VLSI LAB Digital Assignment 3

Submitted by:

Swarup Tripathy — 19BEE0167

School of Electrical Engineering

Faculty: **Professor Balamurugan S**

Course: **EEE-4028**

Course Name: **VLSI Lab**

Lab Slot: **L43 + L44**

This work is submitted in partial fulfilment of the requirement of the award of the degree of Bachelor of Technology in EEE

February 27, 2022

4-BIT CARRY SAVE ARRAY MULTIPLIER

Objectives

1. To provide students with the background needed to design, develop, and test digital arithmetic circuits using IEEE standard Verilog HDL.
2. To provide an understanding complex arithmetic circuit design principles and its architecture design.

Outcomes

1. After completion of this course the students will be familiar with design and implementation of Digital Arithmetic building blocks using Verilog HDL and Modelsim Software.

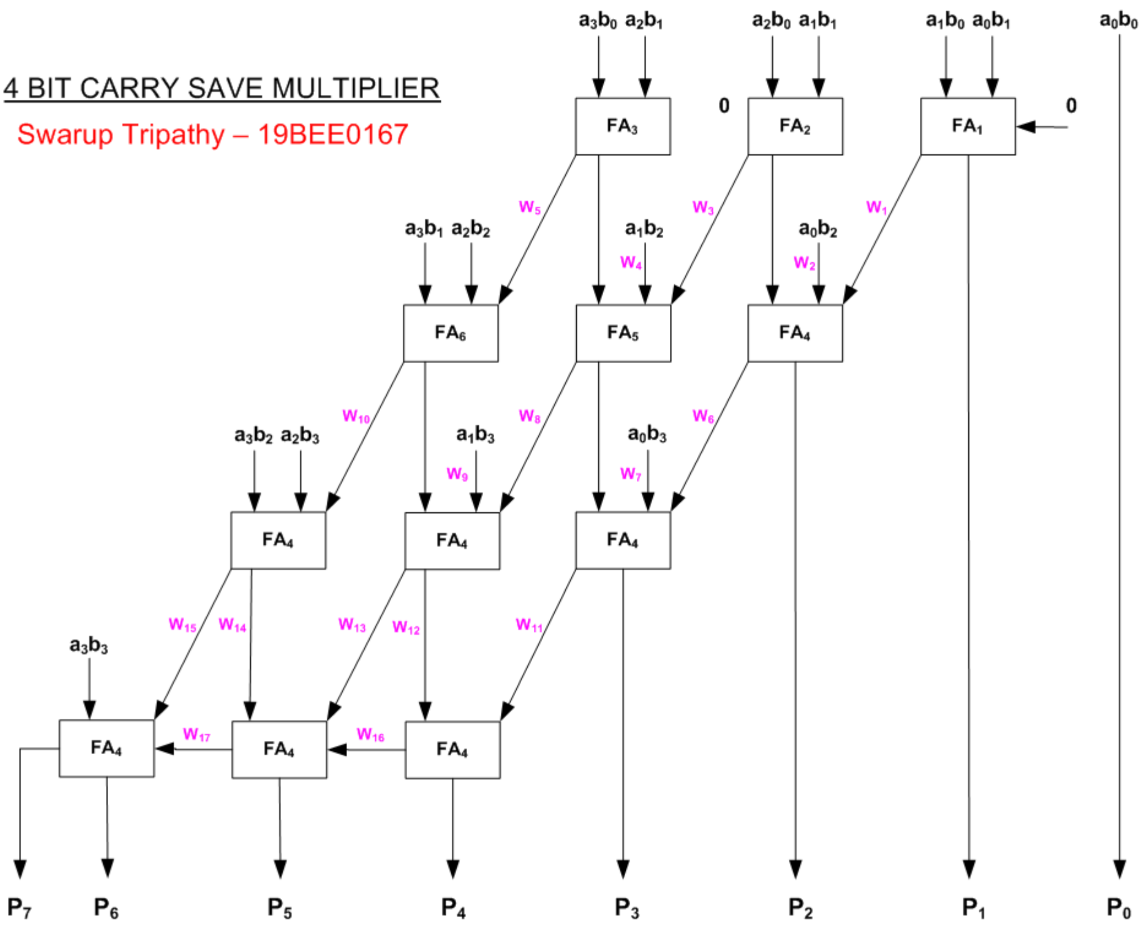
AIM

1. Design a 4 bit carry save array multiplier using 12 full adders.

REQUIRED SOFTWARE

1. Model sim software for simulation
2. Microsoft Visio for making flowchart
3. Documentation to be done using $\LaTeX$

CIRCUIT DIAGRAM



Design Code

1. **Verilog Module** — cla.v

```
module csa_4bit_19BEE0167(input[3:0]a,b,
  output[7:0]p);
wire[17:1]w;
supply0 zero;
assign p[0] = a[0]&b[0];

fa_df_19BEE0167 fa1((a[1]&b[0]),(a[0]&b[1]),zero,p[1],w[1]);
fa_df_19BEE0167 fa2((a[2]&b[0]),(a[1]&b[1]),zero,w[2],w[3]);
fa_df_19BEE0167 fa3((a[3]&b[0]),(a[2]&b[1]),zero,w[4],w[5]);

fa_df_19BEE0167 fa4(w[2],(a[0]&b[2]),w[1],p[2],w[6]);
fa_df_19BEE0167 fa5(w[4],(a[1]&b[2]),w[3],w[7],w[8]);
fa_df_19BEE0167 fa6((a[3]&b[1]),(a[2]&b[2]),w[5],w[9],w[10]);

fa_df_19BEE0167 fa7(w[7],(a[0]&b[3]),w[6],p[3],w[11]);
```

```
fa_df_19BEE0167 fa8(w[9],(a[1]&b[3]),w[8],w[12],w[13]);
fa_df_19BEE0167 fa9((a[3]&b[2]),(a[2]&b[3]),w[10],w[14],w[15]);

fa_df_19BEE0167 fa10(w[12],zero,w[11],p[4],w[16]);
fa_df_19BEE0167 fa11(w[14],w[16],w[13],p[5],w[17]);
fa_df_19BEE0167 fa12((a[3]&b[3]),w[17],w[15],p[6],p[7]);

endmodule
```

2. **Fulladder.v**

```
//Full adder using Dataflow
module fa_df_19BEE0167(input a,b,cin,output sum,cout);
assign sum=a^b^cin;
assign cout=(a^b)&cin|(a&b);
endmodule
```

3. **Test Fixture** — csaTest.v

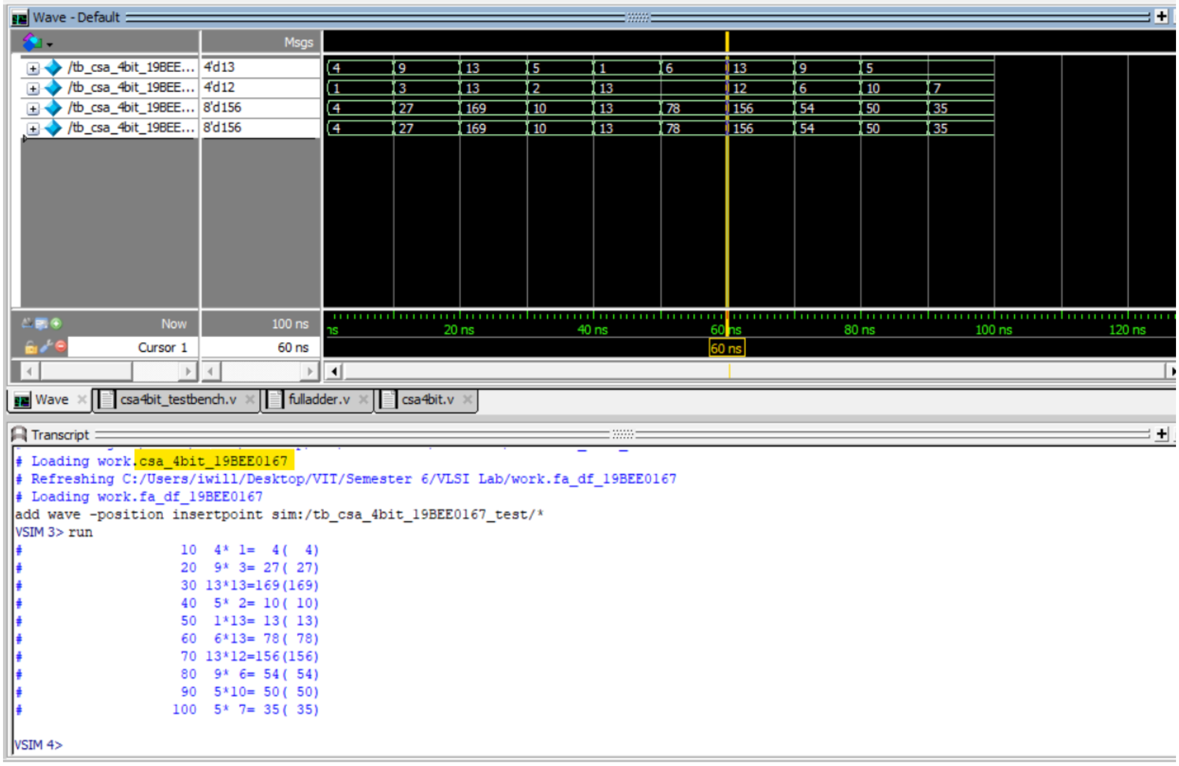
```
module tb_csa_4bit_19BEE0167_test;
reg [3:0]A,B;

wire [7:0]S;
reg [7:0]check;

csa_4bit_19BEE0167 uut(A,B,S);
initial repeat(10) begin
A = $random;
B = $random;

check = A*B;
#10 $display($time," %d*%d=%d(%d)",A,B,S,check);
end
endmodule
```

4. **Output**



```
# Loading work.csa_4bit_19BEE0167
# Refreshing C:/Users/iwill/Desktop/VIT/Semester 6/VLSI Lab/work.fa_df_19BEE0167
# Loading work.fa_df_19BEE0167
add wave -position insertpoint sim:/tb_csa_4bit_19BEE0167_test/*
VSIM 3> run
#                    10  4* 1=  4(  4)
#                    20  9* 3= 27( 27)
#                    30 13*13=169(169)
#                    40  5* 2= 10( 10)
#                    50  1*13= 13( 13)
#                    60  6*13= 78( 78)
#                    70 13*12=156(156)
#                    80  9* 6= 54( 54)
#                    90  5*10= 50( 50)
#                   100  5* 7= 35( 35)

VSIM 4>
```

5. **Console Output**

```
#                    10  4* 1=  4(  4)
#                    20  9* 3= 27( 27)
#                    30 13*13=169(169)
#                    40  5* 2= 10( 10)
#                    50  1*13= 13( 13)
#                    60  6*13= 78( 78)
#                    70 13*12=156(156)
#                    80  9* 6= 54( 54)
#                    90  5*10= 50( 50)
#                   100  5* 7= 35( 35)
```

Result

1. Successfully the 4-bit carry save array multiplier has been designed and the output was verified.

Inference

1. In this experiment learnt about how to construct multiple-bit adders.