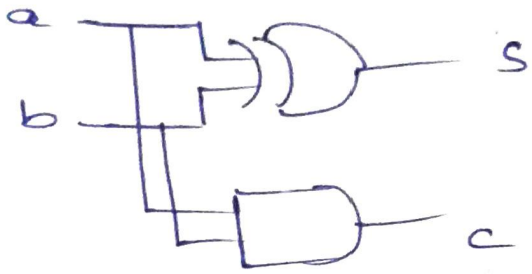
RIPPLE CARRY ADDER

→
```
module HA(a, b, S, C);
input a, b;
output S, C;
assign S = a ^ b;
assign C = a & b;
endmodule
```

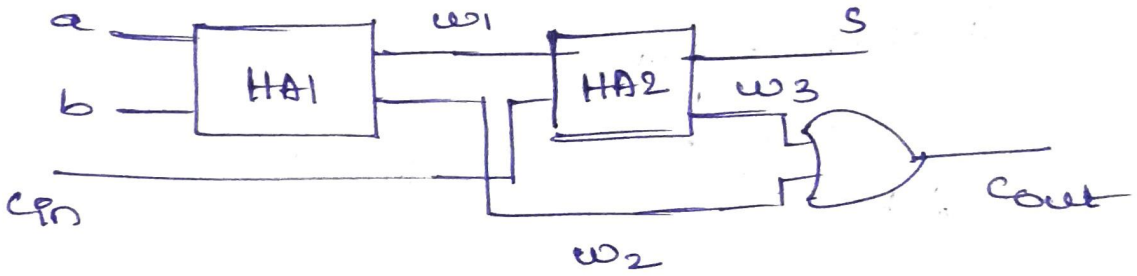
⟹
```
module FA (a, b, cin, S, Cout);
input a, b, cin;
output S, C;
wire w1, w2, w3;
HA1 HA(a, b, w1, w2);
HA2 HA(w1, cin, S, w3);
Or(Cout, w2, w3);
endmodule
```

⟹
```
module RCA (a, b, S, Cout);
input [3:0] a;
input [3:0] b; input cin;
output [3:0] S;
output Cout;
wire c1, c2, c3;
FA0 FA(a[0], b[0], cin, S[0], c1);
FA1 FA(a[1], b[1], c1, S[1], c2);
FA2 FA(a[2], b[2], c2, S[2], c3);
FA3 FA(a[3], b[3], c3, S[3], Cout);
endmodule
```

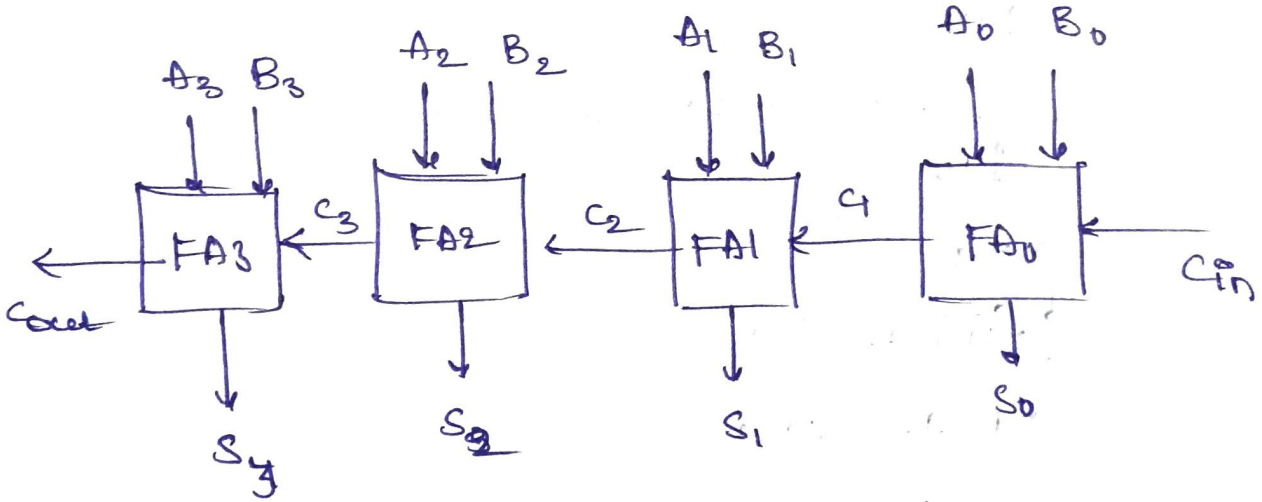
HALF ADDER :-



FULL ADDER using HALF ADDER :-



RIPPLE CARRY ADDER :-



Testbench:-

```
module RCA_tb ( );
 reg [3:0] a, b;
 reg cin;
 wire [3:0] S;
 wire cout;
 RCA uut (a, b, cin, S, cout);
 Initial begin
  a = 0; b = 0; cin = 1;
  #10
  a = 1; b = 1; cin = 0;
  #10
  a = 0; b = 1; cin = 1;
  #10
  a = 1; b = 1; cin = 1;
  #10
 end
 $finish( );
 endmodule
```