

### **VERILOG ASSIGNMENT 2**

Name: Muhammed Said

Surname: Kaya

Student Number: 21627428

**Section**: 02

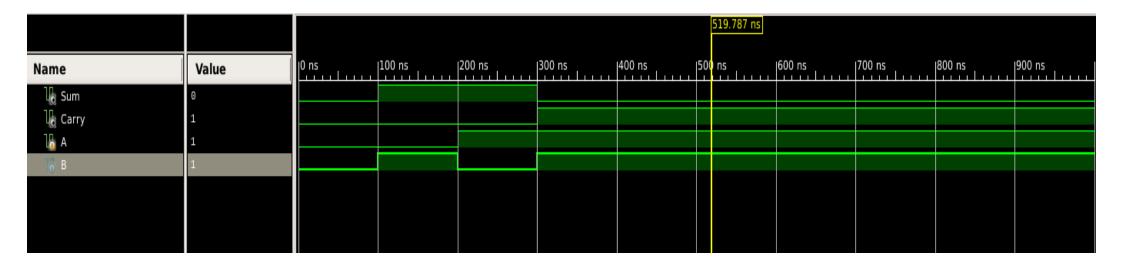
**Subject**: Implementing Half Adder - 1 Bit Full Adder - 4 Bit Full Adder

```
Module of Half Adder
                                                                    .Carry(Carry),
                                                                    .A(A),
`timescale 1ns / 1ps
                                                                    .B(B)
module HalfAdder(
                                                             );
  output Sum, Carry,
  input A,B
                                                             initial begin
  );
                                                     // Initialize Inputs
xor(Sum,A,B);
                                                                    A = 0;
and(Carry,A,B);
                                                                    B = 0;
                                                     // Wait 100 ns for global reset to finish
endmodule
                                                                    #100;
                                                     // Initialize Inputs
TestBench of Half Adder
                                                                    A = 0;
                                                                    B = 1;
                                                     // Wait 100 ns for global reset to finish
`timescale 1ns / 1ps
                                                                    #100;
                                                     // Initialize Inputs
module HalfAdder_TestBench;
                                                                    A = 1;
                                                                    B = 0;
// Inputs
                                                     // Wait 100 ns for global reset to finish
       reg A;
       reg B;
                                                     // Initialize Inputs
                                                                    A = 1;
// Outputs
                                                                    B = 1;
       wire Sum;
       wire Carry;
                                                                    // Add stimulus here
                                                             end
// Instantiate the Unit Under Test (UUT)
       HalfAdder uut (
```

endmodule

.Sum(Sum),

# **Waveform of Half Adder**



```
Module of 1 Bit Adder
                                                                     .Carry(Carry),
                                                                     .Sum(Sum),
                                                                     .A(A),
                                                                      .B(B),
`timescale 1ns / 1ps
                                                                      .Cin(Cin)
                                                             );
module HalfAdder( output Sum, output
Carry, input A , input B );
                                                             initial begin
                                                                     // Initialize Inputs
       xor(Sum,A,B);
                                                                     A = 0;
        and(Carry,A,B);
                                                                     B = 0;
                                                                     Cin = 0;
endmodule
                                                                     // Wait 100 ns for global reset
                                                      to finish
module OneBitFullAdder(
                                                                     #100;
                                                                     // Initialize Inputs
        output Carry, Sum,
                                                                     A = 0;
       input A,B,Cin
                                                                     B = 0;
  );
                                                                     Cin = 1;
                                                                     // Wait 100 ns for global reset
        wire sum1,carry1,carry2;
                                                      to finish
                                                                     #100;
        HalfAdder ha1(sum1,carry1,A,B);
                                                                     // Initialize Inputs
        HalfAdder ha2(Sum ,carry2 ,sum1
                                                                     A = 0;
,Cin);
                                                                     B = 1;
                                                                     Cin = 0;
        or(Carry ,carry2 ,carry1);
                                                                     // Wait 100 ns for global reset
                                                      to finish
                                                                     #100;
endmodule
                                                                     // Initialize Inputs
                                                                     A = 0;
                                                                     B = 1;
TestBench Of 1 Bit Adder
                                                                     Cin = 1;
                                                                     // Wait 100 ns for global reset
                                                      to finish
                                                                     #100;
`timescale 1ns / 1ps
                                                                     // Initialize Inputs
                                                                     A = 1;
module OneBitFullAdder_TestBench;
                                                                     B = 0;
                                                                     Cin = 0;
       // Inputs
                                                                     // Wait 100 ns for global reset
       reg A;
                                                      to finish
       reg B;
                                                                     #100:
       reg Cin;
                                                                     // Initialize Inputs
                                                                     A = 1;
       // Outputs
                                                                     B = 0;
       wire Carry;
                                                                     Cin = 1;
       wire Sum;
                                                                     // Wait 100 ns for global reset
                                                      to finish
       // Instantiate the Unit Under Test
                                                                     #100:
(UUT)
                                                                     // Initialize Inputs
        OneBitFullAdder uut (
```

```
A = 1;
                                                                     Cin = 1;
               B = 1;
                                                                     // Wait 100 ns for global reset
               Cin = 0;
                                                      to finish
               // Wait 100 ns for global reset
                                                                     #100;
to finish
                                                                     // Add stimulus here
               #100;
               // Initialize Inputs
                                                              end
               A = 1;
               B = 1;
                                                      endmodule
```

# **Waveform of 1 Bit Full Adder**

		672.464 ns										
Name	Value	0 ns	100 ns	200 ns	300 ns	400 ns	500 ns	600 ns		700 ns	800 ns	900 ns
<b>№</b> Sum	0											
🎚 Carry	1											
<b>1</b> A	1											
<b>Љ</b> В	1											
1₀ Cin	0											

#### **Module Of 4 Bit Full Adder**

```
`timescale 1ns / 1ps
module HalfAdder( output Sum, output
Carry, input A , input B );
       xor(Sum,A,B);
       and(Carry,A,B);
endmodule
module OneBitFullAdder( output Carry,
output Sum, input A, input B, input Cin );
        wire sum1,carry1,carry2;
        HalfAdder ha1(sum1,carry1,A,B);
        HalfAdder ha2(Sum ,carry2 ,sum1
,Cin);
        or(Carry ,carry2 ,carry1);
endmodule
module FourBitFullAdder(
        output [3:0]Sum,
  output Cout,
  input [3:0]A,B,
        input Cin
```

);

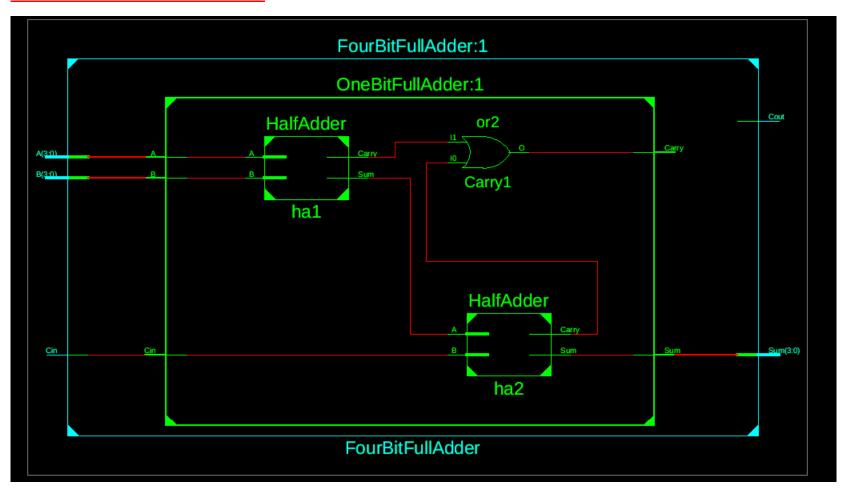
```
wire carry1, carry2, carry3;
       OneBitFullAdder fa1(carry1, Sum[0],
A[0], B[0], Cin);
       OneBitFullAdder fa2(carry2, Sum[1],
A[1], B[1], carry1);
       OneBitFullAdder fa3(carry3, Sum[2],
A[2], B[2], carry2);
       OneBitFullAdder fa4(Cout , Sum[3],
A[3], B[3], carry3);
endmodule
TestBench Of 4 Bit Full Adder
`timescale 1ns / 1ps
module TestBench;
       // Inputs
       reg [3:0] A;
       reg [3:0] B;
       reg Cin;
       // Outputs
       wire [3:0] Sum;
       wire Cout;
       // Instantiate the Unit Under Test
(UUT)
       FourBitFullAdder uut (
               .Sum(Sum),
               .Cout(Cout),
```

```
.A(A),
                                                                       // Wait 100 ns for global reset
                                                       to finish
                .B(B),
                                                                       #100;
                .Cin(Cin)
                                                                       // Initialize Inputs
       );
                                                                       A = 4'b0000;
                                                                       B = 4'b0101;
       initial begin
                                                                       Cin=0;
                // Initialize Inputs
                                                                       // Wait 100 ns for global reset
                A = 4'b1010;
                                                       to finish
                B = 4'b0101;
                                                                       #100;
                Cin=0;
                // Wait 100 ns for global reset
                                                                       // Add stimulus here
to finish
                #100;
                                                               end
                // Initialize Inputs
                A = 4'b1111;
                                                       endmodule
                B = 4'b0000;
                Cin=1;
                // Wait 100 ns for global reset
to finish
                #100;
                // Initialize Inputs
                A = 4'b1111;
                B = 4'b0001;
                Cin=1;
                // Wait 100 ns for global reset
to finish
                #100;
                // Initialize Inputs
                A = 4'b1111;
                B = 4'b0101;
                Cin=0;
```

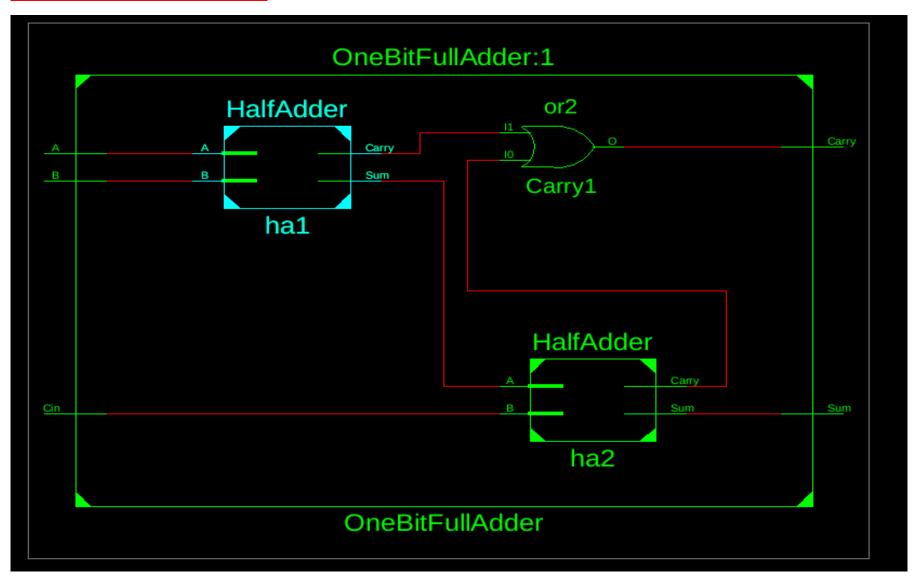
### **Waveform of 4 Bit Full Adder**



## **RTL Schematic of 4 Bit Full Adder**



## **RTL Schematic Of 1 Bit Full Adder**



## **RTL Schematic Of Half Adder**

