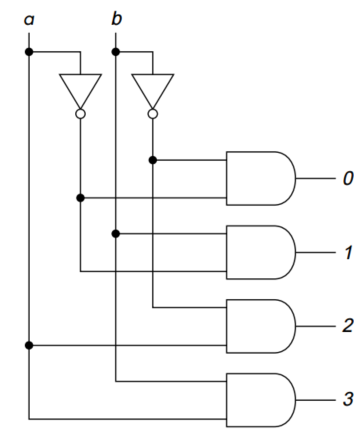
9주차 예비보고서

전공: 물리학과 학년: 3학년 학번: 20191286 이름: 김나현

1. Decoder에 대해 조사하시오.

디코더란 n비트에 입력에 대해 최대 2n비트의 출력을 갖는 논리 회로를 의미하고 encoding된 데이터를 이전으로 되돌리기 위한 과정을 수행하는 논리 회로를 의미합니다. 아래 <Figure 1>의Two-input (four-output) 디코더를 예시로 확인해보면 a와 b라는 1비트 입력 2개가 있을 때, 출력은 0, 1, 2, 3 네 개가 되고 ab가 00이면 0의 출력이 1이 되고, 01이면 1의 출력이 1이 되고, 10이면 2의 출력이 1이 되고, 11이면 3의 출력이 1이 되는 것을 알 수 있습니다.

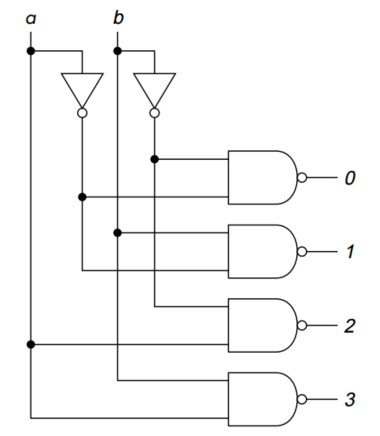


테이블이(가) 표시된 사진

자동 생성된 설명

<Figure 1> 2-input, 4-output active high decoder

<Figure 1>과 같은 decoder은 입력에 따라 하나의 출력만 1이 되고 나머지는 0이므로 active high decoder라고 부르고 반대로 ab가 00일 때 0만 0이고 나머지가 1이 되는 decoder는 active low decoder라고 부릅니다. Active low decoder은 다음 <Figure 2>와 같고, AND 게이트를 이용하여 구현된 high decoder와 달리 NAND 게이트로 구현된 것을 확인할 수 있습니다.

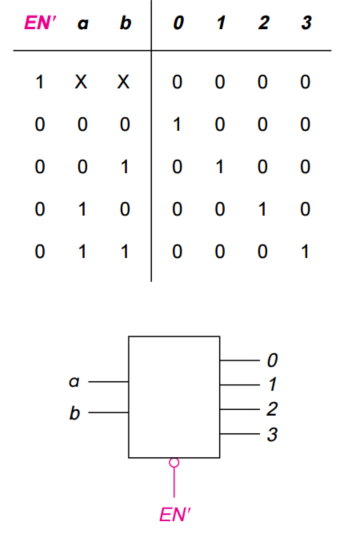
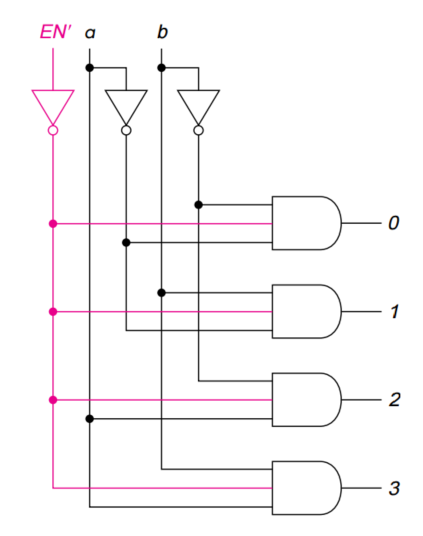


테이블이(가) 표시된 사진

자동 생성된 설명

<Figure 2> 2-input, 4-output active low decoder

대부분의 디코더에는 하나 이상의 enable 입력이 존재하는데 그러한 입력이 active되면 디코더는 위에서 언급한 것과 같이 동작하고 inactive되면 디코더의 모든 출력이 inactive됩니다. 이러한 enable 입력은 주로 active low인데 입력 EN’이 1이면 inactive를 의미하고 EN’이 0이면 active가 됩니다. 따라서 이러한 입력이 존재하는 디코더를 decoder with enable이라고 하고 회로로 나타내면 다음 <Figure 3>과 같습니다.

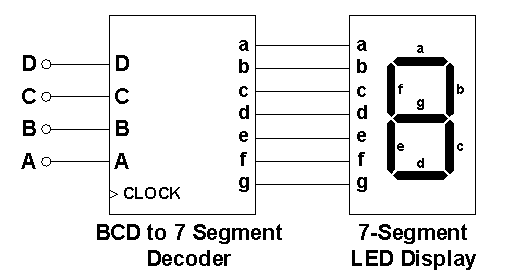


<Figure 3> 2-input, 4-output decoder with enable

위의 <Figure 3>의 진리표처럼 EN’이 1이면 모든 AND 게이트에 0이 인가되기 때문에 AND 게이트의 출력은 0이 되고, EN’이 0이면 디코더의 출력은 EN’에 영향을 받지 않고 a와 b에 의해서만 결정됩니다.

74138과 같이 8개의 출력을 갖는 디코더의 경우, active low인 8개의 출력과 3개의 enable 입력이 존재합니다. 이때, EN1은 active high이고 나머지 두 개, EN2’, EN3’는 active low이므로 디코더가 동작하기 위해서는 EN1이 1, EN2’과 EN3’이 0어야 하고, 그렇지 않으면 디코더는 inactive되어 모든 출력이 1을 나타내게 됩니다.

디코더는 각각의 주소를 가지고 있는 많은 장치들 중 하나를 고르는 역할을 하기도 합니다. 이때, 장치들의 각 주소는 디코더의 입력이 되고 하나의 출력이 active되면 해당 주소의 장치가 선택됩니다.



<Figure 4> BCD 디코더와 7-Segment Display를 연결한 모습

지난 실험에서는 7-segment display에 관해 설명하면서 위의 <Figure 4>와 같이 BCD의 4비트를 7비트로 나누어 주는 BDC 디코더에 대해 언급하였습니다. 4비트 이진수를 디코더를 거쳐 7비트로 바꿔준 후, 7-Segment Display의 입력으로 연결하면 입력 D, C, B, A가 의미하는 십진수를 7-Segment Display에서 육안으로 관찰이 가능합니다. BCD 디코더에 인가되는 입력과 7-Segment에 인가되는 a,b,c,d,e,f,g에 대해 진리표를 작성해보면 다음과 같습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

<Figure 5> BCD 디코더와 7-Segment Display의 진리표

위에서는 active low, high decoder를 통해 m개의 출력에 대해 하나의 출력만 0이나 1이 되는 경우를 설명하였지만 원하는 결과가 있다면 목적에 맞게 위의 <Figure 5>와 같이 여러 개의 출력이 1이나 0이 되도록 디코더 회로를 구현할 수도 있습니다.

1. Encoder에 대해 조사하시오.

Encoder는 디코더와 정반대 역할을 하는 논리 회로라고 말할 수 있는데, 데이터를 보안이나 축소의 목적으로 원래의 비트 수보다 작게 압축하는 논리 회로라고 설명할 수 있습니다.

만약, A0, A1, A2, A3 중 하나의 입력이 1일 때 다음 <Figure 6>과 같이 해당 encoder의 진리표를 작성할 수 있습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

<Figure 6> four-line encoder 진리표

A0이 1일 때는 Z0Z1이 00이 되고, A1이 1일 때는 Z0Z1이 01이 되고, A2가 1일 때는 Z0Z1이 10이 되고, A3가 1일 때는 Z0Z1이 11이 되는 것입니다. Z0와 Z1에 대해 논리 함수를 작성해보면 Z0=A2+A3이고 Z1=A1+A3입니다.

앞서 말했듯이 인코더는 디코더와 정반대 역할을 하므로 8 to 3 인코더와 3 to 8 디코더가 있다고 가정하면 다음과 같은 관계가 성립됩니다.

테이블이(가) 표시된 사진

자동 생성된 설명

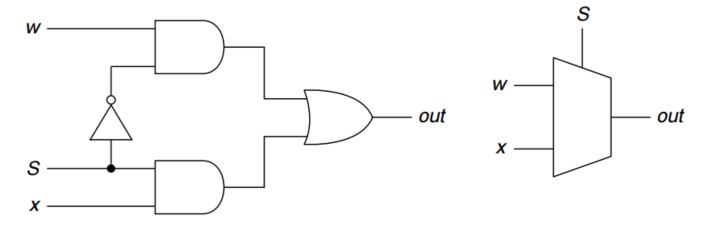
<Figure 7> encoder와 decoder의 관계

따라서, 인코더에서 암호화된 데이터들은 디코더를 거치면 원래의 입력으로 바뀌므로 해독되었다고 말할 수 있습니다.

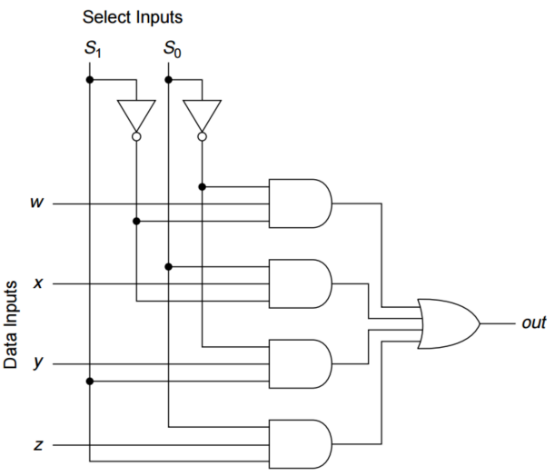
1. Mux(Multiplexer)에 대해 조사하시오.

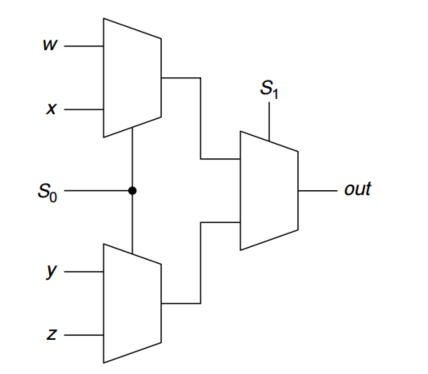
Multiplexer는 흔히 mux라고 줄여서 말하고, 입력을 모으는 역할을 하는, 여러 개의 입력 중 하나를 골라 출력으로 내보내는 스위치의 역할을 하는 논리 회로입니다.

다음은 Two-way 멀티플렉서와 그 기호를 나타낸 그림입니다. 이때, S가 0이면 출력은 w가 되고 S가 1이면 출력은 x가 됩니다.



<Figure 8> Two-way(2X1) multiplexer

Four-way 멀티플렉서는 AND와 OR 게이트를 이용하여 다음과 같이 나타내고 3개의 two-way 멀티플렉서를 이용하여 만들 수 있습니다.



<Figure 9> Four-way(4X1) multiplexer

S1S0이 00이면 w가 출력이 되고, 01이면 x가, 10이면 y가, 11이면 z가 출력이 됩니다. 이 회로는 디코더 회로와 매우 비슷하지만 여러 개의 입력 중 하나를 고르기 위해 OR 게이트가 있다는 차이가 있습니다.

74151은 1-bit 8X1 멀티플렉서로, 하나의 active low enable 입력, EN’이 존재하고 active high 출력과 active low 출력이 동시에 존재합니다. 입력 데이터는 A7부터 A0가지 있고, 출력은 Y와 Y’이 있고 select input에는 S2, S1, S0이 있습니다.

74153은 두 개(A, B)의 4X1 멀티플렉서를 포함한 것으로 각각 active low enable, ENA’과 ENB’을 가집니다. 첫 번째 멀티플렉서의 입력은 A3부터 A0까지고 그 출력은 YA이며 두 번째 멀티플렉서의 입력은 B3부터 B0까지고 그 출력은 YB입니다. 마찬가지로, 두 개의 select line, S1과 S0가 있고 같은 select 신호가 두 멀티플렉서 모두에 사용됩니다.

74157는 네 개의 2X1 멀티플렉서로, 하나의 active low enable 입력, EN’과 하나의 공통적인 select input, S가 사용됩니다. 각 멀티플렉서는 A, B, C, D로 이름이 붙여지고 첫 번째 멀티플렉서에 대해 입력은 A0, A1이며 출력은 YA입니다.

1. Priority Encoder에 대해 조사하시오.

테이블이(가) 표시된 사진

자동 생성된 설명Priority encoder는 하나 이상의 입력이 동시에 발생했을 때, 즉 하나 이상의 입력이 1을 나타낼 때 우선순위에 따라 출력을 결정하는 encoder을 의미합니다. 출력은 active된 입력 중 가장 우선순위가 높은 입력을 따라 결정되고, 그러한 우선순위는 일반적으로 입력의 순서에 대해 오름차순이나 내림차순으로 결정됩니다. 8개의 입력에 대해 오름차순으로 우선순위가 결정된다고 가정할 때 진리표는 다음 <Figure 10>과 같습니다.

<Figure 10> priority encoder의 진리표 예시

위의 <Figure 10> 진리표에서 출력 NR은 request가 없다는 것을 의미하고 NR이 1일 때는 입력들을 신경쓰지 않습니다. 따라서, NR는 NR=A0’A1’A2’A3’A4’A5’A6’A7’으로 나타낼 수 있습니다. 만약 7번째 장치가 active 신호를 갖는다면 출력은 7을 나타내는 이진수여야 하고 다른 입력들은 신경쓰지 않아도 됩니다. 위의 진리표에서 Z0, Z1, Z2에 대한 논리 함수도 구해보면 Z0은 Z0=A4+A5 +A6+A7이라고 나타낼 수 있고, Z1은 Z1=A6+A7+(A2+A3)A4’A5’이라고 나타낼 수 있고, Z2는 Z2= A7+A5A6’+A3A4’A6’+A1A2’A4’A6’라고 나타낼 수 있습니다.

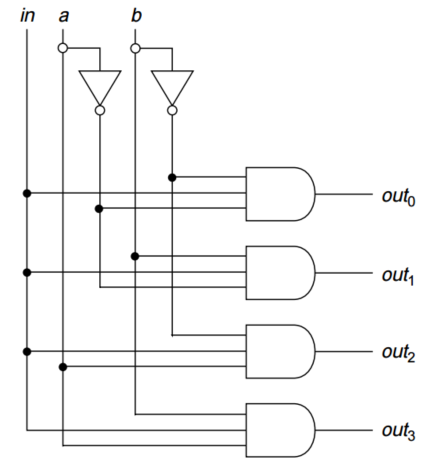
74147은 BCD encoder로, 9개의 active low 입력에 대해 4개의 active low 출력을 갖습니다. 입력은 1’부터 9’으로 번호가 매겨지고, 출력은 A’부터 D’가 됩니다. 74147은 9’이 가장 높은 우선순위를 갖는 priority encoder이기 때문에 1’부터 9’이 모두 1일 때는 inactive이므로 D’부터 A’ 역시 inactive되어 1111이 됩니다. 만약, 9’이 0이면 가장 우선순위가 높은 것이 active된 것이므로 다른 입력은 고려할 필요가 없고 D’C’B’A’가 0110이 됩니다. 74147의 진리표는 다음과 같습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

<Figure 11> 74147 priority encoder의 진리표

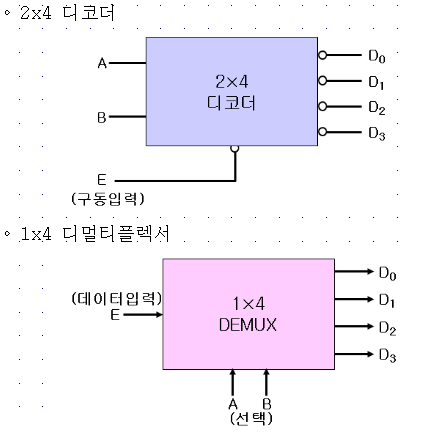
1. DeMux와 Decoder의 차이를 설명하시오.

Demultiplexer는 deMux라고 줄여서 말하며 encoder와 정반대 역할을 하는 decoder와 같이, 멀티플렉서와 정반대 역할을 하는 논리 회로입니다. Demux는 한 곳에서 발생한 신호를 여러 곳 중 한 곳으로 전달하는 역할을 합니다. 다음 <Figure 12>는 four-way demux로, a와 b는 신호가 가는 곳을 결정하는 select input이고 in은 입력을 의미합니다.

<Figure 12> four-way demux

위의 <Figure 12> 회로는 EN을 in 입력으로 대체한 four-way decoder와 같습니다.

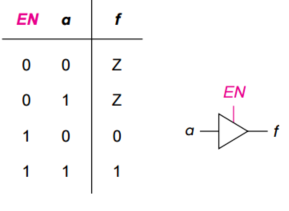
디코더는 n개의 입력에 따라 최대 2n개 중 하나의 출력을 결정하는 논리 회로이고, 디멀티플렉서는 위의 <Figure 12>의 in과 같은 하나의 입력을 받고, select input, a와 b를 입력 받아 a와 b를 통해 결정된 출력에서 in으로 들어온 입력 데이터를 출력하는 것입니다. 다음은 2X4 디코더와 1X4 멀티플렉서를 비교한 그림입니다.



<Figure 13> 2X4 디코더와 1X4 멀티플렉서

<Figure 13>과 같이 enable input이 있는 디코더는 멀티플렉서로도 활용할 수 있으므로 enable이있는 디코더와 멀티플렉서는 정확히 구분하기 어렵지만 주로 인코더를 통해 암호화된 데이터를 복구하는 목적으로 쓰이는 논리 회로는 디코더로, 멀티플렉서를 통해 여러 개의 입력 중 하나만을 선택해 출력으로 보낸 것을 다시 여러 개의 출력으로 만들기 위한 목적으로 쓰이는 논리 회로는 디멀티플렉서라고 말합니다.

1. 기타 이론

지금까지 다룬 논리 회로들은 입력과 출력이 모두 0과 1이었지만 0, 1 외에도 Z라는 하나의 상태가 더 추가된 Three-state gates(tristate gates)라는 논리 회로가 존재합니다. Three-state gate에서는 enable 입력이 하나 있고 이 입력이 active면 게이트는 원래대로 동작을 하고 이 입력이 inactive면 논리 회로의 output은 open circuit처럼 연결되지 않은 것처럼 동작하여 Z라는 상태로 표현됩니다.

<Figure 14> three-state gate

이때, Z는 하이 임피던트 상태를 의미하고 EN이 1이라는 것은 마치 스위치가 닫히는 것과도 같으므로 EN=1일 때, a의 값이 f로 전달되고 EN=0일 때는 스위치가 열려 있어 a의 값이 f에 도달하지 못한다고 생각할 수 있습니다.

Three-state gate는 EN가 active되면 a가 f로 전달되는 것이기 때문에 buffer라고도 부르고, 멀티플렉서를 구현하기 위해 three-state gate를 사용할 수도 있습니다.

시계이(가) 표시된 사진

자동 생성된 설명

<Figure 15> three-state gates를 이용한 2X1 멀티플렉서

위의 <Figure 15>에서 EN이 1이면 b가 출력되고 0이면 a가 출력되기 때문에 <Figure 15>는 2X1 멀티플렉서와 동일하다고 할 수 있습니다.