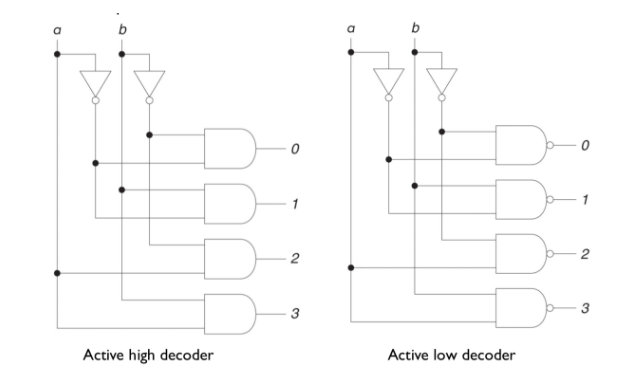
**9주차 예비보고서**

**전공: 생명과학과 학년: 4학년 학번: 20182186 이름: 김승원**

**1. decoder**

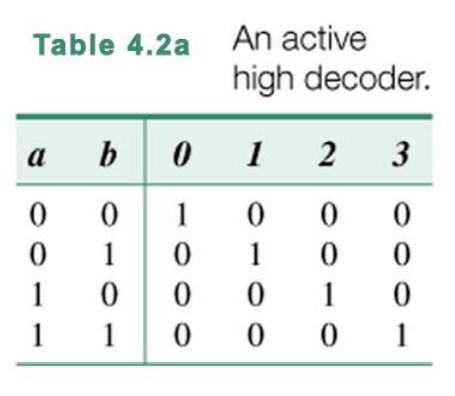
Binary Decoder의 개념은 입력신호를 받아서 여러 출력선들 중에서 하나를 선택하고자 할 때 사용한다.



**<fig 1. Active high decoder and active low decoder>**

위는 Active high decoder와 Active low decoder의 회로다. Active high decoder에서 AND gate를 NAND gate로 바꿔주면 된다.

**<Table 1. Active high decoder의 truth table>**



위는 active high decoder의 진리표다.

**<Table 2. Active low decoder의 Truth table>**

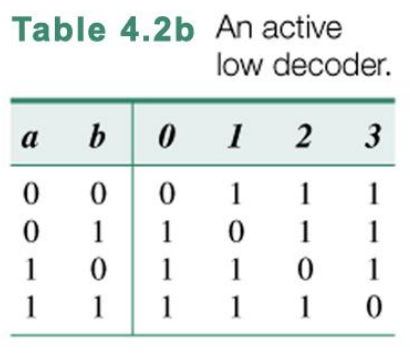
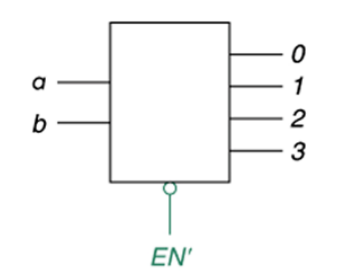
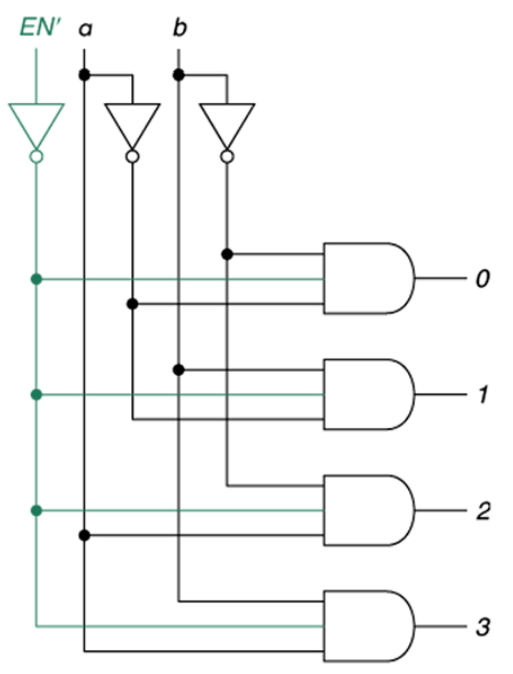


Table 2는 active low decoder의 진리표다. Active high decoder와 비교해서 반대의 값들을 가지는 것을 알 수 있다. 위의 진리표는 Input a, b를 받아서 0, 1, 2, 3중 하나의 출력선을 정해서 한 선만 1로 출력하는 시스템을 만들고자 한 것이다. Table 1에서 알 수 있듯이, Input a, b가 각각 0, 0 일 경우 0에 해당하는 출력선에서 1을 출력한다. Input a, b가 각각 0, 1일 경우 1에 해당하는 출력선에서 1을 출력한다. 그리고 2, 3에 해당하는 출력선도 input을 조절해서 한 출력선만 선택하는 것이 가능하다. Active low decoder의 경우는 반대로 생각하면 된다.



**<fig 2. Decoder with enable>**

위는 Decoder에 하나의 추가적인 선을 연결해서 일종의 스위치 역할을 할 수 있도록 하는 선을 추가한 것이다. EN(Enable)선이 0이냐 1이냐에 따라 decoder system이 동작할 수 있도록 할지 말지 결정할 수 있도록 하는 것이다. 만약 decoder system을 동작하고자 한다면 EN’의 input을 0으로 출력하고, 동작을 원하지 않는 다면 1을 input으로 하여 0~3까지의 모든 출력을 0으로 출력하도록 할 수 있다.

**<Table 3. Decoder with Enable>**

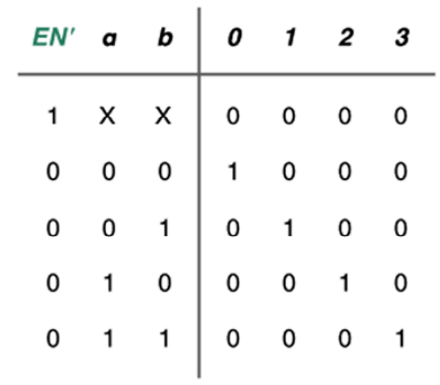


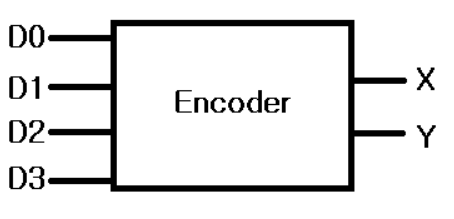
Table 3은 Enable선을 추가한 decoder의 Truth table이다. input값에 Enable 선만 추가된 형태로 EN’에 입력을 1을 한다면 모든 출력선이 0으로 출력되는 것을 확인할 수 있다.

**2. encoder**

Encoder는 2n개로 입력선 중에서 하나를 선택하여 그에 따라서 n개의 출력선이 나오도록 하는 것이다. 이를 통해서 data를 전달할 때 data를 받은 장치가 잘 알아들을 수 있도록 바꿔주는 역할을 한다.

**<Table 4. 4개의 input을 가지는 encoder의 truth table>**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| D3 | D2 | D1 | D0 | X | Y |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |

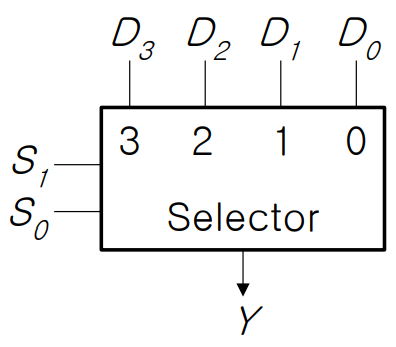


**<fig 3. Encoder의 graphic symbol>**

Encoder의 graphic symbol이다. Decoder와 다르게 decoder의 input과 output이 반대인 것처럼 보인다. 특이한 점은 Encoder의 경우 gate가 존재하지 않는다. 이는 간단하게 OR gate로 구성할 수 있기 때문에 IC로 제작할 필요가 없다. 또한 Encoder는 형태나 형식을 변환시켜서 표준화해주거나, 보안, 처리속도 향상이나 용량 압축을 위해 사용되기도 한다.

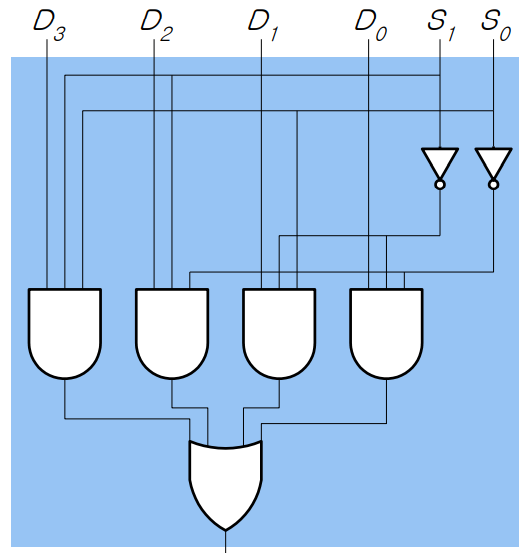
**3. Multiplexer**

Multiplexer는 여러 입력이 들어올 때 하나를 선택하고자 할 때 사용한다. 이러한 기능을 하여 data selectors라고도 하며, Mux라고 말하기도 한다. 즉, n개의 input을 받으면 1개만을 output으로 내고, log2n개의 select signal을 가지고 있다.



**<fig 4. 4x1 Mux의 graphic symbol>**

Figure 4는 4x1 Mux의 graphic symbol이다.



**<fig 5. 4x1 Mux의 logic diagram>**

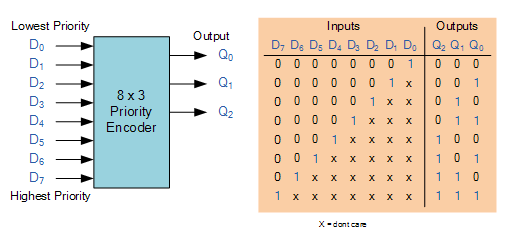
**<Table 5. 4x1 Mux의 truth table>**

|  |  |  |
| --- | --- | --- |
| *S1* | *S0* | Y |
| 0 | 0 | *D0* |
| 0 | 1 | *D1* |
| 1 | 0 | *D2* |
| 1 | 1 | *D3* |

Fig 5는 4x1 Mux의 logic diagram이고, Table 3는 4x1 Mux의 truth table이다. 4개의 data를 받아서 y로 출력되는 형태이다. Boolean expression으로는 Y=S1’S0’D0 + S1’S0D1 + S1S0’D2 + S1S0D3으로 표현할 수 있다. 이러한 Multiplexer를 사용하면 신호의 출력을 Demux()의 입력신호로 연결하여 여러 연결을 하나의 신호로 통합할 수 있어서 비용적인 측면에서 절약적이다.

**4. Priority Encoder**

Priority Encoder는 각 입력에 대해 우선 순위 레벨을 할당해서 사용한다. Priority encoder의 출력의 경우 우선 순위가 가장 높은 현재의 active input에 해당하는 결과가 나타나는 것이다. 즉, 우선 순위가 가장 높은 입력만이 입력되고 그 결과로 그에 해당하는 출력이 나타난다. 다양한 형태가 있으나 8-input priority encoder를 예시로 보자.



**<fig 6. 8-to-3 Bit Priority Encoder>**

가령 D2, D3, D5 입력이 들어오려고 한다면, 가장 높은 우선 순위를 가지는 D5가 적용되어서 Output으로 Q2, Q1, Q0에 각각 1, 0, 1이 출력될 것이다. 그리고 D5가 제거된다면 다음 순위인 입력 값으로 D3가 적용된다.

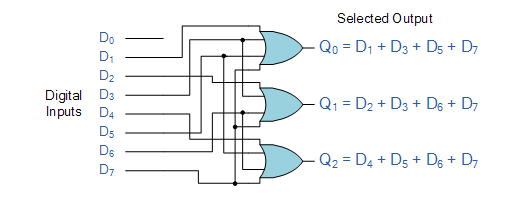
진리표에 따라 Q2, Q1, Q0를 Boolean expression으로 나타내면 다음과 같다.

Q0 = ∑(1, 3, 5, 7)

Q1 = ∑(2, 3, 6, 7)

Q2 = ∑(4, 5, 6, 7)

이를 통해서 OR gate를 사용해서 간단하게 Encoder로 구성할수도 있다.

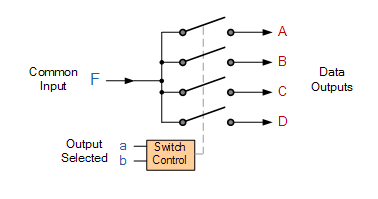


**<fig 7. Logic gate를 사용하여 만든 encoder(OR gate)>**

Fig 7은 priority encoder 출력 표현식을 이용해서 구현한 간단한 Encoder이다.

**5. DeMux와 Decoder의 차이**

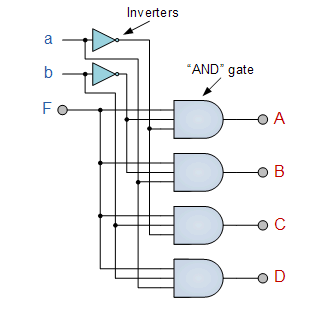
Demux는 Multiplexer의 반대되는 역할을 하는 회로이다. 하나의 단일 입력 data line을 취해서 한 번에 하나씩 여러 출력 라인 들 중 하나로 전환시킨다.



**<fig 8. 1-to-4 channel Demux>**

1-to-4 Demux를 Boolean expression으로 나타내면, F = (ab)’A + ab’B + a’bC +abD이다. 1-to-4를 기준으로 보면, 공통된 데이터 입력라인 4개 중에서 출력 데이터 라인 A~D 중 하나로 전환시키는 역할을 한다.

텍스트, 시계, 게이지이(가) 표시된 사진

자동 생성된 설명

**<fig 9. The Demux symbol, logic gates>**

Fig 8은 Demux의 symbol이다. a, b로 라인을 추가해서 입력이 선택된 출력으로 전달되는 것을 select하기 위해서 사용하기도 한다.

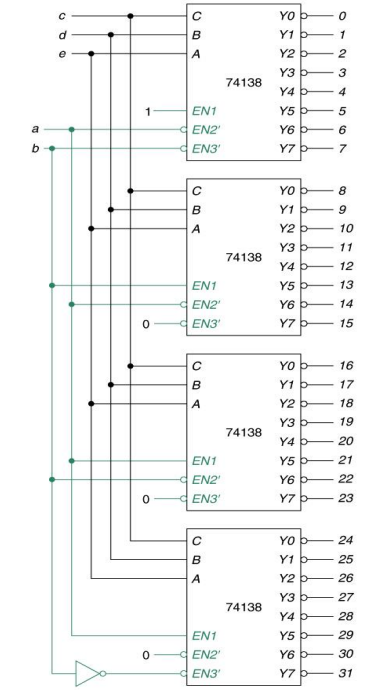
**<Table 6. Truth table (Demux)>**

텍스트, 옅은이(가) 표시된 사진

자동 생성된 설명

따라서 Demux와 Decoder의 차이는 Demux의 경우에는 입력 선택선을 통해 출력선을 결정해서 1개의 데이터 입력으로 들어온 신호를 내보낸다. 즉, 입력된 데이터를 어떤 선이 선택됐는지에 따라 출력을 결정하는 데이터 분배기로서 사용된다. 그리고 Decoder는 입력에 따라서 출력을 결정하는 것은 비슷한 역할을 한다고 볼 수 있으나, 출력이 하나가 1이면 나머지는 0이고, binary 입력 코드를 출력으로 변환시키는 해독기의 역할을 한다고 볼 수 있다.

**6. 기타이론**



**<Fig 10. 5x32 decoder>**

Figure 10은 3x8 decoder를 이어 붙여서 5x32 decoder로 만든 것이다. a와 b에 직접적으로 입력을 줘서 더 큰 decoder로 확장하는 것이 가능한데, 원래는 c와 d와 e로 3개 입력을 받아서 8개 중 하나를 선택하여 출력하는 decoder인데, a, b로 추가입력을 주면서 4개를 연결하여 32개 중 하나를 선택할 수 있는 decoder로 만든 것이다.

테이블이(가) 표시된 사진

자동 생성된 설명

**<fig 11. Decoder and minterms>**

Decoder의 경우 truth table 측면에서 봐도 그렇지만, 각각의 출력이 minterms에 해당한다는 것을 알 수 있다.