

## 1. Decoder 에 대해 조사하시오.

Decoder(디코더)는 활성화되었을 때, 코딩된 입력신호에 기초해 여러 개의 출력 중에 하나를 선택하는 장치이다. 즉, 만약 인코딩 된 데이터가 입력되었다면, 이를 다시 원래의 데이터로 변환하는 작업을 수행한다. 일부 디코더는 디코더의 동작을 활성화시키는 enable 신호를 가지고 있다. 이에 대한 예시도 아래에 있다.

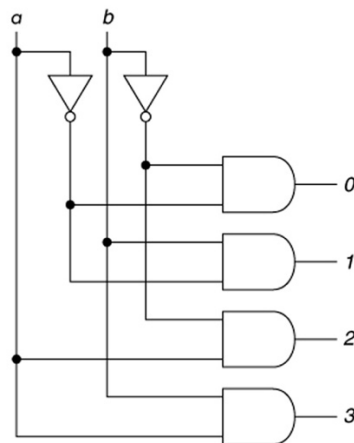
다음은 active high decode, active low decoder, decoder with enable의 예시이다.

(참고자료: Introduction to Logic and Computer Design, Alan B. Marcovitz)

**Table 4.2a** An active high decoder.

<i>a</i>	<i>b</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

**Figure 4.8a** An active high decoder.



**Table 4.2b** An active low decoder.

<i>a</i>	<i>b</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

**Figure 4.8b** An active low decoder.

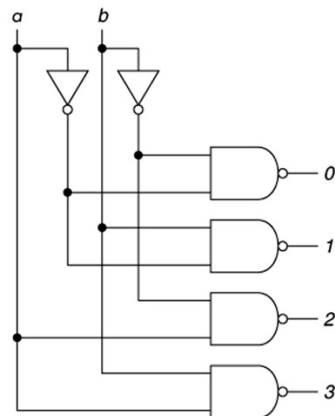
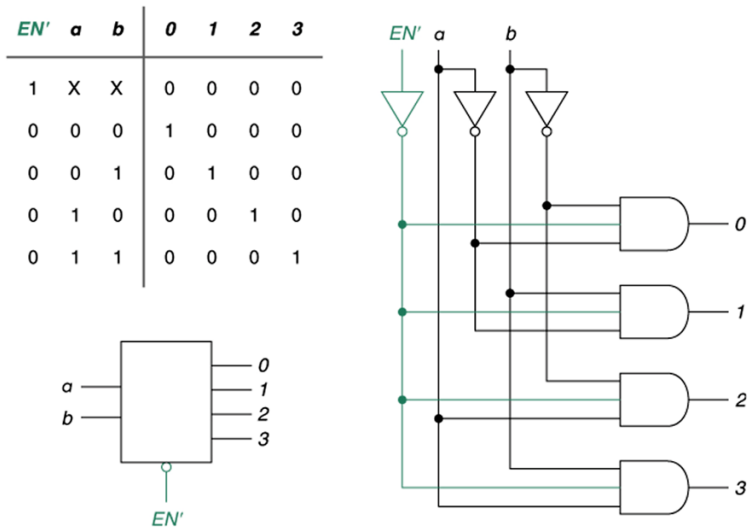


Figure 4.9 Decoder with enable.



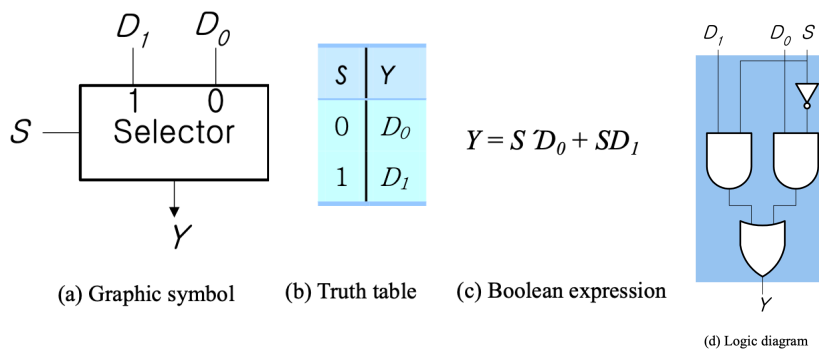
## 2. Encoder 에 대해 조사하시오.

인코더는 이진 인코더의 반대되는 동작을 수행한다. 여러 개의 장치가 컴퓨터에 연결된 선에 1(High)를 전달했을 때 유용하게 쓰인다. -> 장치의 번호를 생성한다. 즉,  $2^n$ 개의 입력이 존재하면  $n$ 개의 출력이 나오게 된다. 다음은 인코더에 가해지는 입출력의 예시이다. (4-line 인코더일 때)

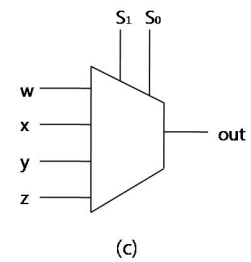
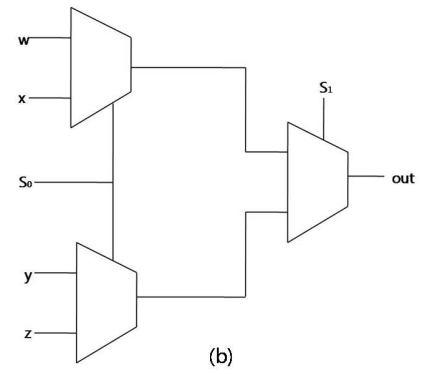
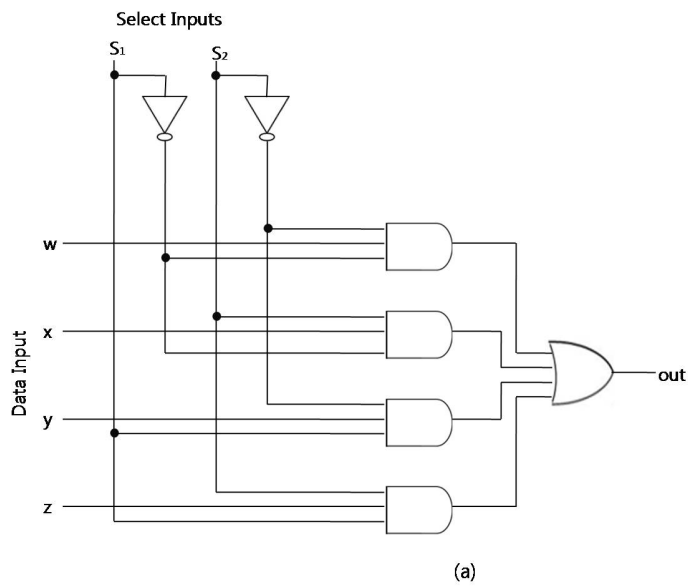
$A_0$	$A_1$	$A_2$	$A_3$	$Z_0$	$Z_1$
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

## 3. Mux(Multiplexer) 에 대해 조사하시오.

Multiplexer(멀티플렉서)는 줄여서 MUX라고도 부르며, 여러 입력 줄 하나를 선택해 입력 값에 따라 출력에 연결하는 장치(스위치)라 볼 수 있다. 많은 수의 다중 비트 숫자 중 하나를 택하기 위해 MUX의 집합이 사용되기도 한다.

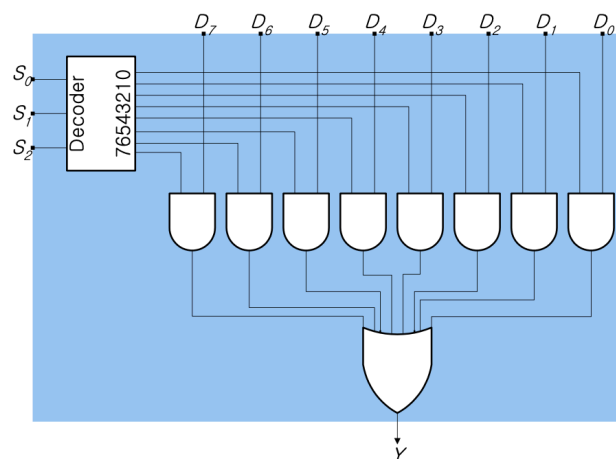


위는 2 X 1 multiplexer의 예시이다. 4 X 1, 8 X 1 multiplexer도 이를 조합하여 만들 수 있다.



↑ 4 X 1 MUX

$S_2$	$S_1$	$S_0$	$Y$
0	0	0	$D_0$
0	0	1	$D_1$
0	1	0	$D_2$
0	1	1	$D_3$
1	0	0	$D_4$
1	0	1	$D_5$
1	1	0	$D_6$
1	1	1	$D_7$



(a) Truth table

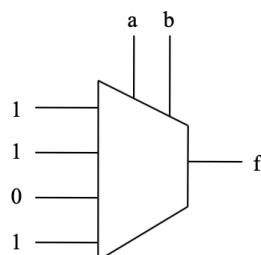
(b) Implementation with a decoder

↑ 8 X 1 MUX

멀티플렉서는 논리 회로 구현에 사용될 수 있는데, 멀티플렉서를 이용한 단순한 논리 회로의 구현 방법은 선택 입력을 디코더처럼 이용하고, 입력에 0 or 1을 연결하는 것이다.

ex)  $f(a, b) = \sum m(0, 1, 3)$ 일 때:

$a$	$b$	$f$
0	0	1
0	1	1
1	0	0
1	1	1



#### 4. Priority Encoder 에 대해 조사하시오.

Priority Encoder (우선순위 인코더)는 일반적인 인코더에 우선순위를 적용한 형태를 띤다. 만약 하나 이상의 입력이 동시에 1 이 될 수 있다면, 입력 간 우선순위를 설정해야 한다. 우선순위 인코더의 출력은 값이 1 인 입력 중 가장 높은 우선순위 입력의 번호를 표시한다. 우선순위는 가장 크거나 가장 작은 입력 번호에 대하여 최고의 우선순위를 배정한다. 다음은 8-입력 우선순위 디코더의 진리표 예시이다.

$A_0$	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$	$A_7$	$Z_0$	$Z_1$	$Z_2$	$NR$
0	0	0	0	0	0	0	0	X	X	X	1
X	X	X	X	X	X	X	1	1	1	1	0
X	X	X	X	X	X	1	0	1	1	0	0
X	X	X	X	X	1	0	0	1	0	1	0
X	X	X	X	1	0	0	0	1	0	0	0
X	X	X	1	0	0	0	0	0	1	1	0
X	X	1	0	0	0	0	0	0	1	0	0
X	1	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0

출력 중 NR은 어떠한 입력도 1이 아니었다는 것을 나타낸다.

각 출력에 대한 식은 아래와 같이 나타낼 수 있다.

$$NR = A_0' A_1' A_2' A_3' A_4' A_5' A_6' A_7'$$

$$Z_0 = A_4 + A_5 + A_6 + A_7$$

$$Z_1 = A_6 + A_7 + (A_2 + A_3)A_4' A_5'$$

$$Z_2 = A_7 + A_5 A_6' + A_3 A_4' A_6' + A_1 A_2' A_4' A_6'$$

#### 5. DeMux와 Decoder의 차이를 설명하시오.

DeMux(디멀티플렉서)는 멀티플렉서의 반대되는 동작을 수행하며, 입력된 신호를 한 장소에서 여러 장소 중 하나로 출력한다.

디코더는 입력의 개수가 상관없지만 디멀티플렉서는 반드시 1개의 입력을 가져야 한다. (선택 입력 제외) 거꾸로 디멀티플렉서의 출력에는 제한이 없지만 디코더는 입력의 개수에 따라 출력의 개수가 달라지게 된다. 아래에 DeMux와 Decoder의 차이를 표로 정리하였다.

	DeMux	Decoder
정의	1개의 입력 2 <sup>n</sup> 개의 출력	n개의 입력 2 <sup>n</sup> 개의 출력, n개의 제어 입력
특징	입력을 출력에 연결	이진 값을 디코딩하여 2 <sup>n</sup> 개의 출력 중 하나 선택
반대 개념	멀티플렉서	인코더

