RTL CODING

#DAY2 Yash Ekhande

Aim - To make a 4 Bit Ripple Adder using module instantiation

CODE

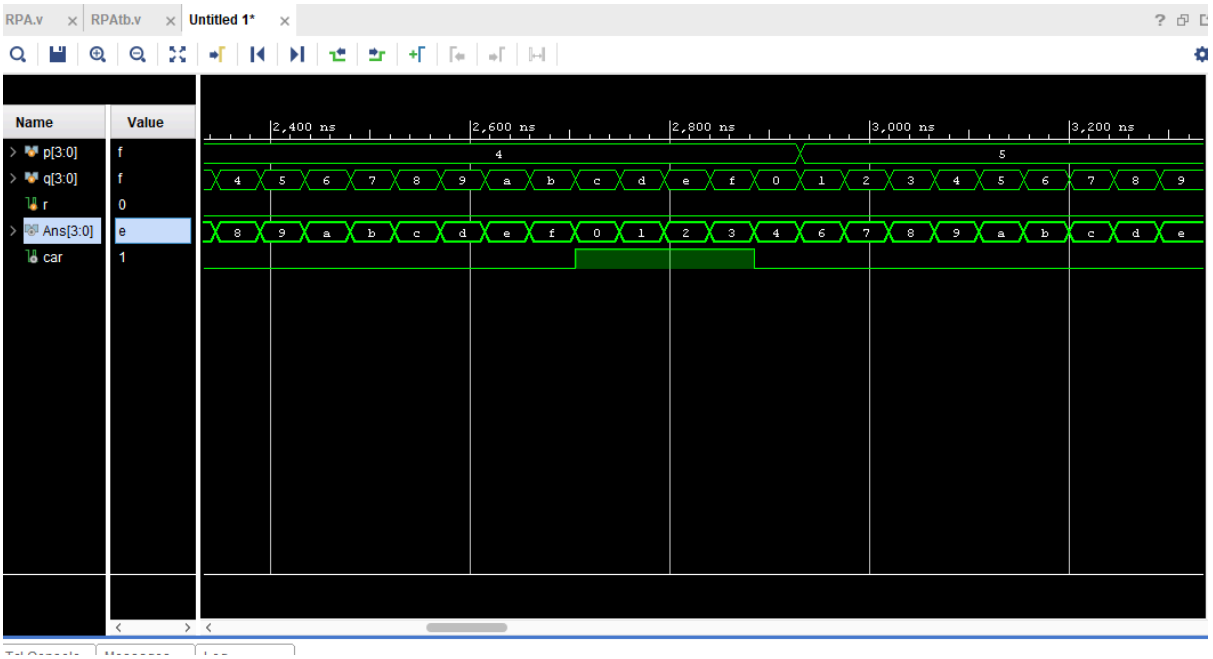
MODULE INSTANTIATION (for checking Full adder module pls check my Day1 code)

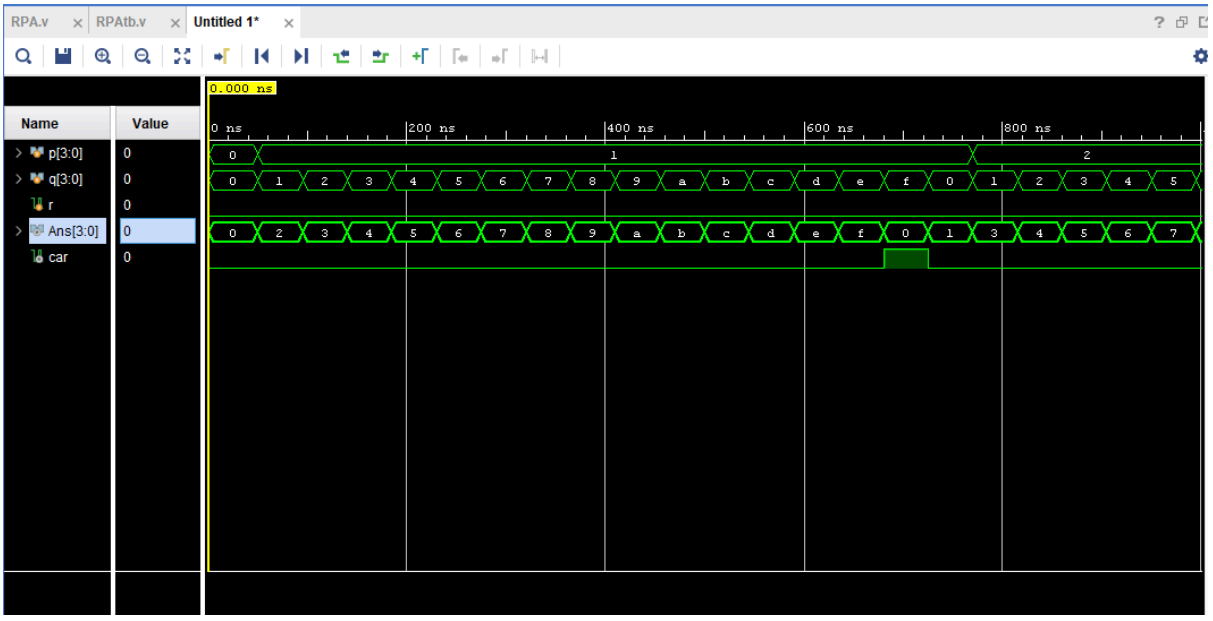
```
1  `timescale 1ns / 1ps
2  module RPA(
3      input [3:0] a,
4      input [3:0] b,
5      input c,
6      output [3:0] sum,
7      output cout
8      );
9      wire c1,c2,c3;
10     FA_gate d0(a[0],b[0],c,sum[0],c1);     //INSTANTIATION OF FULL ADDER
11     FA_gate d1(a[1],b[1],c1,sum[1],c2);
12     FA_gate d2(a[2],b[2],c2,sum[2],c3);
13     FA_gate d3(a[3],b[3],c3,sum[3],cout);
14 endmodule
```

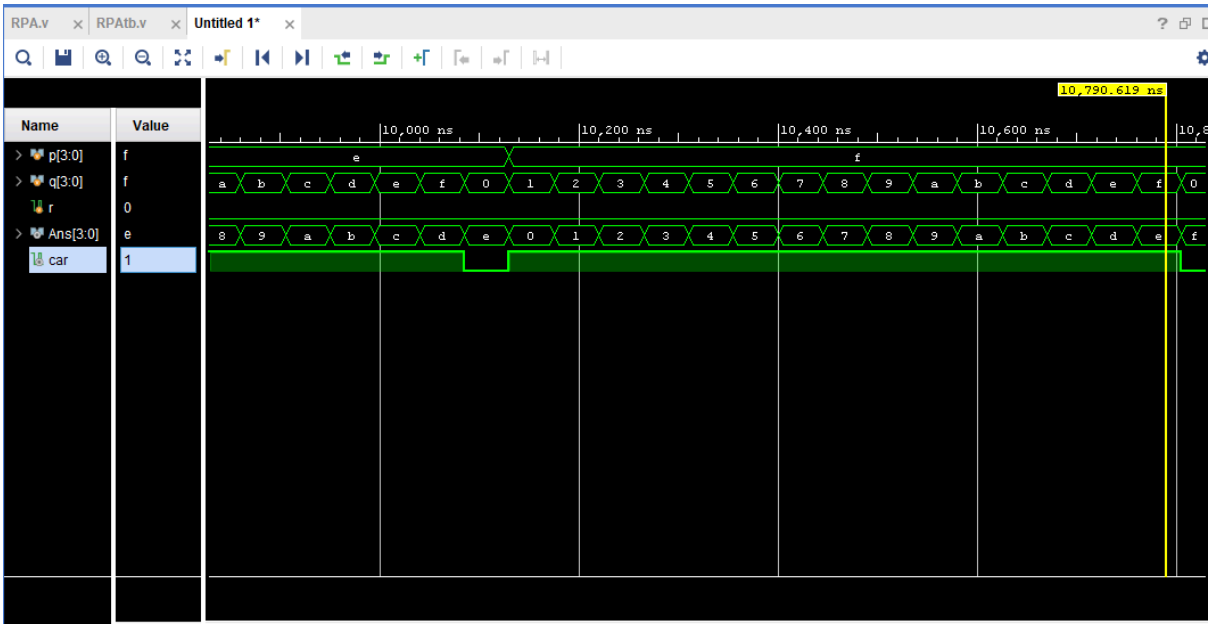
TESTBENCH

```
1      module RPAtb;
2      reg [3:0] p;
3      reg [3:0] q;
4      reg r;
5      wire [3:0] Ans;
6      wire car;
7      RPA h1(p,q,r,Ans,car);
8      integer i,j; //data type declaration for loops\
9      initial begin
10     p=4'b0000; //initial value declration(p,a)
11     q=4'b0000; //initial value declration(q,b)
12     r=1'b0; //initial value declaration of c
13     #50;
14     for(i=0;i<16;i=i+1) //changing values from 0 to 16 of a/p
15         begin
16         p=p+1; //acts as counter
17         for(j=0;j<16;j=j+1) // changing values for 0 to 16 of q/b
18         begin
19             q=q+1;
20             #45;
21         end
22     end
23     end
24     endmodule
25
```

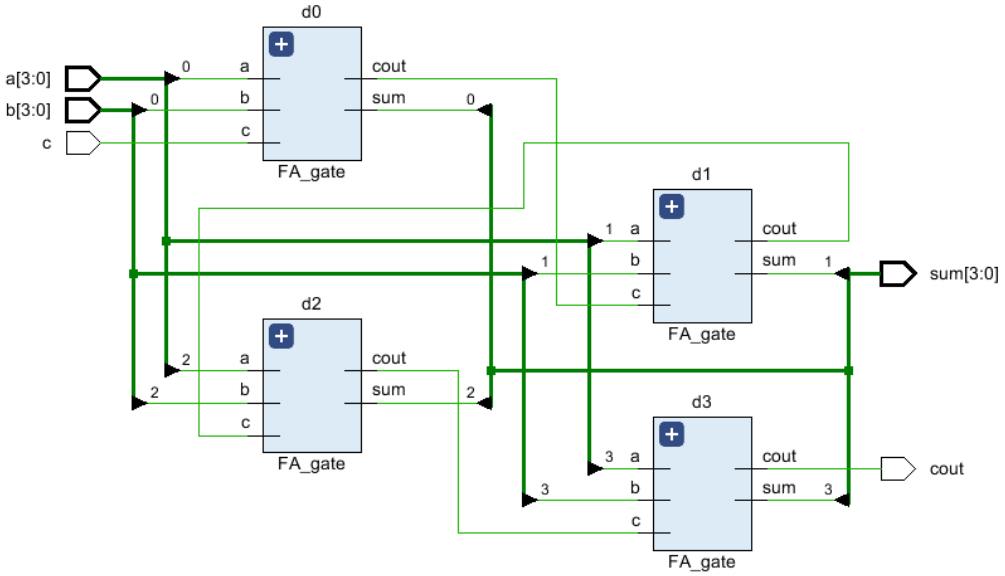
WAVEFORM


RPA.v
RPAtb.v
Untitled 1*
Name
Value
p[3:0]
q[3:0]
r
Ans[3:0]
car
2,400 ns
2,600 ns
2,800 ns
3,000 ns
3,200 ns


RPA.v
RPAtb.v
Untitled 1*
0.000 ns
Name
Value
p[3:0]
q[3:0]
r
Ans[3:0]
car
0 ns
200 ns
400 ns
600 ns
800 ns


RPA.v
RPAtb.v
Untitled 1*
10,790.619 ns
Name
Value
p[3:0]
q[3:0]
r
Ans[3:0]
car
10,000 ns
10,200 ns
10,400 ns
10,600 ns

SCHEMATIC


d0
a[3:0]
b[3:0]
c
0
0
a
b
c
cout
sum
0
FA_gate
d1
1
1
a
b
c
cout
sum
1
FA_gate
sum[3:0]
d2
2
2
a
b
c
cout
sum
2
FA_gate
d3
3
3
a
b
c
cout
sum
3
FA_gate
cout