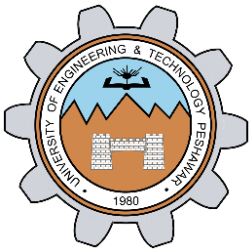
# DSD LAB REPORT

## LAB # 02




**Spring 2021**

**CSE-308L Digital System Design Lab**

Submitted by: **ASHIQ ULLAH**

Registration No. : **18PWCSE1695**

Class Section: **B**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

Submitted to:

**Engr. Madiha Sher**

April 24, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**OBJECTIVES:**

- Learn top down and bottom up design methodologies
- Data flow level modeling

**TASK01 (a):**

Design 4-bit Ripple Carry Adder

1. First implement a Full adder using data gate level modeling.
2. Simulate the Full adder with a test bench.
3. Instantiate the Full adder four times and connect the circuit as shown.
4. Now again write a test bench and simulate the 4-bit RCA.

**CODE:**

**Step1:**

```
ln #
1 module SUM(A, B, Cin, Sum);
2          input A,B,Cin;
3          output Sum;
4          xor g0(Sum, A, B, Cin);
5 endmodule
6
```

```
ln #
 1 module CARRY(A, B, Cin, Cout);
 2          input A,B,Cin;
 3          output Cout;
 4          wire c1,c2,c3;
 5          and g0(c1, A, Cin);
 6          and g1(c2, B, Cin);
 7          and g2(c3, A, B);
 8          or g3(Cout, c1, c2, c3);
 9 endmodule
10
```

```
ln #
1 module FullAdder(A, B, Cin, Sum, Cout);
2          input A,B,Cin;
3          output Sum, Cout;
4
5          SUM S(A, B, Cin, Sum);
6          CARRY C(A, B, Cin, Cout);
7 endmodule
8
```

**Step2:**

**TestBench:**

```
E:/6TH SEMEST
 1 module testB_FA;
 2         reg A,B,Cin;
 3         wire Sum,Cout;
 4         FullAdder FA5(A, B, Cin, Sum, Cout);
 5         initial
 6         begin
 7                 $display("\tA\t|\tB\t|\tCin\t|\tCarry\t|\tSum");
 8                 A = 0; B = 0; Cin = 0;
 9                 #1 $monitor("\t%b\t|\t %b\t|\t %b  \t|\t  %b   \t|\t%b",A,B,Cin,Cout,Sum);
10                 #5;
11
12                 A = 0; B = 0; Cin = 1;
13                 #5;
14                 A = 0; B = 1; Cin = 0;
15                 #5;
16
17                 A = 0; B = 1; Cin = 1;
18                 #5;
19                 A = 1; B = 0; Cin = 0;
20                 #5;
21
22                 A = 1; B = 0; Cin = 1;
23                 #5;
24                 A = 1; B = 1; Cin = 0;
25                 #5;
26                 A = 1; B = 1; Cin = 1;
27                 #5;
28         end
29 endmodule
30
```

**Step3:**

```
 1 module RCA(A, B, S, Cout);
 2          input [0:3] A, B;
 3          output [0:3] S;
 4          output Cout;
 5          wire [0:2] C;
 6
 7          FullAdder FA1(A[0], B[0], 0, S[0], C[0]);
 8          FullAdder FA2(A[1], B[1], C[0], S[1], C[1]);
 9          FullAdder FA3(A[2], B[2], C[1], S[2], C[2]);
10          FullAdder FA4(A[3], B[3], C[2], S[3], Cout);
11 endmodule
12
```

**Step4:**

**TestBench:**

E:/6TH SEMESTER/DSD LAB/Lab2/test_RCA.v

```
module testB_RCA;
        reg [3:0] A,B;
        wire [3:0] S;
        wire Cout;
        RCA RCA1(A, B, S, Cout);
        initial
        begin
                $display("              Decimal                   |                  Binary");
                $display("  Num1 |  Num2 | Carry | Sum  |    Num1 |  Num2 |   Carry   | Sum");
                A[3] = 0; A[2] = 1; A[1] = 0; A[0] = 0;
                B[3] = 1; B[2] = 0; B[1] = 1; B[0] = 0;
                #5;
                #1 $monitor("     %d    |    %d    |    %d    |    %d   |   %b   |   %b   |     %b     | %b",A,B,Cout,S,A,B,Cout,S);
                #5;

                A[3] = 0; A[2] = 1; A[1] = 1; A[0] = 0;
                B[3] = 1; B[2] = 1; B[1] = 0; B[0] = 0;
                #5;

                A[3] = 0; A[2] = 1; A[1] = 0; A[0] = 1;
                B[3] = 1; B[2] = 0; B[1] = 1; B[0] = 1;
                #5;

                A[3] = 0; A[2] = 0; A[1] = 1; A[0] = 1;
                B[3] = 1; B[2] = 1; B[1] = 1; B[0] = 1;
                #5;

                A[3] = 0; A[2] = 0; A[1] = 0; A[0] = 1;
                B[3] = 1; B[2] = 1; B[1] = 1; B[0] = 0;
                #5;

                A[3] = 0; A[2] = 1; A[1] = 1; A[0] = 1;
                B[3] = 1; B[2] = 0; B[1] = 1; B[0] = 0;
                #5;

        end
endmodule

```

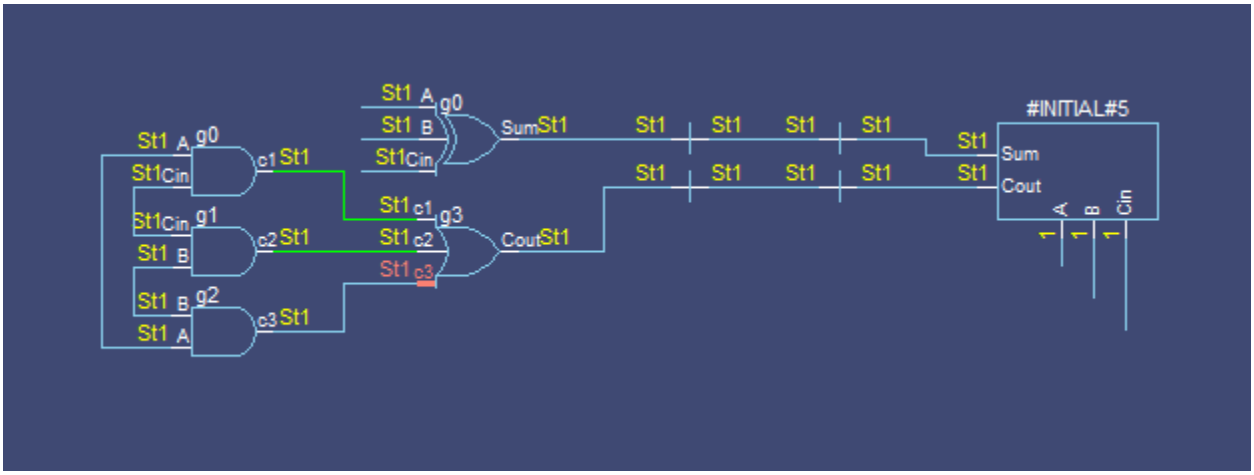
## OUTPUTS

**FULL ADDER:**

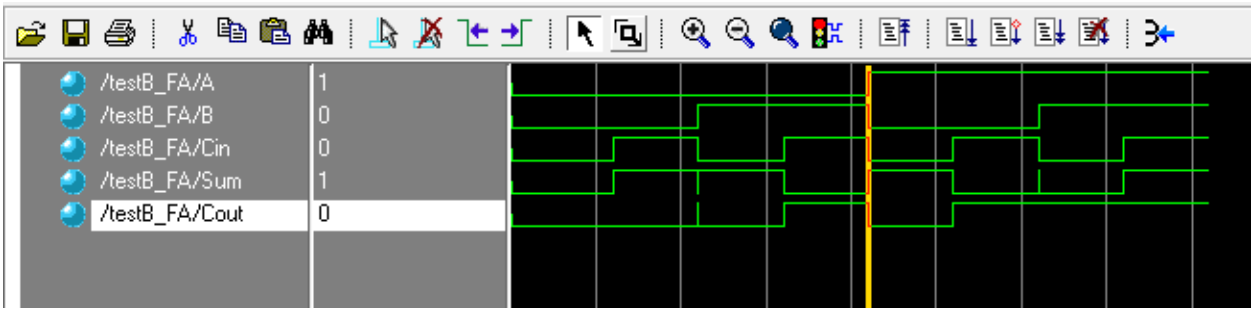
**Truth Table:**

```
# A | B | Cin | Carry | Sum
# 0| 0| 0  |  0   |0
# 0| 0| 1  |  0   |1
# 0| 1| 0  |  0   |1
# 0| 1| 1  |  1   |0
# 1| 0| 0  |  0   |1
# 1| 0| 1  |  1   |0
# 1| 1| 0  |  1   |0
# 1| 1| 1  |  1   |1
```

**Circuit Design:**




**Wave Form:**

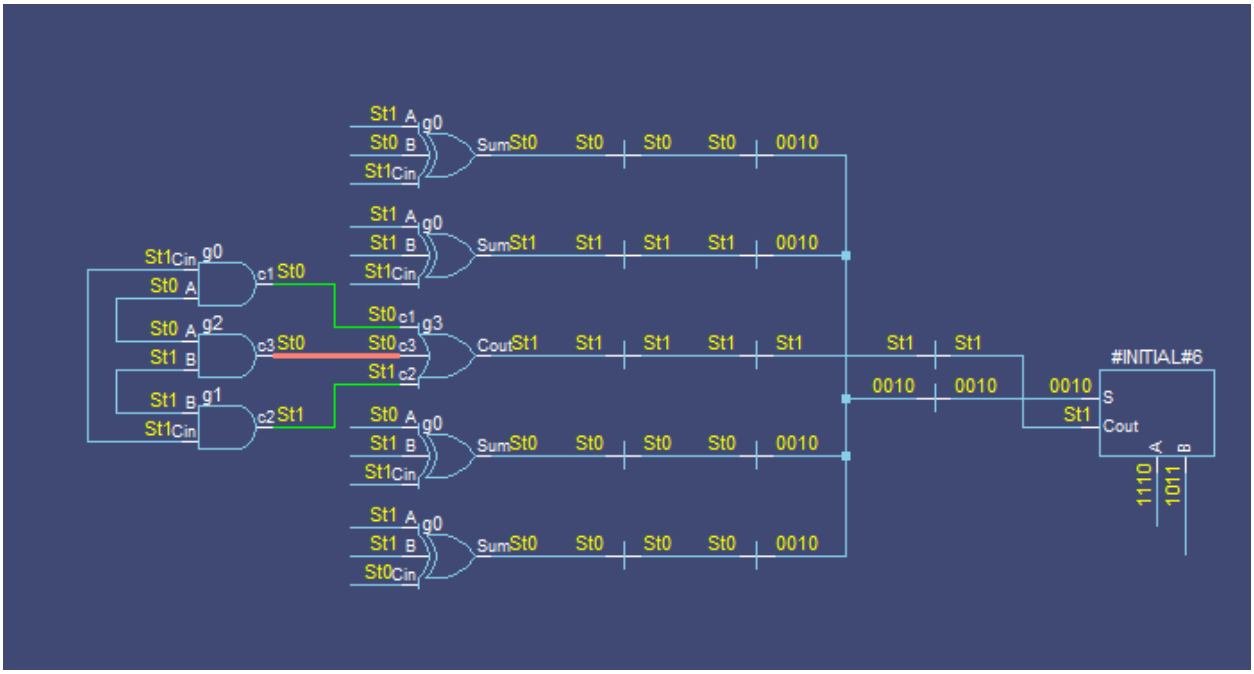



**RIPPLE CARRY ADDER:**

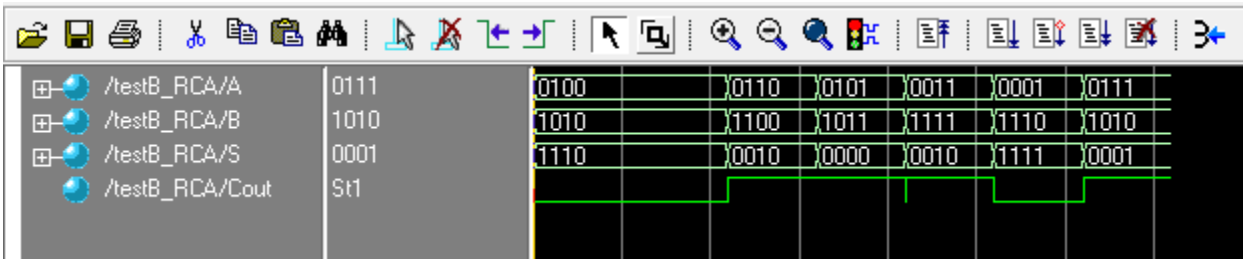
**Truth Table:**

| Decimal | | | | Binary | | | |
|---|---|---|---|---|---|---|---|
| Num1 | Num2 | Carry | Sum | Num1 | Num2 | Carry | Sum |
| 4 | 10 | 0 | 14 | 0100 | 1010 | 0 | 1110 |
| 6 | 12 | 1 | 2 | 0110 | 1100 | 1 | 0010 |
| 5 | 11 | 1 | 0 | 0101 | 1011 | 1 | 0000 |
| 3 | 15 | 1 | 2 | 0011 | 1111 | 1 | 0010 |
| 1 | 14 | 0 | 15 | 0001 | 1110 | 0 | 1111 |
| 7 | 10 | 1 | 1 | 0111 | 1010 | 1 | 0001 |

**Circuit Design:**




**Wave Form:**



**TASK01 (b):**

Design the 4-bit full adder using data flow level modeling.

**Step1:**

**CODE:**

```
ln #
 1 module FA_DFlow(A, B, Cin, Sum, Cout);
 2         input A,B,Cin;
 3         output Sum, Cout;
 4         wire c1,c2,c3;
 5
 6         assign Sum=(A ^ B ^ Cin);
 7         assign c1=(A & B);
 8         assign c2=(A & Cin);
 9         assign c3=(B & Cin);
10         assign Cout=(c1 | c2 | c3);
11 endmodule
12
```

**Step2:**

**TestBench:**

```
ln #                                                                    E:/6TH SEMESTER/
 1 module testB_FA_DFlow;
 2         reg A,B,Cin;
 3         wire Sum,Cout;
 4         FA_DFlow FA6(A, B, Cin, Sum, Cout);
 5         initial
 6         begin
 7                 $display("\tA\t|\tB\t|\tCin\t|\tCarry\t|\tSum");
 8                 A = 0; B = 0; Cin = 0;
 9                 #1 $monitor("\t%b\t|\t %b\t|\t %b  \t|\t  %b   \t|\t%b",A,B,Cin,Cout,Sum);
10                 #5;
11
12                 A = 0; B = 0; Cin = 1;
13                 #5;
14                 A = 0; B = 1; Cin = 0;
15                 #5;
16
17                 A = 0; B = 1; Cin = 1;
18                 #5;
19                 A = 1; B = 0; Cin = 0;
20                 #5;
21
22                 A = 1; B = 0; Cin = 1;
23                 #5;
24                 A = 1; B = 1; Cin = 0;
25                 #5;
26                 A = 1; B = 1; Cin = 1;
27                 #5;
28         end
29 endmodule
```

**Step3:**

```
ln #
 1 module RCA_D(A, B, S, Cout);
 2         input [3:0] A, B;
 3         output [3:0] S;
 4         output Cout;
 5         wire [2:0] C;
 6
 7         FA_DFlow FA1(A[0], B[0], 1'b0, S[0], C[0]);
 8         FA_DFlow FA2(A[1], B[1], C[0], S[1], C[1]);
 9         FA_DFlow FA3(A[2], B[2], C[1], S[2], C[2]);
10         FA_DFlow FA4(A[3], B[3], C[2], S[3], Cout);
11 endmodule
12
```

**Step4:**

**TestBench:**

```
ln #                                                                    E:/6TH SEMESTER/DSD LAB/Lab2/TestB_RPC_DFlow.v
 1 module testB_RCA_D;
 2         reg [3:0] A,B;
 3         wire [3:0] S;
 4         wire Cout;
 5         RCA_D RCA1(A, B, S, Cout);
 6         initial
 7         begin
 8                 $display("              Decimal                 |                 Binary");
 9                 $display("  Num1 |  Num2 | Carry | Sum  |     Num1 |  Num2 |   Carry   | Sum");
10                 A[3] = 0; A[2] = 1; A[1] = 0; A[0] = 0;
11                 B[3] = 1; B[2] = 0; B[1] = 1; B[0] = 0;
12                 #5;
13                 #1 $monitor("     %d    |    %d    |    %d    |    %d   |   %b   |   %b   |     %b     | %b",A,B,Cout,S,A,B,Cout,S);
14                 #5;
15
16                 A[3] = 0; A[2] = 1; A[1] = 1; A[0] = 0;
17                 B[3] = 1; B[2] = 1; B[1] = 0; B[0] = 0;
18                 #5;
19
20                 A[3] = 0; A[2] = 1; A[1] = 0; A[0] = 1;
21                 B[3] = 1; B[2] = 0; B[1] = 1; B[0] = 1;
22                 #5;
23
24                 A[3] = 0; A[2] = 0; A[1] = 1; A[0] = 1;
25                 B[3] = 1; B[2] = 1; B[1] = 1; B[0] = 1;
26                 #5;
27
28                 A[3] = 0; A[2] = 0; A[1] = 0; A[0] = 1;
29                 B[3] = 1; B[2] = 1; B[1] = 1; B[0] = 0;
30                 #5;
31
32                 A[3] = 0; A[2] = 1; A[1] = 1; A[0] = 1;
33                 B[3] = 1; B[2] = 0; B[1] = 1; B[0] = 0;
34                 #5;
35
36         end
37 endmodule
38
```

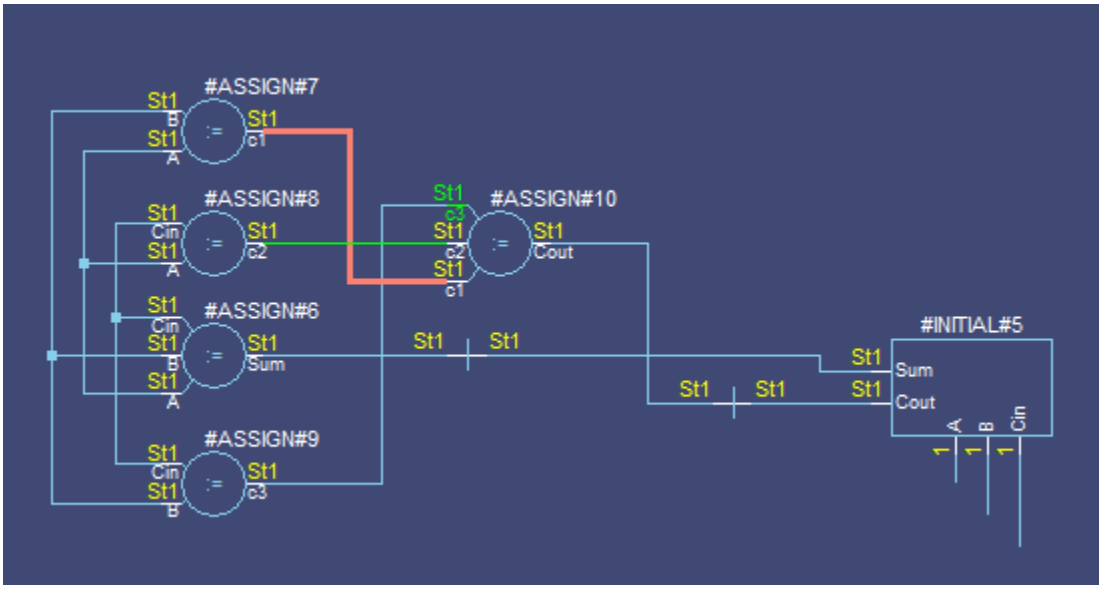
# OUTPUTS

## Full Adder:

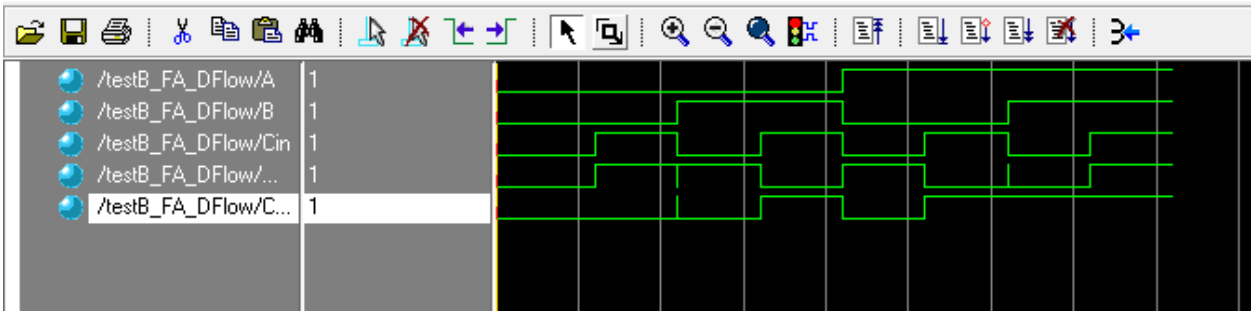
## Truth Table:

```
# A | B | Cin | Carry | Sum
# 0 | 0 | 0   |  0    | 0
# 0 | 0 | 1   |  0    | 1
# 0 | 1 | 0   |  0    | 1
# 0 | 1 | 1   |  1    | 0
# 1 | 0 | 0   |  0    | 1
# 1 | 0 | 1   |  1    | 0
# 1 | 1 | 0   |  1    | 0
# 1 | 1 | 1   |  1    | 1
```

## Circuit Design:




## Wave Form:

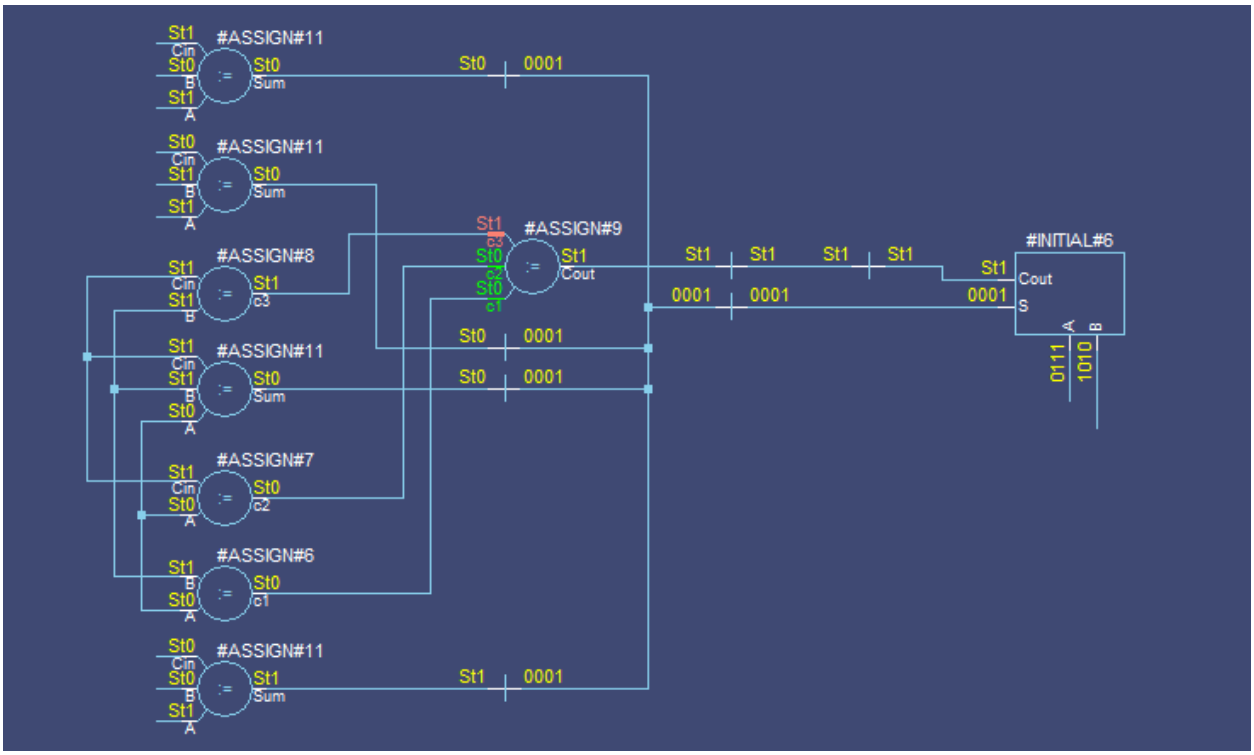


**Ripple Carry Adder:**

**Truth Table:**

| Decimal | | | | Binary | | | |
|---|---|---|---|---|---|---|---|
| Num1 | Num2 | Carry | Sum | Num1 | Num2 | Carry | Sum |
| 4 | 10 | 0 | 14 | 0100 | 1010 | 0 | 1110 |
| 6 | 12 | 1 | 2 | 0110 | 1100 | 1 | 0010 |
| 5 | 11 | 1 | 0 | 0101 | 1011 | 1 | 0000 |
| 3 | 15 | 1 | 2 | 0011 | 1111 | 1 | 0010 |
| 1 | 14 | 0 | 15 | 0001 | 1110 | 0 | 1111 |
| 7 | 10 | 1 | 1 | 0111 | 1010 | 1 | 0001 |

**Circuit Design:**




**Wave Form:**