```
module Lab3(input [0 : 3] X0, X1, input s, output [0 : 6]  X0_output, X1_output, output [0 : 3]  final,
output [0 : 6]  final_output, output count, output [0 : 6]  count_output, h2,h4);

assign h2 = 7'b1111111;
assign h4 = 7'b1111111;

FourBitAdder f1(X0, X1, 0, final, count, s);

// Displaying X0_output

assign X0_output[6] = ~((~X0[0] & ~X0[1] & X0[2] & ~X0[3]) | (~X0[0] & ~X0[1] & X0[2] & X0[3]) |
(~X0[0] & X0[1] & ~X0[2] & X0[3]) | (~X0[0] & X0[1] & X0[2] & ~X0[3]) | (~X0[0] & X0[1] & X0[2]
& X0[3]) | (X0[0] & ~X0[1] & ~X0[2] & ~X0[3]) | (X0[0] & ~X0[1] & ~X0[2] & X0[3]) | (X0[0] &
~X0[1] & X0[2] & ~X0[3]) | (X0[0] & X0[1] & ~X0[2] & ~X0[3]) | (X0[0] & X0[1] & X0[2] & ~X0[3])
| (X0[0] & X0[1] & X0[2] & X0[3]) | (~X0[0] & ~X0[1] & ~X0[2] & ~X0[3]));

assign X0_output[5] = ~((~X0[0] & ~X0[1] & ~X0[2] & X0[3]) | (~X0[0] & ~X0[1] & X0[2] & ~X0[3])
| (~X0[0] & ~X0[1] & X0[2] & X0[3]) | (~X0[0] & X0[1] & ~X0[2] & ~X0[3]) | (~X0[0] & X0[1] &
X0[2] & X0[3]) | (X0[0] & ~X0[1] & ~X0[2] & ~X0[3]) | (X0[0] & ~X0[1] & ~X0[2] & X0[3]) | (X0[0]
& ~X0[1] & X0[2] & ~X0[3]) | (X0[0] & X0[1] & ~X0[2] & X0[3]) | (~X0[0] & ~X0[1] & ~X0[2] &
~X0[3]));

assign X0_output[4] = ~((~X0[0] & ~X0[1] & ~X0[2] & X0[3]) | (~X0[0] & ~X0[1] & X0[2] & X0[3]) |
(~X0[0] & X0[1] & ~X0[2] & ~X0[3]) | (~X0[0] & X0[1] & ~X0[2] & X0[3]) | (~X0[0] & X0[1] &
X0[2] & ~X0[3]) | (~X0[0] & X0[1] & X0[2] & X0[3]) | (X0[0] & ~X0[1] & ~X0[2] & ~X0[3]) | (X0[0]
& ~X0[1] & ~X0[2] & X0[3]) | (X0[0] & ~X0[1] & X0[2] & ~X0[3]) | (X0[0] & ~X0[1] & X0[2] &
X0[3]) | (X0[0] & X0[1] & ~X0[2] & X0[3]) | (~X0[0] & ~X0[1] & ~X0[2] & ~X0[3]));

assign X0_output[3] = ~((~X0[0] & ~X0[1] & X0[2] & ~X0[3]) | (~X0[0] & ~X0[1] & X0[2] & X0[3]) |
(~X0[0] & X0[1] & ~X0[2] & X0[3]) | (~X0[0] & X0[1] & X0[2] & ~X0[3]) | (X0[0] & ~X0[1] &
~X0[2] & ~X0[3]) | (X0[0] & ~X0[1] & ~X0[2] & X0[3]) | (X0[0] & ~X0[1] & X0[2] & X0[3]) | (X0[0]
& X0[1] & ~X0[2] & ~X0[3]) | (X0[0] & X0[1] & ~X0[2] & X0[3]) | (X0[0] & X0[1] & X0[2] &
~X0[3]) | (~X0[0] & ~X0[1] & ~X0[2] & ~X0[3]));

assign X0_output[2] = ~((~X0[0] & ~X0[1] & X0[2] & ~X0[3]) | (~X0[0] & X0[1] & X0[2] & ~X0[3]) |
(X0[0] & ~X0[1] & ~X0[2] & ~X0[3]) | (X0[0] & ~X0[1] & X0[2] & ~X0[3]) | (X0[0] & ~X0[1] &
X0[2] & X0[3]) | (X0[0] & X0[1] & ~X0[2] & ~X0[3]) | (X0[0] & X0[1] & ~X0[2] & X0[3]) | (X0[0] &
X0[1] & X0[2] & ~X0[3]) | (X0[0] & X0[1] & X0[2] & X0[3]) | (~X0[0] & ~X0[1] & ~X0[2] &
~X0[3]));

assign X0_output[1] = ~((~X0[0] & X0[1] & ~X0[2] & ~X0[3]) | (~X0[0] & X0[1] & ~X0[2] & X0[3]) |
(~X0[0] & X0[1] & X0[2] & ~X0[3]) | (X0[0] & ~X0[1] & ~X0[2] & ~X0[3]) | (X0[0] & ~X0[1] &
~X0[2] & X0[3]) | (X0[0] & ~X0[1] & X0[2] & ~X0[3]) | (X0[0] & ~X0[1] & X0[2] & X0[3]) | (X0[0] &
X0[1] & ~X0[2] & ~X0[3]) | (X0[0] & X0[1] & X0[2] & ~X0[3]) | (X0[0] & X0[1] & X0[2] & X0[3]) |
(~X0[0] & ~X0[1] & ~X0[2] & ~X0[3]));

assign X0_output[0] = (~X0[0] & ~X0[1] & ~X0[2]) | (~X0[0] & X0[1] & X0[2] & X0[3]) | (X0[0] &
X0[1] & ~X0[2] & ~X0[3]);
```

```
// Displaying X1_output

assign X1_output[6] = ~((~X1[0] & ~X1[1] & X1[2] & ~X1[3]) | (~X1[0] & ~X1[1] & X1[2] & X1[3]) |
(~X1[0] & X1[1] & ~X1[2] & X1[3]) | (~X1[0] & X1[1] & X1[2] & ~X1[3]) | (~X1[0] & X1[1] & X1[2]
& X1[3]) | (X1[0] & ~X1[1] & ~X1[2] & ~X1[3]) | (X1[0] & ~X1[1] & ~X1[2] & X1[3]) | (X1[0] &
~X1[1] & X1[2] & ~X1[3]) | (X1[0] & X1[1] & ~X1[2] & ~X1[3]) | (X1[0] & X1[1] & X1[2] & ~X1[3])
| (X1[0] & X1[1] & X1[2] & X1[3]) | (~X1[0] & ~X1[1] & ~X1[2] & ~X1[3]));

assign X1_output[5] = ~((~X1[0] & ~X1[1] & ~X1[2] & X1[3]) | (~X1[0] & ~X1[1] & X1[2] & ~X1[3])
| (~X1[0] & ~X1[1] & X1[2] & X1[3]) | (~X1[0] & X1[1] & ~X1[2] & ~X1[3]) | (~X1[0] & X1[1] &
X1[2] & X1[3]) | (X1[0] & ~X1[1] & ~X1[2] & ~X1[3]) | (X1[0] & ~X1[1] & ~X1[2] & X1[3]) | (X1[0]
& ~X1[1] & X1[2] & ~X1[3]) | (X1[0] & X1[1] & ~X1[2] & X1[3]) | (~X1[0] & ~X1[1] & ~X1[2] &
~X1[3]));

assign X1_output[4] = ~((~X1[0] & ~X1[1] & ~X1[2] & X1[3]) | (~X1[0] & ~X1[1] & X1[2] & X1[3]) |
(~X1[0] & X1[1] & ~X1[2] & ~X1[3]) | (~X1[0] & X1[1] & ~X1[2] & X1[3]) | (~X1[0] & X1[1] &
X1[2] & ~X1[3]) | (~X1[0] & X1[1] & X1[2] & X1[3]) | (X1[0] & ~X1[1] & ~X1[2] & ~X1[3]) | (X1[0]
& ~X1[1] & ~X1[2] & X1[3]) | (X1[0] & ~X1[1] & X1[2] & ~X1[3]) | (X1[0] & ~X1[1] & X1[2] &
X1[3]) | (X1[0] & X1[1] & ~X1[2] & X1[3]) | (~X1[0] & ~X1[1] & ~X1[2] & ~X1[3]));

assign X1_output[3] = ~((~X1[0] & ~X1[1] & X1[2] & ~X1[3]) | (~X1[0] & ~X1[1] & X1[2] & X1[3]) |
(~X1[0] & X1[1] & ~X1[2] & X1[3]) | (~X1[0] & X1[1] & X1[2] & ~X1[3]) | (X1[0] & ~X1[1] &
~X1[2] & ~X1[3]) | (X1[0] & ~X1[1] & ~X1[2] & X1[3]) | (X1[0] & ~X1[1] & X1[2] & X1[3]) | (X1[0]
& X1[1] & ~X1[2] & ~X1[3]) | (X1[0] & X1[1] & ~X1[2] & X1[3]) | (X1[0] & X1[1] & X1[2] &
~X1[3]) | (~X1[0] & ~X1[1] & ~X1[2] & ~X1[3]));

assign X1_output[2] = ~((~X1[0] & ~X1[1] & X1[2] & ~X1[3]) | (~X1[0] & X1[1] & X1[2] & ~X1[3]) |
(X1[0] & ~X1[1] & ~X1[2] & ~X1[3]) | (X1[0] & ~X1[1] & X1[2] & ~X1[3]) | (X1[0] & ~X1[1] &
X1[2] & X1[3]) | (X1[0] & X1[1] & ~X1[2] & ~X1[3]) | (X1[0] & X1[1] & ~X1[2] & X1[3]) | (X1[0] &
X1[1] & X1[2] & ~X1[3]) | (X1[0] & X1[1] & X1[2] & X1[3]) | (~X1[0] & ~X1[1] & ~X1[2] &
~X1[3]));

assign X1_output[1] = ~((~X1[0] & X1[1] & ~X1[2] & ~X1[3]) | (~X1[0] & X1[1] & ~X1[2] & X1[3]) |
(~X1[0] & X1[1] & X1[2] & ~X1[3]) | (X1[0] & ~X1[1] & ~X1[2] & ~X1[3]) | (X1[0] & ~X1[1] &
~X1[2] & X1[3]) | (X1[0] & ~X1[1] & X1[2] & ~X1[3]) | (X1[0] & ~X1[1] & X1[2] & X1[3]) | (X1[0] &
X1[1] & ~X1[2] & ~X1[3]) | (X1[0] & X1[1] & X1[2] & ~X1[3]) | (X1[0] & X1[1] & X1[2] & X1[3]) |
(~X1[0] & ~X1[1] & ~X1[2] & ~X1[3]));

assign X1_output[0] = (~X1[0] & ~X1[1] & ~X1[2]) | (~X1[0] & X1[1] & X1[2] & X1[3]) | (X1[0] &
X1[1] & ~X1[2] & ~X1[3]);

// Displaying count_output

assign count_output[6] = 1;
assign count_output[5] = ~(count & ~s);
assign count_output[4] = ~(count & ~s);
assign count_output[3] = 1;
assign count_output[2] = 1;
assign count_output[1] = 1;
assign count_output[0] = ~(~count & s);
```

```
// Displaying final_output

assign final_output[6] = (((~final[0] & ~final[1] & ~final[2] & final[3]) | (~final[0] & final[1] & ~final[2]
& ~final[3]) | (final[0] & ~final[1] & final[2] & final[3]) | (final[0] & final[1] & ~final[2] & final[3])) &
((count & s) | ~s)) ^ (((~final[0] & ~final[1] & final[2] & final[3]) | (~final[0] & final[1] & ~final[2] &
final[3]) | (final[0] & final[1] & ~final[2] & ~final[3]) | (final[0] & final[1] & final[2] & final[3])) &
(~count & s));

assign final_output[5] = (((~final[0] & final[1] & ~final[2] & final[3]) | (~final[0] & final[1] & final[2] &
~final[3]) | (final[0] & ~final[1] & final[2] & final[3]) | (final[0] & final[1] & ~final[2] & ~final[3]) |
(final[0] & final[1] & final[2] & ~final[3]) | (final[0] & final[1] & final[2] & final[3])) & ((count & s) |
~s)) ^ (((~final[0] & ~final[1] & ~final[2] & final[3]) | (~final[0] & ~final[1] & final[2] & ~final[3]) |
(~final[0] & final[1] & ~final[2] & ~final[3]) | (~final[0] & final[1] & ~final[2] & final[3]) | (final[0] &
~final[1] & final[2] & ~final[3]) | (final[0] & ~final[1] & final[2] & final[3])) & (~count & s));

assign final_output[4] = (((~final[0] & ~final[1] & final[2] & ~final[3]) | (final[0] & final[1] & ~final[2]
& ~final[3]) | (final[0] & final[1] & final[2] & ~final[3]) | (final[0] & final[1] & final[2] & final[3])) &
((count & s) | ~s)) ^ (((~final[0] & ~final[1] & ~final[2] & final[3]) | (~final[0] & ~final[1] & final[2] &
~final[3]) | (~final[0] & final[1] & ~final[2] & ~final[3]) | (final[0] & final[1] & final[2] & ~final[3])) &
(~count & s));

assign final_output[3] = (((~final[0] & ~final[1] & ~final[2] & final[3]) | (~final[0] & final[1] & ~final[2]
& ~final[3]) | (~final[0] & final[1] & final[2] & final[3]) | (final[0] & ~final[1] & final[2] & ~final[3]) |
(final[0] & final[1] & final[2] & final[3])) & ((count & s) | ~s)) ^ (((~final[0] & ~final[1] & ~final[2] &
final[3]) | (~final[0] & final[1] & final[2] & ~final[3]) | (final[0] & ~final[1] & ~final[2] & final[3]) |
(final[0] & final[1] & ~final[2] & ~final[3]) | (final[0] & final[1] & final[2] & final[3])) & (~count & s));

assign final_output[2] = (((~final[0] & ~final[1] & ~final[2] & final[3]) | (~final[0] & ~final[1] & final[2]
& final[3]) | (~final[0] & final[1] & ~final[2] & ~final[3]) | (~final[0] & final[1] & ~final[2] & final[3]) |
(~final[0] & final[1] & final[2] & final[3]) | (final[0] & ~final[1] & ~final[2] & final[3])) & ((count & s) |
~s)) ^ (((~final[0] & final[1] & final[2] & final[3]) | (final[0] & ~final[1] & ~final[2] & final[3]) |
(final[0] & ~final[1] & final[2] & final[3]) | (final[0] & final[1] & ~final[2] & ~final[3]) | (final[0] &
final[1] & ~final[2] & final[3]) | (final[0] & final[1] & final[2] & final[3])) & (~count & s));

assign final_output[1] = (((~final[0] & ~final[1] & ~final[2] & final[3]) | (~final[0] & ~final[1] & final[2]
& ~final[3]) | (~final[0] & ~final[1] & final[2] & final[3]) | (~final[0] & final[1] & final[2] & final[3]) |
(final[0] & final[1] & ~final[2] & final[3])) & ((count & s) | ~s)) ^ (((~final[0] & ~final[1] & final[2] &
final[3]) | (final[0] & ~final[1] & ~final[2] & final[3]) | (final[0] & final[1] & ~final[2] & final[3]) |
(final[0] & final[1] & final[2] & ~final[3]) | (final[0] & final[1] & final[2] & final[3])) & (~count & s));

assign final_output[0] = (((~final[0] & ~final[1] & ~final[2]) | (~final[0] & final[1] & final[2] & final[3])
| (final[0] & final[1] & ~final[2] & ~final[3])) & ((count & s) | ~s)) ^ (((~final[0] & final[1] & ~final[2] &
~final[3]) | (final[0] & ~final[1] & ~final[2] & final[3]) | (final[0] & final[1] & final[2] & final[3])) &
(~count & s));

endmodule
```

```
module FourBitAdder(input [0 : 3] X0, X1, input cin, output [0 : 3] r, output count, input s);
        wire w1, w2, w3, w4, w5, w6, w7;
        xor(w7, X1[3], s);
        xor(w6, X1[2], s);
        xor(w5, X1[1], s);
        xor(w4, X1[0], s);
        FA f1(X0[3], w7, s, r[3], w1);
        FA f2(X0[2], w6, w1, r[2], w2);
        FA f3(X0[1], w5, w2, r[1], w3);
        FA f4(X0[0], w4, w3, r[0], count);
endmodule

module FA(input X0, X1, cin, output r, count);
        assign r = X0 ^ X1 ^ cin;
        assign count = (X0 ^ X1) & cin | X0 & X1;
endmodule
```