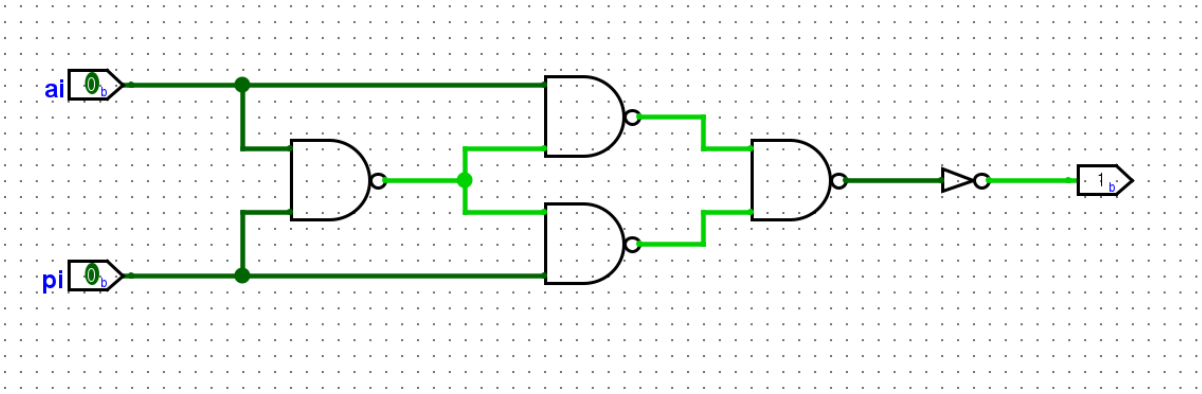
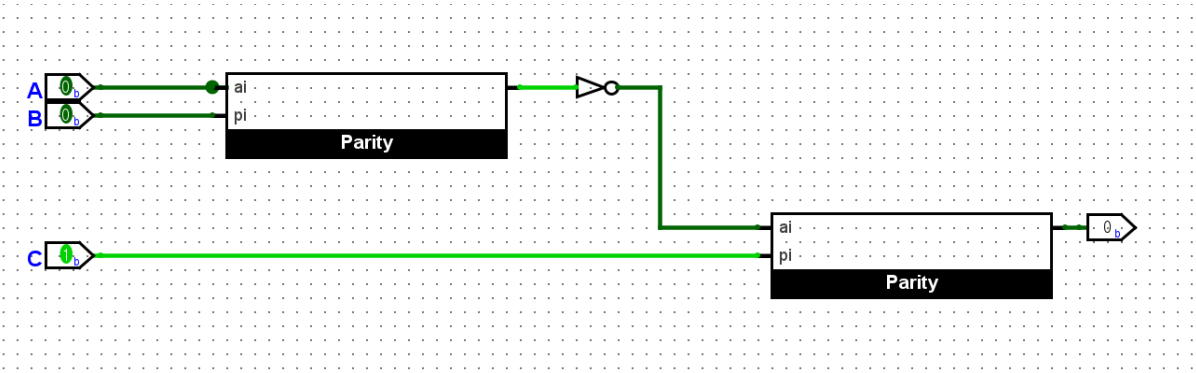
<1-a번>

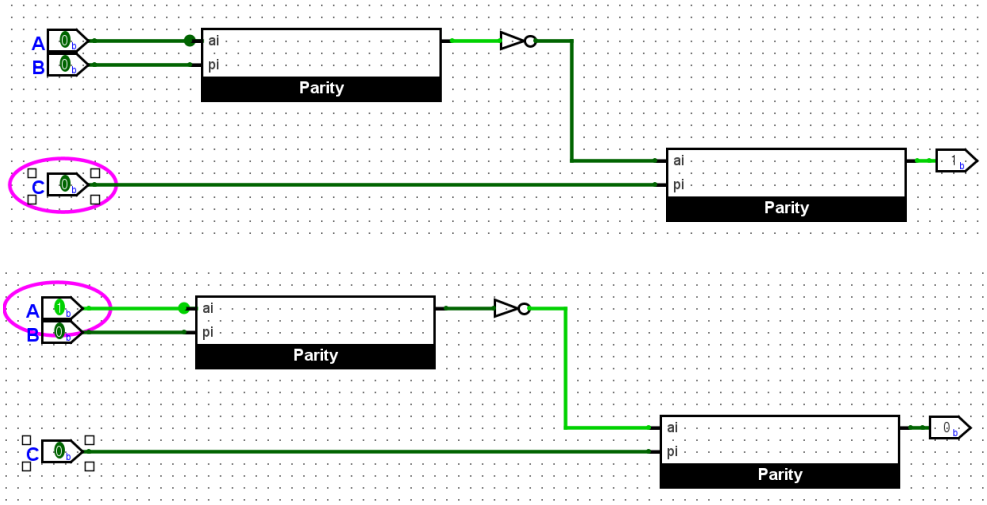


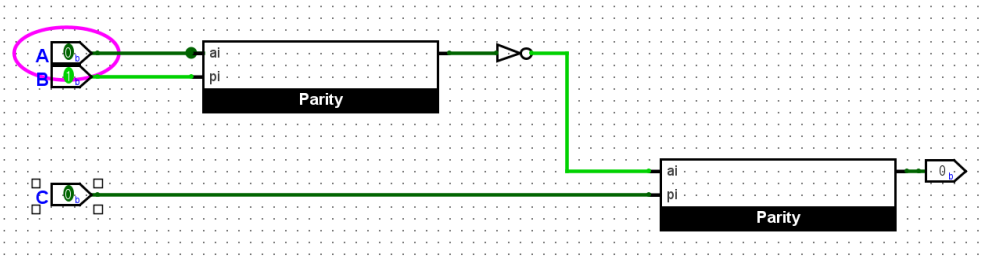
<1-b번>

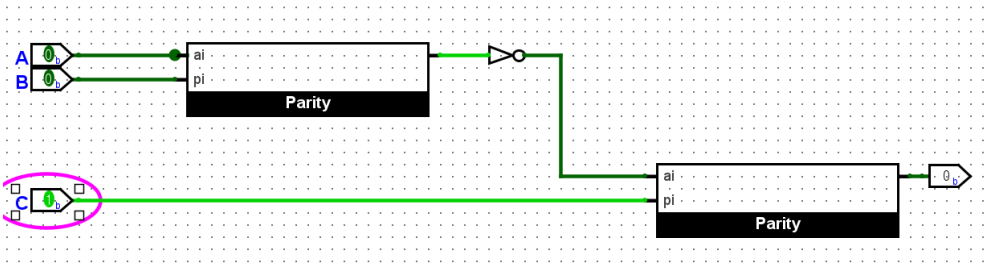
: (a)번에서 1bit parity 구현한 것을 (b)번에서 활용함 이름은 Parity로 지정했음.

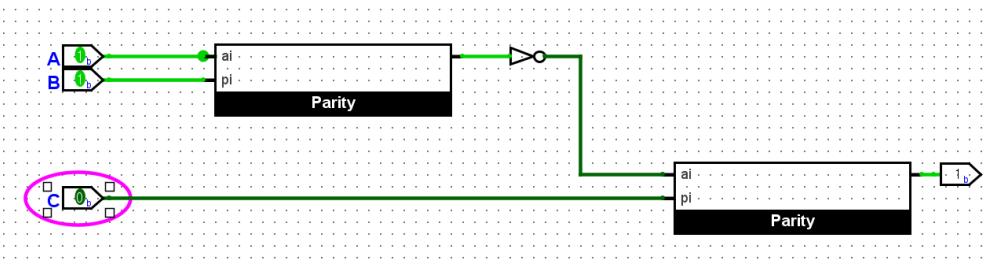


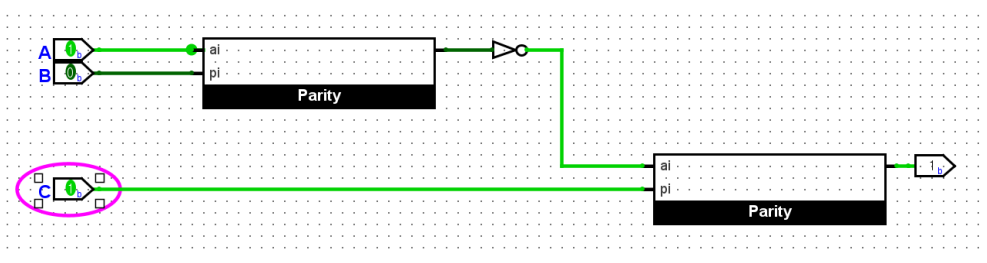
<test.vector대신 모든 상황 캡쳐>

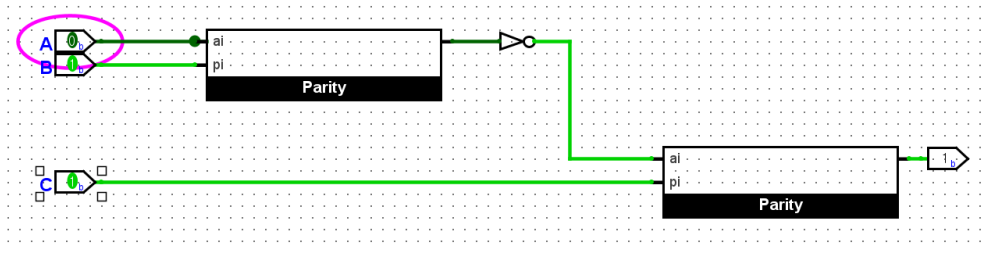


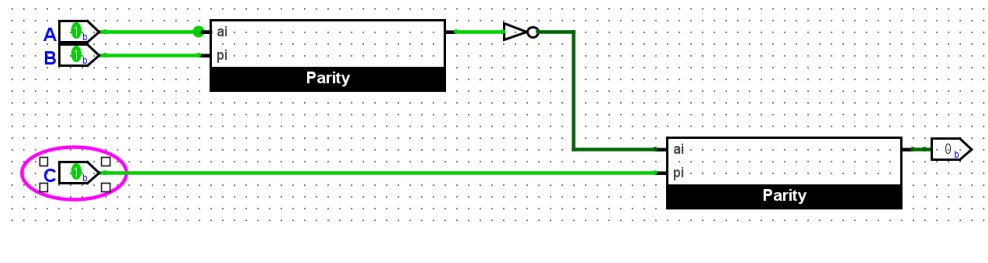

A
B
C
ai
pi
Parity
ai
pi
Parity
0


A
B
C
ai
pi
Parity
ai
pi
Parity
0


A
B
C
ai
pi
Parity
ai
pi
Parity
1

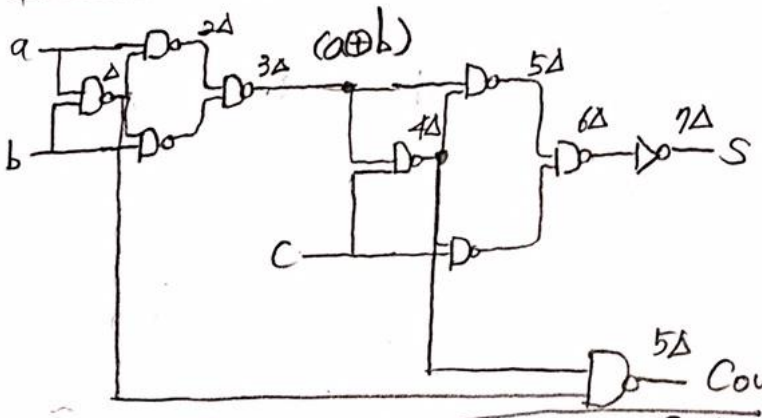

A
B
C
ai
pi
Parity
ai
pi
Parity
1


A
B
C
ai
pi
Parity
ai
pi
Parity
1

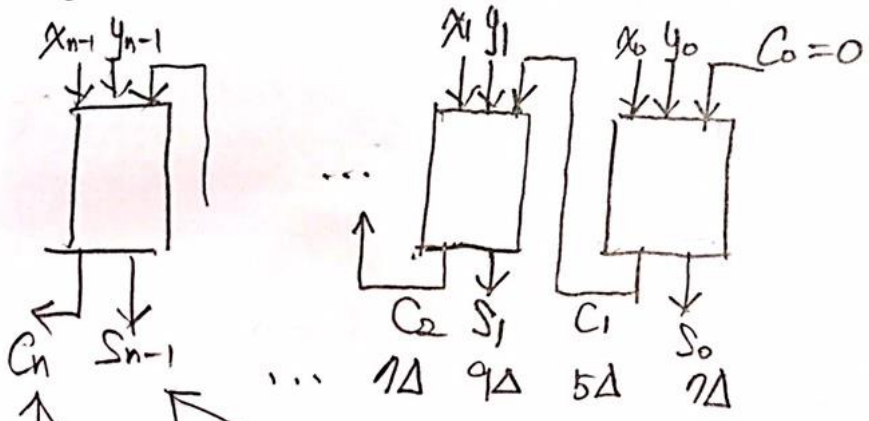

A
B
C
ai
pi
Parity
ai
pi
Parity
0

(c). Determine the delay to compute an odd parity of an n-bit binary number.

$$Cout = (a \oplus b) \cdot C + a \cdot b$$



$x(y) \Rightarrow S : 7\Delta$ $Cin \Rightarrow S : 4\Delta$ $x(y) \Rightarrow Cout : 5\Delta$ $Cin \Rightarrow Cout : 2\Delta$



$5\Delta + 2(n-1)\Delta$

$5\Delta + 2(n-2)\Delta + 4\Delta$

$= (2n+5)\Delta$

For $n = 1$, delay $= 7\Delta$

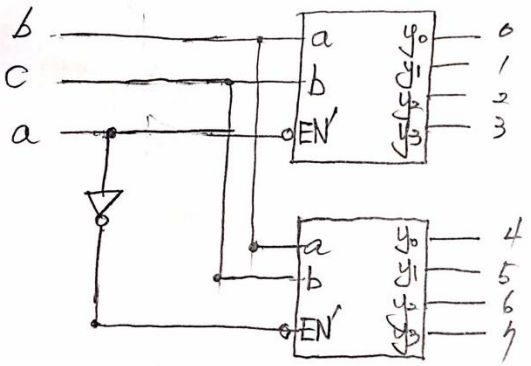
For $n \geq 2$, delay $= (2n+5)\Delta$

2. (a).

2-to-4 decoder를 이용해야하므로

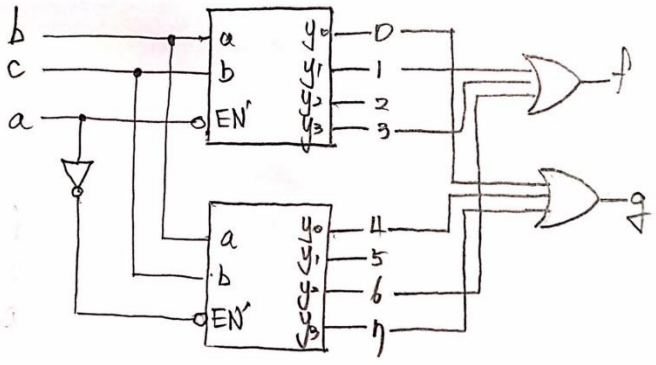
a & c 중 b, c를 input으로 설정한다.

(MSB) (LSB)



(b).

active high 3-to-8 decoder.



3. 4-to-2 priority encoder.

priority ↑

A3	A2	A1	A0	Z1	Z0
0	x	x	x	1	1
1	0	x	x	1	0
1	1	0	x	0	1
1	1	1	0	0	0

<Z1>

A3A2 \ A1A0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0		0
10	1	1	1	1

$Z1 = A3' + A2'$

<Z0>

A3A2 \ A1A0	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1		0
10	0	0	0	0

$Z0 = A3' + A2A1'$