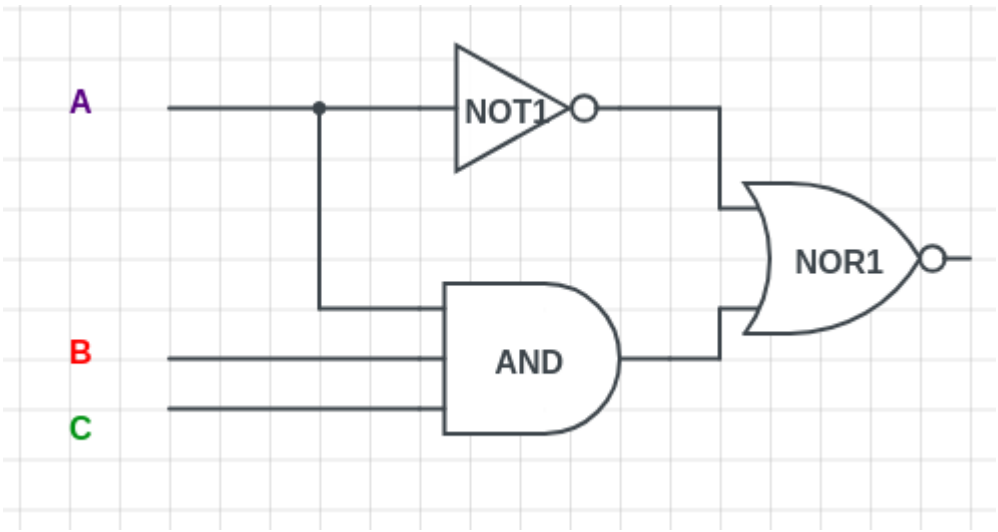
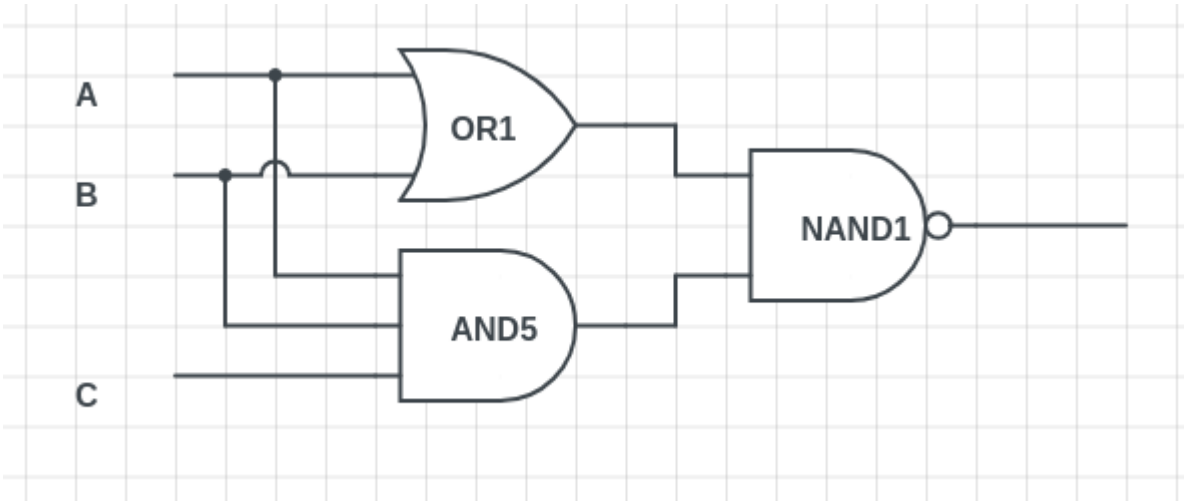
1.



truth table:

A	B	C	OUTPUT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

B.



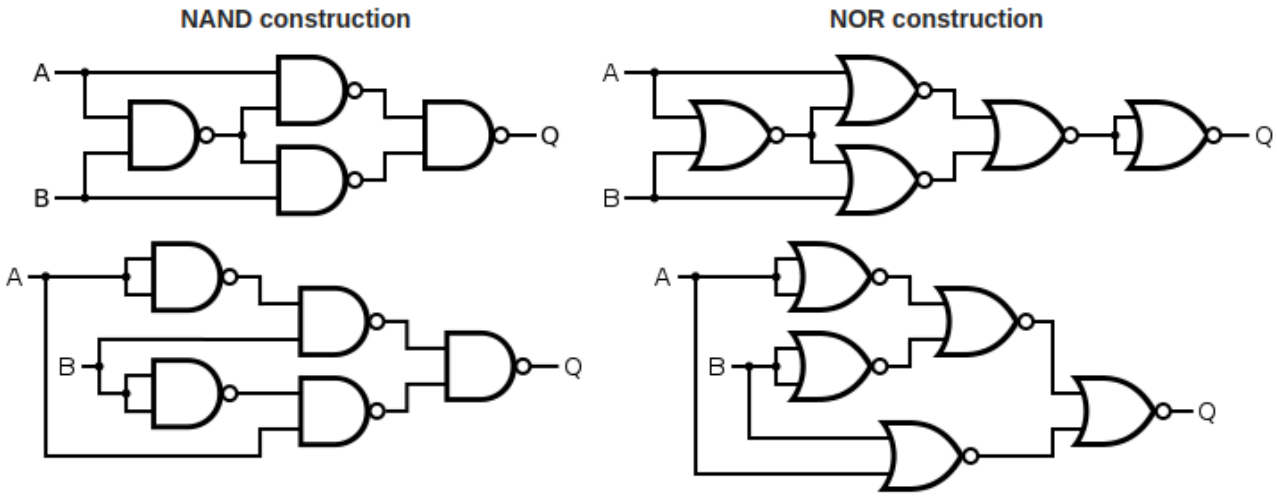
truth table

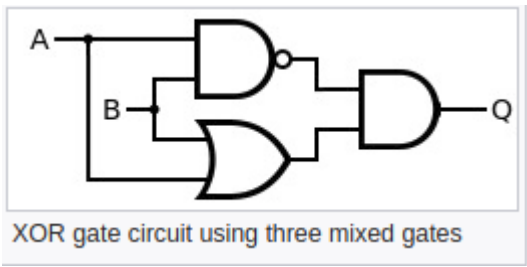
A	B	C	OUTPUT
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

2.

In my opinion, the XOR gate should be implemented in $A\overline{B} + \overline{A}B$ of course, which use $2 * 2 + 6 * 2 + 6 = 22$ transistor.

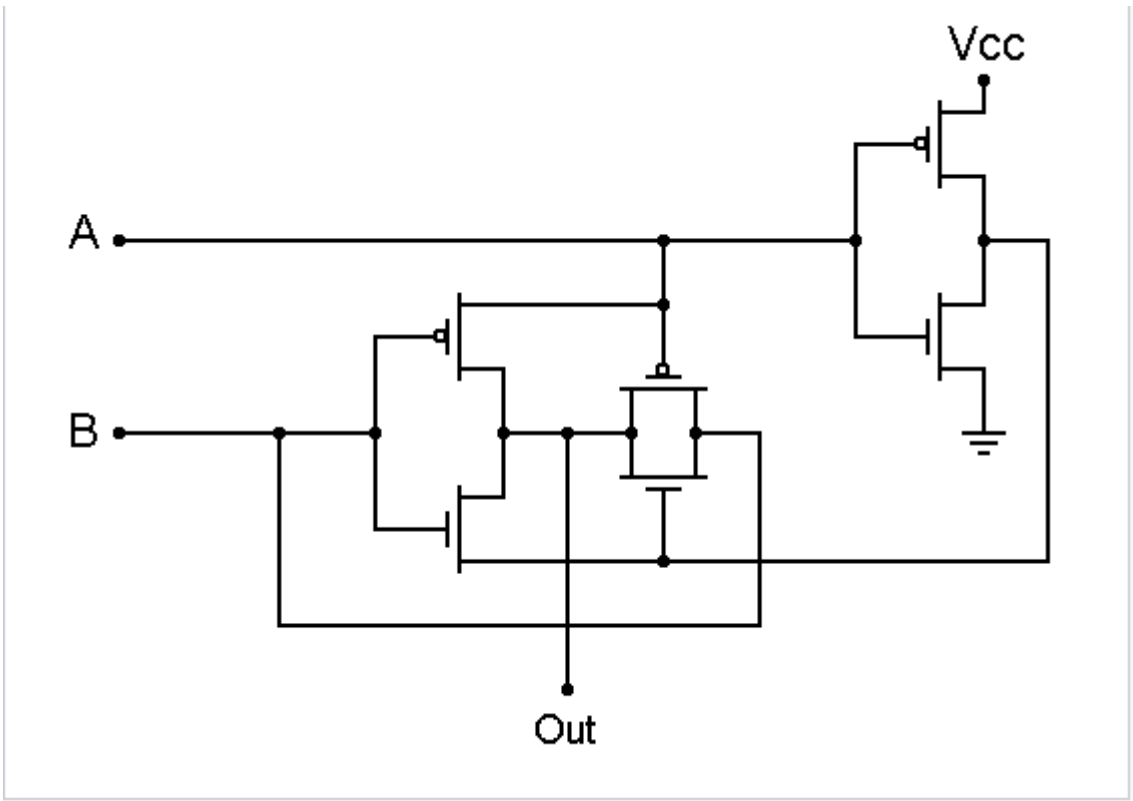
emmmm . But in wiki and other ways I find it is much more ways to implement it



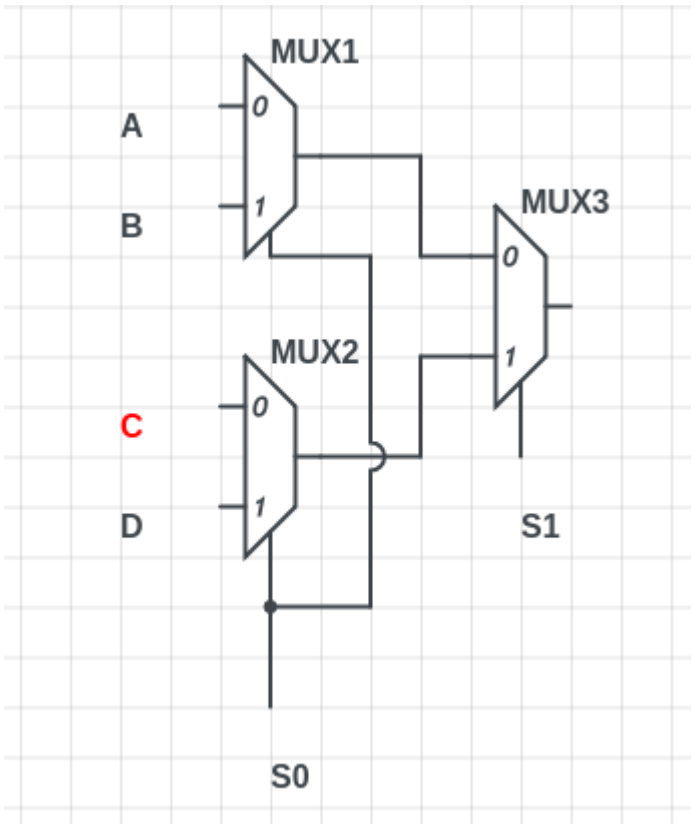


XOR gate circuit using three mixed gates

And in wiki I find XOR can be constructed using MOSFETs using only 6 transistor. How shocking !



b.



and the truth table :

S0	S1	output
0	0	A
0	1	B
1	0	C
1	1	D

3.

infer from the completed truth entries, we can get the bleow data

A: 3 4

B: 2

C: 1 5

and the truth table

A	B	C	OUTPUT
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

4.

when A is low volt.

$$D_1 = 1 * C + 0 * B = C$$
$$D_2 = \overline{0} = 1$$

and when A goes from 0 to 1

$$D_1 = 0 * C + 1 * B = B$$
$$D2 = \overline{\overline{B} * 1} = B$$

D2 circuits ouput is influenced by the last value of A while D1 is not

5.

a. 3 (for input **s** propagates through one NOT, one AND and one OR gate)

b. 3*4 = 12(from c0 to c4)

c. acually I can not find any way to reduce the delay for $Z = ABCDE$ can not be simplized....

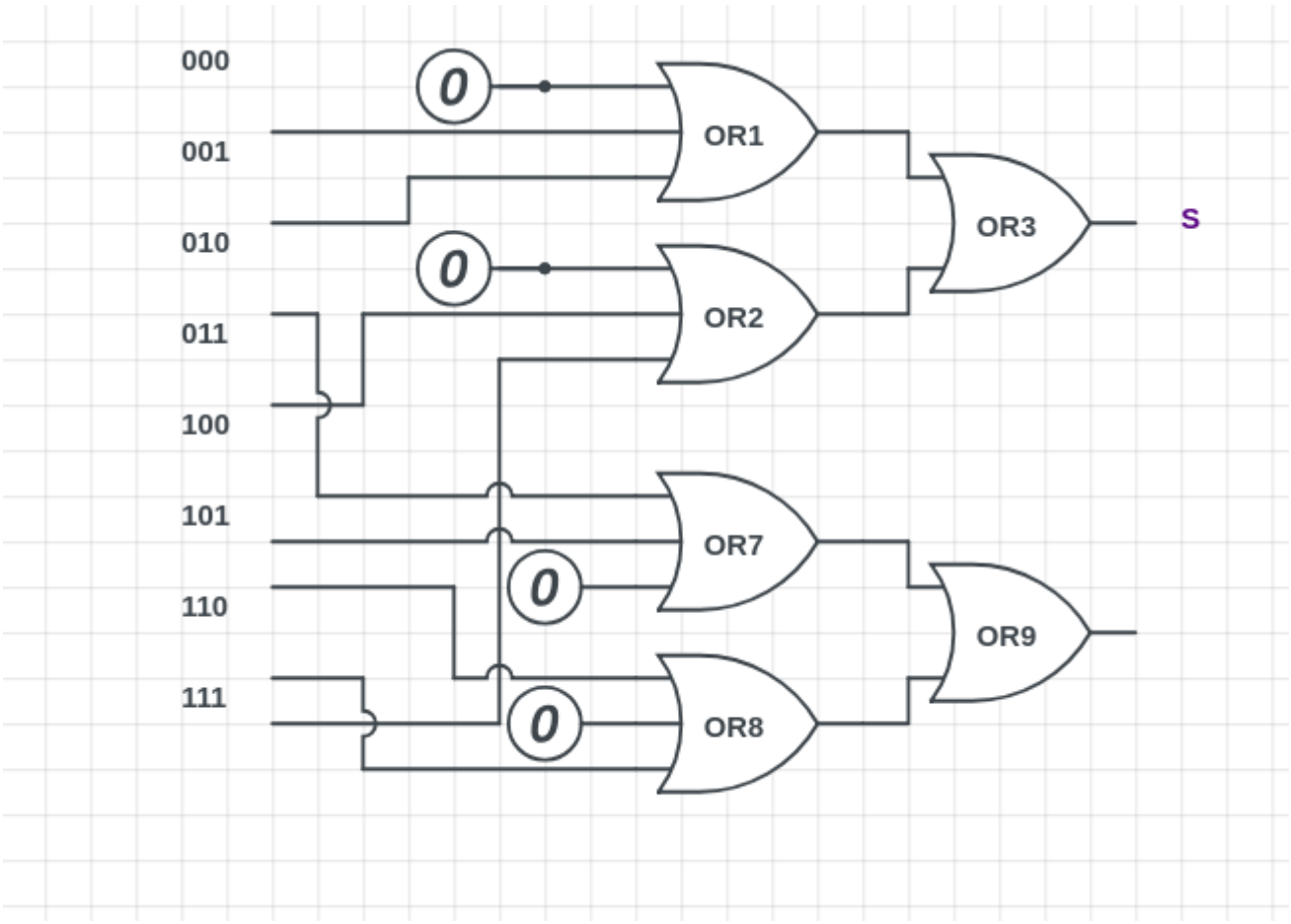
6.

throuth the truth table, we can considered each output port of 3to8 decoded as an item of 最小项表达式.

ABC	S	out
000	0	0
001	1	0
010	1	0
011	0	1
100	1	0
101	0	1
110	0	1
111	1	1

thus we can get that **s** = '001' + '010' + '100' + '111' ports and so it is **out** port.

(comment : I use two 3-input OR and one 2-input OR gate instead of the 6-input gate)



7.

Z=AC+AD+BD+~BC=A(C+D)+BD+~BC

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

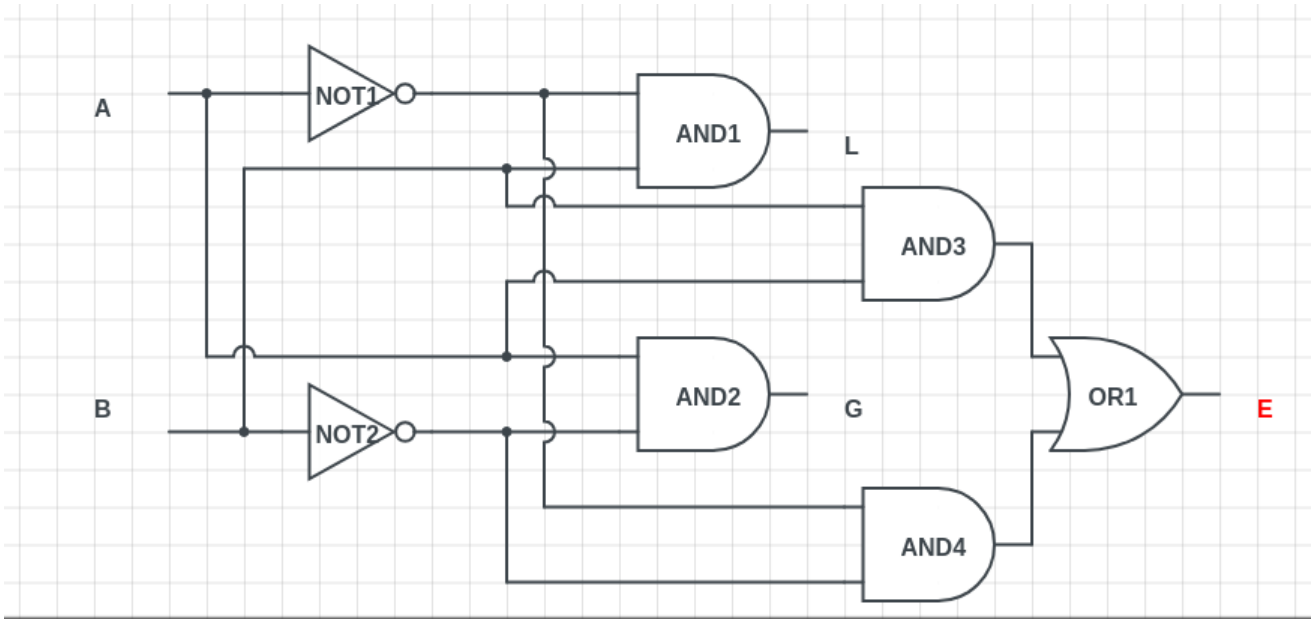
8.

a. truth table

A	B	L	G	E
0	0	0	0	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

b. circuit

$$L = \overline{A}B$$
$$G = A\overline{B}$$
$$E = \overline{A}\overline{B} + AB$$



c.

output = G[3]+E[3]G[2]+E[3]E[2]G[1]+E[3]E[2]E[1]G[1]

9.

staute change :

000000 -> 000000 -> 100000 -> 110000 -> 111000 - > 111100 -> 111110 -> 111111 ->111111 -> 111111 -> 011111 -> 001111 -> 000111 -> 000011 -> 000001 -> 000000

T = 7clk clycle , so after 50 cycle the status of circuit is 50mod7 = 1 -> 100000

10.

11.

a. truth table

2bit BCD code + 1

00: 1

01: 2

10: 3

11 : 4

current	desired	target
00	00	00
00	01	00
00	10	10
00	11	11
01	00	10
01	01	10
01	10	10
01	11	11
10	00	00
10	01	10
10	10	10
10	11	10
11	00	00
11	01	01
11	10	10
11	11	11

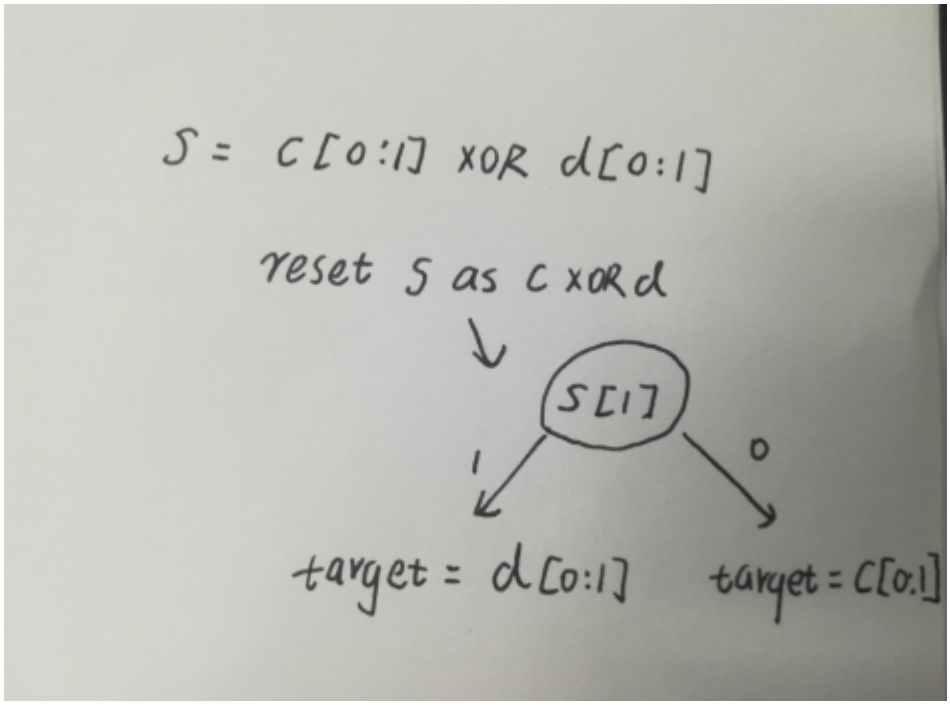
b.

input data : current floor (stored in 2 bit) . . . . . c[1:0]

desired floor (sored in 2 bi t) . . . . . d[1:0]

output data : target floor

FSM:


S = c[0:1] XOR d[0:1]
reset S as c XOR d
S[1]
1
0
target = d[0:1]
target = C[0:1]