

RTL CHALLENGE

DAY-8:- Verilog Code for Implementation of FULL SUBTRACTOR using GateLevel, DataFlow & Structural.

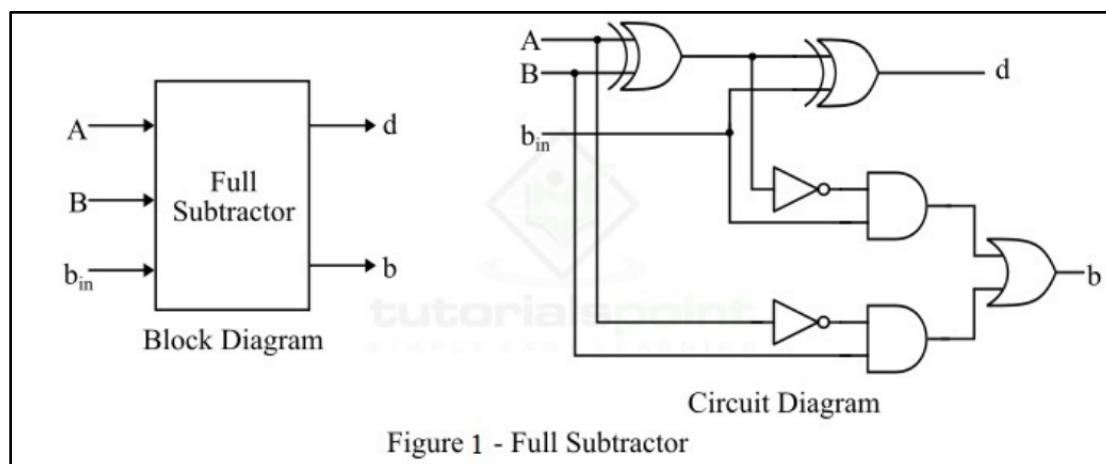
Software used:- Xilinx Vivado(2023.1)

Theory:

Full Subtractor:-

A full-subtractor is a combinational circuit that has three inputs A, B, b_{in} and two outputs d and b. Where, A is the minuend, B is subtrahend, b_{in} is borrow produced by the previous stage, d is the difference output and b is the borrow output. As we know that the half-subtractor can only be used for subtraction of LSB (least significant bit) of binary numbers. If there is any borrow during the subtraction of the LSBs of two binary numbers, then it will affect the subtraction of next stages. Therefore, the subtraction with borrow are performed by a full subtractor.

The block diagram and circuit diagram of a full-subtractor is shown in Figure-1.



Truth Table of Full-Subtractor:

Inputs			Outputs	
A	B	Borrow _{in}	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

DESIGN CODES & TESTBENCHES:-

Code for Full Subtractor using GATELEVEL:

```
// Design Name: GateLevel
// Module Name: FullSubtractorGateLevel
// Project Name: RTL Challenge

module FullSubtractorGateLevel(
    input a,b,c,
    output difference,borrow
);

    wire W1,W2,W3;
    xor (W1,a,b);
    and (W2,!a,b);
    xor (difference,W1,c);
    and (W3,!W1,c);
    or (borrow,W2,W3);

endmodule
```

TEST BENCH:

```
// Design Name:  
// Module Name: FullSubtractorGateLevel_tb  
// Project Name:  
  
module FullSubtractorGateLevel_tb();  
reg a,b,c;  
wire difference,borrow;  
FullSubtractorGateLevel uut(.a(a),.b(b),.c(c),.difference(difference),.borrow(borrow));  
initial  
begin  
  
a=0;b=0;c=0;#100;  
a=0;b=0;c=1;#100;  
a=0;b=1;c=0;#100;  
a=0;b=1;c=1;#100;  
a=1;b=0;c=0;#100;  
a=1;b=0;c=1;#100;  
a=1;b=1;c=0;#100;  
a=1;b=1;c=1;#100;  
  
end  
endmodule
```

Code for Full Subtractor using DATAFLOW:

```
// Design Name: DATAFLOW  
// Module Name: FullSubtractorDataFlow  
// Project Name: RTL SCHEMATIC  
  
module FullSubtractorDataFlow(  
    input a,b,c,  
    output difference,borrow  
);  
  
    wire W1,W2,W3;  
    assign W1 = (a^b);  
    assign W2 = (!a&b);  
    assign difference = (W1^c);  
    assign W3 = (!W1&c);  
    assign borrow = (W2|W3);  
  
endmodule
```

TEST BENCH:

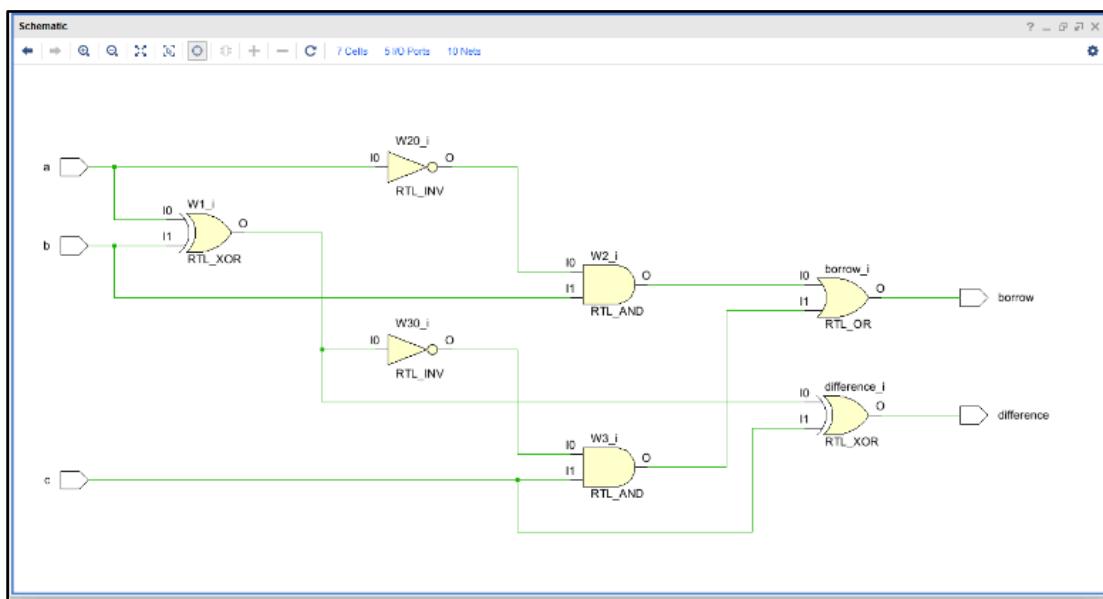
```
// Design Name: TESTBENCH
// Module Name: FullSubtractorDataFlow_tb
// Project Name: RTL SCHEMATIC

module FullSubtractorDataFlow_tb();
reg a,b,c;
wire difference,borrow;
FullSubtractorDataFlow uut(.a(a),.b(b),.c(c),.difference(difference),.borrow(borrow));
initial
begin

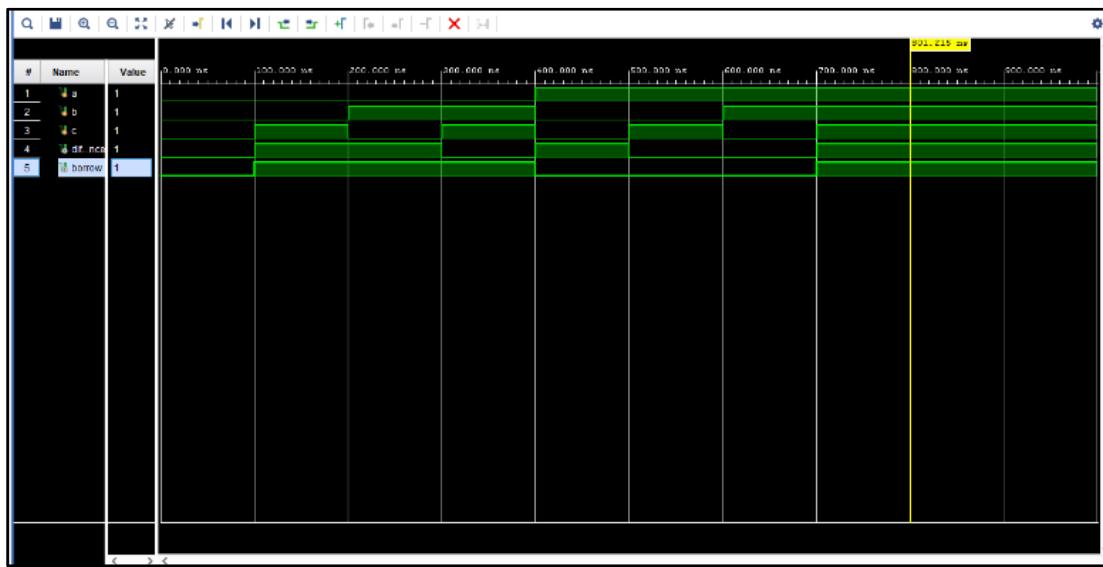
    a=0;b=0;c=0;#100;
    a=0;b=0;c=1;#100;
    a=0;b=1;c=0;#100;
    a=0;b=1;c=1;#100;
    a=1;b=0;c=0;#100;
    a=1;b=0;c=1;#100;
    a=1;b=1;c=0;#100;
    a=1;b=1;c=1;#100;

end
endmodule
```

RTL SCHEMATIC FOR FULL SUBTRACTOR USING GATELEVEL & DATAFLOW:



OUTPUT WAVEFORM:



Code for Full Subtractor using STRUCTURAL:

```
// Design Name: STRUCTURAL
// Module Name: FullSubtractorStructural
// project Name: RTL SCHEMATIC

module FullSubtractorStructural(
    input a,b,c,
    output difference,borrow
);
    wire W1,W2,W3;

    xorgate xl (.a(a),.b(b),.y(W1));
    andgate al (.a(!a),.b(b),.y(W2));
    xorgate x2 (.a(W1),.b(c),.y(difference));
    andgate a2 (.a(!W1),.b(c),.y(W3));
    orgate ol (.a(W3),.b(W2),.y(borrow));

endmodule
```

TEST BENCH:

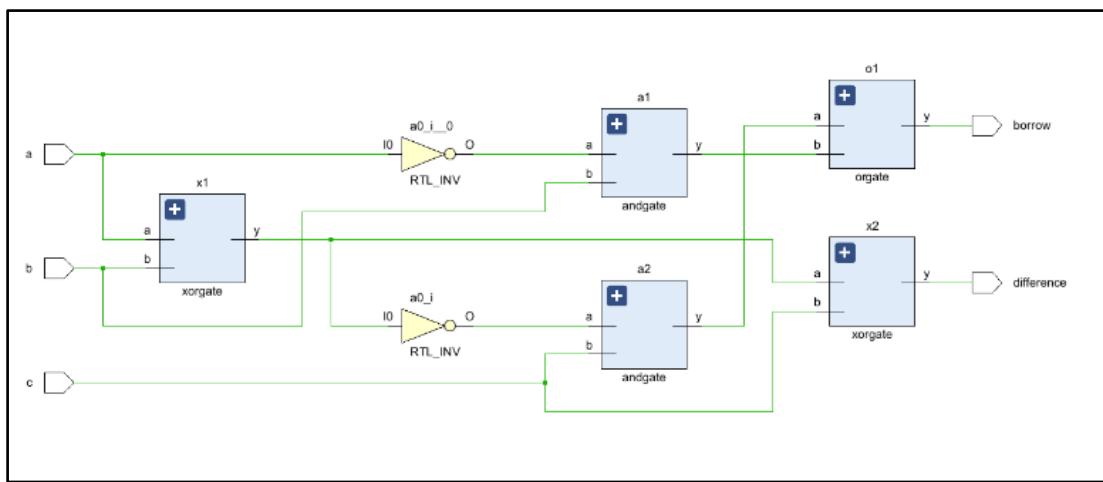
```
// Design Name: TESTBENCH
// Module Name: FullSubtractorStructural_tb
// Project Name: RTL SCHEMATIC

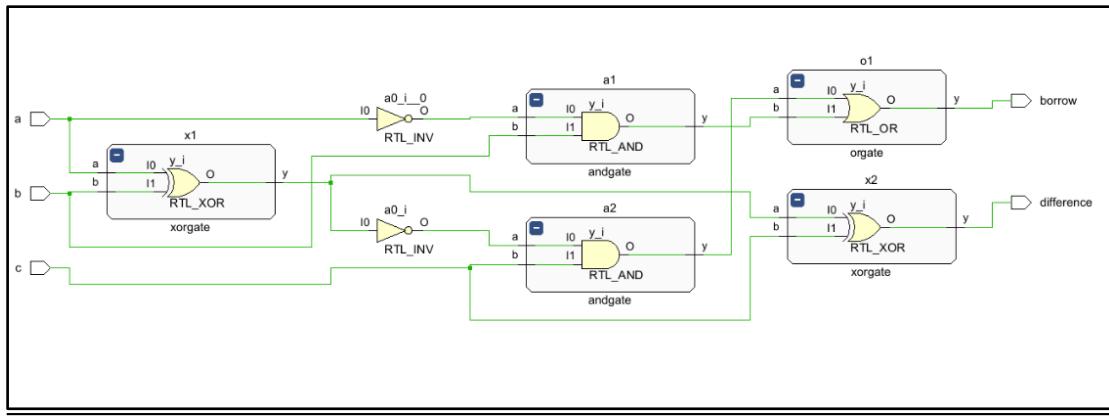
module FullSubtractorStructural_tb();
reg a,b,c;
wire difference,borrow;
FullSubtractorStructural uut(.a(a),.b(b),.c(c),.difference(difference),.borrow(borrow));
initial
begin

    a=0;b=0;c=0;#100;
    a=0;b=0;c=1;#100;
    a=0;b=1;c=0;#100;
    a=0;b=1;c=1;#100;
    a=1;b=0;c=0;#100;
    a=1;b=0;c=1;#100;
    a=1;b=1;c=0;#100;
    a=1;b=1;c=1;#100;

end
endmodule
```

RTL SCHEMATIC FOR FULL SUBTRACTOR USING STRUCTURAL:





OUTPUT WAVEFORM:

