**Experiment 2. Half adder and full adder**

Date: 14/05/21

**AIM**

a) Development of Verilog modules for half adder in 3 modeling styles (dataflow/  
structural/behavioural).

**VERILOG CODE**

**Gate level**

module halfadder(

input a,b,

output s,cout);

xor x1(s,a,b);

and A1(cout,a,b);

endmodule

**Data flow level**

module halfadder(

input wire a,b,

output wire s,cout);

assign s = a^b;

assign cout = a&b;

endmodule

**Behavioural level**

module halfadder(

input wire a,b,

output reg s,cout);

always @ (a or b)

begin

if((a==1'b0 && b ==1'b1)||(a==1'b1 && b==1'b0))

begin

s = 1'b1;

end

else

s = 1'b0;

if(a==1'b1 && b ==1'b1)

begin

cout = 1'b1;

end

else

cout = 1'b0;

end

endmodule

**Testbench**

// Code your testbench here

// or browse Examples

`timescale 1ns/1ps

module test\_half;

reg t\_a,t\_b;

wire t\_s,t\_cout;

halfadder uut(.a(t\_a),.b(t\_b),.s(t\_s),.cout(t\_cout));

initial

begin

$dumpvars(1,test\_half);

t\_a=1'b0;

t\_b=1'b0;

#10;

t\_a=1'b0;

t\_b=1'b1;

#10;

t\_a=1'b1;

t\_b=1'b0;

#10;

t\_a=1'b1;

t\_b=1'b1;

#10;

$stop;

end

endmodule

**Output**

