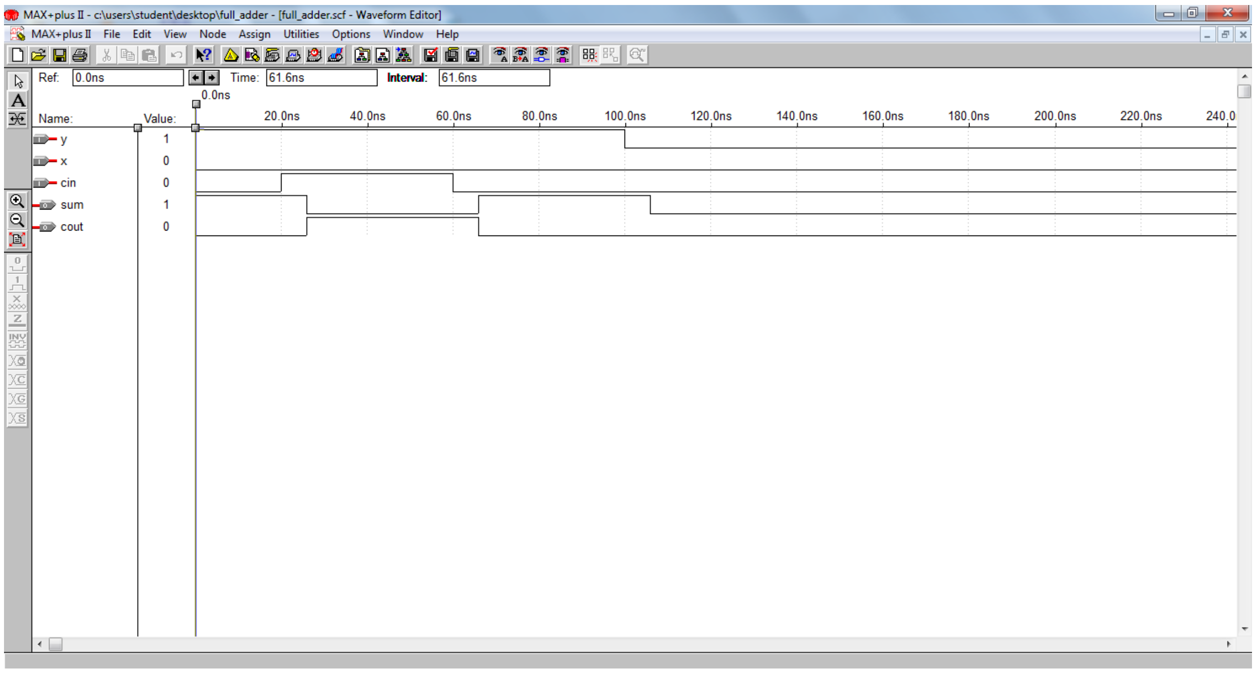
Exercise:

Write behavioural 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).

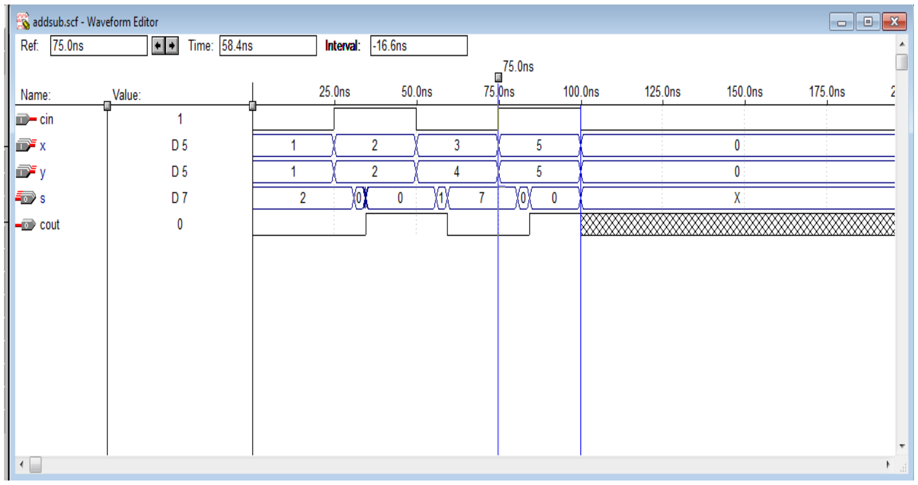
Solution 1:

```
module full_adder(cin,x,y,sum,cout);

input cin,x,y;

output sum,cout;


assign sum=cin^x^y;

assign cout=(x&y)|(y&cin)|(cin&x);


endmodule
```


MAX+plus II - c:\users\student\desktop\full_adder - [full_adder.scf - Waveform Editor]
MAX+plus II File Edit View Node Assign Utilities Options Window Help
Ref: 0.0ns
Time: 61.6ns
Interval: 61.6ns
0.0ns
Name:
Value:
20.0ns
40.0ns
60.0ns
80.0ns
100.0ns
120.0ns
140.0ns
160.0ns
180.0ns
200.0ns
220.0ns
240.0
y 1
x 0
cin 0
sum 1
cout 0

Solution 2:

```
module addsub(cin,x,y,s,cout);
  input cin;
  input[3:0]x,y;
  output[3:0]s;
  output cout;
  wire[3:1]c;
  fulladder stage0 (cin,x[0],y[0]^cin,s[0],c[1]);
  fulladder stage1 (c[1],x[1],y[1]^cin,s[1],c[2]);
  fulladder stage2 (c[2],x[2],y[2]^cin,s[2],c[3]);
  fulladder stage3 (c[3],x[3],y[3]^cin,s[3],cout);
endmodule

module  fulladder(cin,x,y,sum,cout);
  input cin,x,y;
  output sum,cout;
  assign sum = cin^x^y;
  assign cout =(x&y) | (x&cin) |(y&cin);
endmodule
```



addsub.scf - Waveform Editor

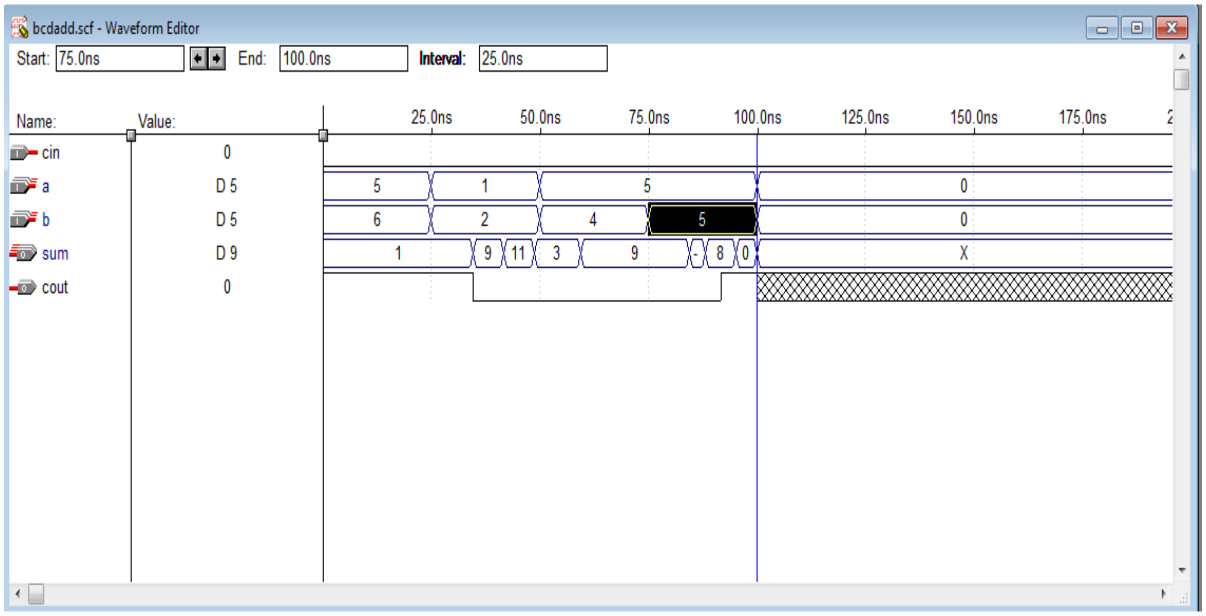
Ref: 75.0ns   Time: 58.4ns   Interval: -16.6ns

Name:	Value:
cin	1
x	D 5
y	D 5
s	D 7
cout	0

Solution 3:

```
module bcdadd(cin,a,b,sum,cout);
  input [3:0]a,b;
  input cin;
  output [3:0]sum;
  output cout;
 wire[3:0]z;
 wire m;
 wire[3:1]k;
 wire[3:0]h;
 adderm1 stage0 (cin,a,b,z,m);
 assign k[1]=z[3]&z[2];
 assign k[2]=z[3]&z[1];
 assign k[3]=(m|k[1]|k[2]);
 assign h[0]=0,h[1]=k[3],h[2]=k[3],h[3]=0;
 adderm1 stage1 (cin,z,h,sum,cout);
endmodule


module adderm1(cin,x,y,s,cout);
  input cin;
  input [3:0]x,y;
  output cout;
  output [3:0]s;
  wire[3:1]c;
  fulladder stage0 (cin,x[0],y[0],s[0],c[1]);
  fulladder stage1 (c[1],x[1],y[1],s[1],c[2]);
  fulladder stage2 (c[2],x[2],y[2],s[2],c[3]);
  fulladder stage3 (c[3],x[3],y[3],s[3],cout);
```

```
endmodule

module  fulladder(cin,x,y,sum,cout);
  input cin,x,y;
  output sum,cout;
  assign sum = cin^x^y;
  assign cout =(x&y) | (x&cin) |(y&cin);
endmodule
```


bcdadd.scf - Waveform Editor
Start: 75.0ns
End: 100.0ns
Interval: 25.0ns
Name:
Value:
25.0ns
50.0ns
75.0ns
100.0ns
125.0ns
150.0ns
175.0ns
cin 0
a D 5
b D 5
sum D 9
cout 0
5 1 5 0
6 2 4 5 0
1 9 11 3 9 - 8 0 X