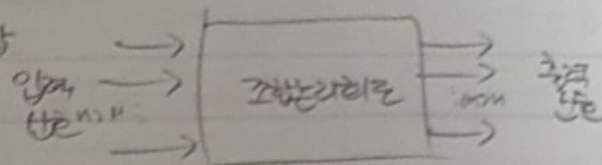


< 디지털공학 과제 3 > 20190123 오늘

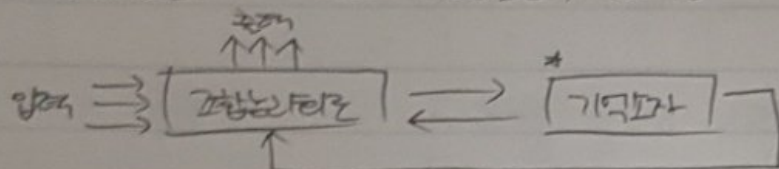
조합 논리회로

- 기본 논리 게이트 (AND, OR, NOT)를 조합
- 출력신호는 입력신호에 의해서만 결정
- 입력변수, 논리게이트, 출력변수로 구성



순차 논리회로

- 출력신호가 입력신호 뿐만 아니라 이전 상태의 논리값에 의해 결정
- 조합논리회로와 기억장치(플립플롭)로 구성



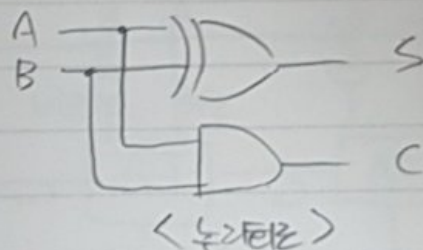
[조합논리회로]

① 가산기

1) 반가산기 → 2비트 덧셈

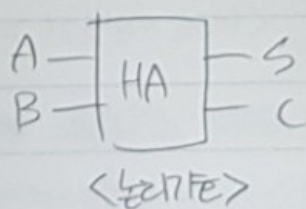
→ $A+B \Rightarrow \text{sum (합), carry (가라울림)}$

입력		출력	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



$$S = \overline{A}B + A\overline{B} = A \oplus B$$

$$C = A \cdot B$$



< 반가산기 논리식 >

2) 전가산기 → (하위비트 가라울림) + 2비트 덧셈

→ $C_{in} + A + B \rightarrow \text{sum, carry}$

입력			출력	
A	B	C_{in}	S	Count
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

$$S = \overline{A}\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\overline{C}_{in} + ABC_{in}$$

$$= A(\overline{B}C_{in} + B\overline{C}_{in}) + A(\overline{B}\overline{C}_{in} + BC_{in})$$

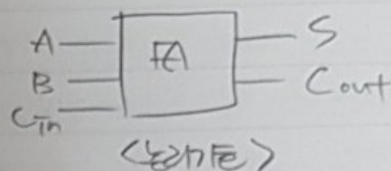
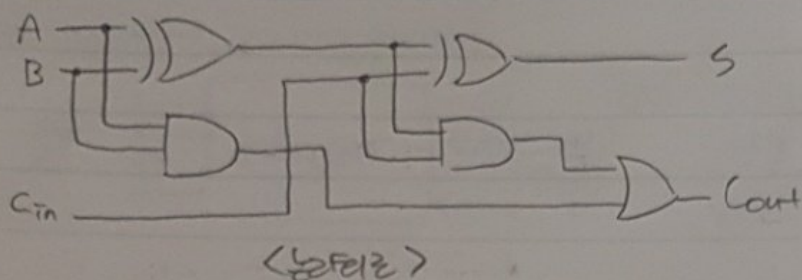
$$= A(B \oplus C_{in}) + A(\overline{B \oplus C_{in}})$$

$$= A \oplus (B \oplus C_{in}) = (A \oplus B) \oplus C_{in}$$

$$Count = \overline{A}BC_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\overline{C}_{in} + ABC_{in}$$

$$= C_{in}(\overline{A}B + AB) + AB(\overline{C}_{in} + C_{in})$$

$$= C_{in}(A \oplus B) + AB$$

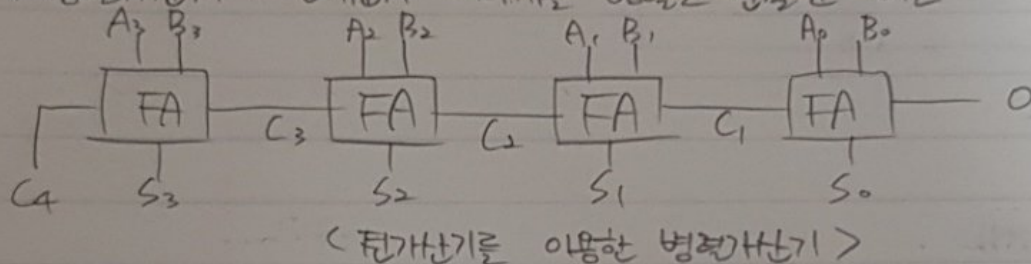


$$S = A \oplus B \oplus C_{in}, \quad C_{out} = C_{in}(A \oplus B) + AB$$

* 1비트 가산기는 2비트 OR 게이트를 이용하여 구성된다.

3) 병렬가산기

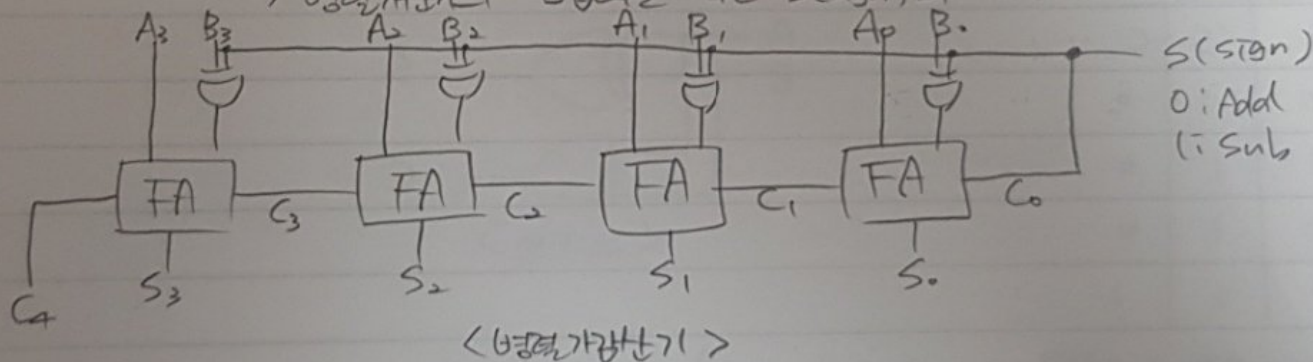
* 병렬가산기 : 1비트 가산기 여러개를 병렬로 연결한 회로



< 1비트 가산기를 이용한 병렬가산기 >

* 병렬가산기 → 덧셈과 뺄셈 모두 가능

→ 병렬가산기의 B입력을 부른 S(sign)와 XOR해서 1비트 가산기의 입력으로 사용



< 병렬가산기 >

4) 곱셈가산기 → 병렬가산기의 자리올림을 전달로 인한 속도저하의 단점을 극복한 가산기

* 캐리예측가산기 (Carry-look-ahead-adder, CLA) ⇒ 부분제곱기 + 캐리예측회로

→ 자리올림을 미리 계산한다.

* A/B의 입력이 들어오는 각 비트에 대해서 다음의 연산을 수행하여 6타 P를 얻어낸다.

$$G(A, B) = A \cdot B$$

$$P(A, B) = A \oplus B$$

* G: 자리올림 생성

P: 자리올림 전달

* 이렇게 얻어진 G, P는 S와 C를 만드는 다음의 식으로 활용된다.

$$S_i = P_i \oplus C_i, \quad C_{i+1} = G_i + (P_i \cdot C_i)$$

② 비교기 (Comparator)

* 1비트 비교기

입력		출력			
A	B	$A=B$ F_1	$A \neq B$ F_2	$A > B$ F_3	$A < B$ F_4
0	0	1	0	0	0
0	1	0	1	0	1
1	0	0	1	1	0
1	1	1	0	0	0

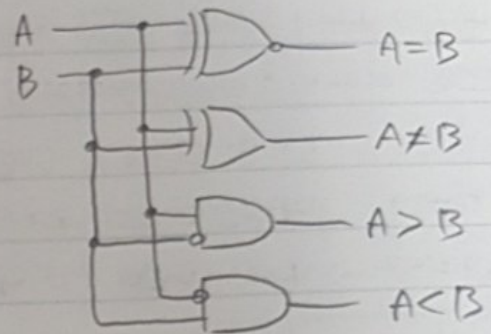
$$F_1 = \overline{A \oplus B}$$

$$F_2 = A \oplus B$$

$$F_3 = A \overline{B}$$

$$F_4 = \overline{A} B$$

(진리표 & 논리식)



< 회로 >

* 2비트 비교기

$$F_1 = \overline{(A_1 \oplus B_1)} (\overline{A_2 \oplus B_2})$$

$$F_2 = (A_1 \oplus B_1) + (A_2 \oplus B_2)$$

$$F_3 = A_1 \overline{B_1} + A_2 \overline{B_1} \overline{B_2} + A_1 A_2 \overline{B_2}$$

$$F_4 = \overline{A_1} B_1 + \overline{A_1} \overline{A_2} B_2 + \overline{A_2} B_1 B_2$$

* IC 7485 (4비트 비교기)

→ $A_3 \sim A_0$ 와 $B_3 \sim B_0$ 의 크기를 비교하는 해

→ 출력

- AGBO : $A > B$ 일 때 1
- ALBO : $A < B$ 일 때 1
- AEBO : $A = B$ 일 때 1

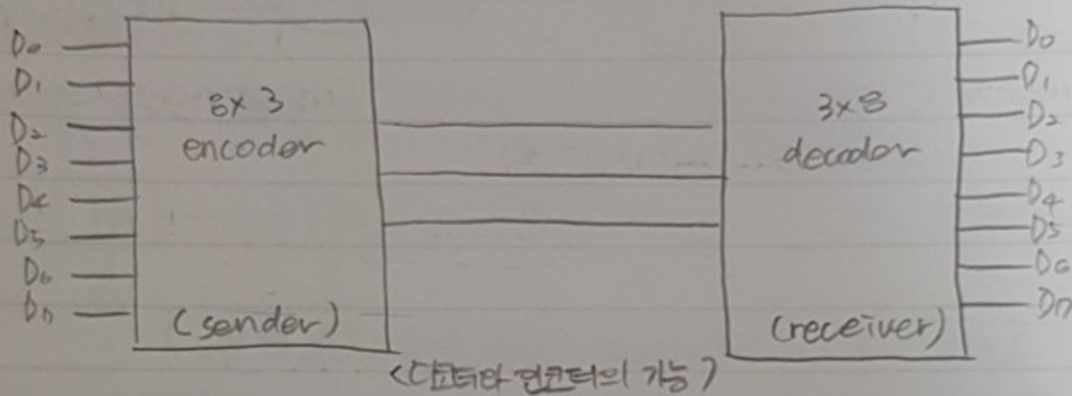
→ 확장 입력 AGBI, ALBI, AEBI 가 LSB로 입력

- 아래단의 AGBO, ALBO, AEBO 출력이 위단의 AGBI, ALBI, AEBI로 입력
- 맨 아래단의 AGBI, ALBI는 0을 AEBI는 1을 입력

③ 디코더

- 디코더: n비트의 2진코드를 입력 → 최대 2ⁿ개의 서로 다른 형태로 변환
- 인에이블(enable) 단자를 가진 경우는 디멀티플렉서의 기능도 수행
- 병용 IC는 인에이블(enable) 단자를 여러 디코더, 디멀티플렉서 병용

- IC 74138 : 3x8 디코더 / 디멀티플렉서
- IC 74139 : 독립된 2개의 2x4 디코더 / 디멀티플렉서
- IC 74154 : 4x16 디코더 / 디멀티플렉서

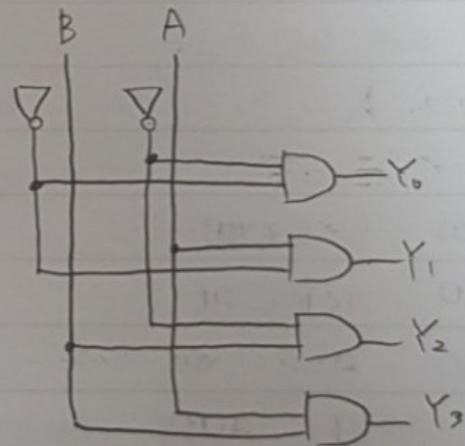


(1) 2x4 디코더

입력		출력			
B	A	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$Y_0 = \overline{B}A, Y_1 = \overline{B}\overline{A}$$

$$Y_2 = B\overline{A}, Y_3 = BA$$



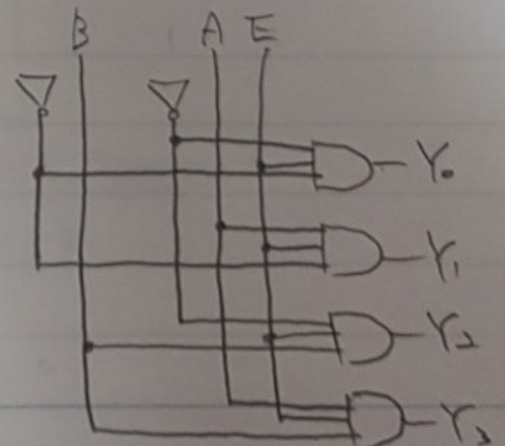
< 리본 >

* 실제 IC들은 AND게이트가 아닌 NAND 게이트로 구성.

2) 인에이블이 있는 2x4 디코더 → E=1일때만 출력이 동작

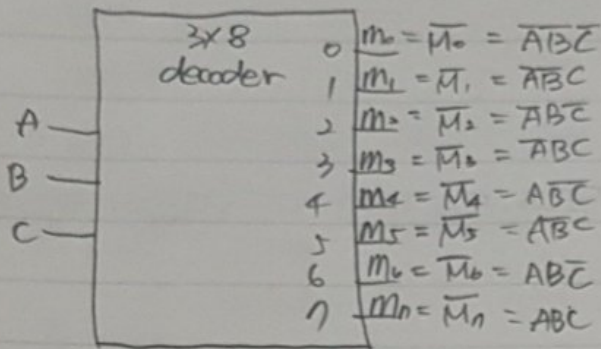
입력			출력			
E	B	A	Y ₃	Y ₂	Y ₁	Y ₀
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$$\begin{aligned} Y_0 &= E\overline{B}A \\ Y_1 &= E\overline{B}\overline{A} \\ Y_2 &= EBA \\ Y_3 &= EBA \end{aligned}$$

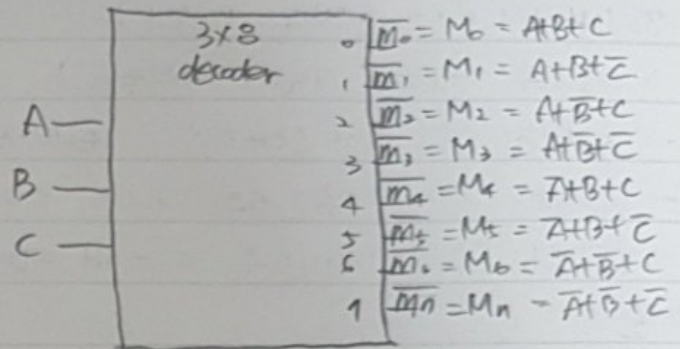


3) 디코더를 이용한 조합논리회로

* 3x8 디코더를 이용하는 경우

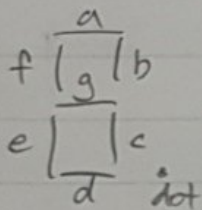


<3x8 디코더 출력>

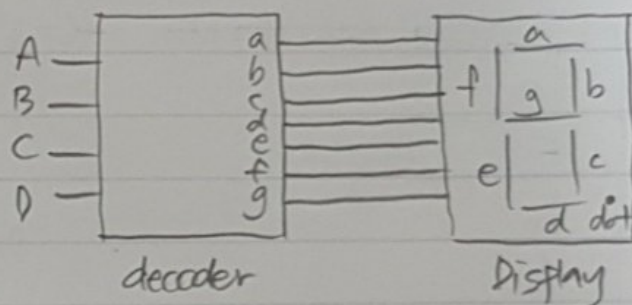


<3x8 디코더 반진출력>

4) BCD-7-시그널 디코더 \Rightarrow 숫자 표시용 장치



<7-시그널 구성>

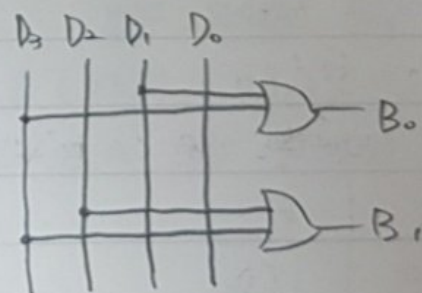


<7-시그널 디코더의 연결>

4) 인코더 (encoder)

- 디코더의 반대기능을 수행
 - 2^n 개의 입력신호 \rightarrow n 개의 출력신호
 - 2^n 개중 활성화된 하나의 1비트 입력신호를 받아서 그 숫자에 해당하는 n 비트 2진정보 출력
- * 4x2 인코더 : 입력의 상태에 따라 2개의 2진조합으로 출력된다.

입력				출력	
D_3	D_2	D_1	D_0	B_1	B_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1



<회로도>

$$B_1 = D_2 + D_3, B_0 = D_1 + D_0$$

<진리표와 공식>

* 8x3 인코더 : $8 (= 2^3)$ 개의 입력과 3개의 출력을 가지며, 입력의 상태에 따라 3개의 2진조합으로 출력

* 8x3 우선순위 인코더 : 입력에 우선순위를 정하고, 여러개의 입력이 있을 때, 우선순위가 높은 입력을 출력으로 연딩

* IC 7448 (8x3 우선순위 인코더)

→ 우선순위 인코더 (7번이 가장 높음)

→ 논리반전 입력 8개 (0-7), 논리반전 출력 3개

→ GS 출력 : 데이터 입력 중의 하나가 0이고, EI가 0일 때만 0 (출력값이 유효함)

→ EI와 EO는 7448은 여러개 연결할 때 사용

⑤ 멀티플렉서 (multiplexer or selector)

→ 선택선의 입력값에 따라, 여러개의 데이터 입력 중 하나를 출력선으로 연결

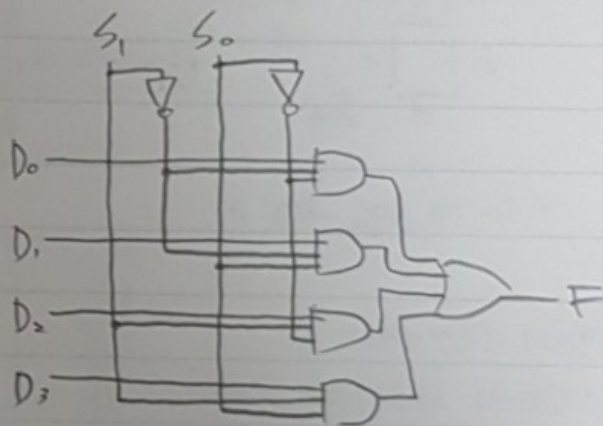
→ 데이터 입력 2ⁿ 비트, 선택비트 n개 → 출력 1비트

* 4x1 멀티플렉서 : 선택선 S₁, S₀ 값에 따라, 4(=2²)개 입력 중 하나를 출력으로 전달

선택선		출력
S ₁	S ₀	F
0	0	D ₀
0	1	D ₁
1	0	D ₂
1	1	D ₃

<진리표>

$$F = \overline{S_1} \overline{S_0} D_0 + \overline{S_1} S_0 D_1 + S_1 \overline{S_0} D_2 + S_1 S_0 D_3$$



<회로도>

* 8x1 멀티플렉서

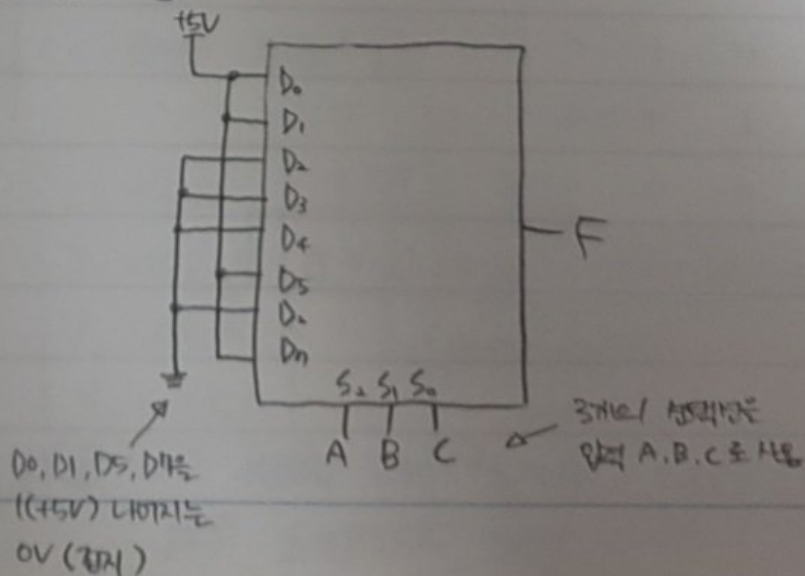
⇒ 8(=2³) 개의 입력중의 하나를 출력으로 보내주는 조합논리회로

* 멀티플렉서를 이용한 조합회로 구현

② F(A,B,C) = Σm(0,1,5,7) 를 8x1 멀티플렉서로 구현하는 경우

↳ 3개의 선택선을 입력 A,B,C로 사용

A	B	C	F
0	0	0	1 (D ₀)
0	0	1	1 (D ₁)
0	1	0	0 (D ₂)
0	1	1	0 (D ₃)
1	0	0	0 (D ₄)
1	0	1	1 (D ₅)
1	1	0	0 (D ₆)
1	1	1	1 (D ₇)

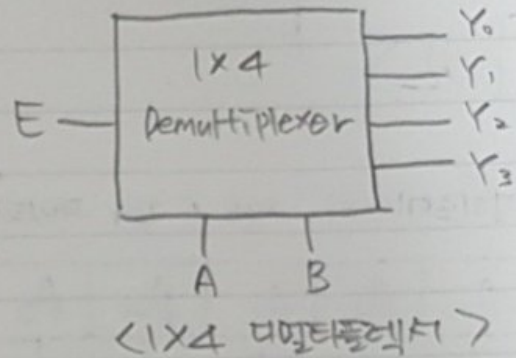
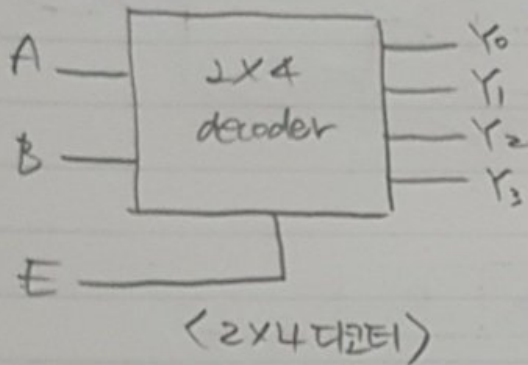


⑥ 디멀티플렉서

→ 1개의 인메이블 입력을 가지고 있는 디코더는 디멀티플렉서로 사용 가능

→ 정보를 한 선으로 보내기 2^n 개의 가능한 출력 선들 중 하나를 선택하여, 받은 정보를 전송하는 회로이다.

→ 1개의 선택입력을 이용하여 출력을 제어



⑦ 코드변환기

1) 2진 코드 - 그레이 코드 변환

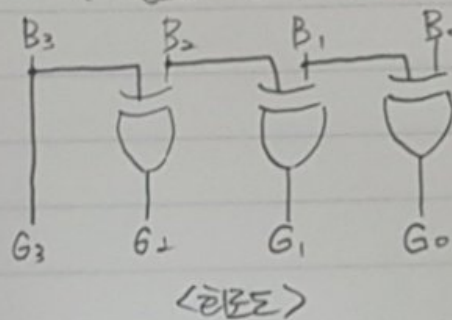
(2진코드 = B, 그레이코드 = G)

$$G_3 = B_3$$

$$G_2 = B_3 \oplus B_2$$

$$G_1 = B_2 \oplus B_1$$

$$G_0 = B_1 \oplus B_0$$



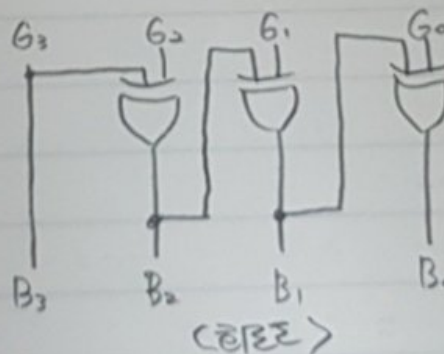
2) 그레이코드 - 2진코드 변환

$$B_3 = G_3$$

$$B_2 = G_3 \oplus G_2$$

$$B_1 = B_2 \oplus G_1$$

$$B_0 = B_1 \oplus G_0$$



3) BCD 코드 - 3진라 코드 변환

$$E_3 = B_3 + B_2 B_1 + B_2 B_0$$

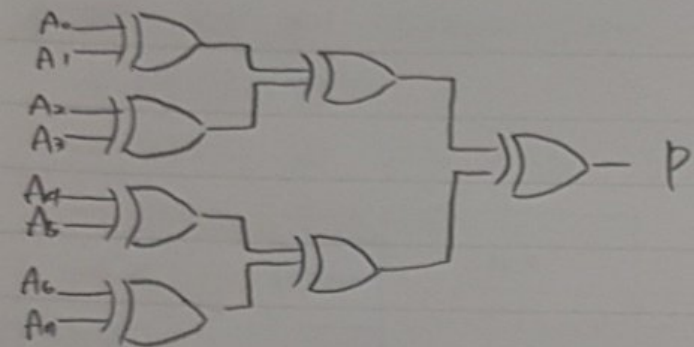
$$E_2 = \overline{B_2} B_1 + \overline{B_2} B_0 + B_2 \overline{B_1} \overline{B_0}$$

$$E_1 = \overline{B_1 \oplus B_0}$$

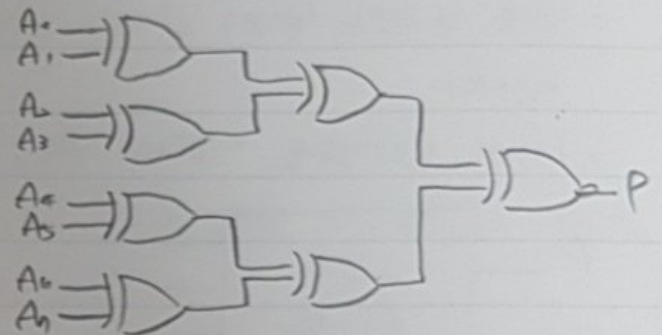
$$E_0 = \overline{B_0}$$

⑧ 패리티 발생기 / 검출기

1) 짝수 패리티 발생회로



2) 홀수 패리티 발생회로



3) 8비트 직렬회로에서의 짝수 / 홀수 패리티 발생

