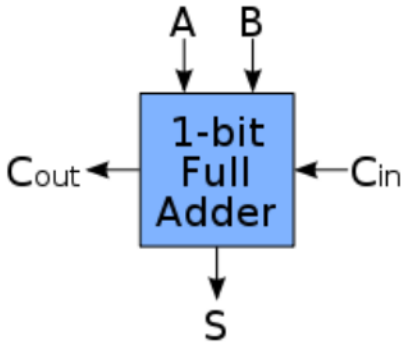
Lab No 10: **Date:**

FUNCTIONS OF A FULL ADDER USING THREE MODELLING STYLES

THEORY In electronics, an adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic units, but also in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators and similar operations.

A **full adder** adds binary numbers and accounts for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as A, B, and C_{in}; A and B are the operands, and C_{in} is a bit carried in from the previous less significant stage. The full adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. bit binary numbers. The circuit produces a two-bit output, output carry and sum typically represented by the signals C_{out} and S. The one-bit full adder's truth table is:



Block Diagram:

Expression:

```
sum = a_in ⊕ b_in ⊕ c_in;
carry= (a_in.b_in)+(b_in.c_in)+(a_in.b_in);
```

Truth Table:

Inputs			Outputs	
a_in	b_in	c_in	sum	carry
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

a) Verilog code for full adder in DATAFLOW Style

```
module fulladder(a_in, b_in, c_in, sum, carry);
input a_in, b_in,c_in;
output sum, carry;
assign sum = a_in^b_in^c_in;
assign carry = (a_in & b_in)|(b_in & c_in)|(a_in & c_in);
endmodule
```

b) Verilog code for full adder in Behavioral Style

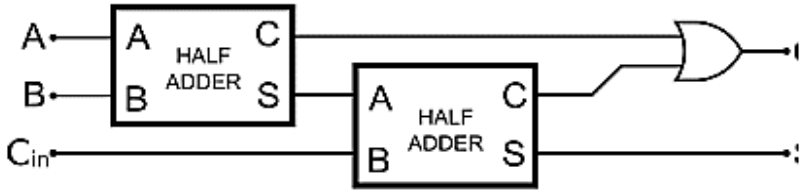
```
module fulladder(abc, sum, carry);
input [2:0] abc;
output reg sum,carry;

always@(abc)
begin
case (abc)
3'b000: begin  sum=1'b0; carry=1'b0; end
3'b001: begin sum=1'b1; carry=1'b0; end
3'b010: begin sum=1'b1; carry=1'b0; end
3'b011: begin sum=1'b0; carry=1'b1; end
3'b100: begin sum=1'b1; carry=1'b0;end
3'b101: begin sum=1'b0; carry=1'b1;end
3'b110:begin sum=1'b0; carry=1'b1;end
3'b111:begin sum=1'b1; carry=1'b1;end
endcase
end
endmodule
```

c) Verilog Code to describe Full adder using Structural Style

Block Diagram:

Logic Diagram:



Truth Table:

Inputs			Outputs	
a_in	b_in	c_in	sum	carry
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

VERILOG CODE :

```
module fulladder(a_in, b_in, c_in, sum, carry);
input a_in,b_in, c_in;
output sum, carry;
wire temp1, temp2, temp3;
halfadder ha1 (a_in, b_in, temp1, temp2);
 halfadder ha2 (c_in, temp1, sum, temp3);
 assign carry= temp3 | temp2;
endmodule

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