HACETTEPE
University

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

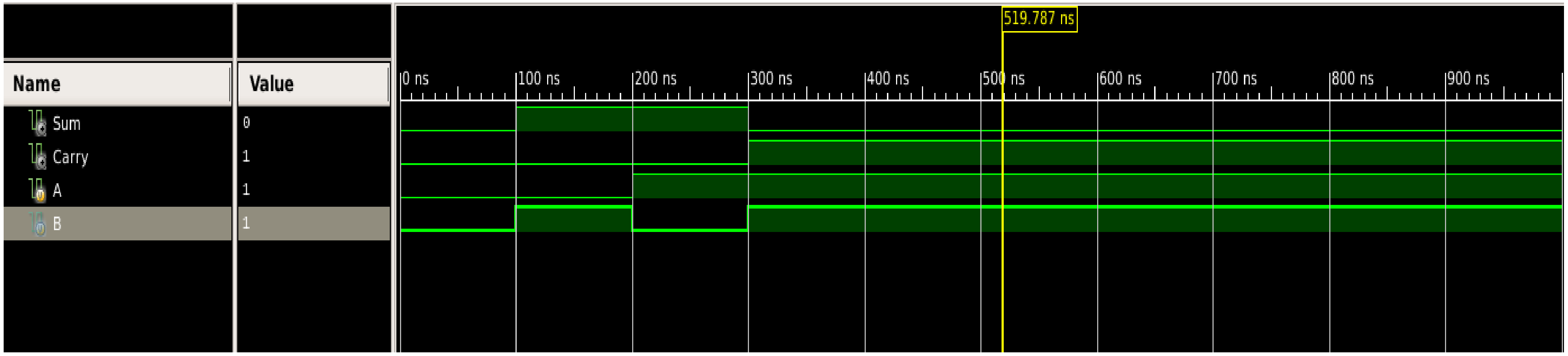
Module of Half Adder

```
`timescale 1ns / 1ps
module HalfAdder(
   output Sum,Carry,
   input A,B
   );
xor(Sum,A,B);
and(Carry,A,B);

endmodule
```

TestBench of Half Adder

```
`timescale 1ns / 1ps

module HalfAdder_TestBench;

// Inputs
          reg A;
          reg B;

// Outputs
          wire Sum;
          wire Carry;

// Instantiate the Unit Under Test (UUT)
          HalfAdder uut (
                    .Sum(Sum),
                    .Carry(Carry),
                    .A(A),
                    .B(B)
          );

          initial begin
// Initialize Inputs
                    A = 0;
                    B = 0;
// Wait 100 ns for global reset to finish
                    #100;
// Initialize Inputs
                    A = 0;
                    B = 1;
// Wait 100 ns for global reset to finish
                    #100;
// Initialize Inputs
                    A = 1;
                    B = 0;
// Wait 100 ns for global reset to finish
                    #100;
// Initialize Inputs
                    A = 1;
                    B = 1;

                    // Add stimulus here
          end

endmodule
```

Waveform of Half Adder



Module of 1 Bit Adder

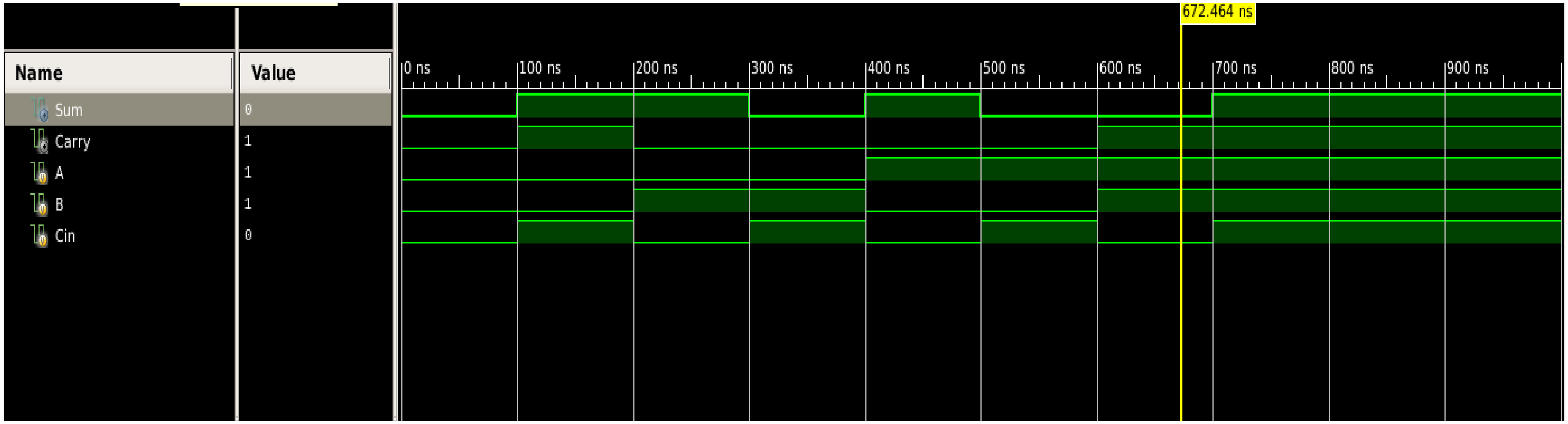
```
`timescale 1ns / 1ps

module HalfAdder( output Sum, output
Carry, input A , input B );

	xor(Sum,A,B);
	and(Carry,A,B);

endmodule

module OneBitFullAdder(

	 output Carry,Sum,
	input A,B,Cin
   );

	 wire sum1,carry1,carry2;

	 HalfAdder ha1(sum1 ,carry1  ,A     ,B);
	 HalfAdder ha2(Sum  ,carry2  ,sum1
,Cin);

	 or(Carry ,carry2 ,carry1);


endmodule
```

TestBench Of 1 Bit Adder

```
`timescale 1ns / 1ps

module OneBitFullAdder_TestBench;

	// Inputs
	reg A;
	reg B;
	reg Cin;

	// Outputs
	wire Carry;
	wire Sum;

	// Instantiate the Unit Under Test
(UUT)
	OneBitFullAdder uut (
		.Carry(Carry),
		.Sum(Sum),
		.A(A),
		.B(B),
		.Cin(Cin)
	);

	initial begin
		// Initialize Inputs
		A = 0;
		B = 0;
		Cin = 0;
		// Wait 100 ns for global reset
to finish
		#100;
		// Initialize Inputs
		A = 0;
		B = 0;
		Cin = 1;
		// Wait 100 ns for global reset
to finish
		#100;
		// Initialize Inputs
		A = 0;
		B = 1;
		Cin = 0;
		// Wait 100 ns for global reset
to finish
		#100;
		// Initialize Inputs
		A = 0;
		B = 1;
		Cin = 1;
		// Wait 100 ns for global reset
to finish
		#100;
		// Initialize Inputs
		A = 1;
		B = 0;
		Cin = 0;
		// Wait 100 ns for global reset
to finish
		#100;
		// Initialize Inputs
		A = 1;
		B = 0;
		Cin = 1;
		// Wait 100 ns for global reset
to finish
		#100;
		// Initialize Inputs
```

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

		end

endmodule
```

Waveform of 1 Bit Full Adder



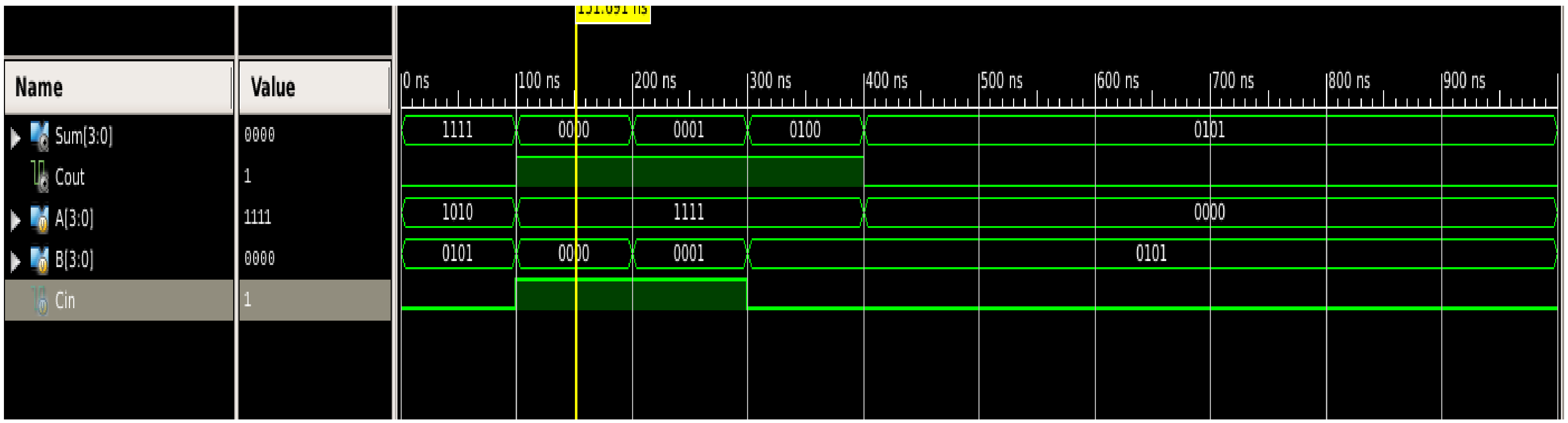
Module Of 4 Bit Full Adder

```
`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
```

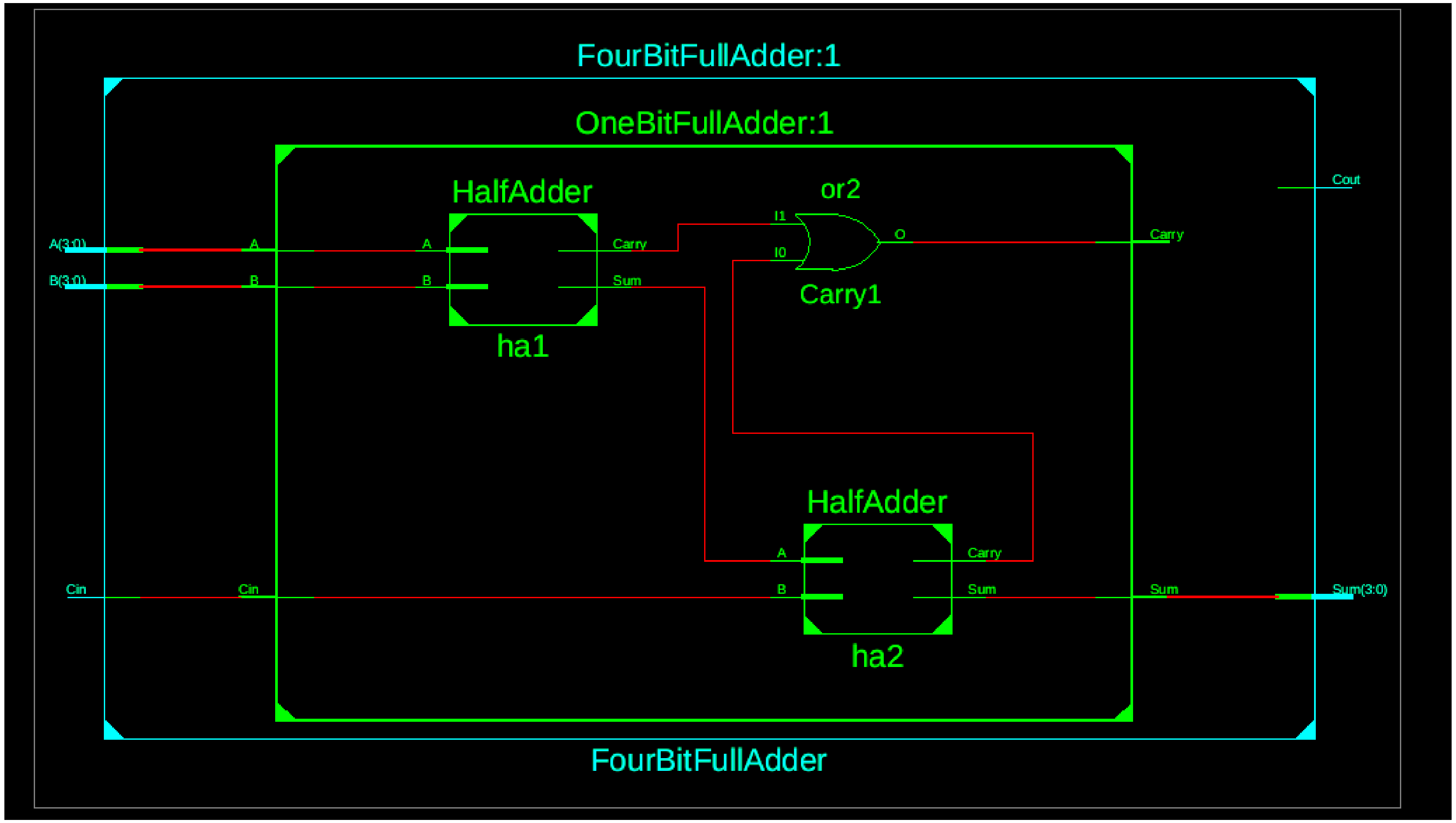
TestBench Of 4 Bit Full Adder

```
`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),
		.B(B),
		.Cin(Cin)
	);

	initial begin
		// Initialize Inputs
		A = 4'b1010;
		B = 4'b0101;
		Cin=0;
		// Wait 100 ns for global reset
to finish
		#100;
		// Initialize Inputs
		A = 4'b1111;
		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;
		// Wait 100 ns for global reset
to finish
		#100;
		// Initialize Inputs
		A = 4'b0000;
		B = 4'b0101;
		Cin=0;
		// Wait 100 ns for global reset
to finish
		#100;

		// Add stimulus here

	end

endmodule
```

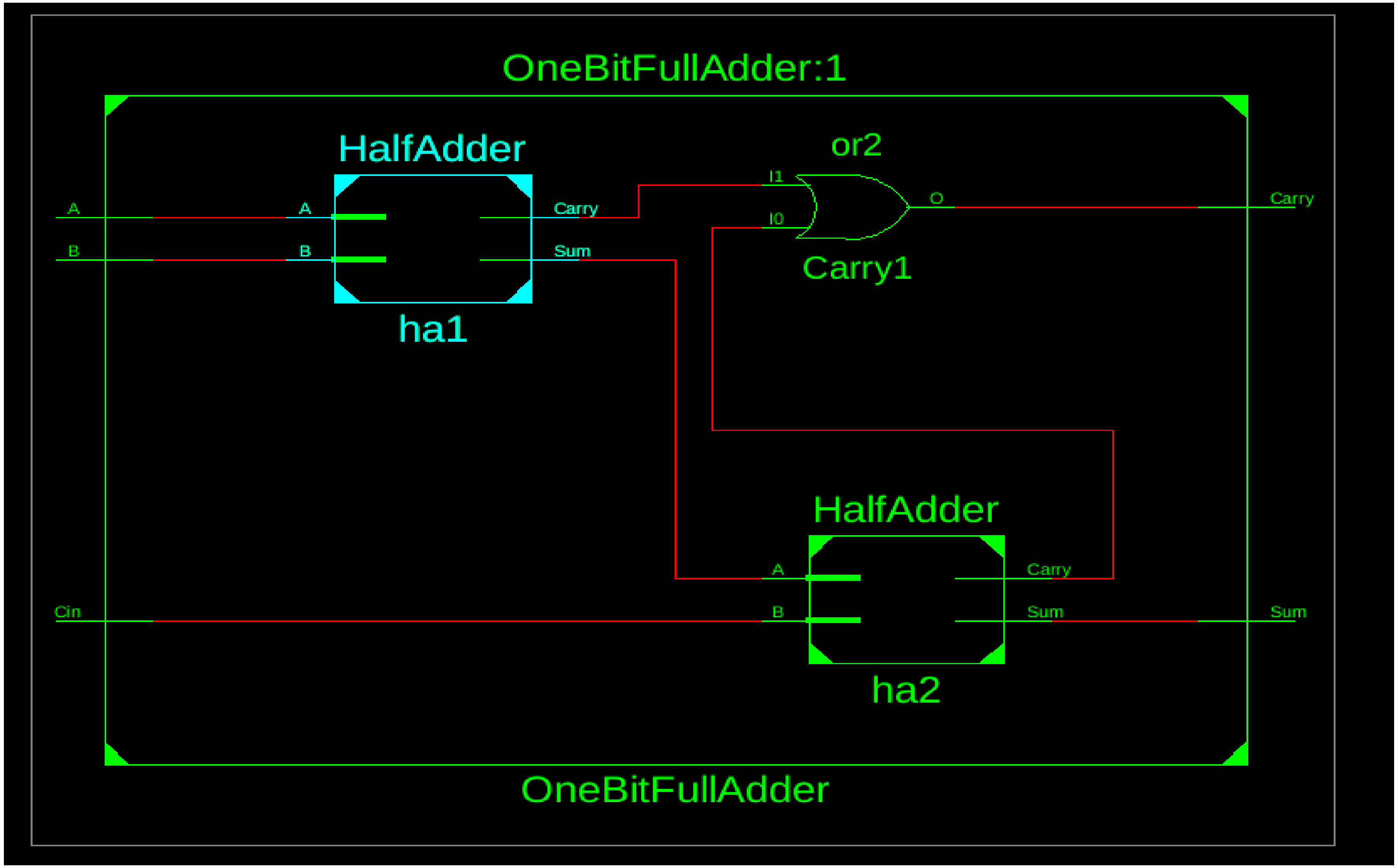
Waveform of 4 Bit Full Adder



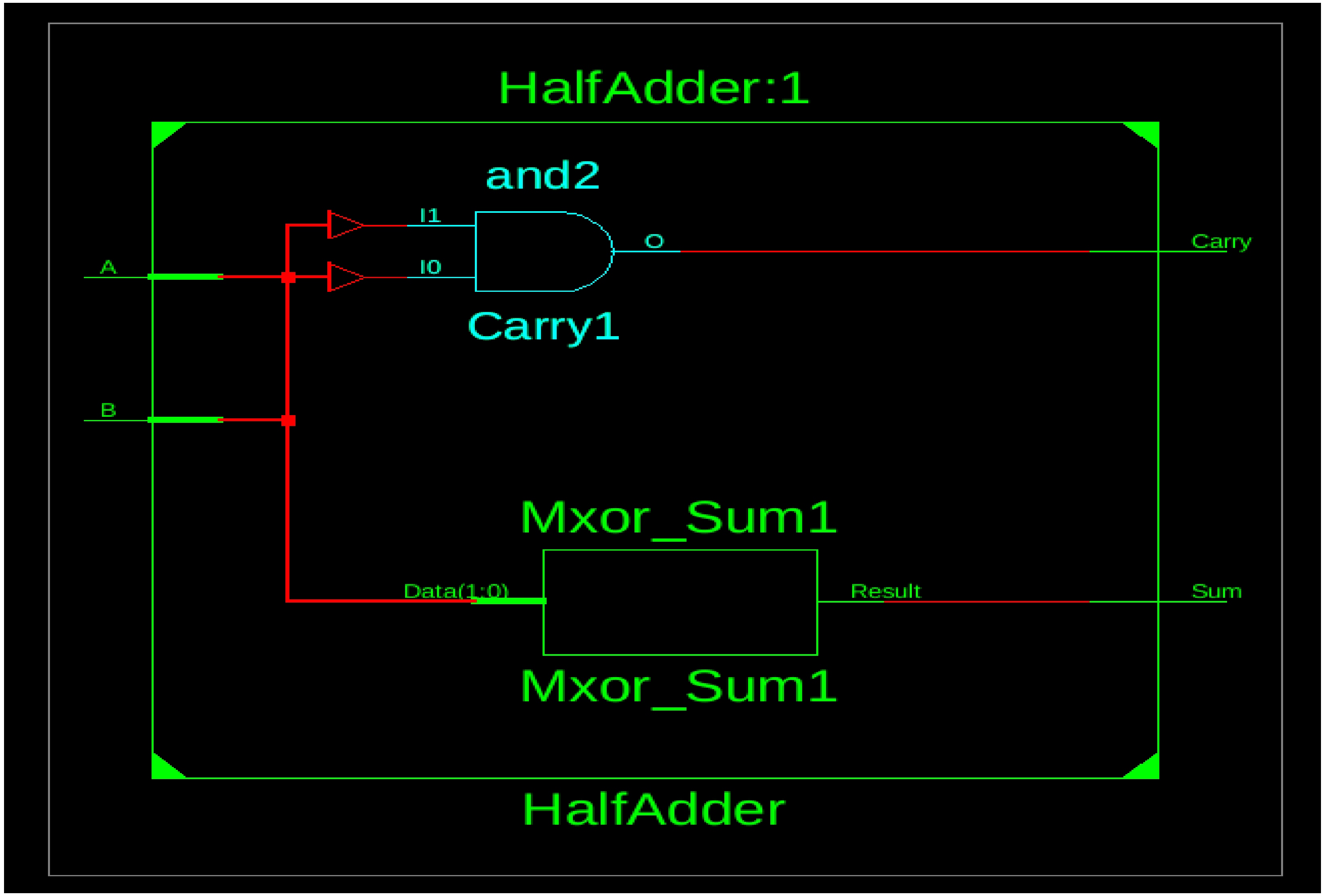
RTL Schematic of 4 Bit Full Adder



RTL Schematic Of 1 Bit Full Adder



RTL Schematic Of Half Adder


HalfAdder:1
and2
I1
I0
O
A
Carry
Carry1
B
Mxor_Sum1
Data(1:0)
Result
Sum
Mxor_Sum1
HalfAdder