Experiment-4

Design Verilog HDL to implement Binary Adder-Subtractor- Half and full adder, Half and full subtractor.

Aim: To Design Verilog HDL to implement Half adder, full adder, Half subtractor and full subtractor.

a. Half Adder

Symbol

Truth table

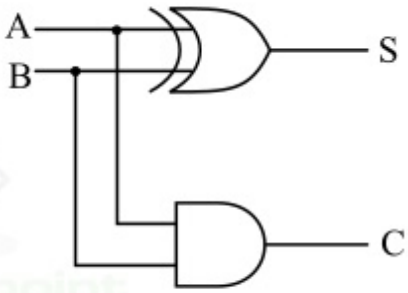
INPUT		OUTPUT	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Expression

$S=A'B+AB'=A \oplus B$

$C=AB$

Logic diagram



Circuit Diagram

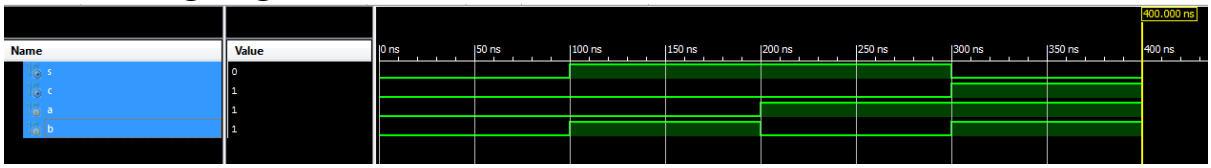
Verilog code

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

Testbench code

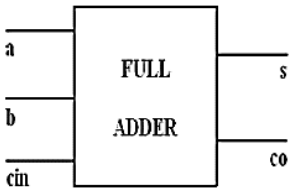
```
module half_adder_tb;
reg a,b;
wire s,c;
half_adder uut (.a(a),.b(b),.s(s),.c(c));
initial begin
a = 0; b = 0;
#100
a = 0; b = 1;
#100
a = 1; b = 0;
#100
a = 1; b = 1;
#100
$finish();
end
endmodule
```

Timing Diagram



b. Full Adder

Symbol



Truth table

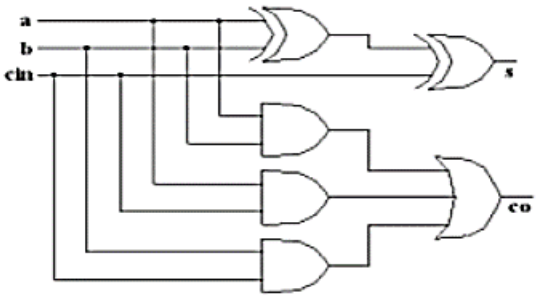
INPUT			OUTPUT	
A	B	Cin	S	CO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Expression

$$S = A \oplus B \oplus Cin$$

$$Cout = (A \& B)|(B \& Cin)|(Cin \& A)$$

Logic Diagram



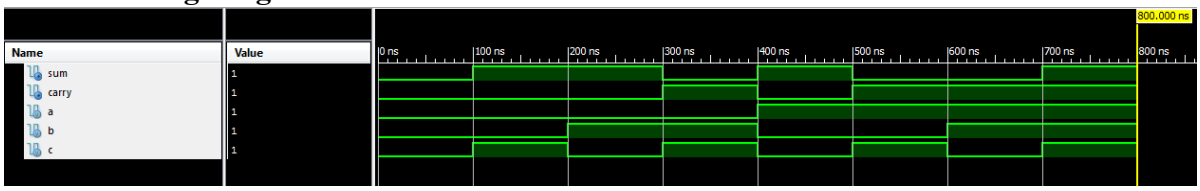
Verilog code

```
module full_adder (a,b,cin,s,cout);
input cin, a, b;
output s, cout;
assign s = a ^ b ^ cin;
assign cout = (a & b) | (b & cin) | (cin & a);
endmodule
```

Testbench code

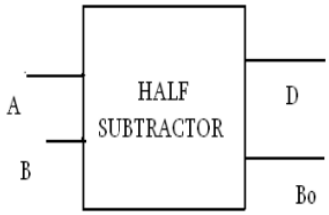
```
module full_adder_tb;
reg a,b,cin;
wire s,cout;
full_adder uut(.a(.a),.b(b),.cin(cin),.s(s),.cout(cout));
initial begin
a = 0; b = 0; cin = 0;
#100
a = 0; b = 0; cin = 1;
#100
a = 0; b = 1; cin = 0;
#100
a = 0; b = 1; cin = 1;
#100
a = 1; b = 0; cin = 0;
#100
a = 1; b = 0; cin = 1;
#100
a = 1; b = 1; cin = 0;
#100
a = 1; b = 1; cin = 1;
#100
$finish();
end
```

Timing Diagram



c. Half Subtractor

Symbol



Truth table

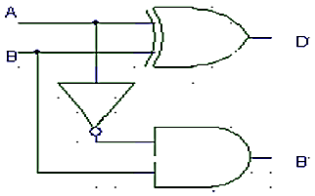
INPUT		OUTPUT	
A	**B**	**Difference**	**Borrow**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Expression

$Diff = A'B + AB' = A \oplus B$

$Borrow = A'B$

Logic Diagram



Verilog Code

```
module half_sub( a, b, diff, borrow );
input a, b;
output diff, borrow;
assign diff=a^b;
assign borrow=(~a)&b;
endmodule
```

Testbench code

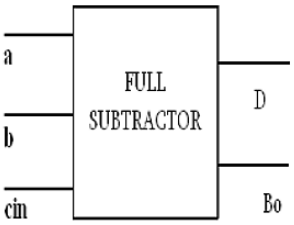
```
module half_sub_tb;
reg a,b;
wire diff,borrow;
```

```
half_sub uut (.a(a),.b(b),.diff(diff),.borrow(borrow));
initial begin
a = 0; b = 0;
#100
a = 0; b = 1;
#100
a = 1; b = 0;
#100
a = 1; b = 1;
#100
$finish();
end
endmodule
```

Timing Diagram

d. Full Subtractor

Symbol



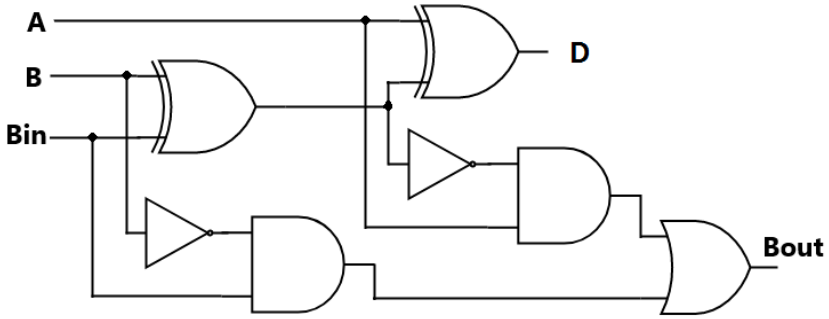
Truth table

A	B	B_{in}	D	B_{out}
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

Expression

$$D = A \oplus B \oplus Bin$$
$$Bout = A' Bin + A' B + B Bin$$

Logic Diagram



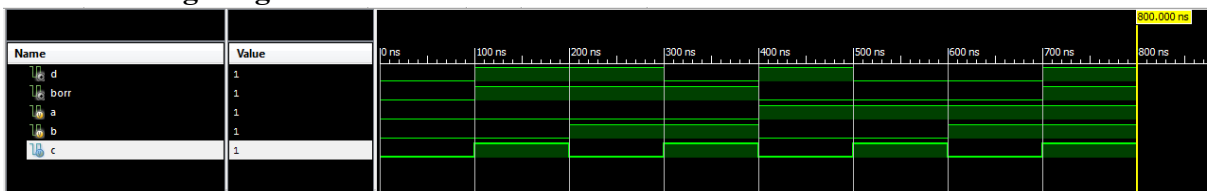
Verilog code

```
module full_sub(a, b, bin, d, bout);
input a, b, bin;
output d, bout;
assign d = a^b^bin;
assign bout= ((~a) & bin) | ((~a) & b)) | (b & bin);
endmodule
```

Testbench code

```
module full_sub_tb;
reg a,b,bin;
wire d,bout;
full_sub uut(.a(.a),.b(b),.bin(bin),.d(d),.bout(bout));
initial begin
a = 0; b = 0; bin = 0;
#100
a = 0; b = 0; bin = 1;
#100
a = 0; b = 1; bin = 0;
#100
a = 0; b = 1; bin = 1;
#100
a = 1; b = 0; bin = 0;
#100
a = 1; b = 0; bin = 1;
#100
a = 1; b = 1; bin = 0;
#100
a = 1; b = 1; bin = 1;
#100
$finish();
end
```

Timing Diagram



Result

Verilog HDL code for Half adder, full adder, half subtractor and full subtractor is designed and simulated.