

Decoder/Encoder and MUX/DeMUX

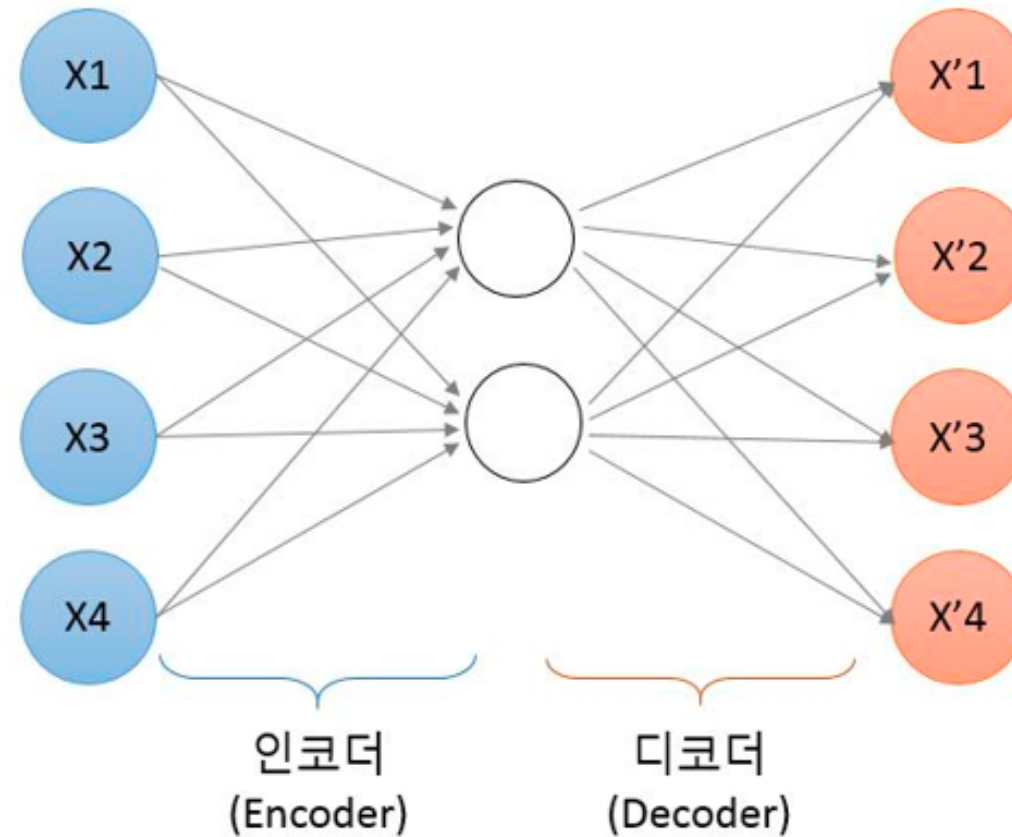
2조 (9주차)

20161205 한윤도

20221572 심은

목차

1. 디코더와 인코더 소개
2. 4 to 2 인코더 <> 2 to 4 디코더 예시
3. BCD <> Decimal 인코더/디코더 예시
4. MUX와 2×1 MUX, 4×1 MUX, 8×1 MUX 예시
5. 작은 MUX를 이용한 큰 MUX의 구현
6. DeMUX와 1 x 4 예시
7. 출처 및 기여도



Encoder

보안이나 데이터 크기 축소 등의 목적으로
특정 데이터를 다른 형태나 형식으로
변환하는 회로

Decoder

인코딩된 데이터를 원래 상태로
돌려놓기 위해 변환 및 처리하는 회로.

반대의 역할을 수행한다

예시) 4 to 2 인코더

D2+D3 D1+D3

D0	D1	D2	D3	Y0	Y1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

인코딩 할 원본

인코딩 된 결과(연결된 선 번호)

A0	A1	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

디코딩 된 원본

	00	01	11	10
00	X	0	X	0
01	1	X	X	X
11	X	X	X	X
10	1	X	X	X

D3

D2

	00	01	11	10
00	X	1	X	0
01	1	X	X	X
11	X	X	X	X
10	0	X	X	X

D3

D1

예시) 2 to 4 디코더

D0	D1	D2	D3	Y0	Y1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

인코딩 할 원본

인코딩 된 결과(연결된 선 번호)

A0	A1	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

디코딩 된 원본

	0	1
0	1	0
1	0	0

$A_0'A_1'$

	0	1
0	0	0
1	1	0

$A_0'A_1$

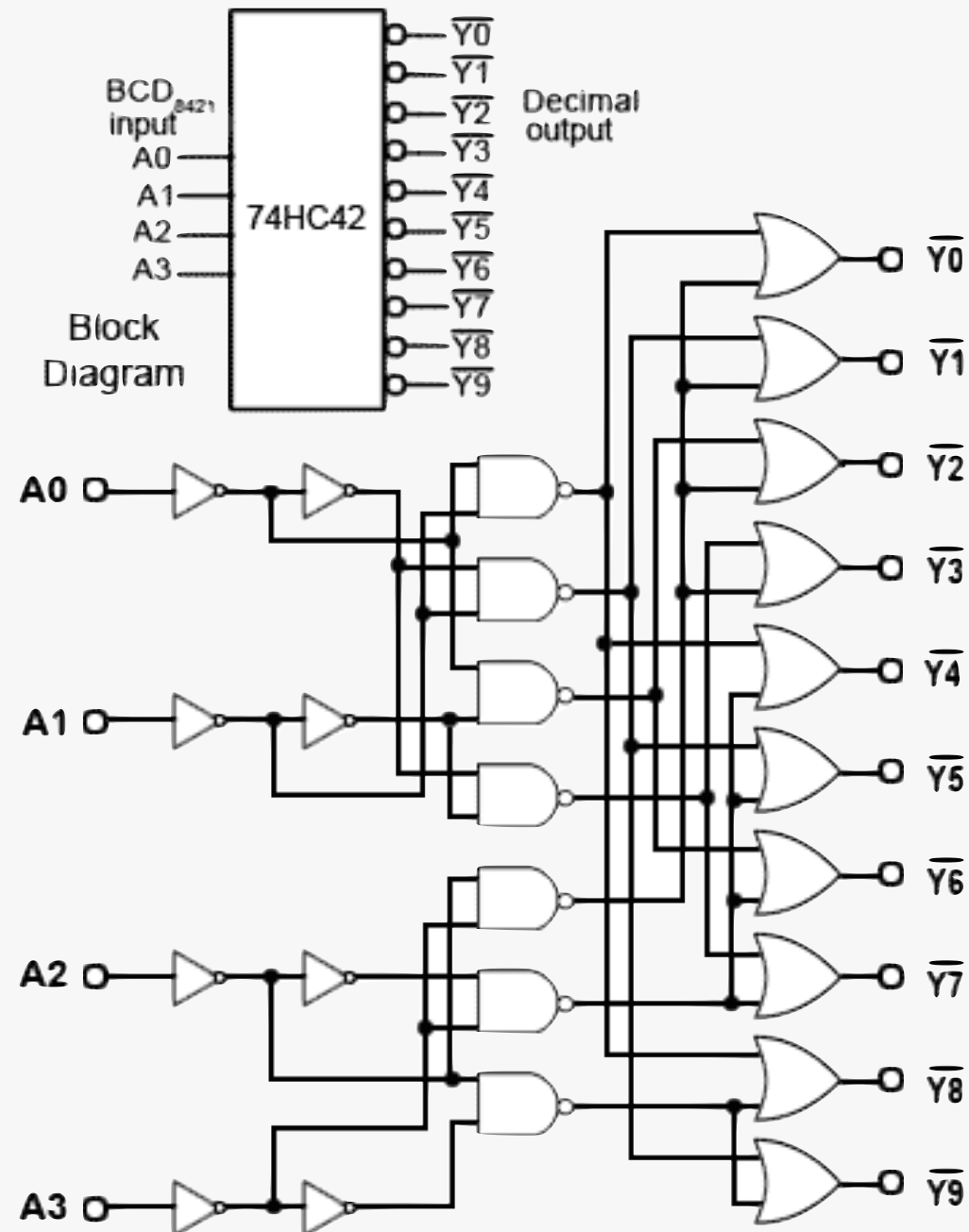
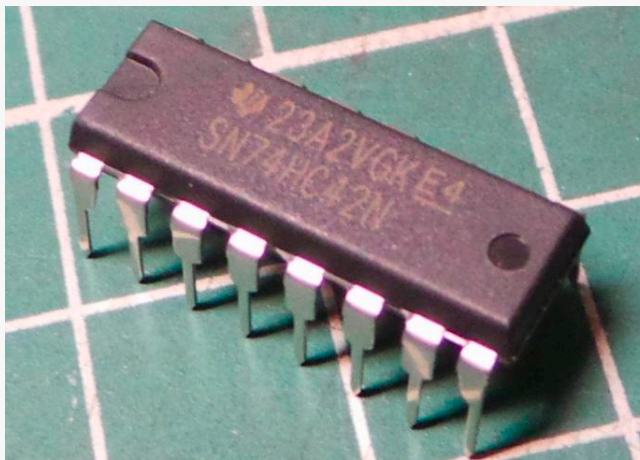
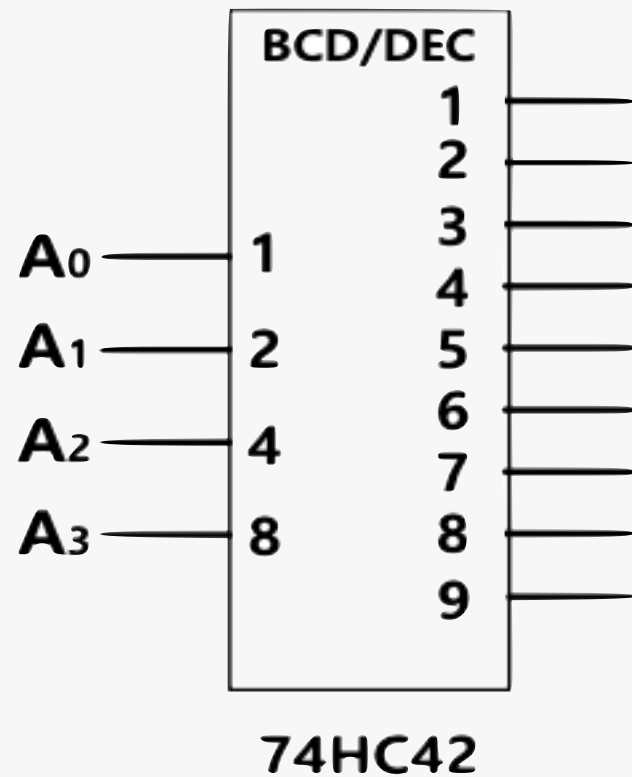
	0	1
0	0	1
1	0	0

A_0A_1'

	0	1
0	0	0
1	0	1

A_0A_1

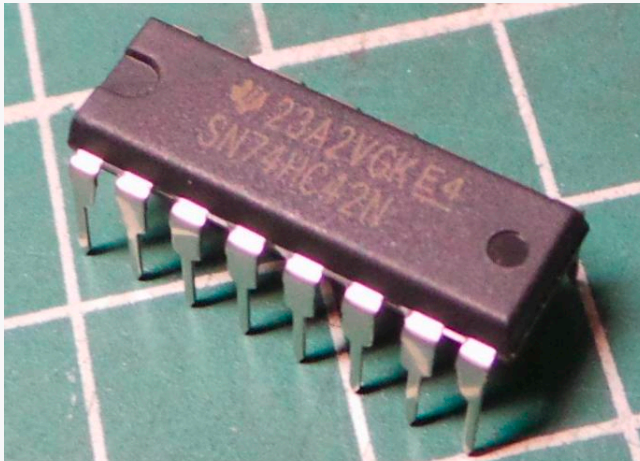
예시) BCD to Decimal Decoder



대체 어디다가 썼을까?

원래는 위에서 보이는 cold cathode Display에 썼다고 함. 하지만 Demultiplexer에도 유용하게 쓰일 수 있음

예시) BCD to Decimal Decoder



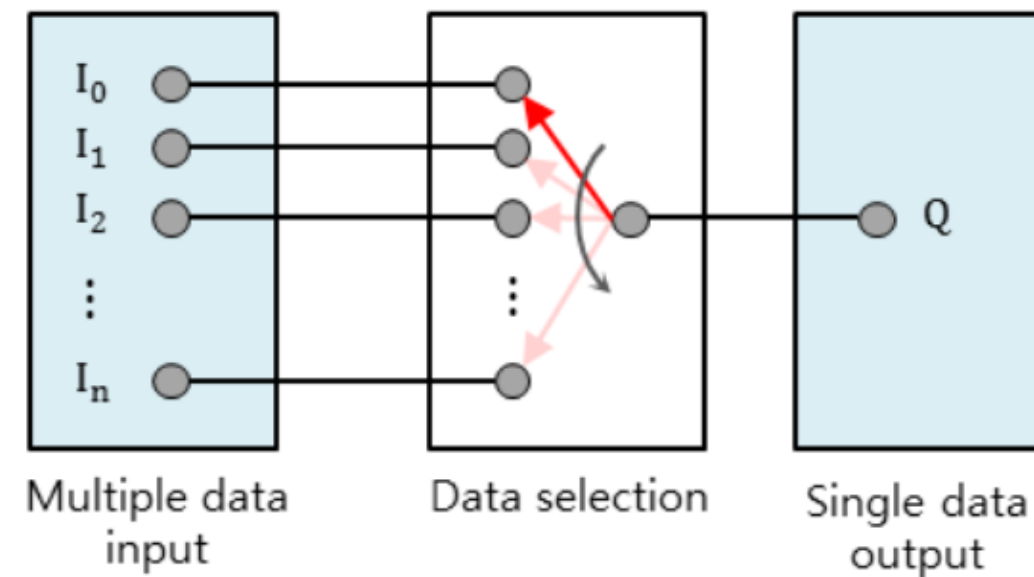
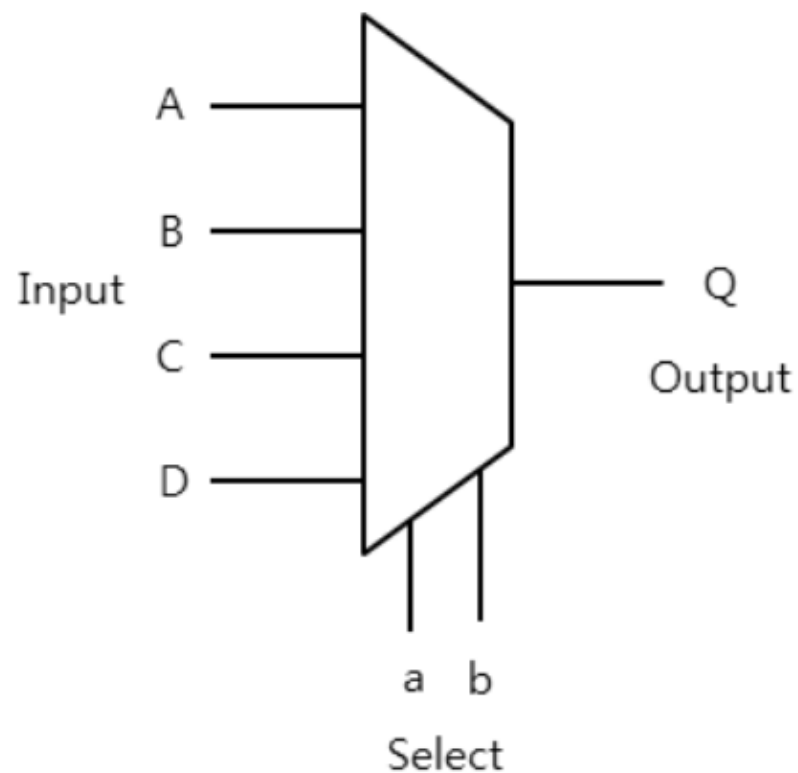
A3	A2	A1	A0	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

만약 1001보다 큰 값이 들어오면?

BCD는 4비트로 10진수 0~9까지 표현하는 코드이기 때문에 1010~1111은 표준 BCD 값이 아니다.
74HC42에서는 D0~D9가 모두 0으로 나온다고 함.

MUX(Multiplexer)

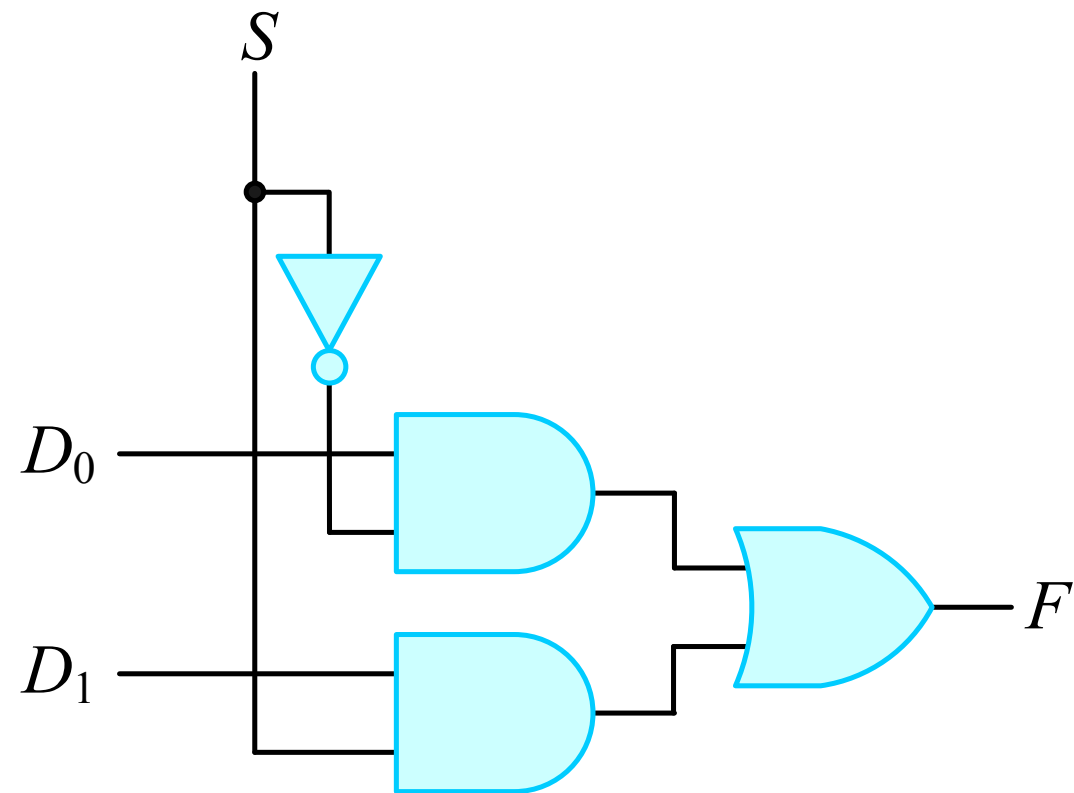
- 여러 개의 입력선들 중에서 하나를 선택하여 출력선에 연결하는 조합논리회로
- 2^n 개의 입력 신호 중 n 개의 선택 신호를 이용하여 입력 신호 중 하나를 선택하여 출력



2×1 Multiplexer

선택선	출력
S	F
0	D_0
1	D_1

$$F = \bar{S}D_0 + SD_1$$

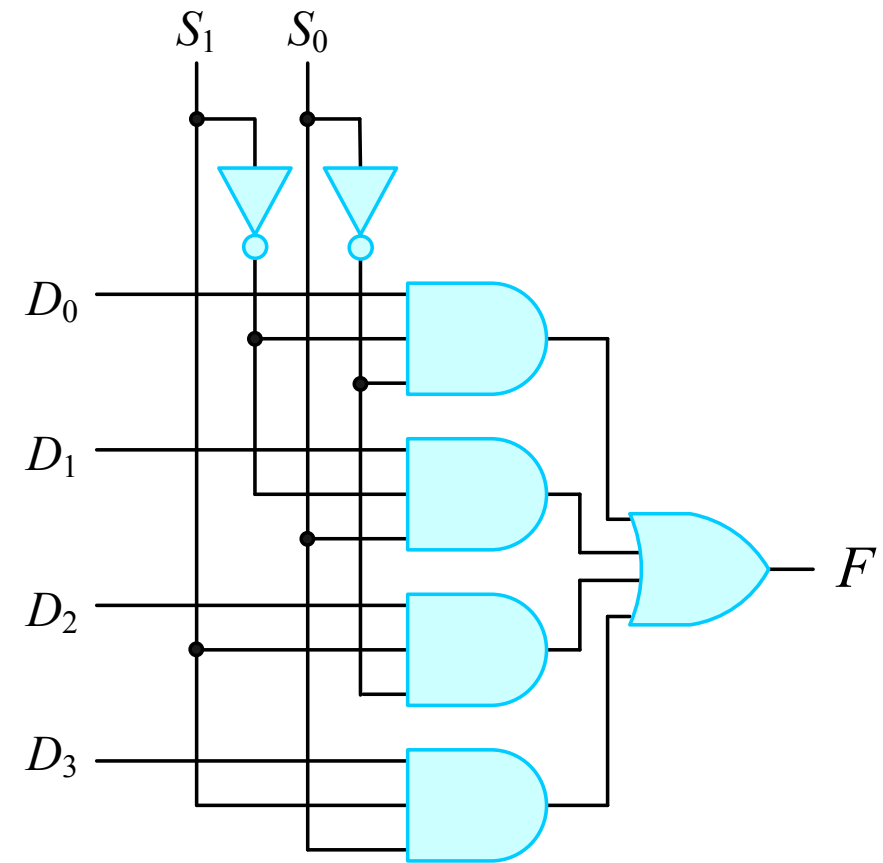


: 2($=2^1$)개의 입력중의 하나를 선택선 S 에 입력된 값에 따라서 출력으로 보내주는 조합회로

4×1 Multiplexer

선택선		출력
S_1	S_0	F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

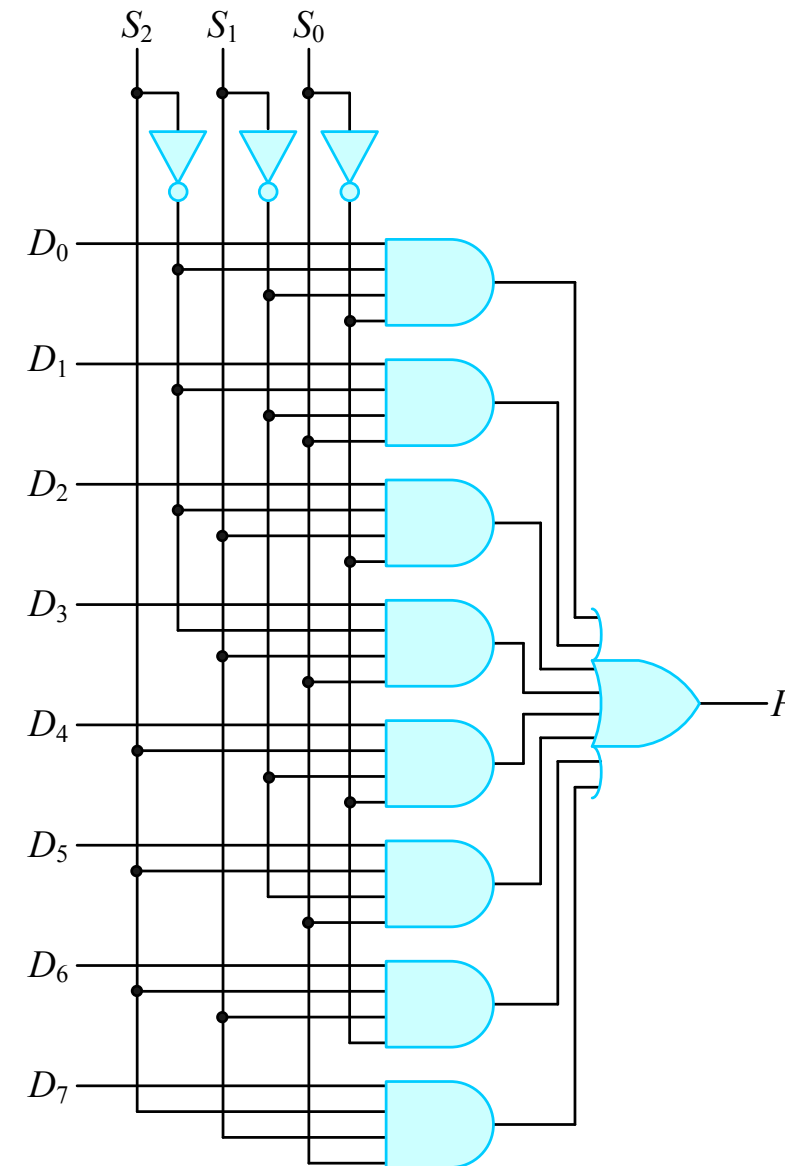
$$F = \overline{S_1}\overline{S_0}D_0 + \overline{S_1}S_0D_1 + S_1\overline{S_0}D_2 + S_1S_0D_3$$



: 4(=2²)개의 입력중의 하나를 선택선 S_1 과 S_0 에 입력된 값에 따라서 출력으로 보내주는 조합회로

8×1 Multiplexer

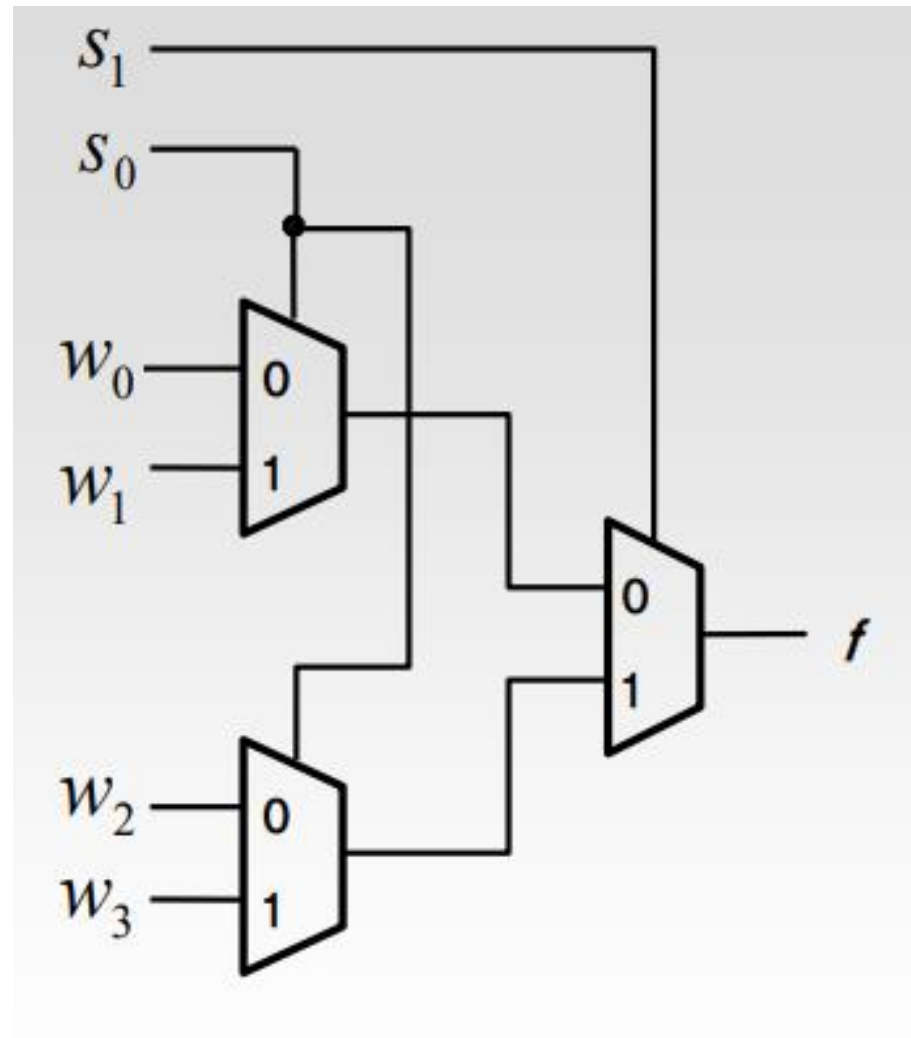
선택선			출력
S_2	S_1	S_0	F
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7



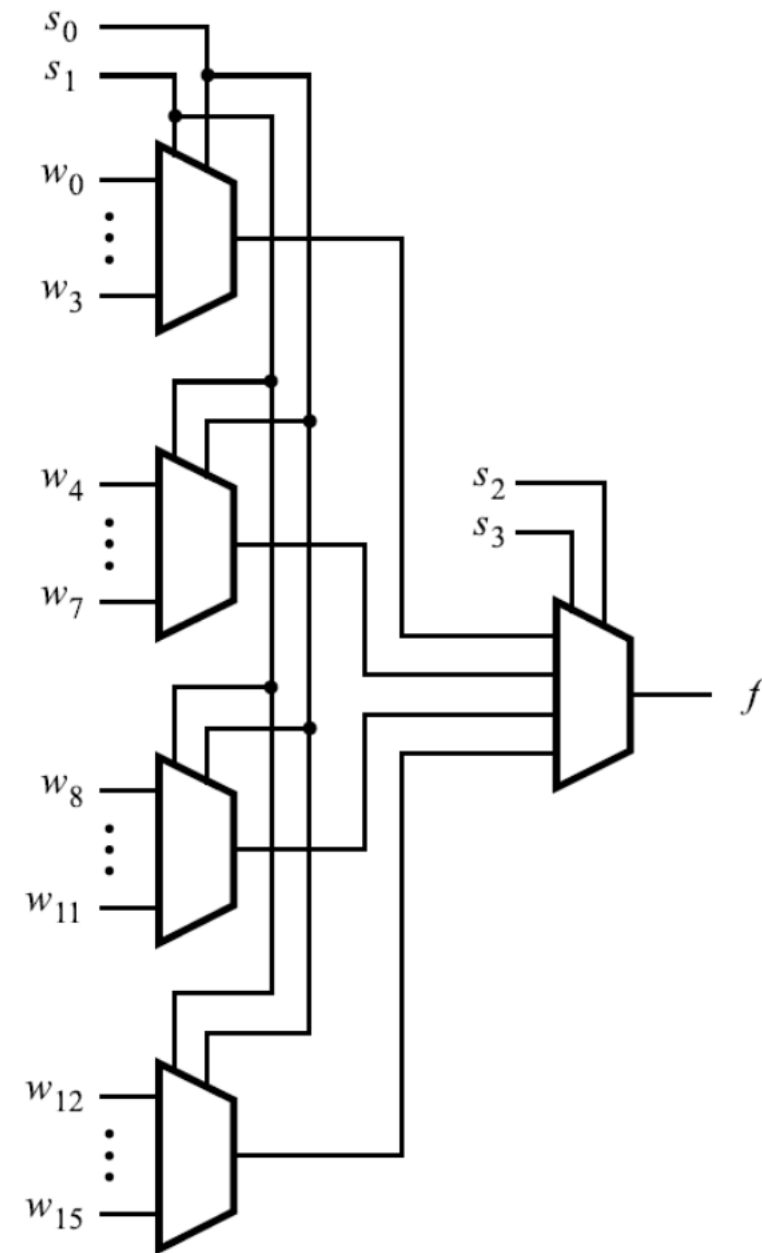
$$F = \overline{S_2}\overline{S_1}\overline{S_0}D_0 + \overline{S_2}\overline{S_1}S_0D_1 + \overline{S_2}S_1\overline{S_0}D_2 + \overline{S_2}S_1S_0D_3 + S_2\overline{S_1}\overline{S_0}D_4 + S_2\overline{S_1}S_0D_5 + S_2S_1\overline{S_0}D_6 + S_2S_1S_0D_7$$

: 8(=2³)개의 입력중의 하나를 출력으로 보내주는 조합논리회로

작은 MUX를 이용한 큰 MUX의 구현



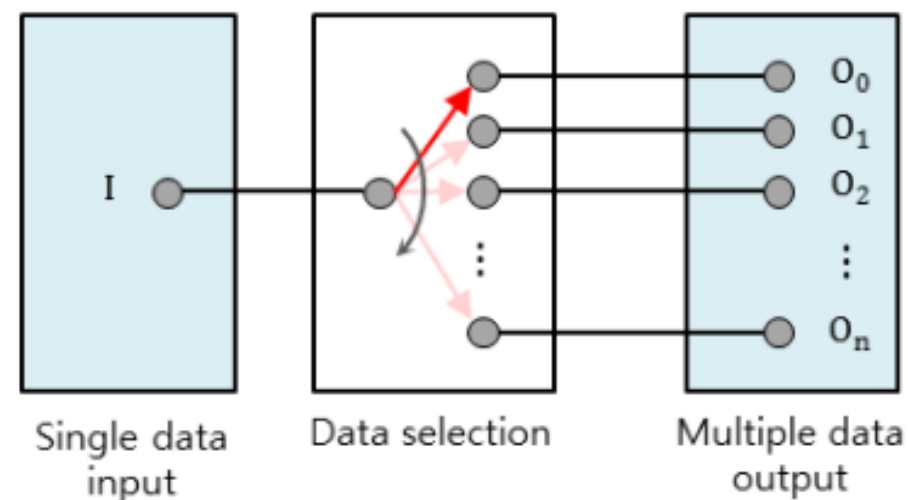
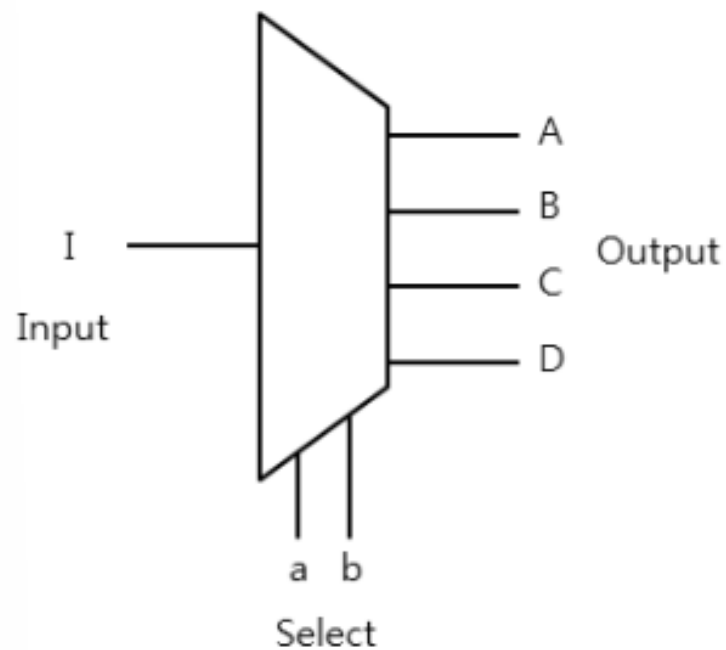
2×1 멀티플렉서를 이용한 4×1 멀티플렉서



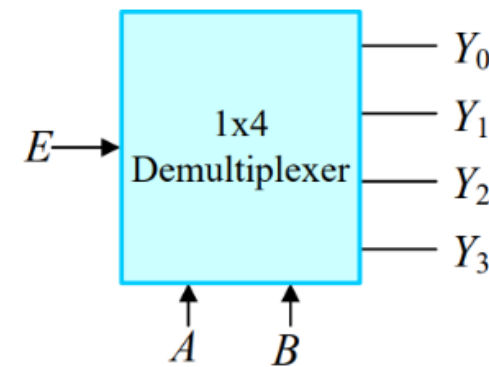
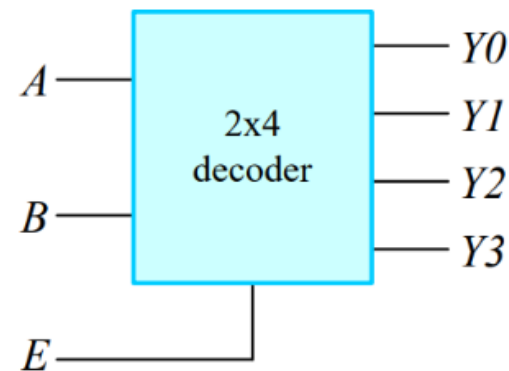
4×1 멀티플렉서를 이용한 16×1 멀티플렉서

DeMUX(Demultiplexer)

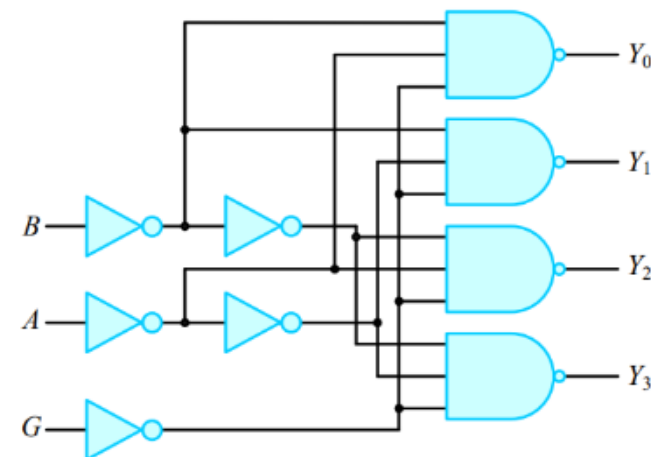
- 한 개의 입력을 어느 출력단에 내보낼 지 선택할 수 있는 기능을 가진 논리 회로
- 한 개의 데이터 입력에 대해 n 개의 선택 단자를 통해 2^n 개의 출력 단자 중 어느 단자에 출력할지를 선택하는 회로
- 멀티플렉서와 반대로 동작함



1 × 4 Demultiplexer



입력		출력			
<i>B</i>	<i>A</i>	<i>Y</i> ₃	<i>Y</i> ₂	<i>Y</i> ₁	<i>Y</i> ₀
0	0	1	1	1	<i>G</i>
0	1	1	1	<i>G</i>	1
1	0	1	<i>G</i>	1	1
1	1	<i>G</i>	1	1	1



- Enable 입력이 있는 디코더는 디멀티플렉서의 역할을 수행할 수 있음
- 디코더를 통해 디멀티플렉서를 구현할 수 있음

출처 및 팀원별 기여도

1. 인코더-디코더를 설명하는 이미지

<https://taehojo.github.io/deeplearning-for-everyone/auto-encoder.html>

2. BCD-to-Decimal Decoder 논리 회로, Usage

<https://learnabout-electronics.org/Digital/dig44.php>

3. 신종홍, 『컴퓨터 구조와 원리 2.0 비주얼 컴퓨터 아키텍처』, 2011

20161205 한윤도 (50%, 인코더, 디코더, BCD-to-Decimal Decoder)

20221572 심은 (50%, Multiplexer, DeMUX, 예시)