Minh Le

Scott Shi

EE M116L

Lab Report # 1

**Combinational Gates Muxed**

**2.** With this module, we used a multiplexer to choose from eight different combinational gate networks that had the output to an LED. We used switches as select bit inputs into the multiplexer and also used separate switches to provide input into the combinational gate networks that we chose.

**3.**

===

\*.v

===

/\*

\* Module: Multiplexed combinational gates

\*

\* Filename: combinational\_gates\_muxed.v

\* Version: 1.0

\*

\* Author: Cejo Konuparamban Lonappan

\*

\* Description: RTL for implementing eight combinational gates with the

\* outputs of the gates multiplexed using an 8:1 Multiplexer. The gates

\* implemented are NAND, AND, NOR, OR, XOR, XNOR, NOT, and a non-inverting

\* buffer.

\*/

module combinational\_gates\_muxed (led, sw);

// Input and output declaration

input [4:0] sw; // Inputs sw[7:5] are unused

output reg led; // Outputs led[7:1] are unused

// Declaring wires for the eight two inpuit basic gates

// Output Select Input Number

wire nand\_out; // 3'b111

wire and\_out; // 3'b110

wire nor\_out; // 3'b101

wire or\_out; // 3'b100

wire xor\_out; // 3'b011

wire xnor\_out; // 3'b010

wire buff\_out; // 3'b001

wire not\_out; // 3'b000

// 8-to-1 Multiplexer Inputs

wire [7:0] MuxIn;

wire [2:0] SelectIn;

// Generate outputs for the eight gates

assign nand\_out = ~(sw[0] & sw[1]); // 3'b111

assign and\_out = sw[0] & sw[1]; // 3'b110

assign nor\_out = ~(sw[0] | sw[1]); // 3'b101

assign or\_out = sw[0] | sw[1]; // 3'b100

assign xor\_out = sw[0] ^ sw[1]; // 3'b011

assign xnor\_out = sw[0] ^~ sw[1]; // 3'b010

assign buff\_out = sw[0]; // 3'b001

assign not\_out = ~sw[0]; // 3'b000

// 3'b111 3'b110 3'b101 3'b100 3'b011 3'b010 3'b001 3'b000

assign MuxIn = {nand\_out, and\_out, nor\_out, or\_out, xor\_out, xnor\_out, buff\_out, not\_out};

// Assigning select input lines for the multiplexer using switch lines SW7, SW6, and SW5

assign SelectIn = sw[4:2];

// Output multiplexer

always @(MuxIn, SelectIn)

led = MuxIn[SelectIn];

endmodule

====

\*TB.v

====

===

\*.ucf

===

## Leds

NET "led" LOC = "U16" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L2P\_CMPCLK, Sch name = LD0

## Switches

NET "sw<0>" LOC = "T10" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L29N\_GCLK2, Sch name = SW0

NET "sw<1>" LOC = "T9" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L32P\_GCLK29, Sch name = SW1

NET "sw<2>" LOC = "V9" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L32N\_GCLK28, Sch name = SW2

NET "sw<3>" LOC = "M8" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L40P, Sch name = SW3

NET "sw<4>" LOC = "N8" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L40N, Sch name = SW4

**4.**

\*\*\*\*\*\*\*\*MISSING\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

=========================================

\* Design Summary \*

=========================================

Top Level Output File Name : combinational\_gates\_muxed.ngc

Primitive and Black Box Usage:

------------------------------

# BELS : 1

# LUT5 : 1

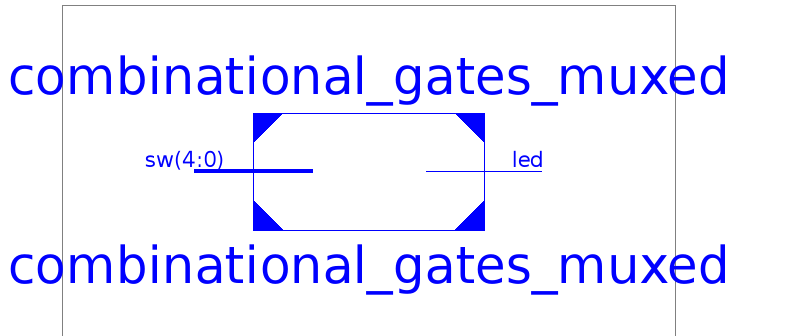
# IO Buffers : 6

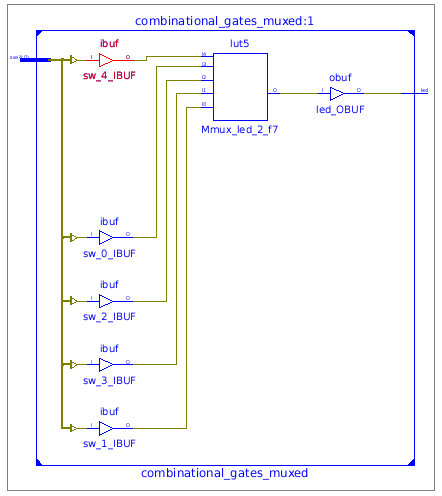
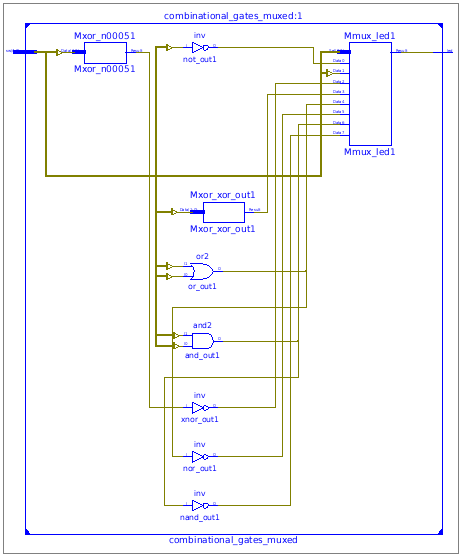
# IBUF : 5

# OBUF : 1

**5.** Design Overview --> Summary --> Device Utilization Summary -- TABLE (only add things that are used)

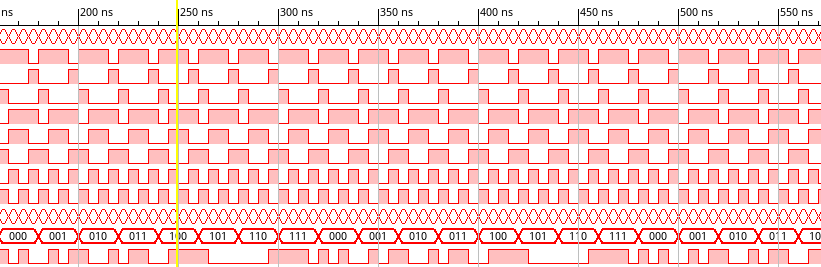
**6.**





**7.** N/A

**8.** Waveform from the IMPACT window (need to include signal names and zoom in)



**Four Bit Counter**

1.

**2.** This module was designed to count from 0 to 15. The counter was implemented using registers and we incremented the value of the counter on the positive edge of every clock tick. The moment the counter hit 15, we reset it back to zero.

**3**.

===

\*.v

===

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 14:32:34 04/06/2016

// Design Name:

// Module Name: my4BitCounter

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module my4BitCounter(rst,a,clk);

input clk;

input rst;

output reg [3:0] a;

always @ (posedge clk)

if (rst)

a <= 4'b0000;

else

a <= a + 1'b1;

Endmodule

====

\*TB.v

====

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 14:49:56 04/06/2016

// Design Name: my4BitCounter

// Module Name: /home/minh/School/spring\_2016/cs\_m152a/my4BitCounter/my4BitCounter\_TB.v

// Project Name: my4BitCounter

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: my4BitCounter

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

`include "my4BitCounter.v"

module my4BitCounter\_TB;

// Inputs

reg rst;

reg clk;

// Outputs

wire[3:0] a;

// Instantiate the Unit Under Test (UUT)

my4BitCounter uut (

.rst(rst),

.a(a),

.clk(clk)

);

initial begin

// Initialize Inputs

rst = 1'b1;

clk = 1'b0;

#2 rst = 1'b0;

end

initial

#200 $finish;

always

#1 clk = ~clk;

endmodule

====

\*.ucf

====

## Clock signal

NET "clk" LOC = "V10" | IOSTANDARD = "LVCMOS33"; #Bank = 2, pin name = IO\_L30N\_GCLK0\_USERCCLK, Sch name = GCLK

Net "clk" TNM\_NET = sys\_clk\_pin;

TIMESPEC TS\_sys\_clk\_pin = PERIOD sys\_clk\_pin 100000 kHz;

## Leds

NET "a<0>" LOC = "U16" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L2P\_CMPCLK, Sch name = LD0

NET "a<1>" LOC = "V16" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L2N\_CMPMOSI, Sch name = LD1

NET "a<2>" LOC = "U15" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L5P, Sch name = LD2

NET "a<3>" LOC = "V15" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L5N, Sch name = LD3

## Buttons

NET "rst" LOC = "B8" | IOSTANDARD = "LVCMOS33"; #Bank = 0, Pin name = IO\_L33P, Sch name = BTNS

**4.**

======================================

\* Low Level Synthesis \*

========================================

Optimizing unit <my4BitCounter> ...

Mapping all equations...

Building and optimizing final netlist ...

Found area constraint ratio of 100 (+ 5) on block my4BitCounter, actual ratio is 0.

Final Macro Processing ...

========================================

Final Register Report

Macro Statistics

# Registers : 4

Flip-Flops : 4

========================================

\* Design Summary \*

========================================

Top Level Output File Name : my4BitCounter.ngc

Primitive and Black Box Usage:

------------------------------

# BELS : 4

# INV : 1

# LUT2 : 1

# LUT3 : 1

# LUT4 : 1

# FlipFlops/Latches : 4

# FDR : 4

# Clock Buffers : 1

# BUFGP : 1

# IO Buffers : 5

# IBUF : 1

# OBUF : 4

Device utilization summary:

---------------------------

Selected Device : 7a100tcsg324-3

Slice Logic Utilization:

Number of Slice Registers: 4 out of 126800 0%

Number of Slice LUTs: 4 out of 63400 0%

Number used as Logic: 4 out of 63400 0%

Slice Logic Distribution:

Number of LUT Flip Flop pairs used: 8

Number with an unused Flip Flop: 4 out of 8 50%

Number with an unused LUT: 4 out of 8 50%

Number of fully used LUT-FF pairs: 0 out of 8 0%

Number of unique control sets: 1

IO Utilization:

Number of IOs: 6

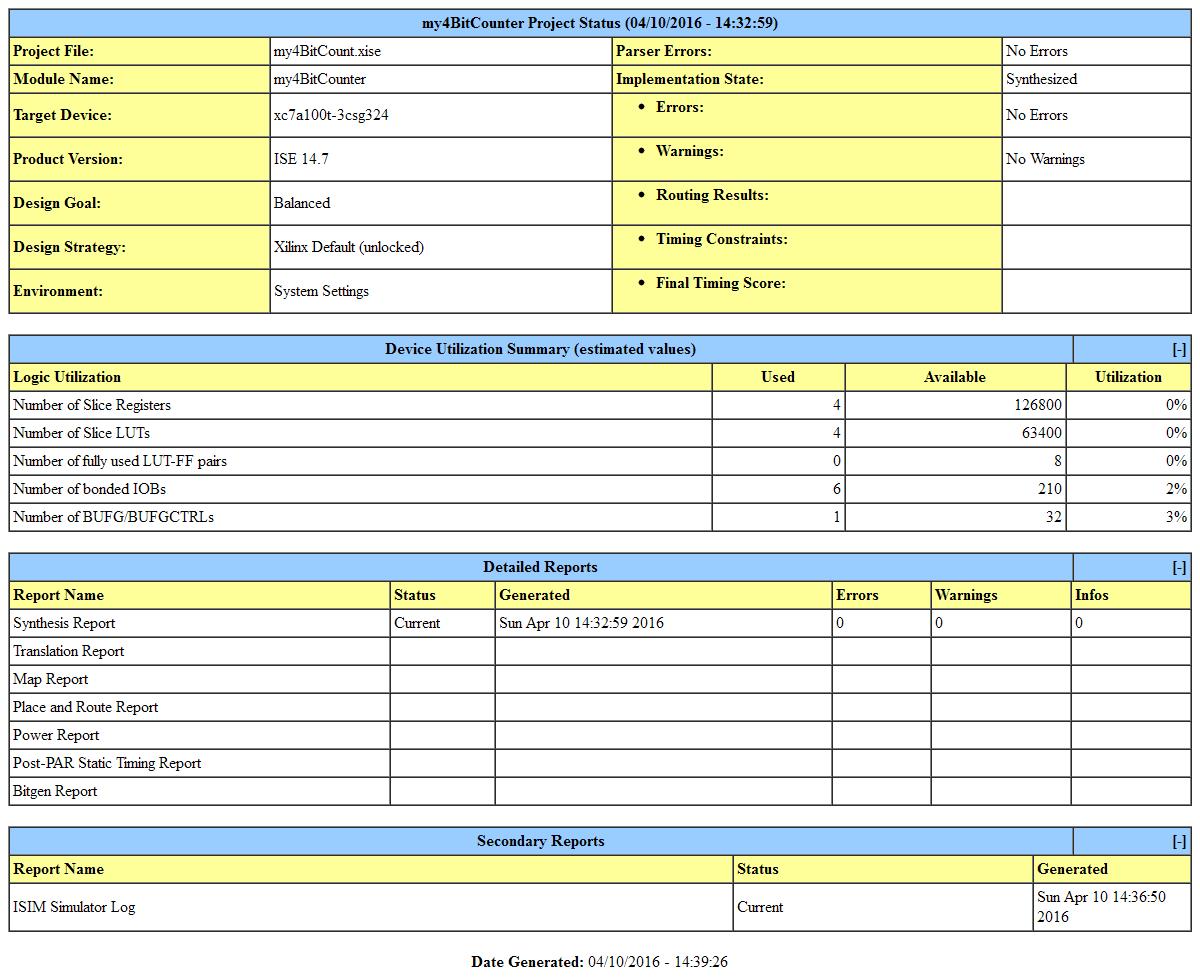
Number of bonded IOBs: 6 out of 210 2%

Specific Feature Utilization:

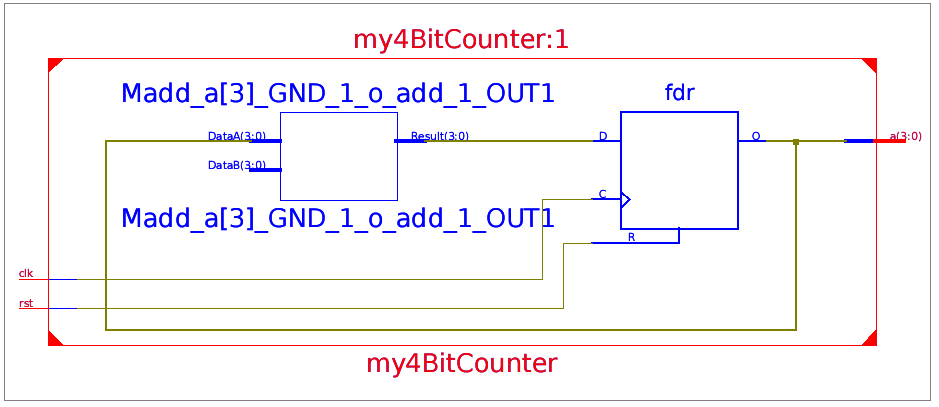
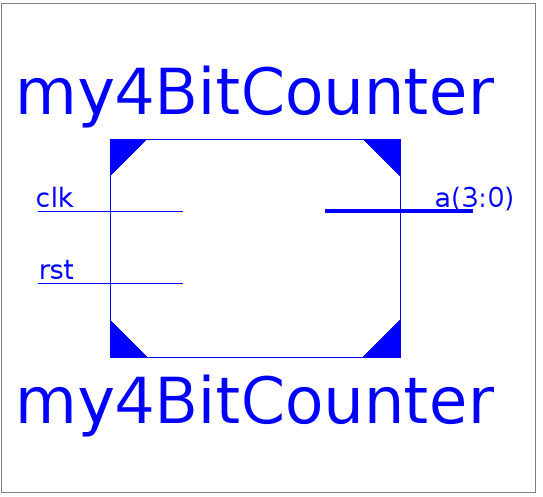
Number of BUFG/BUFGCTRLs: 1 out of 32 3%

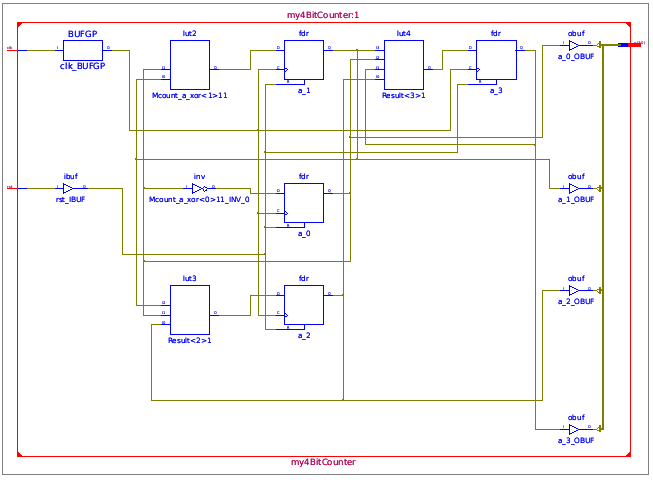
====================================

**5.** Design Overview --> Summary --> Device Utilization Summary



**6.** High level schematic

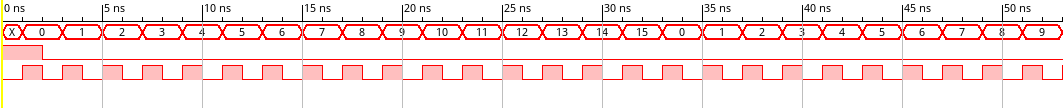




**7.** Detailed Reports → Design Summary → Timing Summary

578.035MHz

**8.** Waveform from the IMPACT window



**Clock Divider**

**1.**

**2.** The goal of this task was to create a 4 bit counter that incremented every second. To do this, we had to implement two counters in the same fashion described above. What was different was that one of the counters had to count to 100 MHz (this is the frequency at which the clock ran). We had the first counter count from 0 to 99,999,999 and on it’s terminal count, we incremented the second clock. Therefore, every 100 M clock pulses, or every second, we incremented the secondary counter having it count up every second.

**3.**

===

\*.v

===

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 14:42:00 04/08/2016

// Design Name:

// Module Name: clock\_divider\_2

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

`define N\_BIT 27

`define DIVIDE 99999999

module clock\_divider\_2(clk,rst,LED);

//input outputs

input wire clk;

input wire rst;

output reg [3:0] LED;

//internal wirings

reg [`N\_BIT-1:0] in\_cnt;

wire trig;

assign trig = (in\_cnt == `DIVIDE);

//main clk to initialize and count inner count

always@(posedge clk, posedge rst)

begin

if (rst | trig)

begin

in\_cnt <= `N\_BIT'b0;

end

else

in\_cnt <= in\_cnt + 1'b1;

end

always @(posedge trig, posedge rst)

begin

if (rst)

LED <= 4'b0000;

else

begin

LED <= LED + 1'b1;

end

end

endmodule

====

\*TB.v

====

`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 14:54:03 04/08/2016

// Design Name: clock\_divider\_2

// Module Name: /home/minh/School/spring\_2016/cs\_m152a/clock\_divider\_2/clock\_divider2\_TB.v

// Project Name: clock\_divider\_2

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: clock\_divider\_2

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

module clock\_divider2\_TB;

// Inputs

reg clk;

reg rst;

// Outputs

wire [3:0] LED;

// Instantiate the Unit Under Test (UUT)

clock\_divider\_2 uut (

.clk(clk),

.rst(rst),

.LED(LED)

);

initial begin

// Initialize Inputs

clk = 0;

rst = 1;

#2 rst = 0;

// Wait 100 ns for global reset to finish

#10000 $finish;

// Add stimulus here

end

always

#1 clk = ~clk;

endmodule

====

\*.ucf

====

## Clock signal

NET "clk" LOC = "V10" | IOSTANDARD = "LVCMOS33"; #Bank = 2, pin name = IO\_L30N\_GCLK0\_USERCCLK, Sch name = GCLK

Net "clk" TNM\_NET = sys\_clk\_pin;

TIMESPEC TS\_sys\_clk\_pin = PERIOD sys\_clk\_pin 100000 kHz;

## Leds

NET "LED<0>" LOC = "U16" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L2P\_CMPCLK, Sch name = LD0

NET "LED<1>" LOC = "V16" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L2N\_CMPMOSI, Sch name = LD1

NET "LED<2>" LOC = "U15" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L5P, Sch name = LD2

NET "LED<3>" LOC = "V15" | IOSTANDARD = "LVCMOS33"; #Bank = 2, Pin name = IO\_L5N, Sch name = LD3

## Buttons

NET "rst" LOC = "B8" | IOSTANDARD = "LVCMOS33"; #Bank = 0, Pin name = IO\_L33P, Sch name = BTNS

**4.** ======================================

\* Low Level Synthesis \*

========================================

Optimizing unit <clock\_divider\_2> ...

Mapping all equations...

Building and optimizing final netlist ...

Found area constraint ratio of 100 (+ 5) on block clock\_divider\_2, actual ratio is 0.

Final Macro Processing ...

================================================

Final Register Report

Macro Statistics

# Registers : 31

Flip-Flops : 31

======

=======================================

\* Design Summary \*

=======================================

Top Level Output File Name : clock\_divider\_2.ngc

Primitive and Black Box Usage:

------------------------------

# BELS : 119

# GND : 1

# INV : 2

# LUT1 : 26

# LUT2 : 29

# LUT3 : 1

# LUT4 : 1

# LUT6 : 5

# MUXCY : 26

# VCC : 1

# XORCY : 27

# FlipFlops/Latches : 31

# FDC : 31

# Clock Buffers : 1

# BUFGP : 1

# IO Buffers : 5

# IBUF : 1

# OBUF : 4

Device utilization summary:

---------------------------

Selected Device : 7a100tcsg324-3

Slice Logic Utilization:

Number of Slice Registers: 31 out of 126800 0%

Number of Slice LUTs: 64 out of 63400 0%

Number used as Logic: 64 out of 63400 0%

Slice Logic Distribution:

Number of LUT Flip Flop pairs used: 64

Number with an unused Flip Flop: 33 out of 64 51%

Number with an unused LUT: 0 out of 64 0%

Number of fully used LUT-FF pairs: 31 out of 64 48%

Number of unique control sets: 2

IO Utilization:

Number of IOs: 6

Number of bonded IOBs: 6 out of 210 2%

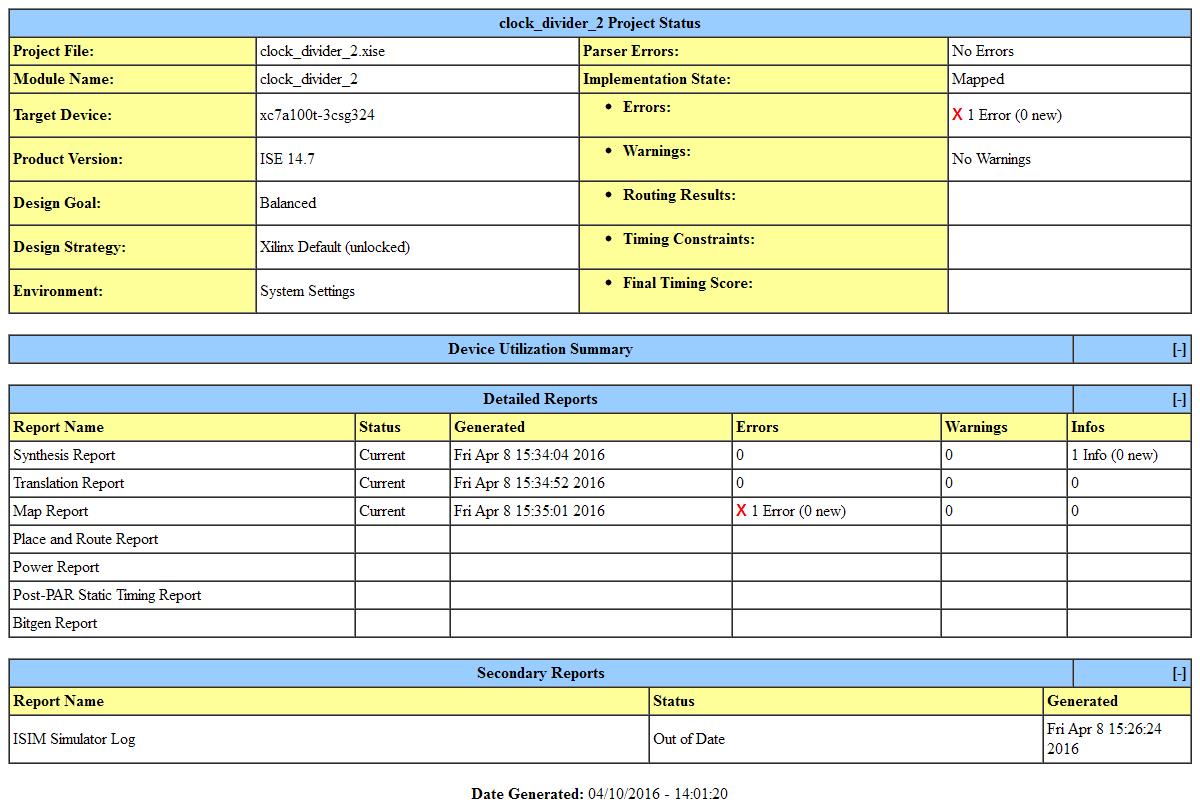
Specific Feature Utilization:

Number of BUFG/BUFGCTRLs: 1 out of 32 3%

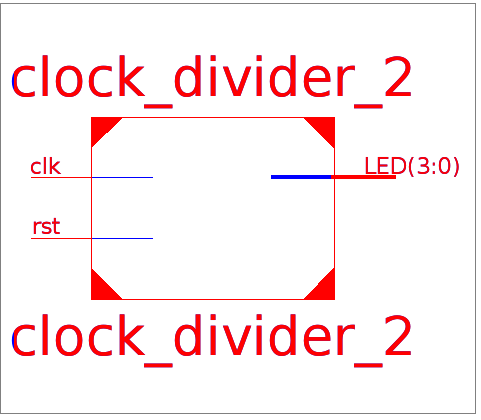
---------------------------

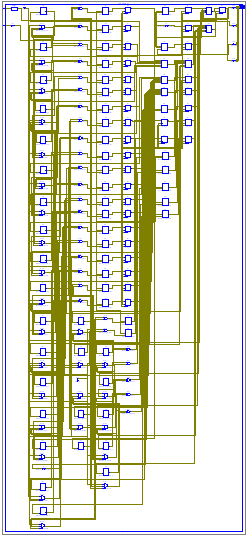
================================================

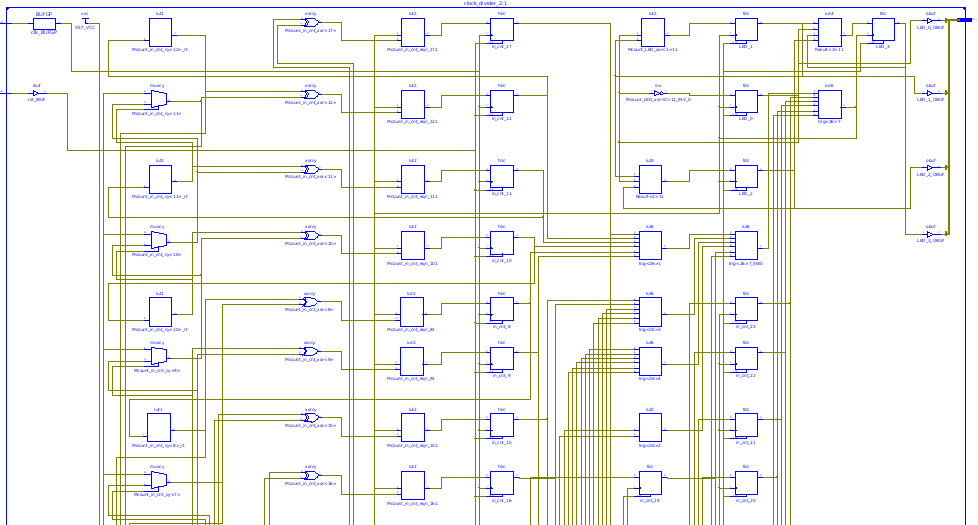
**5.** Design Overview --> Summary --> Device Utilization Summary



**6.** High level schematic







**7.** Detailed Reports --> Design Summary → Timing Summary (in console)

344.947MHz

**8.** Waveform from the IMPACT window

