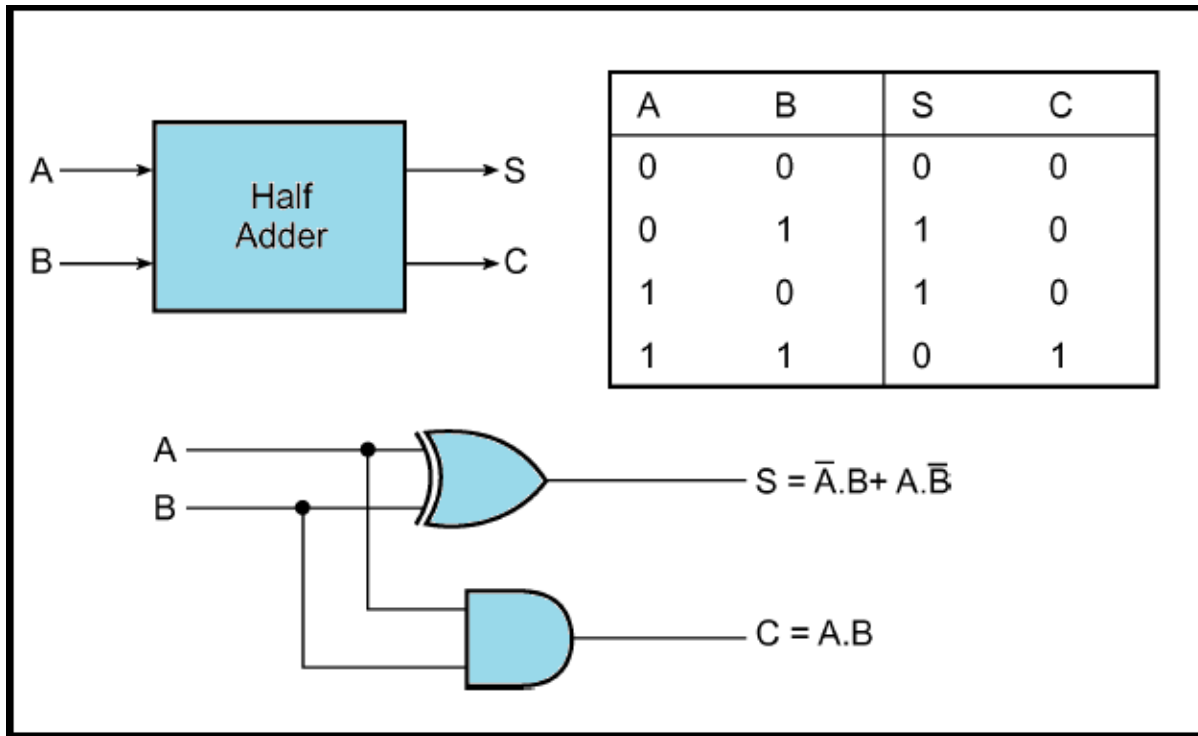
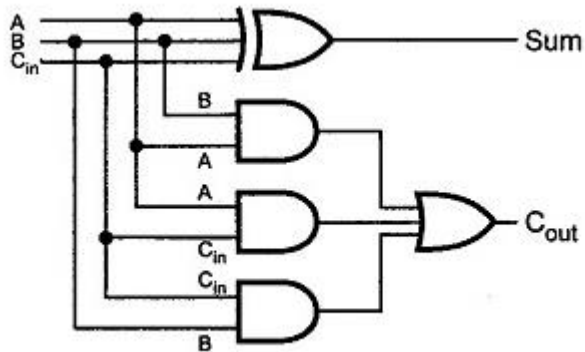


**Experiment No. 3****Aim: Implementation of half and full adder using gate level and data flow modeling in Verilog****Apparatus:** Xilinx ISE Design Suite, Verilog Code**Theory:** Truth table of half adder and full adder

Full Adder



Inputs			Outputs	
$A$	$B$	$C_{in}$	$S$	$C_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

**Procedure:**

- 1) Open the Xilinx Vivado Design Suite
- 2) Go to file and click new project
- 3) Enter the project name and click next
- 4) Select the family name of the FPGA Device, parameter setting and HDL is verilog click next and click finish.
- 5) Click new source.
- 6) Select Verilog module and type file name and click next.
- 7) Assign input and output port and click next.
- 8) Finally, the report is shown click finish.
- 9) Type the program save and check syntax error.
- 10) To see the output waveform select ISim simulator
- 11) Give values to the input variables using force clock or force constant and then click run
- 12) In wave window, click run icon and you can see corresponding output.
- 13) For synthesis of the design, open XST synthesis tool and run the design for synthesis.
- 14) Open RTL schematic and Technology schematic and understand implemented design on FPGA
- 15) Open synthesis result to know resource utilization of the design.

**Code:**

- 1) **HALF ADDER**

```

Module ha_glm (input a, input b, output sum,
output c_out );
  xor x1(sum,a,b);
  and n1(c_out,a,b);

endmodule

```

Gate level modeling

```

module ha_dflm(input a,
input b, output s, output c);
  assign
  s=a^b;
  assign
  c=a&b;

endmodule

```

Dataflow modeling

## 2) FULL ADDER

```

module fa_glm ( input x,
input y,
input cin,
output A,
output
cout );
  wire p,r,s;
  xor(p,x,y)
  ;
  xor (A,p,cin);
  and(r,p,cin);
  and(s,x,y);
  or(cout,r,s);

endmodule

```

Gate level modeling

```

odule fsd_dlm ( input x,
input y, input
cin, output A,
output cout);

  assign A=(x^y)^cin;
  assign cout =((x^y)&cin)|(x&y);

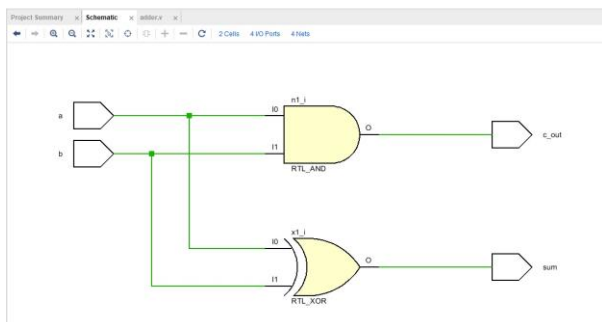
endmodule

```

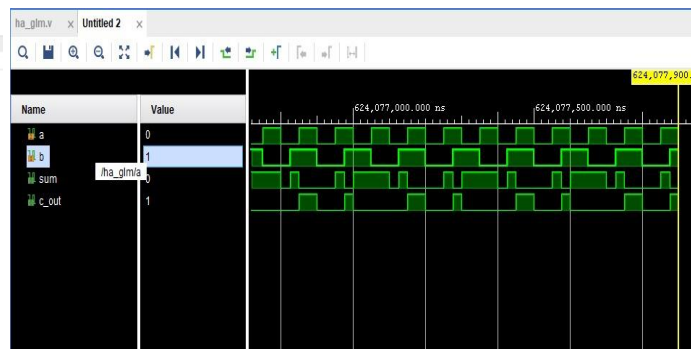
Dataflow modeling

## Diagrams & Simulation Images:

### 1. Half-Adder:



Schematic Diagram



Behavioral Simulation

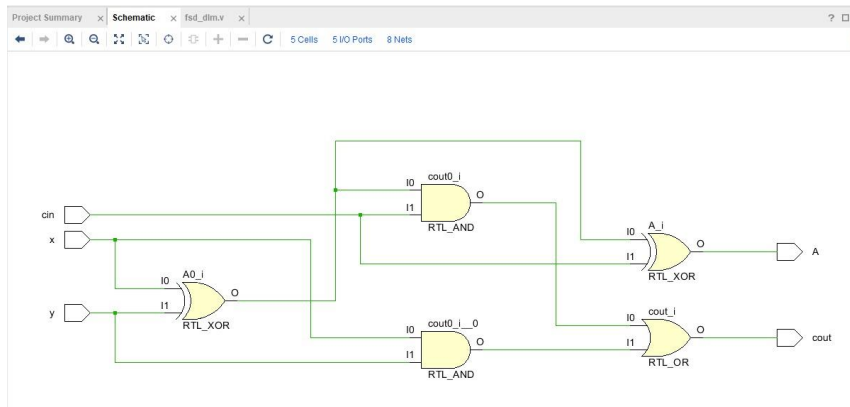
```
module ha_glm(  
    input a,  
    input b,  
    output sum,  
    output c_out  
);  
xor x1(sum,a,b);  
and  
n1(c_out,a,b);  
endmodule  
//22070123043
```

Gate-Level-Model

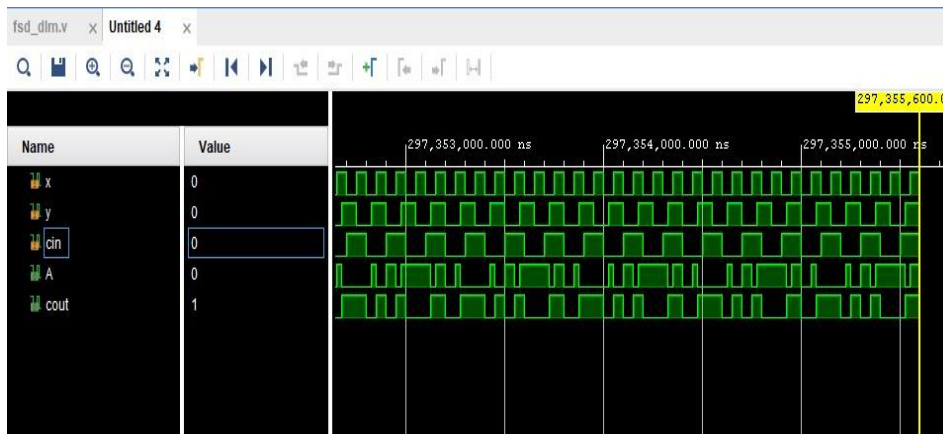
```
module ha_dflm(  
    input a,  
    input b,  
    output s,  
    output c  
);  
assign  
s=a^b;  
assign  
c=a&b;  
endmodule  
//22070123043
```

Data-flow-Level Model

## 2. Full Adder:



Schematic Diagram Full Adder



Full Adder Behavioral Simulation

```
module fsd_glm(  
    input x,  
    input y,  
    input cin,  
    output A,  
    output cout  
);  
    wire p,r,s;  
    xor(p,x,y);  
    xor(A,p,cin);  
    and(r,p,cin);  
    and(s,x,y);  
    or(cout,r,s);  
endmodule  
//22070123043
```

Full-Adder Code

**Conclusion:** In this lab experiment, we successfully explored the design and implementation of Half Adder and Full Adder circuits using gate-level and data-flow level Verilog modeling. Through simulation in Vivado, we verified the functionality and correctness of both adder designs. This hands-on experience provided a deeper understanding of digital circuit design and the advantages of different modeling approaches in Verilog.