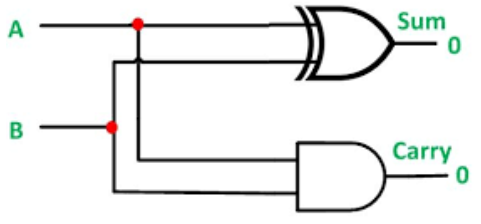
**Hardware Runs**

**Run 1: Half adder**

**Diagram**

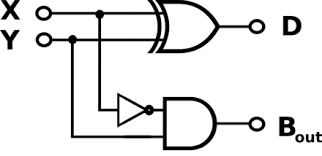
****

**Truth Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **SUM** | **CARRY** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Run 2: Half subtractor**

**Diagram**



**Truth Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **DIFFERENCE** | **BORROW** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

**Software Runs**

**Run 3: Full adder and 4-bit adder**

**Q1:** Write the Verilog code and testbench of Full adder using data flow modeling.

**A: Verilog Code-** [**https://www.edaplayground.com/x/n\_La**](https://www.edaplayground.com/x/n_La)

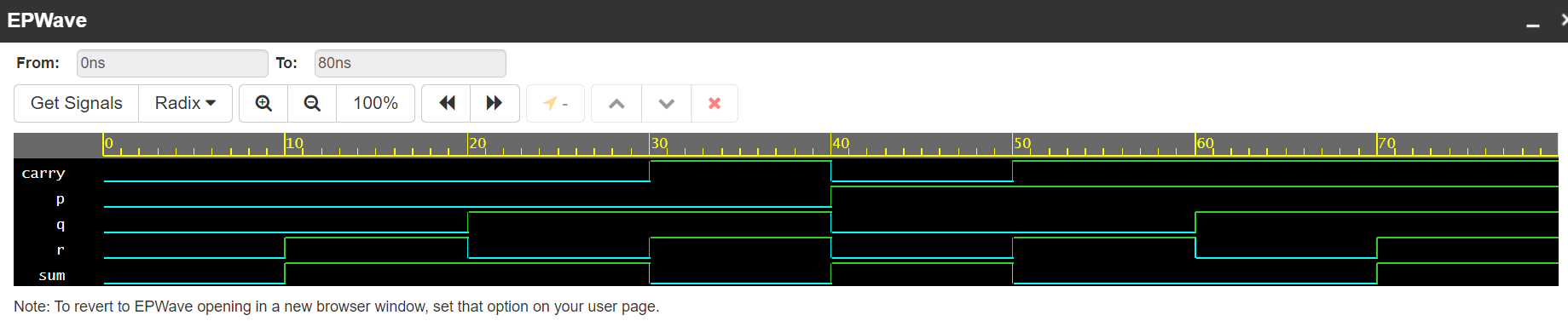
module full\_adder\_dataflow(input a,b,cin,output s, carry);

assign s=a^b^cin;

assign carry=(a&b)|(a&cin)|(b&cin);

endmodule

**Q2:** Paste the Image of your **Simvision** window where you get the waveforms for the above code.

**A:**

**Q3:** Write the Verilog code and testbench for 4- bit parallel adder using structural modeling, use full adder as a building block.

**A: Verilog Code-** [**https://www.edaplayground.com/x/6rk3**](https://www.edaplayground.com/x/6rk3)

module full\_adder\_dataflow(input a,b,cin,output s, carry);

assign s=a^b^cin;

assign carry=(a&b)|(a&cin)|(b&cin);

endmodule

module main(in1,in2,ic,sum,c\_final);

input [3:0]in1;

input [3:0]in2;

input ic;

wire c0,c1,c2;

output [3:0]sum;

output c\_final;

full\_adder\_dataflow fa1(in1[0],in2[0],ic,sum[0],c0);

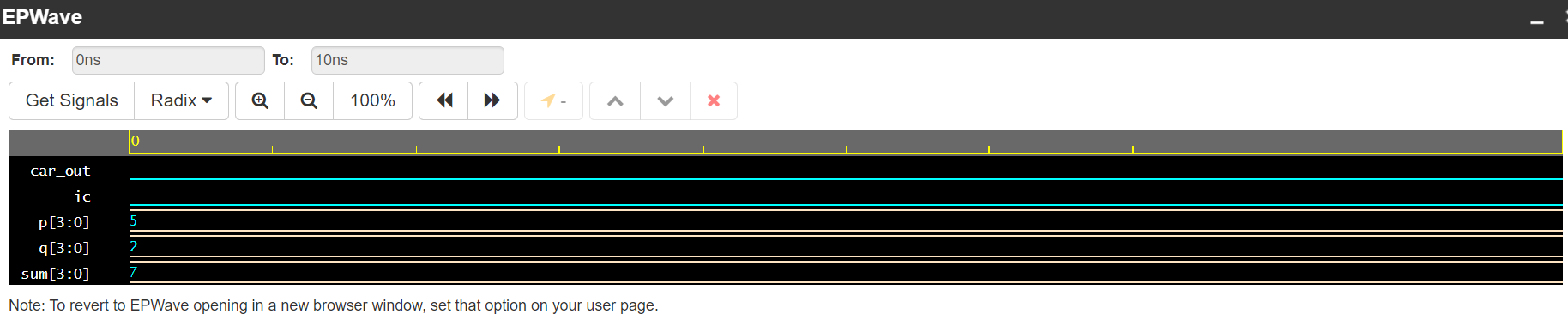
full\_adder\_dataflow fa2(in1[1],in2[1],c0,sum[1],c1);

full\_adder\_dataflow fa3(in1[2],in2[2],c1,sum[2],c2);

full\_adder\_dataflow fa4(in1[3],in2[3],c2,sum[3],c\_final);

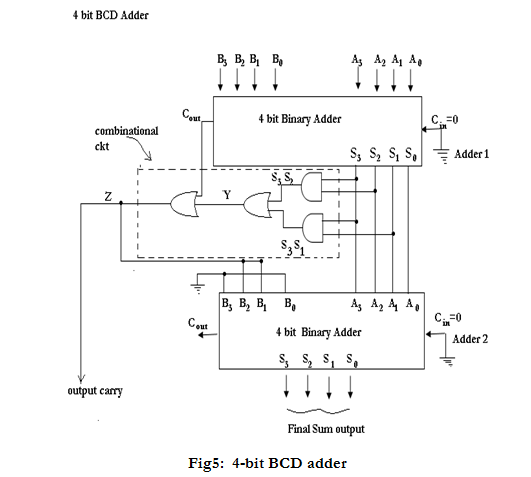
endmodule

**Q4:** Paste the Image of your **Simvision** window where you get the waveforms for the above code.

**A: **

**Run 4: BCD adder**

**Q5:** Write the verilog code and testbench for BCD adder using structural modeling, use 4-bit parallel adder and other gates as building blocks. (Hint refer the image below)



**A: Verilog Code-** [**https://www.edaplayground.com/x/Ghcu**](https://www.edaplayground.com/x/Ghcu)

**module full\_adder\_dataflow(input a,b,cin,output s, carry);**

**assign s=a^b^cin;**

**assign carry=(a&b)|(a&cin)|(b&cin);**

**endmodule**

**module parallel\_adder(in1,in2,ic,sum,c\_final);**

**input [3:0]in1;**

**input [3:0]in2;**

**input ic;**

**wire c0,c1,c2;**

**output [3:0]sum;**

**output c\_final;**

**full\_adder\_dataflow fa1(in1[0],in2[0],ic,sum[0],c0);**

**full\_adder\_dataflow fa2(in1[1],in2[1],c0,sum[1],c1);**

**full\_adder\_dataflow fa3(in1[2],in2[2],c1,sum[2],c2);**

**full\_adder\_dataflow fa4(in1[3],in2[3],c2,sum[3],c\_final);**

**endmodule**

**module BCD(a,b,cin,s,cout);**

**input[3:0]a;**

**input[3:0]b;**

**input cin;**

**output cout;**

**output [3:0]s;**

**parallel\_adder p1(a,b,cin,s,cout);**

**endmodule**

**module main(A,B,Cin,Sum,Cout);**

**input [3:0]A;**

**input [3:0]B;**

**input Cin;**

**output [3:0]Sum;**

**output Cout;**

**wire c\_out, s3s2, s3s1, f;**

**wire [3:0]sum\_int;**

**wire [3:0]next;**

**BCD b1(A,B,Cin,sum\_int,c\_out);**

**and a1(s3s2, sum\_int[3], sum\_int[2]);**

**and a2(s3s1, sum\_int[3], sum\_int[1]);**

**or r1(f,c\_out,s3s2,s3s1);**

**assign next[3]=0;**

**assign next[2]=f;**

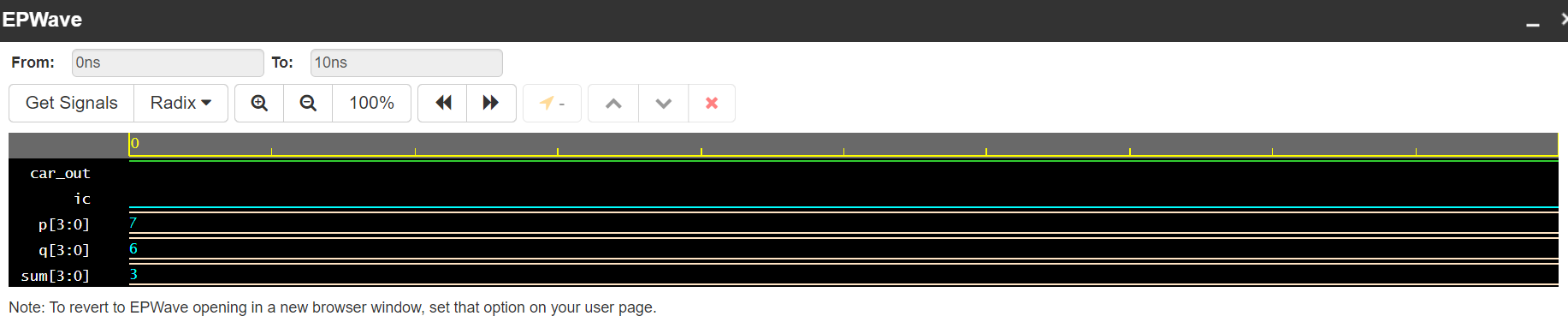
**assign next[1]=f;**

**assign next[0]=0;**

**BCD b2(next,sum\_int,Cin,Sum,Cout);**

**endmodule**

**Q6:** Paste the Image of your **Simvision** window where you get the waveforms for the above code.

**A: **