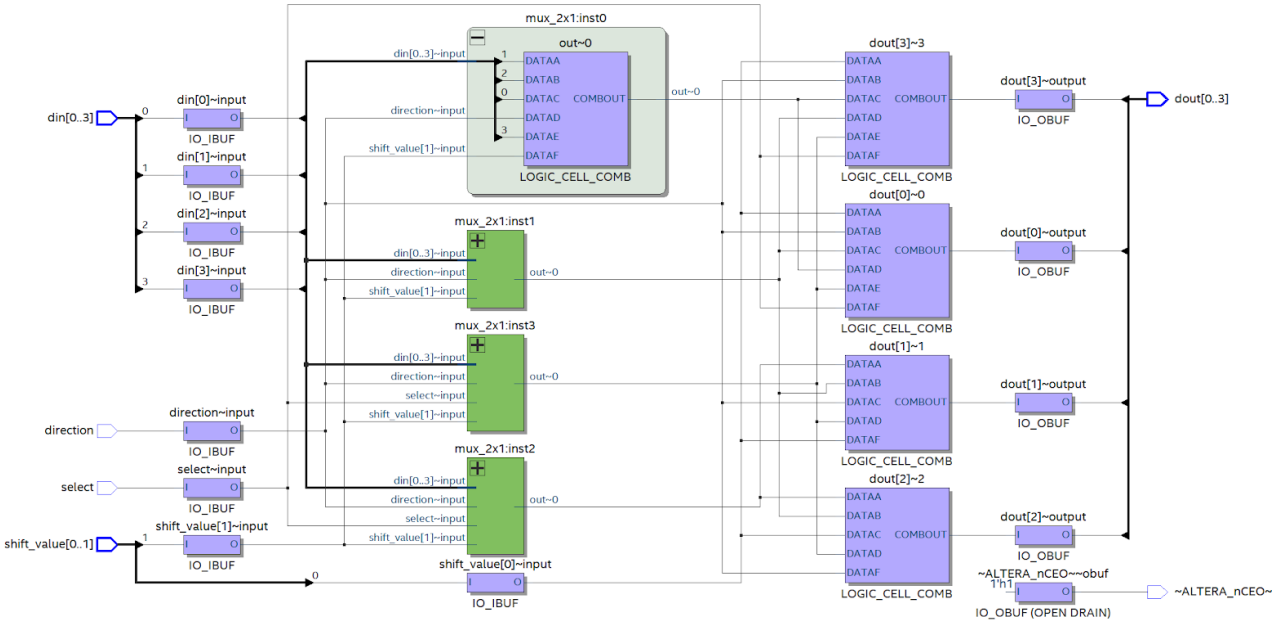
3. RTL schematic



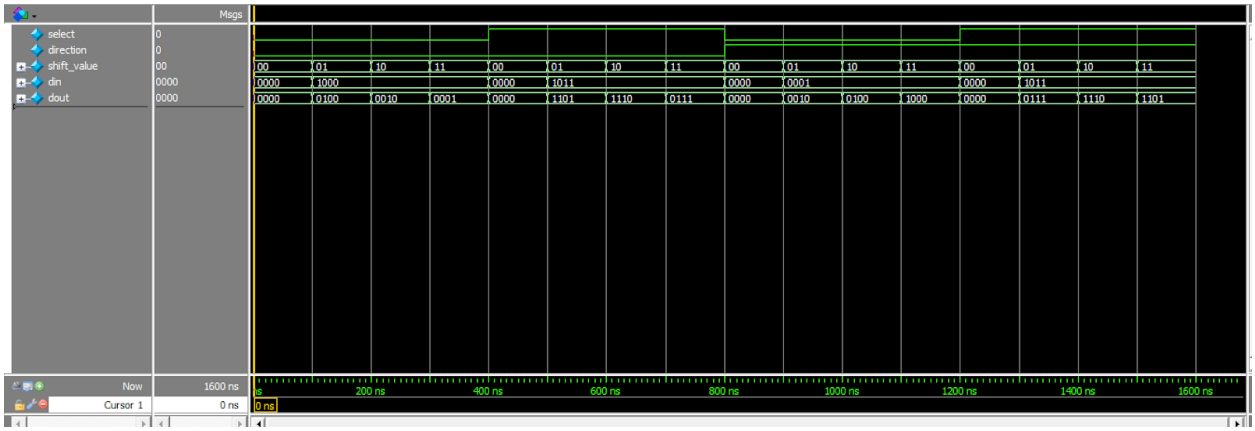
4. Post mapping schematic



5. Post fitting schematic



6. Simulation waveform



7. Transcript

Transcript

```
vlog -reportprogress 300 -work work {C:/Users/user/Desktop/ECE 111 HW/Lab4/Lab4/barrel_shif
ter_testbench.sv}
# Model Technology ModelSim - Intel FPGA Edition vlog 2020.1 Compiler 2020.02 Feb 28 2020
# Start time: 13:57:35 on Feb 08,2021
# vlog -reportprogress 300 -work work C:/Users/user/Desktop/ECE 111 HW/Lab4/Lab4/barrel_shi
fter_testbench.sv
# -- Compiling module barrel_shifter_testbench
#
# Top level modules:
#       barrel_shifter_testbench
# End time: 13:57:35 on Feb 08,2021, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
ModelSim> vsim work.barrel_shifter_testbench
# vsim work.barrel_shifter_testbench
# Start time: 13:57:40 on Feb 08,2021
# Loading sv_std.std
# Loading work.barrel_shifter_testbench
# Loading work.barrel_shifter
# Loading work.mux_2x1
add wave -position insertpoint sim:/barrel_shifter_testbench/*
VSIM 6> run -all
#  time=0,  select=0  shift_value=00  din=0000 dout=0000
#  time=100,  select=0  shift_value=01  din=1000 dout=0100
#  time=200,  select=0  shift_value=10  din=1000 dout=0010
#  time=300,  select=0  shift_value=11  din=1000 dout=0001
#  time=400,  select=1  shift_value=00  din=0000 dout=0000
#  time=500,  select=1  shift_value=01  din=1011 dout=1101
#  time=600,  select=1  shift_value=10  din=1011 dout=1110
#  time=700,  select=1  shift_value=11  din=1011 dout=0111
#  time=800,  select=0  shift_value=00  din=0000 dout=0000
#  time=900,  select=0  shift_value=01  din=0001 dout=0010
#  time=1000,  select=0  shift_value=10  din=0001 dout=0100
#  time=1100,  select=0  shift_value=11  din=0001 dout=1000
#  time=1200,  select=1  shift_value=00  din=0000 dout=0000
#  time=1300,  select=1  shift_value=01  din=1011 dout=0111
#  time=1400,  select=1  shift_value=10  din=1011 dout=1110
#  time=1500,  select=1  shift_value=11  din=1011 dout=1101
# ** Note: $finish    : C:/Users/user/Desktop/ECE 111 HW/Lab4/Lab4/barrel_shifter_testbench
.sv(134)
#    Time: 1600 ns  Iteration: 0  Instance: /barrel_shifter_testbench
# 1
# Break in Module barrel_shifter_testbench at C:/Users/user/Desktop/ECE 111 HW/Lab4/Lab4/ba
rrel_shifter_testbench.sv line 134

VSIM 7>
```

8. Simulation analysis
   The values below are from the waveform simulation found by moving the cursor.

Select	Direction	Shift_value	Operation	Din	Dout
0	0	00	No change	0000	0000
0	0	01	Shift right by 1 bit	1000	0100
0	0	10	Shift right by 2 bits	1000	0010
0	0	11	Shift right by 3 bits	1000	0001
1	0	00	No change	0000	0000
1	0	01	Rotate right by 1 bit	1011	1101
1	0	10	Rotate right by 2 bits	1011	1110
1	0	11	Rotate right by 3 bits	1011	0111
0	1	00	No change	0000	0000
0	1	01	Shift left by 1bit	0001	0010
0	1	10	Shift left by 2 bits	0001	0100
0	1	11	Shift left by 3 bits	0001	1000
1	1	00	No change	0000	0000
1	1	01	Rotate left by 1bits	1011	0111
1	1	10	Rotate left by 2 bits	1011	1110
1	1	11	Rotate left by 3 bits	1011	1101