CS2610: Computer Organization and Architecture
Lab Report 1

Devansh Singh Rathore[11]

Objective: In this lab we were supposed to analyse the efficiency of different **Adder** topologies in terms of **processing delay** and **power consumption**.

Problem: Given **two 4-bit** unsigned numbers **A** and **B** in **two 4-bit** registers. We were required to implement the addition of A and B in **Verilog** for the below mentioned scenarios:

1. Provided with **4 full adders** and **two 4-bit** registers **R0** and **R1**.
2. Provided with only **1 full adder** and **two 4-bit** registers **R0** and **R1**.

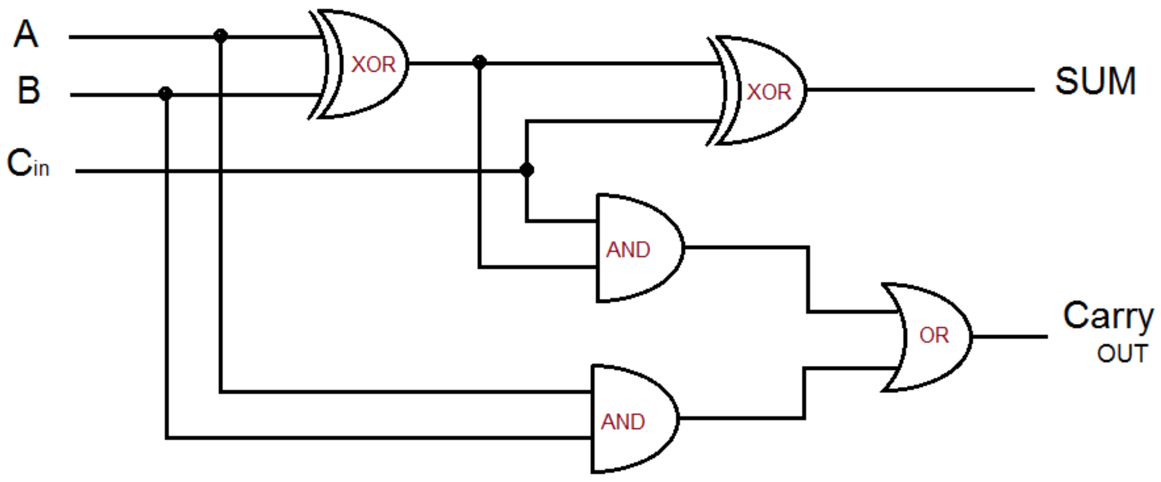
Theory:

A **Full Adder** circuit is a logical circuit that performs an addition operation on **three** single bit binary numbers and result out the carry and the summation.

Equations for **Full Adder**:

Sum = a^b^c

Carry = a&b + (a^b)&c;



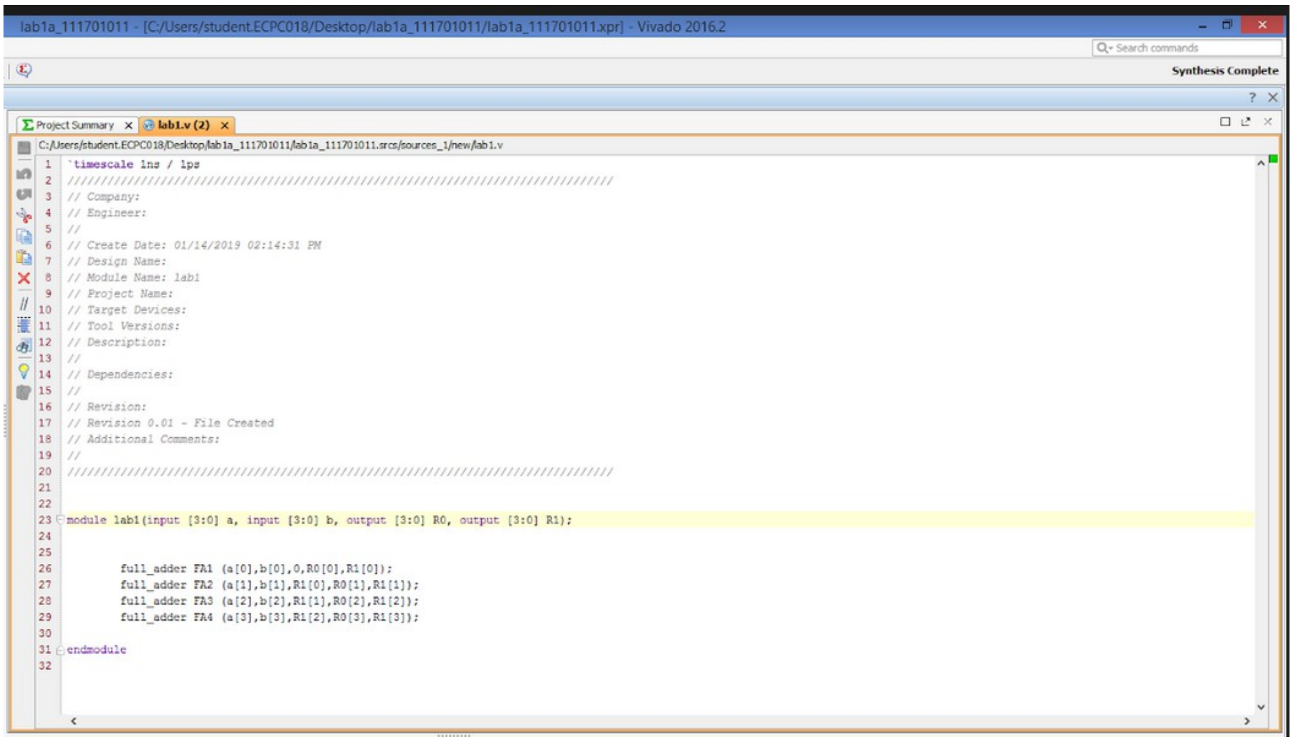
For solving the **first problem**, we can use a **Ripple Carry Adder** circuit to perform a **4 bit** addition, which involves **four serial Adders** passing **carry** value as input to the next **Adder** in series. Initially the carry bit is set equal to **0**.

For solving the **second problem**, we can use the logic of a **Multiplexer** in nested **conditional statements** carring a register value from **00, 01, 10, 11** and so on to add **bitwise** digits of **two** input numbers sequentially by passing through same **Full Adder** and then storing the sum in one of the registers finally. This will run under "**always**" condition triggered on "**posedge clock**" after the primary initialization of **carry** value. And after attaining the addition result, the value of **iterator** will again shift to **00** for the next **clock** cycle.

Procedure:

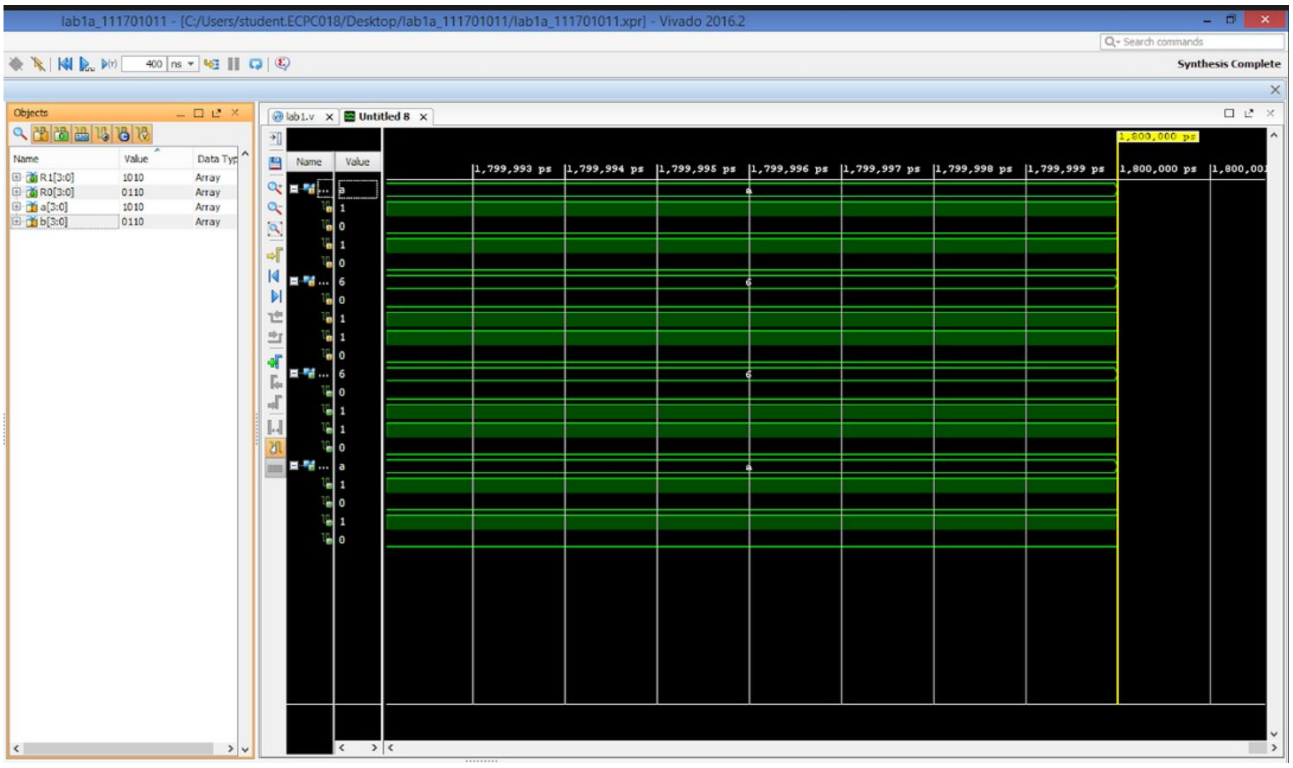
Provided with 4 full adders and two 4-bit registers R0 and R1->

CODE::


lab1a_111701011 - [C:/Users/student.ECPC018/Desktop/lab1a_111701011/lab1a_111701011.xpr] - Vivado 2016.2
Search commands
Synthesis Complete
Project Summary
lab1.v (2)
C:/Users/student.ECPC018/Desktop/lab1a_111701011/lab1a_111701011.srcs/sources_1/new/lab1.v

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 01/14/2019 02:14:31 PM
// Design Name:
// Module Name: lab1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module lab1(input [3:0] a, input [3:0] b, output [3:0] R0, output [3:0] R1);


        full_adder FA1 (a[0],b[0],0,R0[0],R1[0]);
        full_adder FA2 (a[1],b[1],R1[0],R0[1],R1[1]);
        full_adder FA3 (a[2],b[2],R1[1],R0[2],R1[2]);
        full_adder FA4 (a[3],b[3],R1[2],R0[3],R1[3]);

endmodule
```

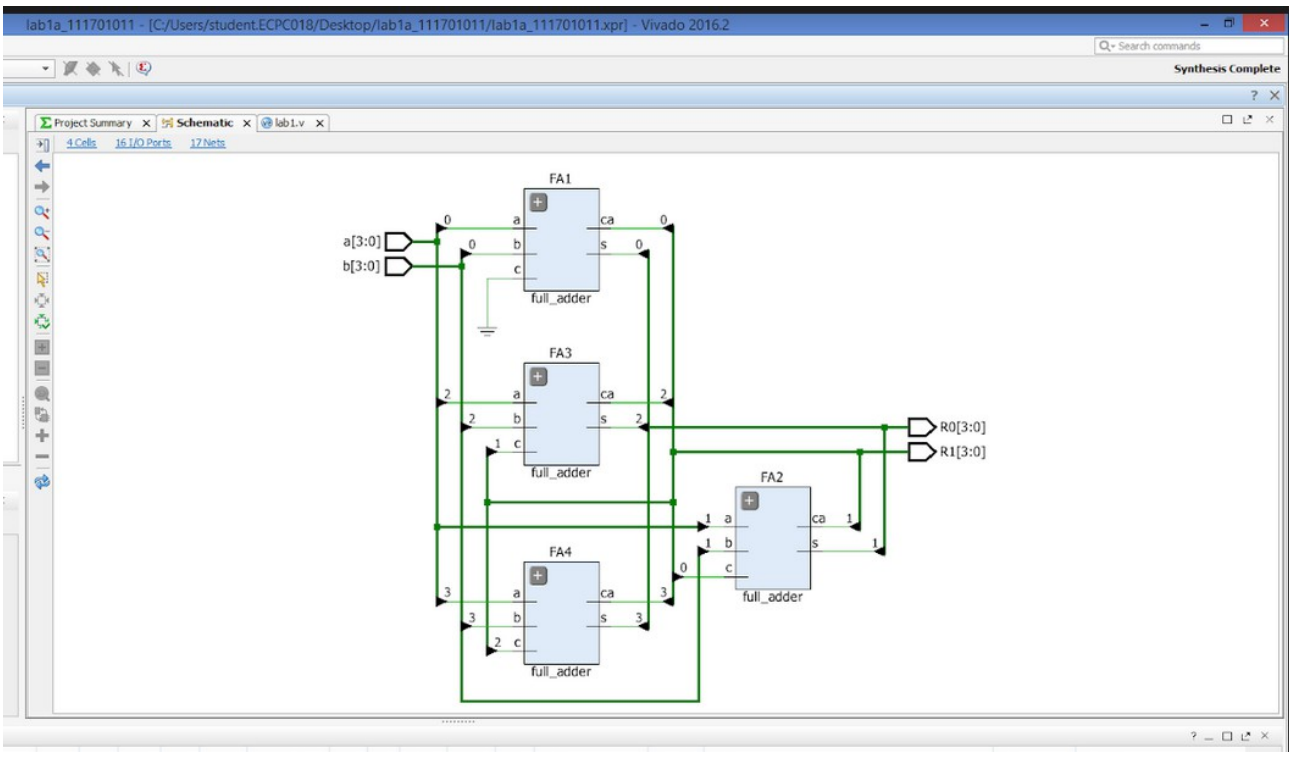
SIMULATION RESULT::


lab1a_111701011 - [C:/Users/student.ECPC018/Desktop/lab1a_111701011/lab1a_111701011.xpr] - Vivado 2016.2

Synthesis Complete

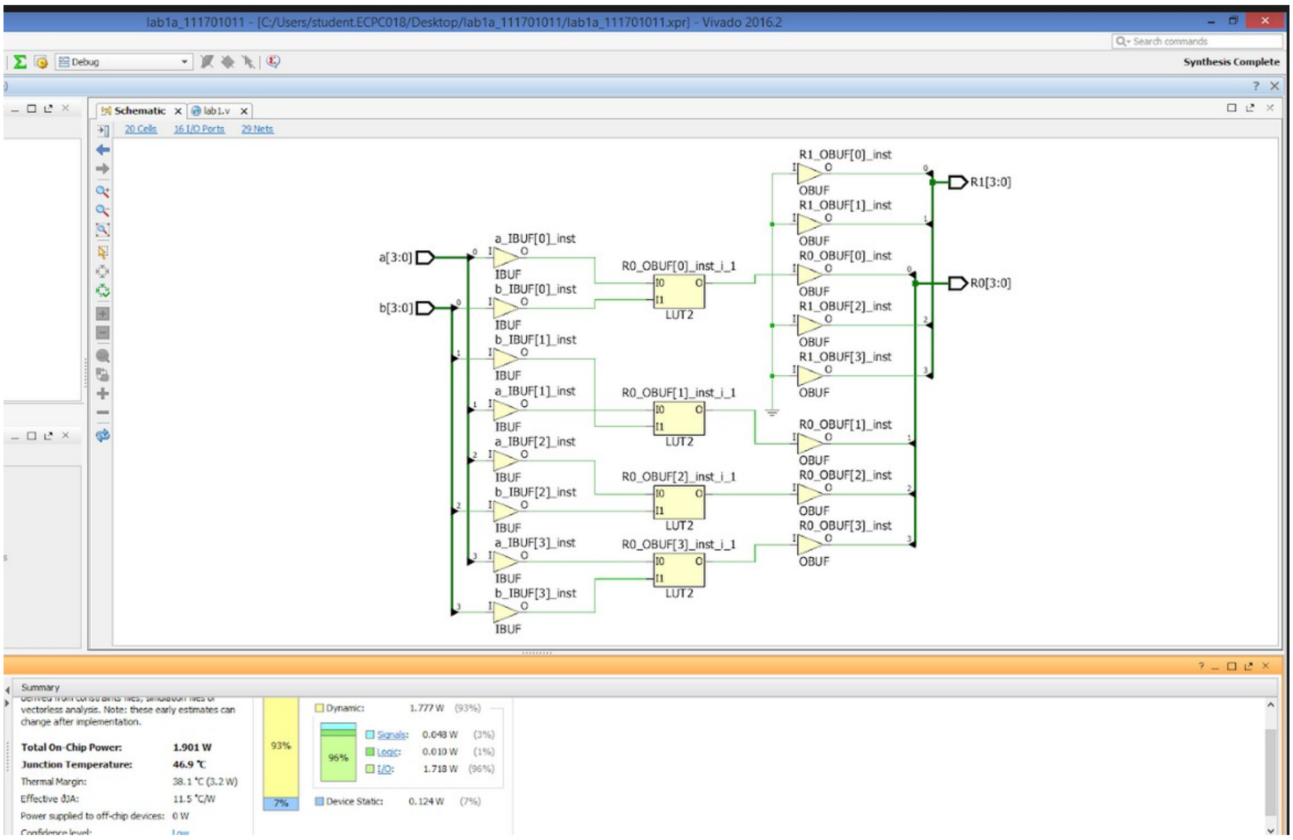
Name	Value	Data Type
R1[3:0]	1010	Array
R0[3:0]	0110	Array
a[3:0]	1010	Array
b[3:0]	0110	Array

RTL SCHEMATIC::


lab1a_111701011 - [C:/Users/student.ECPC018/Desktop/lab1a_111701011/lab1a_111701011.xpr] - Vivado 2016.2

Synthesis Complete

POWER & MEMORY CONSUMTION SCHEMATIC::


lab1a_111701011 - [C:/Users/student.ECPC018/Desktop/lab1a_111701011/lab1a_111701011.xpr] - Vivado 2016.2

Synthesis Complete

Summary

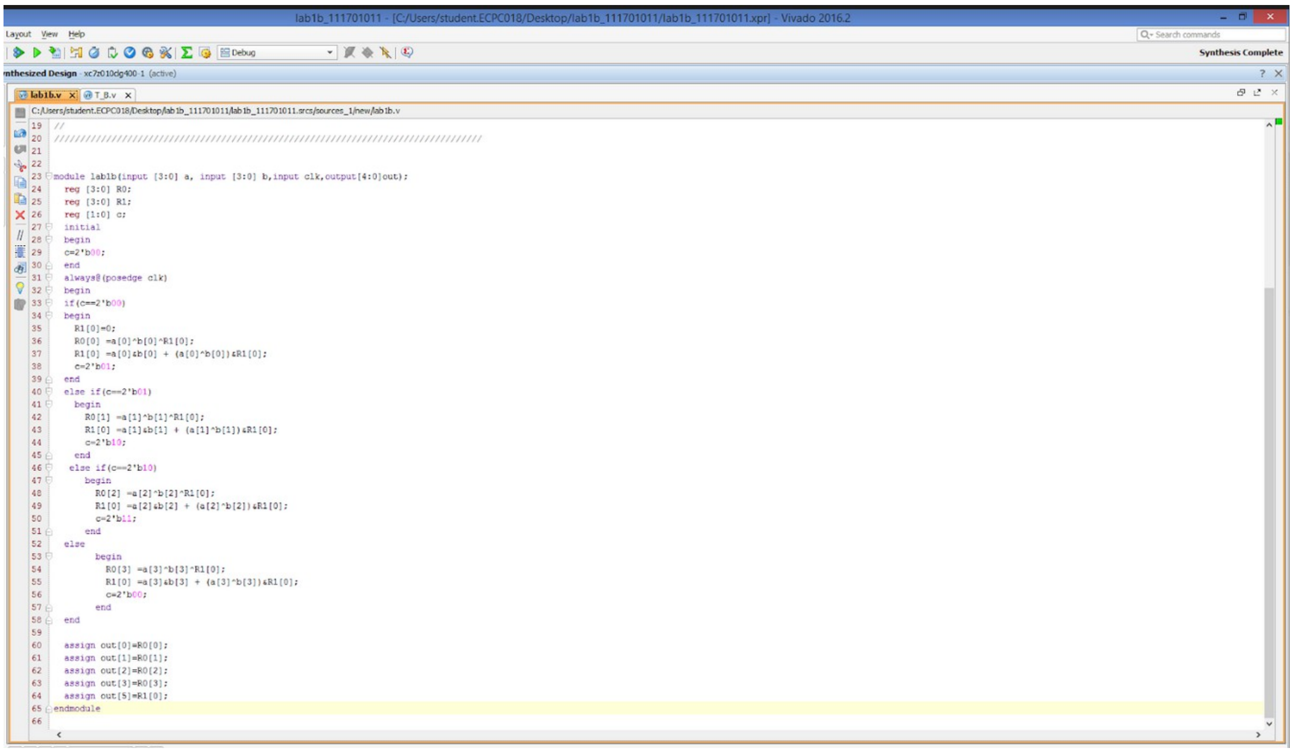
vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power: **1.901 W**
Junction Temperature: **46.9 ℃**
Thermal Margin: 38.1 ℃ (3.2 W)
Effective ϑJA: 11.5 ℃/W
Power supplied to off-chip devices: 0 W
Confidence level: Low

Dynamic: 1.777 W (93%)
Signals: 0.048 W (3%)
Logic: 0.010 W (1%)
I/O: 1.718 W (96%)
Device Static: 0.124 W (7%)

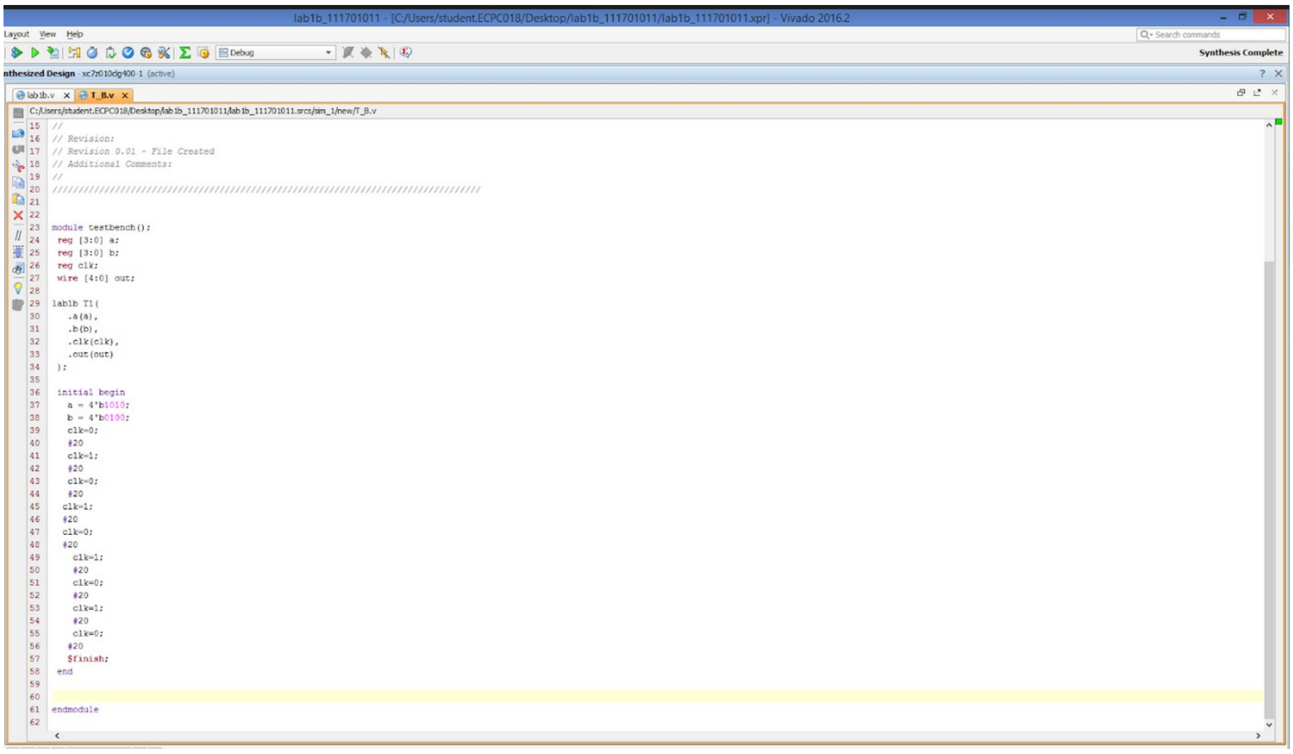
Provided with only 1 full adder and two 4-bit registers R0 and R1->

CODE::



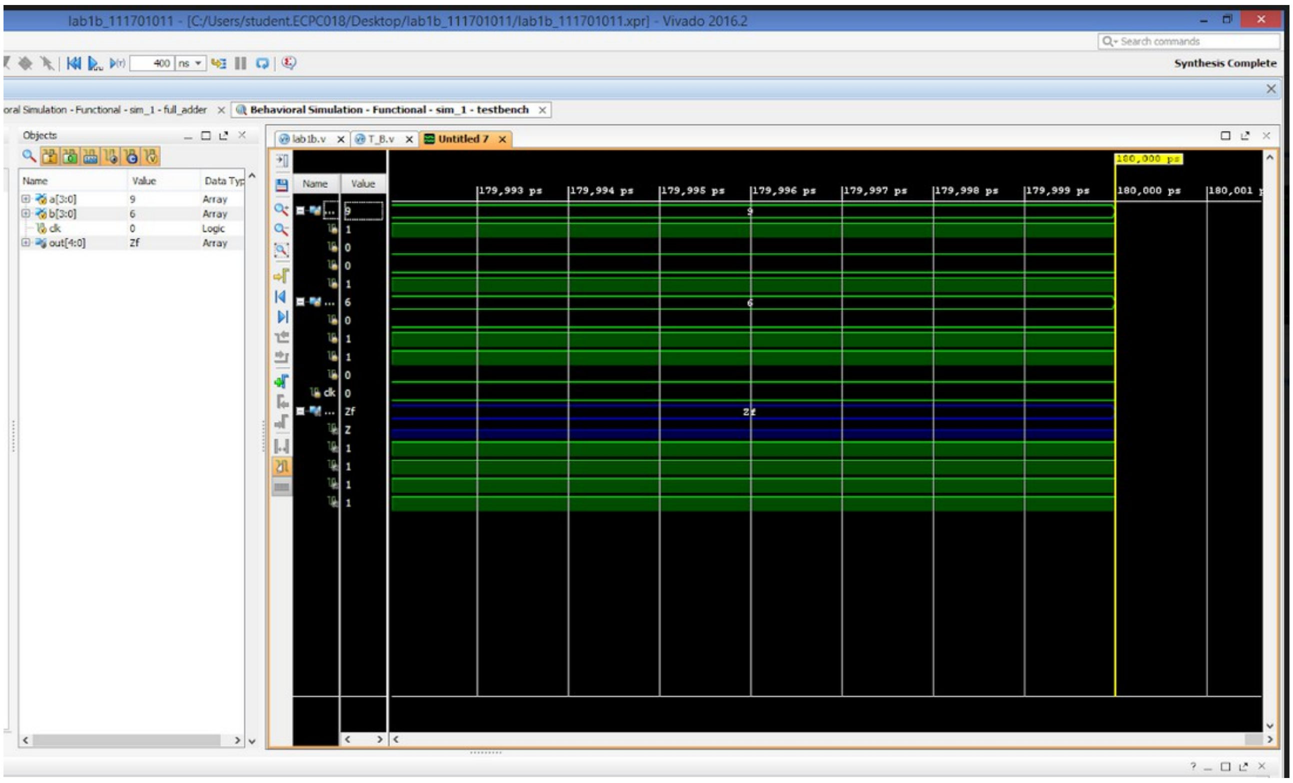
```
//
////////////////////////////////////////////////////////////////////////////////


module lab1b(input [3:0] a, input [3:0] b,input clk,output[4:0]out);
  reg [3:0] R0;
  reg [3:0] R1;
  reg [1:0] c;
  initial
  begin
  c=2'b00;
  end
  always@(posedge clk)
  begin
  if(c==2'b00)
  begin
    R1[0]=0;
    R0[0] =a[0]^b[0]^R1[0];
    R1[0] =a[0]&b[0] + (a[0]^b[0])&R1[0];
    c=2'b01;
  end
  else if(c==2'b01)
    begin
      R0[1] =a[1]^b[1]^R1[0];
      R1[0] =a[1]&b[1] + (a[1]^b[1])&R1[0];
      c=2'b10;
    end
   else if(c==2'b10)
      begin
        R0[2] =a[2]^b[2]^R1[0];
        R1[0] =a[2]&b[2] + (a[2]^b[2])&R1[0];
        c=2'b11;
      end
  else
        begin
          R0[3] =a[3]^b[3]^R1[0];
          R1[0] =a[3]&b[3] + (a[3]^b[3])&R1[0];
          c=2'b00;
        end
  end

  assign out[0]=R0[0];
  assign out[1]=R0[1];
  assign out[2]=R0[2];
  assign out[3]=R0[3];
  assign out[5]=R1[0];
endmodule
```

TEST_BENCH::

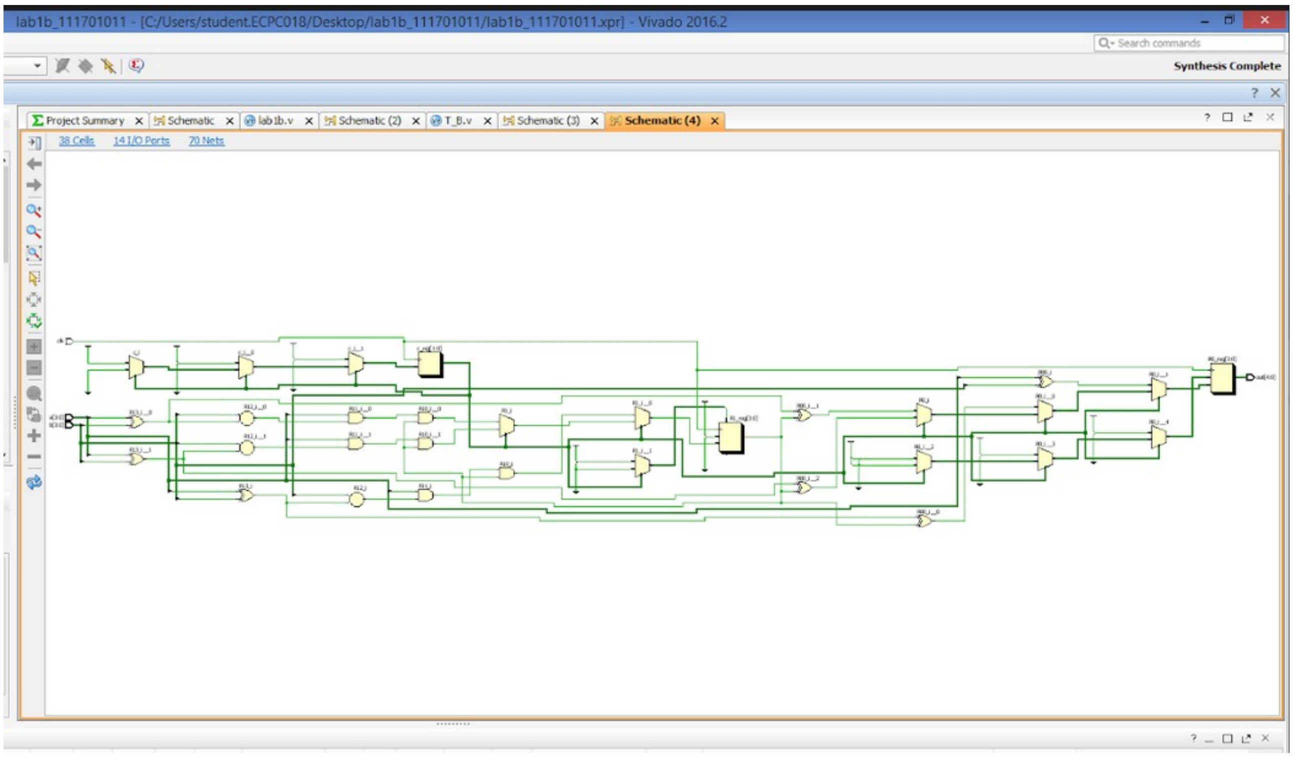


```
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////


module testbench();
 reg [3:0] a;
 reg [3:0] b;
 reg clk;
 wire [4:0] out;

lab1b T1(
   .a(a),
   .b(b),
   .clk(clk),
   .out(out)
 );

 initial begin
   a = 4'b1010;
   b = 4'b0100;
   clk=0;
   #20
   clk=1;
   #20
   clk=0;
   #20
  clk=1;
  #20
  clk=0;
  #20
    clk=1;
    #20
    clk=0;
    #20
    clk=1;
    #20
    clk=0;
   #20
   $finish;
 end


endmodule
```

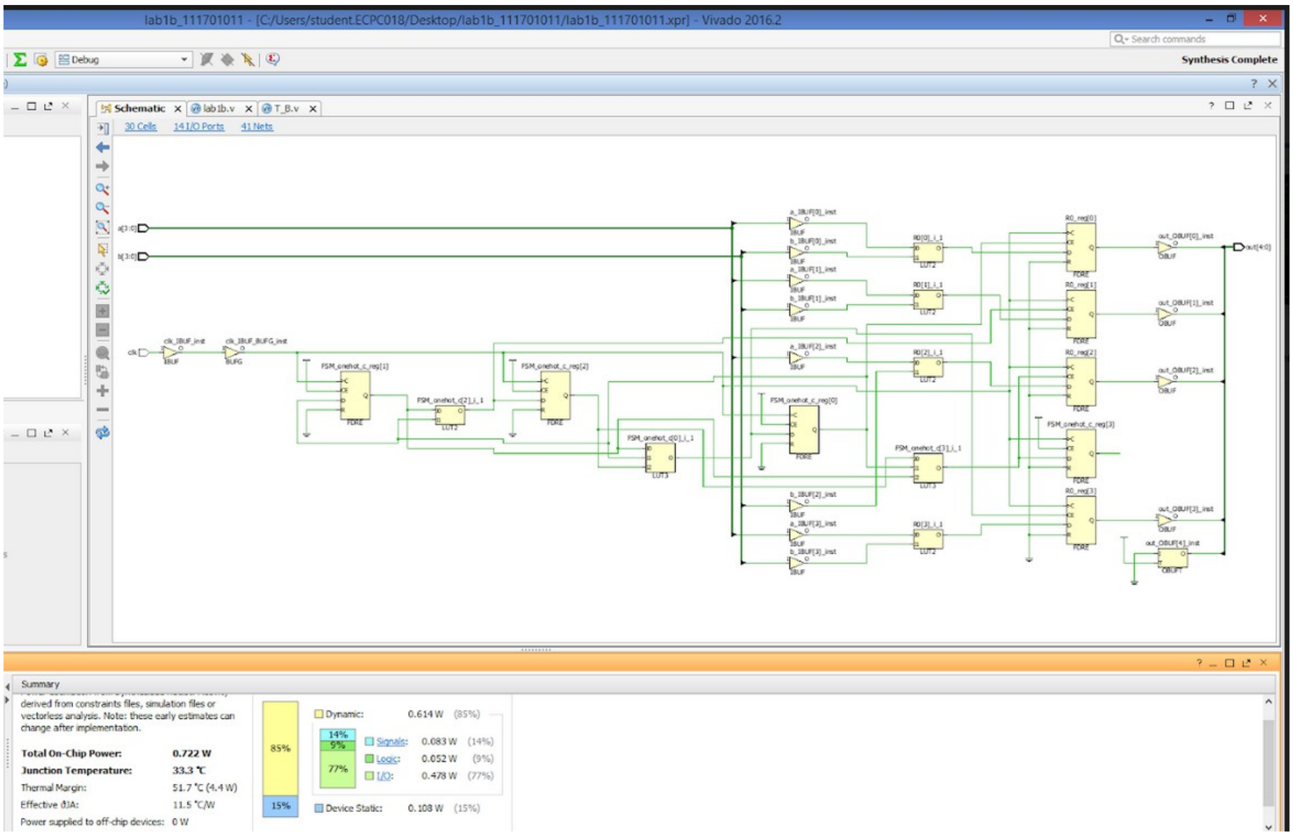
SIMULATION RESULT::



RTL SCHEMATIC::


lab1b_111701011 - [C:/Users/student.ECPC018/Desktop/lab1b_111701011/lab1b_111701011.xpr] - Vivado 2016.2
Synthesis Complete
Project Summary
Schematic (4)
38 Cells
14 I/O Ports
70 Nets

POWER & MEMORY CONSUMTION SCHEMATIC::


lab1b_111701011 - [C:/Users/student.ECPC018/Desktop/lab1b_111701011/lab1b_111701011.xpr] - Vivado 2016.2
Debug
Synthesis Complete
Schematic
lab1b.v
T_B.v
30 Cells
14 I/O Ports
41 Nets
Summary

derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

```
Total On-Chip Power:              0.722 W
Junction Temperature:             33.3 °C
Thermal Margin:                   51.7 °C (4.4 W)
Effective ϑJA:                    11.5 °C/W
Power supplied to off-chip devices: 0 W

Dynamic:          0.614 W   (85%)
   Signals:       0.083 W   (14%)
   Logic:         0.052 W   (9%)
   I/O:           0.478 W   (77%)
Device Static:    0.108 W   (15%)
```

Observation:

Provided with 4 full adders and two 4-bit registers R0 and R1->

Total On-Chip power = 1.901 W

Junction Temperature = 46.9°C

Dynamic power consumption: 1.777W (93%)

Signals: 0.048W (3%)

Logic: 0.010W (1%)

I/O: 1.718W (96%)

Static Power consumption: 0.124W (7%)

Provided with only 1 full adder and two 4-bit registers R0 and R1->

Total On-Chip power = 0.722 W

Junction Temperature = 33.3°C

Dynamic power consumption: 0.614W (85%)

Signals: 0.083W (14%)

Logic: 0.052W (9%)

I/O: 0.478W (77%)

Static Power consumption: 0.108W (15%)

Conclusion:

1> The **first** scenario takes lesser **power consumption** than the **second** one.

2> Difference of **Dynamic power consumption** to the **static power consumption (%)** is **more** in the **first** case. This shows the presence of **more %** of **parallel programing** in the **first** case as compared to the **second** case.

3> In the **second** case, with **lesser** number of **full adders**, there is more distribution of **load** among the **Signals** and **Logic** as compared to the **first** case.

4> The use of predefined modules of **xor, and, or** increases **On-Chip power consumtion**, as compared to when we use direct expression forms of these **operators (^,&,|)** by a visible difference.

5> There was **parallel allocation** in the **first** case, and hence there was **more On-Chip power consumption** in it as compared to the **second** case.

6> There was more **time delay** in the **first** scenario, and hence it took more **On-Chip power consumption** as compared to the **second** case.

Result:

1> Use of **lesser** number of **hierarchical modules** and reducing the **loads** on path of functioning of a program, may **reduce** the total **On-Chip power consumption**.

2> With different scenarios of design of circuits; **power consumtion**, **Junction temperature**, **Dynamic/Static power consumption** difference can vary.

3> **Parallel computing increases** the total **On-Chip power consumption**.

4> **Time delay** varies for different **paths** in a same circuit.

………………………………………………………………………