EE 465
Ahn Ho and Aaron Pederson
Lab Report

Lab 4

The goals of this lab:

- Synthesize a netlist fom Verilog Code
- Analyze the timing, power and area of the resulting circuit.

Synthesis of the Verilog Code:

First, we had to make modifications to our original code from lab so we will have D flip flops storing the input variables.

Here is the Verilog file from lab 1:

```
1   module lab1(iCLK, iRST_N, iSEL,  iA0, iA1, iB0, iB1, oRESULT);
2   input iCLK, iRST_N, iSEL;
3   input iA0, iA1, iB0, iB1;
4   output oRESULT;
5
6   wire iCLK, iRST_N;
7   wire [7:0] iA0, iA1, iB0, iB1;
8   reg [16:0] oRESULT;
9
10  always @ (posedge iCLK) begin
11    if(iRST_N)begin
12      if(iSEL) begin
13          oRESULT <= iA0 * iB0 + iA1 * iB1;
14      end
15      else begin
16          oRESULT <= iA0 * iB0 + iA1 * iB1 + iA0 * iA1 * iB0 * iB1;
17      end
18    end
19    else begin
20      oRESULT <= 0;
21    end
22  end
23
24  endmodule
```

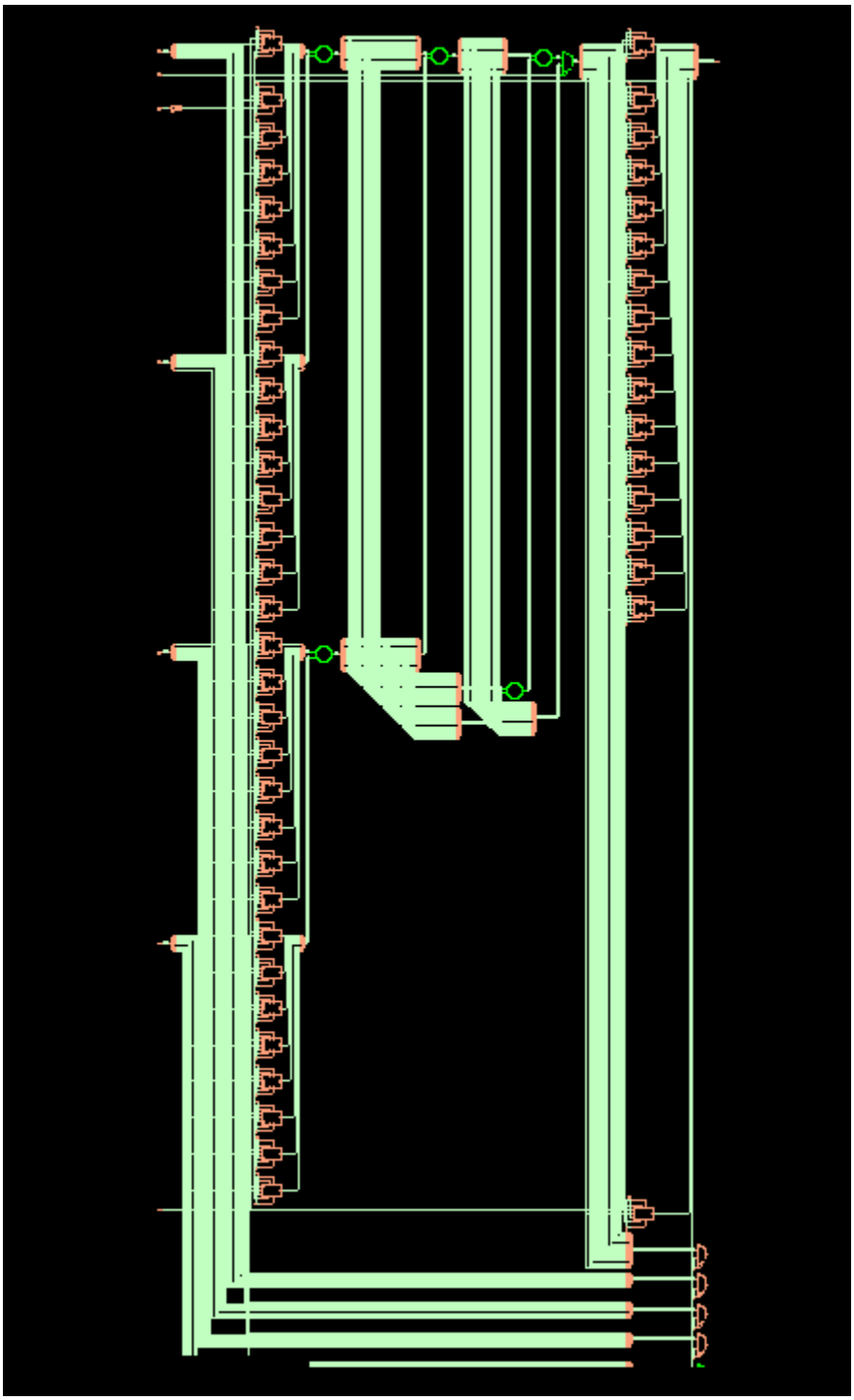
We In order to make this work, we had to modify a few things. We did this with the help of some classmates.

First, we changed the inputs A0 and 1 and B0 and 1 into registers. By doing this, we create d flip flops at the input. But since we did this, we now need intermediate wire variables to carry out the computation. So we add AB0, AB1, ABO1m(product) and AB01p(sum). Then finally the result register is input by the multiplexed wires from the product or sum value.

The resulting code is as follows:

```
module lab1(iCLK, iRST_N, iSEL,  iA0, iA1, iB0, iB1, oRESULT);
input iCLK, iRST_N, iSEL;
input [7:0] iA0, iA1, iB0, iB1;
output reg [16:0] oRESULT;
reg  [7:0] A0, B0, A1, B1;
wire [16:0] re, AB0, AB1, AB01m, AB01p;

assign AB0= A0 * B0;
assign AB1= A1 * B1;
assign AB01m= AB0 * AB1;
assign AB01p= AB0 + AB1;

always @ (posedge iCLK or negedge iRST_N)
        if (iRST_N ==0)
        begin
        A0<=8'h0;
        A1<=8'h0;
        B0<=8'h0;
        B1<=8'h0;
        oRESULT<=17'h0;
        end
        else if (iRST_N ==1)
        begin
        A0<=iA0;
        A1<=iA1;
        B0<=iB0;
        B1<=iB1;
        oRESULT<=re;
        end

assign re=iSEL?(AB01p):(AB01p+AB01m);

endmodule

// Worked with two other classmates to get this lab done
```

Now that we have code that will synthesize properly, we synthesized the Verilog code and it looks like the image on the next page.



Timing Analysis:

When we ran the timing report we received the following results:

Timing

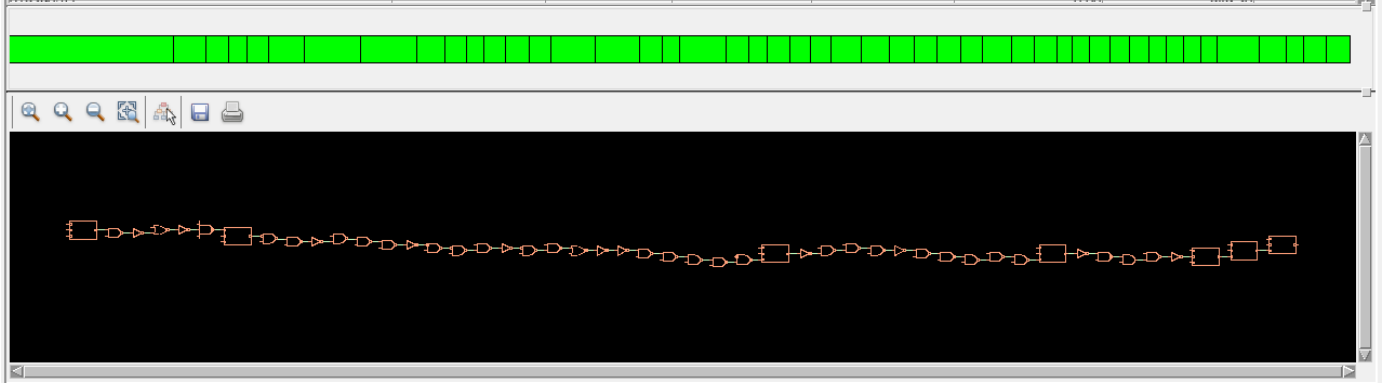
Report:

Detailed Timing Report

Close | Endpoint: oRESULT_reg[15]/D

Endpoint	Slack (ps)	Rise Slew (ps)	Fall Slew (ps)
oRESULT_reg[15]/D	0	46	31

Pin	Type	Fanout	Load (fF)	Slew (ps)	Delay (ps)	Arrival (ps)	
(clock iCLK)	launch					0.00	R
A1_reg[7]/CP				0.00		0.00	R
A1_reg[7]/Q	DFCNQD4	8	13.70	56.60	187.90	187.90	R
g17177/A1					0.00	187.90	
g17177/ZN	CKND2D1	2	2.30	37.80	37.20	225.10	F
g22/I					0.00	225.10	
g22/ZN	CKND1	2	2.20	28.20	26.80	251.90	R
g21/A2					0.00	251.90	
g21/ZN	NR2D1	1	2.30	21.50	21.70	273.60	F
g17818/I					0.00	273.60	
g17818/ZN	INVD2	3	6.20	32.70	25.60	299.20	R
g17819/A2					0.00	299.20	
g17819/ZN	ND3D1	1	1.10	43.70	41.30	340.50	F
g17941/A1					0.00	340.50	
g17941/ZN	IOA21D2	3	4.60	36.90	65.20	405.70	F
g17867/A1					0.00	405.70	
g17867/ZN	IND2D1	1	2.10	36.50	64.70	470.40	F
g15295/A1					0.00	470.40	
g15295/ZN	ND2D2	3	6.80	43.90	32.20	502.60	R
g15294/I					0.00	502.60	
g15294/ZN	CKND2	1	4.80	23.10	25.80	528.40	F
g15291/A2					0.00	528.40	
g15291/ZN	ND2D4	1	4.50	21.60	20.80	549.20	R



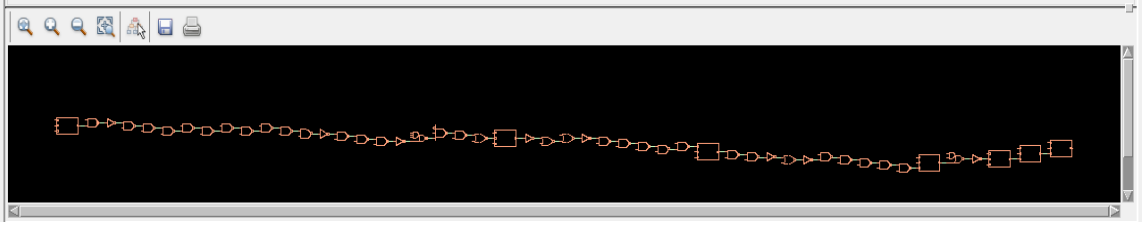
From this, we get a slack value of 0. This means the signal reaches the output at exactly the correct time no later or no earlier. That means this will not be a timing violation. This clock frequency should be the heist possible frequency without error.

Detailed report with FA1D:

Close | Endpoint: oRESULT_reg[16]/D

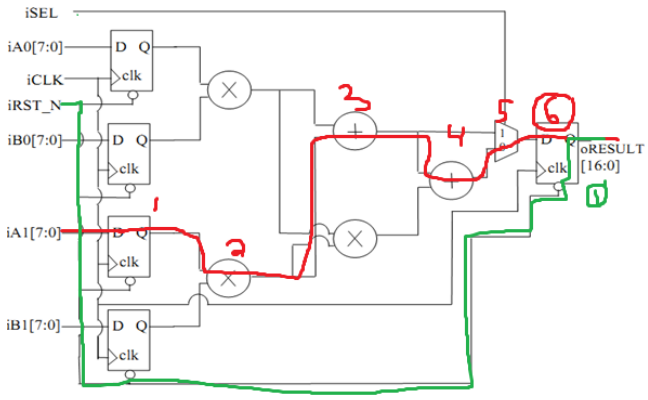
Endpoint	Slack (ps)	Rise Slew (ps)	Fall Slew (ps)
oRESULT_reg[16]/D	-724	46	31

Pin	Type	Fanout	Load (fF)	Slew (ps)	Delay (ps)	Arrival (ps)	
g26096/ZN	CKND2D2	3	4.30	35.30	36.90	227.20	F
g20557/I					0.00	227.20	
g20557/ZN	CKND2	2	2.10	18.30	18.70	245.90	R
g27162/A1					0.00	245.90	
g27162/ZN	CKND2D1	2	2.40	35.90	28.50	274.40	F
g27161/A1					0.00	274.40	
g27161/ZN	ND2D1	1	2.10	31.50	27.70	302.10	R
g27160/A1					0.00	302.10	
g27160/ZN	ND2D2	2	3.70	28.80	27.00	329.10	F
g25143/A2					0.00	329.10	
g25143/ZN	ND2D2	2	3.20	33.70	24.40	353.50	R
g19624/A1					0.00	353.50	
g19624/ZN	ND2D2	1	2.50	23.60	24.10	377.60	F
g19623/A2					0.00	377.60	
g19623/ZN	ND2D2	2	4.40	33.10	25.60	403.20	R
g19622/A1					0.00	403.20	
g19622/ZN	ND2D2	2	7.30	46.60	37.40	440.60	F
g26682/A2					0.00	440.60	
g26682/ZN	ND2D4	2	4.60	24.90	27.10	467.70	R
g26847/A1					0.00	467.70	
g26847/ZN	ND2D2	1	2.10	21.10	20.80	488.50	F
g26846/A1					0.00	488.50	
g26846/ZN	ND2D2	2	4.40	30.60	22.70	511.20	R
g26845/I					0.00	511.20	



The slack time for this timing analysis is -724ps. This means this will not pass the design check for timing. The clock period is

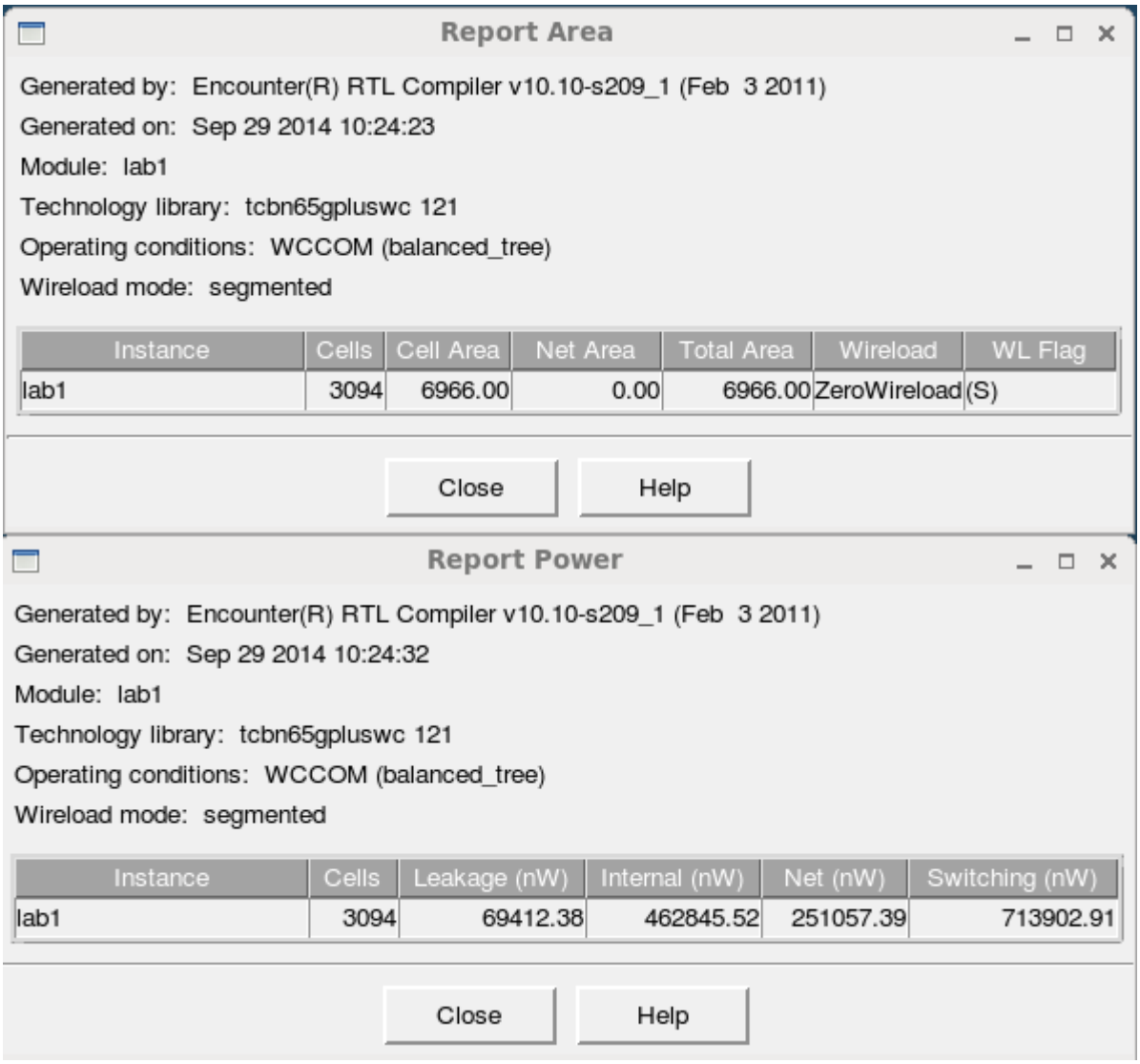
Slack does not improve when FA1D is used.



Anh Ho,
Aaron P.

Area and Power consumption analysis:

The area analysis of the first synthesis: (without FA1D and at 1.6ns period)



Report Area

Generated by: Encounter(R) RTL Compiler v10.10-s209_1 (Feb 3 2011)
Generated on: Sep 29 2014 10:24:23
Module: lab1
Technology library: tcbn65gpluswc 121
Operating conditions: WCCOM (balanced_tree)
Wireload mode: segmented

Instance	Cells	Cell Area	Net Area	Total Area	Wireload	WL Flag
lab1	3094	6966.00	0.00	6966.00	ZeroWireload	(S)

Close Help

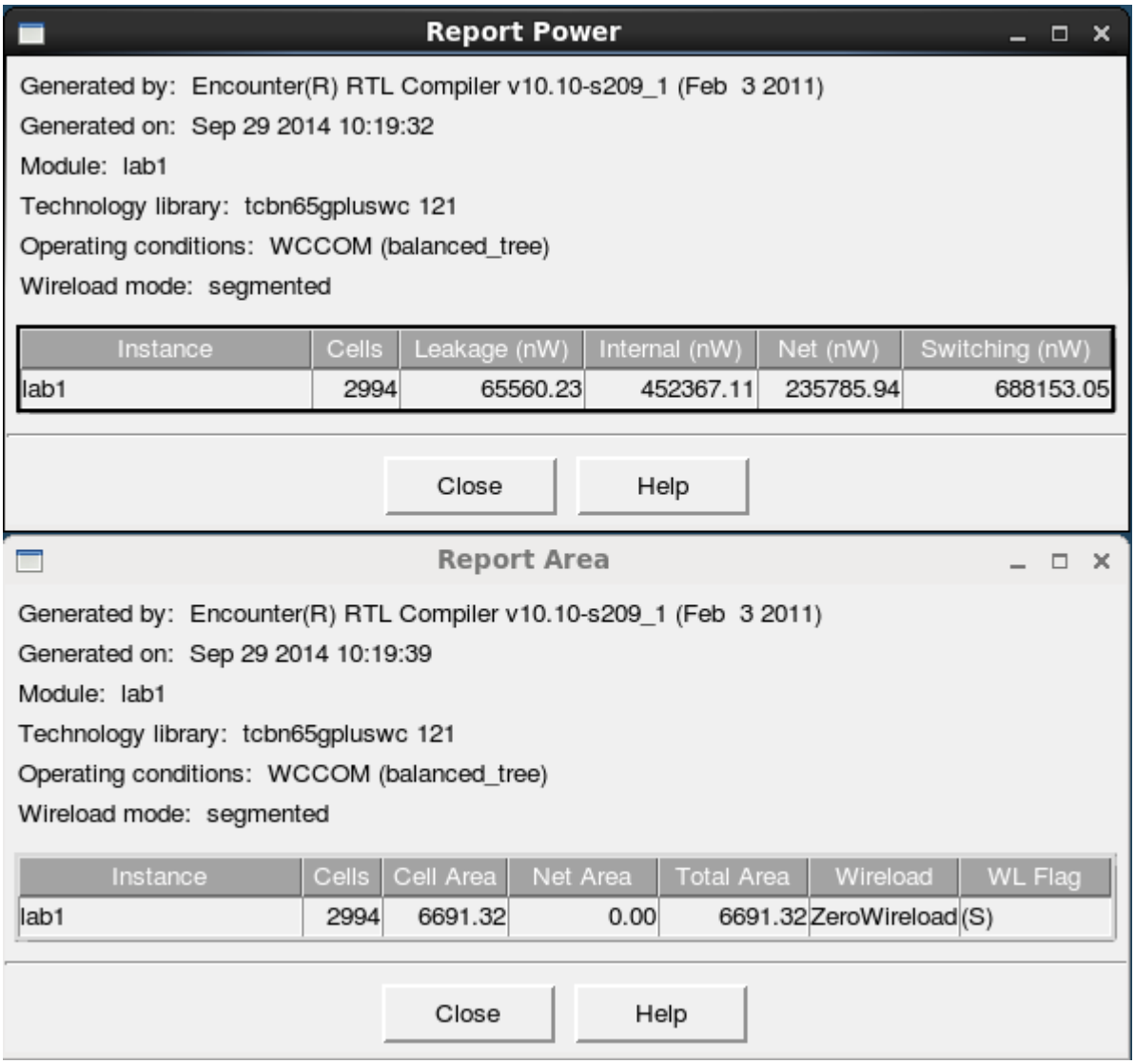
Report Power

Generated by: Encounter(R) RTL Compiler v10.10-s209_1 (Feb 3 2011)
Generated on: Sep 29 2014 10:24:32
Module: lab1
Technology library: tcbn65gpluswc 121
Operating conditions: WCCOM (balanced_tree)
Wireload mode: segmented

Instance	Cells	Leakage (nW)	Internal (nW)	Net (nW)	Switching (nW)
lab1	3094	69412.38	462845.52	251057.39	713902.91

Close Help

Comparing these to the area and power report with FA1D below,



We can see that with FA1D there is a slightly smaller area and power consumption.

Conclusion:

In this lab we synthesized our circuit using a few different methods. One being pure synthesis and the other using FA1D standard cells. We discovered that the FA1D method will result in a slower circuit but will have a smaller area and consume less power. So there are trade-offs for both options.