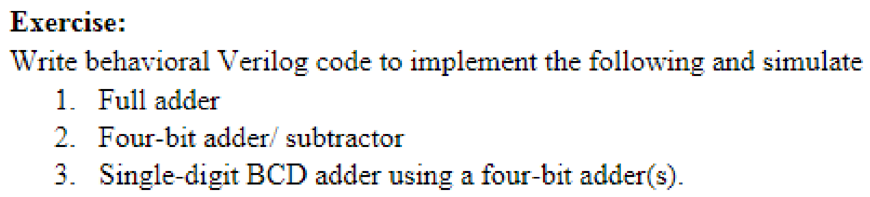
DSD LAB

Week 2

Udeet Mittal

CSE C3

Roll Number 64



Exercise:

Write behavioral Verilog code to implement the following and simulate

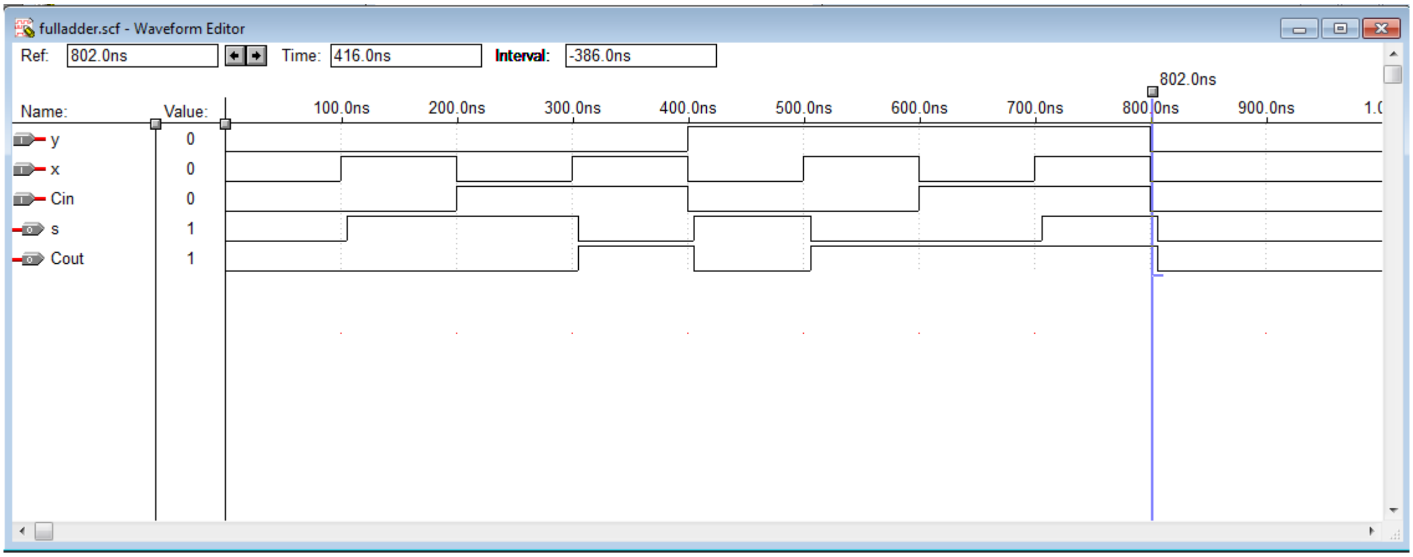
1. Full adder
2. Four-bit adder/ subtractor
3. Single-digit BCD adder using a four-bit adder(s).

1.

Source Code:

```
module fulladder(Cin,x,y,s,Cout);
input Cin,x,y;
output s,Cout;
assign s = x ^ y ^ Cin;
assign Cout = (x & y) | ( x & Cin) | ( y & Cin);
endmodule
```

Output Waveform:



fulladder.scf - Waveform Editor

Ref: 802.0ns Time: 416.0ns Interval: -386.0ns

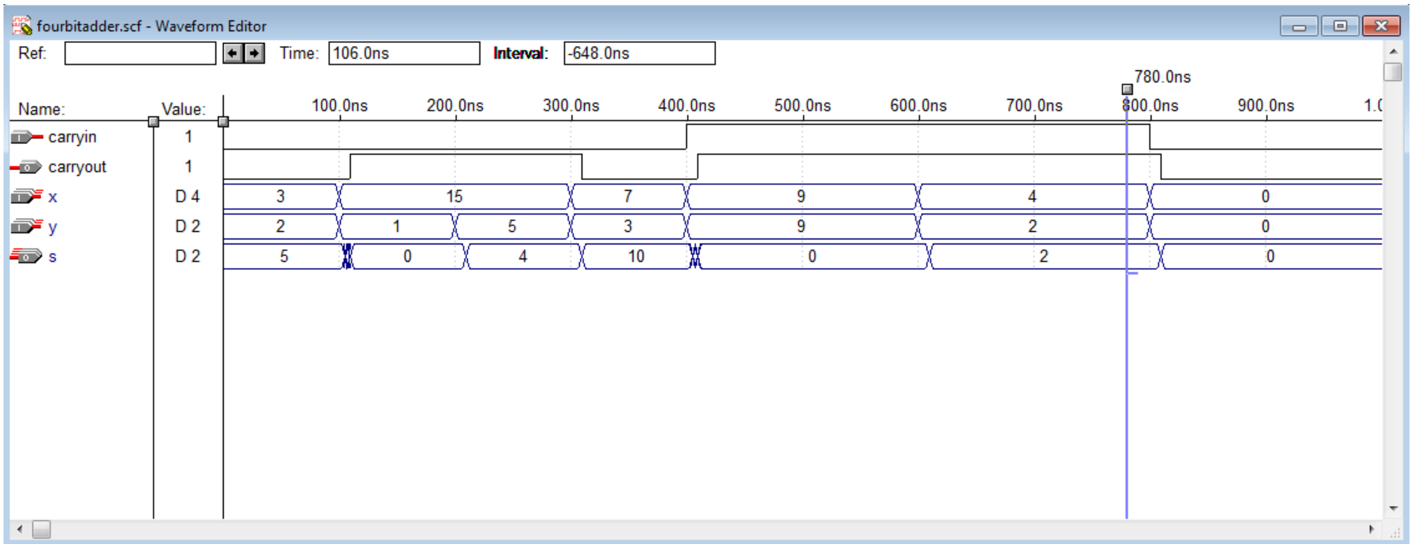
Name:	Value:
y	0
x	0
Cin	0
s	1
Cout	1

2.

Source Code:

```
module fourbitadder(carryin,x,y,s,carryout);
input carryin;
input[3:0] x,y;
output[3:0] s;
output carryout;
wire[3:1] C;
fulladder stage0(carryin , x[0] , y[0] ^ carryin , s[0] , C[1]);
fulladder stage1(C[1] , x[1] , y[1] ^ carryin , s[1] , C[2]);
fulladder stage2(C[2] , x[2] , y[2] ^ carryin , s[2] , C[3]);
fulladder stage3(C[3] , x[3] , y[3] ^ carryin , s[3] , carryout);
endmodule
//using the previous full adder module
```

Output Waveform:



Ref: Time: 106.0ns Interval: -648.0ns

Name:	Value:
carryin	1
carryout	1
x	D 4
y	D 2
s	D 2

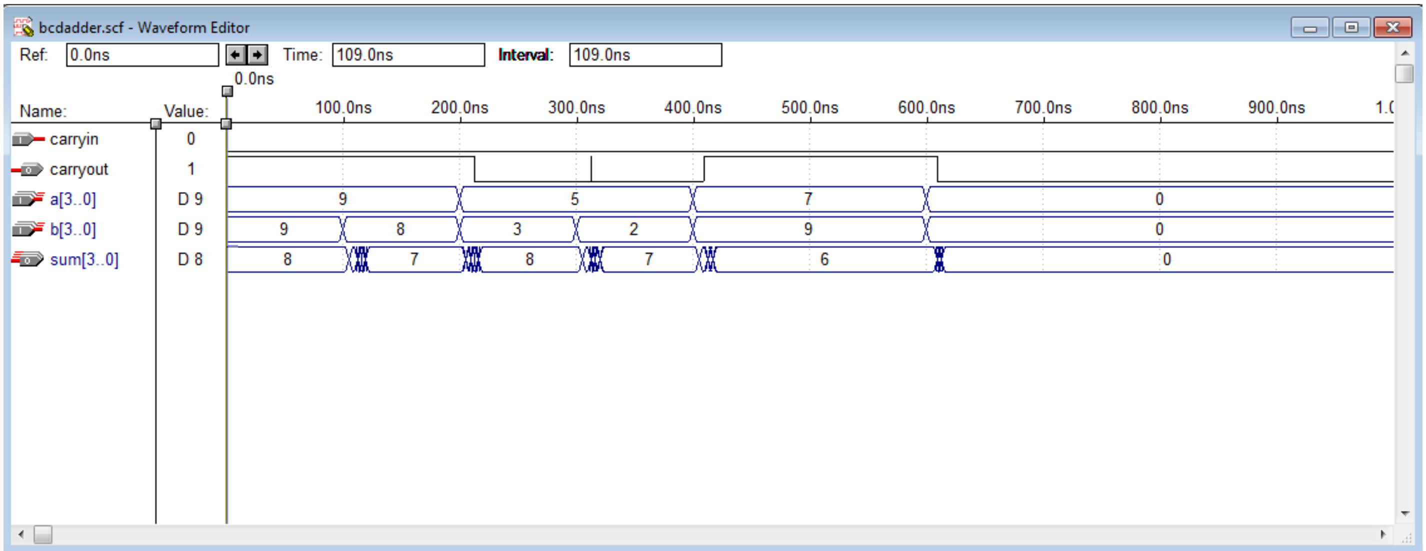
3.

Source Code:

```
module bcdadder(carryin,a,b,sum,carryout);
input [3:0] a,b;
input carryin;
output [3:0]sum;
output carryout;
wire [3:0]z;
wire c4,c41;
wire [3:0]ss;
fourbitadder stage1(carryin,a,b,z,c4);
```

```
assign carryout= (c4|(z[3] & z[2])|(z[3] & z[1]));
assign ss[0] = 1'b0;
assign ss[1] = carryout;
assign ss[2] = carryout;
assign ss[3] = 1'b0;
fourbitadder stage2(0,ss,z,sum,c41);
endmodule
```

Output Waveform:



bcdadder.scf - Waveform Editor

Ref: 0.0ns Time: 109.0ns Interval: 109.0ns

Name: Value:
carryin 0
carryout 1
a[3..0] D 9
b[3..0] D 9
sum[3..0] D 8