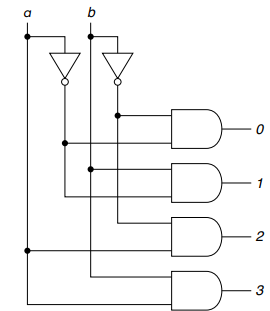
9주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

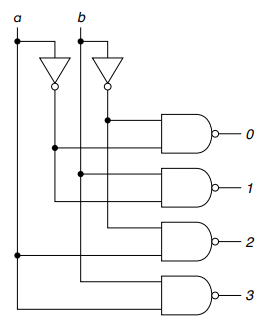
**1. Decoder**

디코더는 n비트의 이진코드 입력이 들어오면, 디코딩을 통해 최대 2^n비트의 출력을 할 수 있는 논리회로이다. 입력의 조합에 따라 활성화 되는 출력이 달라지며, 이처럼 입력 코드에 숨겨져있던 특정의미를 번역하는 과정을 디코딩이라 한다. 필요에 따라 출력 신호는 high-level과 low-level로 나뉜다. 아래그림은 순서대로 2 to 4 디코더의 active-high와 active-low를 나타낸다. 둘은 NOT의 관계에 있다.

테이블이(가) 표시된 사진

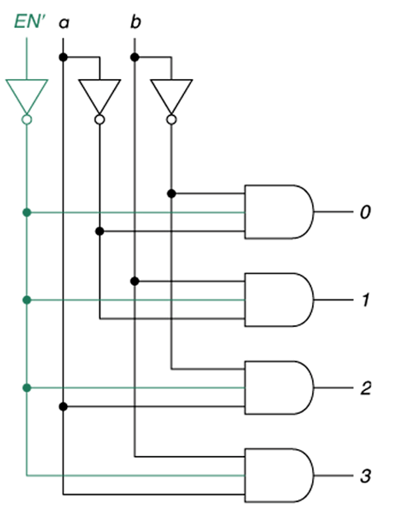
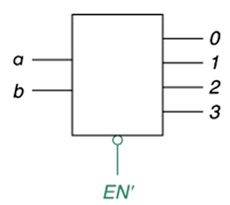
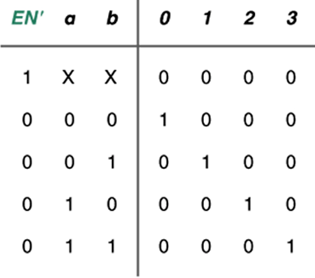
자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

일반적으로 active high와 active low의 회로 중 active low회로를 더 선호한다. 이는 active low회로가 노이즈에 강하며, 반응속도도 빠르기 때문이다. 또한, 회로 구성에 있어 AND게이트와 NAND게이트를 트랜지스터 래밸로 비교했을 때 앞선 주차에서 공부했듯 더 효율적이기 때문이다.

추가적으로 EN을 추가한 회로도 존재한다. EN은 enable의 약자로서 제어신호를 의미한다. en값이 0일 대 동작하며 1일 때 멈춤을 의미한다. 아래그림은 active high에 en을 추가한 모습이다.



칩의 종류로 74138, 74154, 74155가 있다. 또한, 디코더의 종류로 저번시간에 배운 7-Segment나 BCD , 3 to 8 디코더 등이 있다.

**2. Encoder**

인코더는 디코더와는 정반대의 기능을 하는 논리회로이다. 입력으로 최대 2^n비트를 받으며, n개의 출력을 한다. 십진수를 컴퓨터가 이해할 수 있는 이진수로 바꿔주는 것과 같이, 데이터를 특정 형식으로 변환하는 회로이다. 이런 특징으로 정보의 보안 또는, 압축에도 사용된다. 인코더는 OR게이트로 구성되기 때문에 IC로 제작되지 않는다.

종종 컴퓨터를 사용하면 오디오 및 동영상과 같은 곳에서 인코딩이라는 말을 많이 들었을 것이다. 오디오를 인코딩하여 압축하고 디코딩을 통해 원본을 볼 수 있다. 아래 표는 진리표이다.

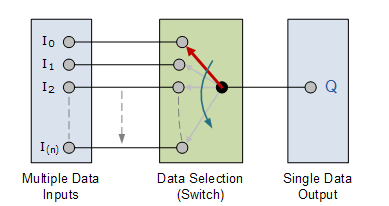
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A0 | A1 | A2 | A3 | Z0 | Z1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |

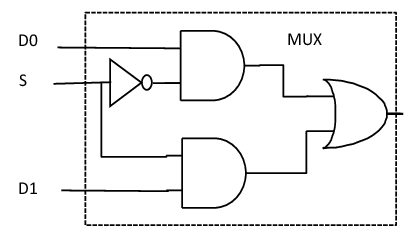
Z0 = A2 + A3 , Z1 = A1 + A3 이다.

**3. Mux(Multiplexer)**

전자제품에서 Mux는 이름처럼 Multiplexing을 수행한다. Multiplexing이란, 다중화를 말한다. Mux는 여러 아날로그나 디지털 입력 신호들 중 하나를 선택하여 하나의 단일 출력선으로 내보낸다. 보통 2의 거듭제곱인 특정 수의 데이터 입력값을 가진다. (만약, n개의 input이 들어오면 하나의 output과 log2 n 의 select signal을 가진다. ) Mux는 Data Selector switch 또는 Multi-way selector switch로도 불린다.

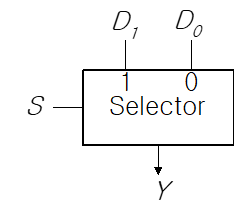
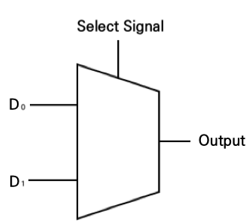
|  |  |
| --- | --- |
| Signal | Y |
| 0 | D0 |
| 1 | D1 |



 Mux는 Multi-throw swtich와 유사한 기능을 한다. 가장 간단한 형태인 2-to-1 Mux는 두개의 입력 데이터 채널 D0, D1이 있으며, 하나의 컨트롤 시그널 S가 있다. 이때, 진리표를 보고 표현식을 SOP형태로 작성할 수 있다. 진리표는 다음 표와 같이 작성 가능하다.

|  |  |  |  |
| --- | --- | --- | --- |
| S | D0 | D1 | Y |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

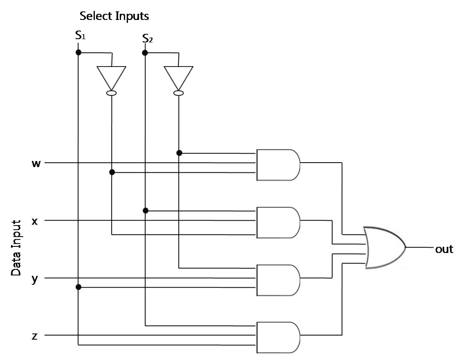
시그널 S가 0이라면 출력은 D0를 하며 반대로 1이라면 D1을 출력한다. 따라서, 불 표현식은 Y = S’D0D1 + S’D0D1 + SD0’D1 + SD0D1 이다. 이를 최소화 하면, Y = S’D0 + SD1 이 된다. Mux로 작성 시 더 간단한 형태의 회로 구현이 가능하다. 진리표는 다음과 같다. 보통 그림처럼 Mux는 사다리꼴로 주로 표현한다.

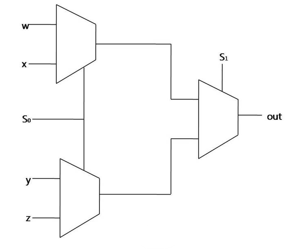
****

4x1 Mux는 3개의 2x1 Mux로 구현할 수 있다. 마찬가지로 SOP를 사용해 아래와 같은 회로와 불 표현식을 작성할 수 있다. 이러한 방식으로 디코더를 활용한 8x1 Mux 등 여러 Mux의 구현이 가능하다. 각 Mux의 Cascading을 통해 Multi-bit Mux의 구현이 가능하다.

|  |  |  |
| --- | --- | --- |
| S1 | S0 | Y |
| 0 | 0 | W |
| 0 | 1 | X |
| 1 | 0 | Y |
| 1 | 1 | Z |

Y = S1’S0’W + S1’S0X + S1S0’Y + S1S0Z

텍스트, 안테나이(가) 표시된 사진

자동 생성된 설명

Mux를 사용하는 목적은 효율성을 최적화하는 것에 있다. 특히 네트워크 최적화의 유용한 도구중 하나이며 과거엔 라디오와 같은 통신 채널의 최적화에 처음 사용되었다. 예로 여러 전화 통화를 하나의 단일 회선을 통해 관리가 가능하다.

**4. Priority Encoder**

Priority는 우선순위로 이름 처럼 인코딩하는 순서에 우선순위가 존재한다. 하나 이상의 입력이 동시에 발생할 경우(1이 여러 개일 경우)가 생길 때 사용한다. 즉, 각 입력값에는 우선순위가 주어지기 때문에 다음과 같이 Don’t care를 사용한 진리표를 작성할 수 있다. 일반적으로 내림차순 또는 오름차순으로 우선순위가 정해진다. (MSB or LSB가 최우선 비트가 된다.) NR은 동작여부를 제어했던 Enable과 비슷한 역할을 한다. A0이 high라면 출력은 0을 나타내는 이진수이다. 아래 표에서 우선순위는 7 > 6 > 5 > 4 > 3 > 2 > 1 > 0 순이다.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | Z0 | Z1 | Z2 | 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 |

NR=A0’A1’A2’A3’A4’A5’A6’A7’

Z0=A4+A5 +A6+A7

Z1=A6+A7+(A2+A3)A4’A5’

Z2= A7+A5A6’+A3A4’A6’+A1A2’A4’A6’

**5. DeMux / Decoder**

텍스트, 시계, 장치, 게이지이(가) 표시된 사진

자동 생성된 설명Encoder와 Decoder의 차이처럼 DeMux는 Mux와는 반대되는 개념이다. Data Selector로 불린 Mux와 달리 Data Distributor로 불린다. 한 개의 입력이 들어왔을 때 시그널에 따라 여러 개의 출력들 중 하나를 선택한다. (만약, 1개의 input이 들어오면 n개의 output과 log2 n 의 select signal을 가진다. ) 이런 특성으로 주로 데이터 수집, 데이터 전송 시스템 등 여러 분야에서 사용된다. DeMux는 Enable이라는 입력을 추가적으로 넣을 수 있다. 이는 켜짐과 꺼짐과 같은 역할을 한다. 4-to-1 Mux의 반대는 당연하게도 1-to-4 DeMux이다. (1x2 DeMux , 1x8 DeMux 등이 존재한다.) Mux와의 조합으로 통신 시스템의 효율성을 증가시킬 수 있다. 그러나, 지연이 발생할 수 있으며, 대역폭 낭비가 발생한다는 단점이 있다.

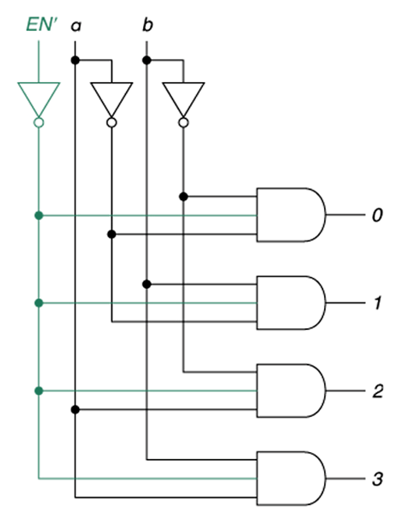
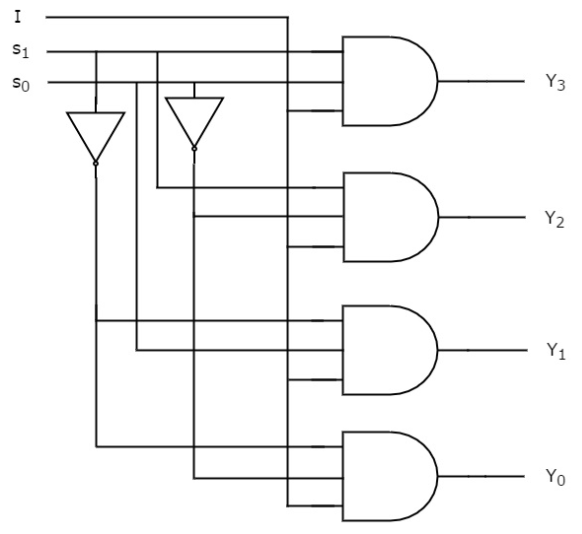
|  |  |  |
| --- | --- | --- |
| b | a | Out |
| 0 | 0 | A |
| 0 | 1 | B |
| 1 | 0 | C |
| 1 | 1 | D |

F = a’bA + ab’B + a’bC + abD

디코더와 DeMux의 차이는 입출력에 있다. 디코더는 앞서 설명했듯이 n개의 bit를 입력으로 하지만, DeMux는 1개 bit와 n개의 선택라인을 가진다. 또한, DeMux는 여러 출력 중 하나를 **선택**해 보내지만, 디코더는 디코딩된 출력을 **생성**해 내보낸다. 다음은 차이를 정리한 표이다.

|  |  |  |
| --- | --- | --- |
|  | DeMux | Decoder |
| 정의 | 하나의 입력, Select Signal로 여러 출력 중 하나로 전환하는 회로(라우팅) | 인코딩 된 데이터를 제어신호의 도움으로 디코딩 하는 회로 |
| Input | 1개의 데이터를 입력 | n개의 제어신호를 입력 |
| Out | Signal 개수에 따라 다름 | 2^n개 중 1개 |
| 유형 | 1 to 4 Demux, 1 to 8 Demux .. | 2 to 4 Decoder, 3 to 8 Decoder.. |
| Signal | bit | X |
|  | DeMux의 Select을 디코더의 입력라인으로 취급할 수 있다. DeMux의 입력은 디코더의 EN으로 간주할 수 있다. DeMux를 디코더로 사용할 수 있다. | |

아래 그림의 왼쪽은 EN이 추가된 디코더이며 오른쪽그림은 4-to-1 DeMux이다. 둘의 역할은 다르지만, EN이 추가된 디코더는 DeMux와 동일한 기능을 하는 것을 볼 수 있다.

**6. 기타이론**

Mux를 사용해 불 함수를 구현할 수 있다. 3개의 input으로 이루어진 전가산기의 Carry를 생각해보자. 진리표는 다음과 같이 작성할 수 있다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Terms | a | b | c | f |
| m0 | 0 | 0 | 0 | 0 |
| m1 | 0 | 0 | 1 | 0 |
| m2 | 0 | 1 | 0 | 0 |
| m3 | 0 | 1 | 1 | 1 |
| m4 | 1 | 0 | 0 | 0 |
| m5 | 1 | 0 | 1 | 1 |
| m6 | 1 | 1 | 0 | 1 |
| m7 | 1 | 1 | 1 | 1 |

기존에는 K-map을 사용해 Minimized 하였다. 이번에는 Mux를 사용하도록 한다. 우선, 8개의 input이 있다면, 가장 쉬운 8 to 1 Mux를 생각해 볼 수 있다. 시그널은 3개가 되며, 이는 a,b,c가 될 것이다. 또한, 입력값과 출력값이 동일해야 하므로 8개의 input은 차례대로 f가 될 것이다.

하얀색, 일, 더러운이(가) 표시된 사진

자동 생성된 설명 조금 더 나아가서 다음과 같은 방법이 있다. K-map을 그릴 때 처럼 표의 왼쪽에 MSB비트를, 그리고 테이블 상단에 나머지 변수를 순차적으로 쓴다. 이후, f가 1이 되는 부분을 체크한다. 한 열에서 모두 체크 된 경우 1이며 아닐 경우 0이 된다. 하나만 체크 될 경우 행에 맞는 a 또는 a’을 적는다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | b’c’(I\_0) | b’c(I\_1) | bc’(I\_2) | bc(I\_3) |
| a’ | 0 | 1 | 2 | 3 |
| a | 4 | 5 | 6 | 7 |
|  | 0 | a | a’ | 1 |

이제 I\_0 ~ I\_3 은 input이 되고 bc가 Signal이 된다.

**7. 참고문헌**

L. -h. Jiang and C. Huang, "Design and realization of multiplexer based on Schematic and VHDL Language," 2010 International Conference on Optics, Photonics and Energy Engineering (OPEE), 2010, pp. 273-276, doi: 10.1109/OPEE.2010.5508134.

디지털회로개론 chap4 Designing Combinational Systems 강의자료

Electronics-tutorials . (n.d.). <https://www.electronics-tutorials.ws/combination/comb_3.html>.