Introduction to Combinational Circuit Simulation Lab: 3

Full Subtractor

Student : Chandani Lapasia

User id : fpga0522-chan50

Code snippet for Full Subtractor DATA flow modelling.

Design:

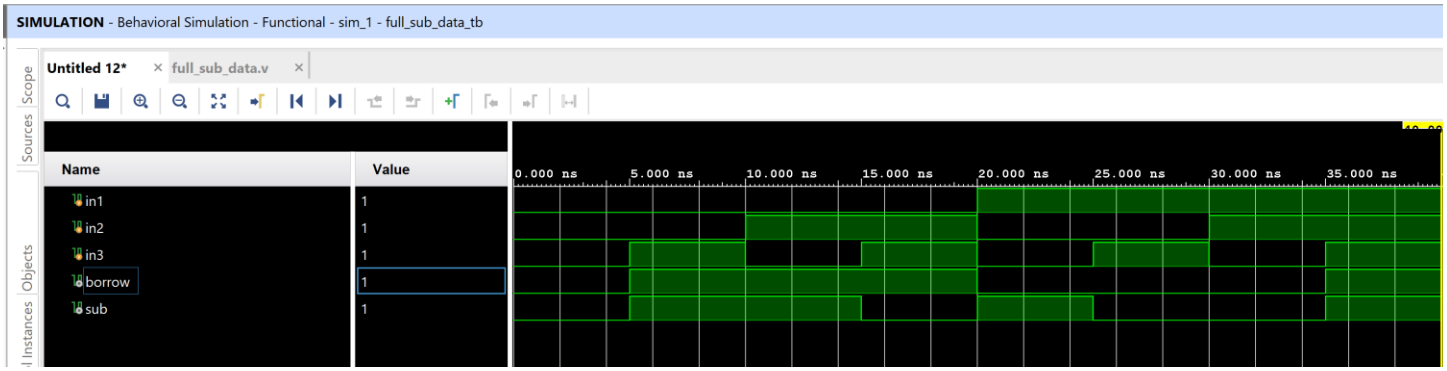
```
module full_sub_data(sub,borrow,in1,in2,in3);

input in1, in2,in3;

output sub,borrow;

assign sub= in1^in2^in3;

assign borrow= (~in1 && in2) || (~in1 && in3) || (in2 && in3);

endmodule
```

Testbench:

```
module full_sub_data_tb();

wire sub,borrow;

reg in1,in2,in3;

full_sub_data u0(sub,borrow,in1,in2,in3);

initial begin

in1=0; in2=0; in3=0;

#5 in1=0; in2=0; in3=1;

#5 in1=0; in2=1; in3=0;

#5 in1=0; in2=1; in3=1;

#5 in1=1; in2=0; in3=0;

#5 in1=1; in2=0; in3=1;

#5 in1=1; in2=1; in3=0;

#5 in1=1; in2=1; in3=1;
```

```
#5 $finish;

end

initial begin

$monitor ($time, "\t" , "in1=%d in2=%d in3=%d borrow=%d sub=%d", in1,in2,in3,borrow,sub);

end

endmodule
```

Output waveform:



Name	Value
in1	1
in2	1
in3	1
borrow	1
sub	1

Output console :

```
 0    in1=0 in2=0 in3=0 borrow=0 sub=0

 5    in1=0 in2=0 in3=1 borrow=1 sub=1

10    in1=0 in2=1 in3=0 borrow=1 sub=1

15    in1=0 in2=1 in3=1 borrow=1 sub=0

20    in1=1 in2=0 in3=0 borrow=0 sub=1

25    in1=1 in2=0 in3=1 borrow=0 sub=0

30    in1=1 in2=1 in3=0 borrow=0 sub=0

35    in1=1 in2=1 in3=1 borrow=1 sub=1
```

Code snippet for Full Subtractor BEHAVORAL flow modelling.

Design code :

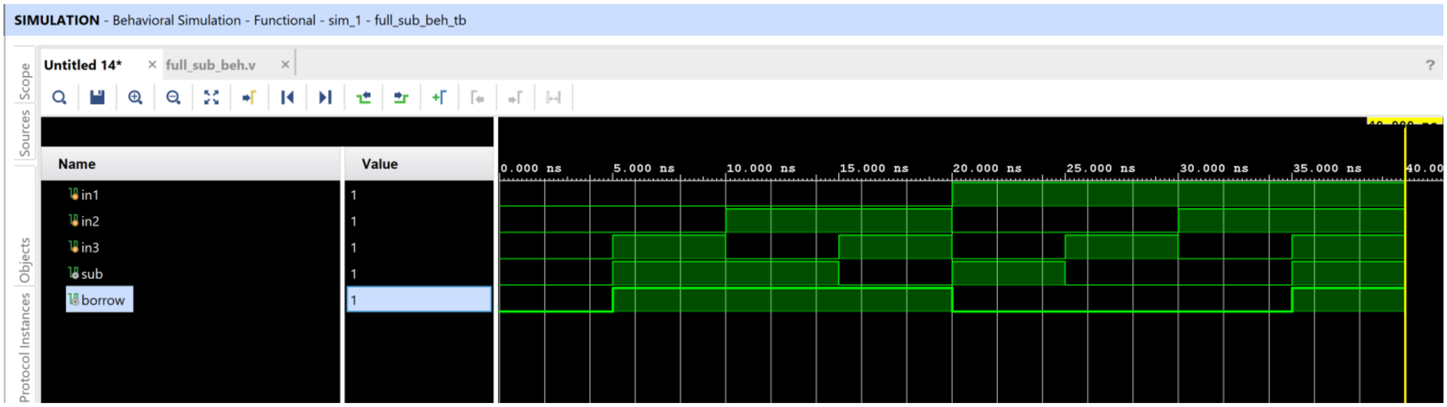
```
module full_sub_beh(sub,borrow,in1,in2,in3);
input in1, in2,in3;
output sub,borrow;
reg sub,borrow;
always @(in1 or in2 or in3)
 begin
  sub= in1^in2^in3;
 borrow = (~in1 && in2) || (~in1 && in3) || (in2 && in3);
  end
endmodule
```

Testbench code :

```
module full_sub_beh_tb();
wire sub,borrow;
reg in1,in2,in3;
full_sub_beh u0(sub,borrow,in1,in2,in3);
initial begin
in1=0; in2=0; in3=0;
#5 in1=0; in2=0; in3=1;
#5 in1=0; in2=1; in3=0;
#5 in1=0; in2=1; in3=1;
#5 in1=1; in2=0; in3=0;
#5 in1=1; in2=0; in3=1;
#5 in1=1; in2=1; in3=0;
#5 in1=1; in2=1; in3=1;
#5 $finish;
end
```

```
initial begin

$monitor ($time, "\t" , "in1=%d in2=%d in3=%d borrow=%d sub=%d", in1,in2,in3,borrow,sub);

end

endmodule
```

Output waveform :



Name	Value
in1	1
in2	1
in3	1
sub	1
borrow	1

Output console :

```
        0    in1=0 in2=0 in3=0 borrow=0 sub=0

        5    in1=0 in2=0 in3=1 borrow=1 sub=1

        10   in1=0 in2=1 in3=0 borrow=1 sub=1

        15   in1=0 in2=1 in3=1 borrow=1 sub=0

        20   in1=1 in2=0 in3=0 borrow=0 sub=1

        25   in1=1 in2=0 in3=1 borrow=0 sub=0

        30   in1=1 in2=1 in3=0 borrow=0 sub=0

        35   in1=1 in2=1 in3=1 borrow=1 sub=1
```

Code snippet for Full Subtractor STRUCTURAL flow modelling.

Design code :

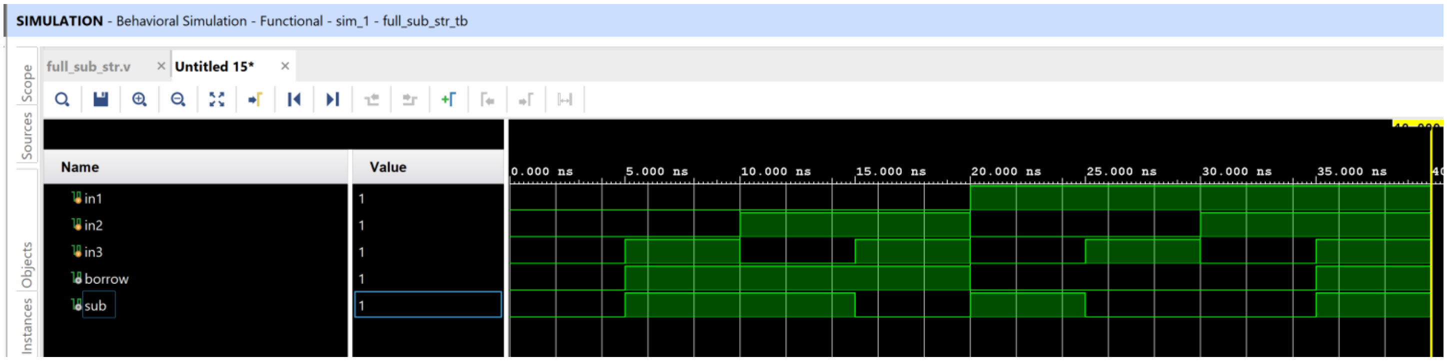
```
module full_sub_str(sub,borrow,in1,in2,in3);
input in1, in2,in3;
output sub,borrow;
wire sub,borrow;
wire temp1,temp2,temp3;
xor(sub,in1,in2,in3);
and(temp3,~in1,in2);
and(temp1,in2,in3);
and(temp2,~in1,in3);
or(borrow,temp3,temp1,temp2);
endmodule
```

Testbench code :

```
module full_sub_str_tb();
wire sub,borrow;
reg in1,in2,in3;
full_sub_str u0(sub,borrow,in1,in2,in3);
initial begin
in1=0; in2=0; in3=0;
#5 in1=0; in2=0; in3=1;
#5 in1=0; in2=1; in3=0;
#5 in1=0; in2=1; in3=1;
#5 in1=1; in2=0; in3=0;
#5 in1=1; in2=0; in3=1;
#5 in1=1; in2=1; in3=0;
#5 in1=1; in2=1; in3=1;
#5 $finish;
end
```

```
initial begin

$monitor ($time, "\t" , "in1=%d in2=%d in3=%d borrow=%d sub=%d", in1,in2,in3,borrow,sub);

end

endmodule
```

Output waveform :



Name	Value
in1	1
in2	1
in3	1
borrow	1
sub	1

Output console :

```
     0    in1=0 in2=0 in3=0 borrow=0 sub=0

     5    in1=0 in2=0 in3=1 borrow=1 sub=1

    10    in1=0 in2=1 in3=0 borrow=1 sub=1

    15    in1=0 in2=1 in3=1 borrow=1 sub=0

    20    in1=1 in2=0 in3=0 borrow=0 sub=1

    25    in1=1 in2=0 in3=1 borrow=0 sub=0

    30    in1=1 in2=1 in3=0 borrow=0 sub=0

    35    in1=1 in2=1 in3=1 borrow=1 sub=1
```

============================== **END**======================================