9주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231523 이름: 김민정

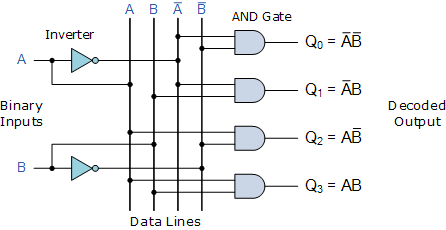
**1. Decoder**

Decoder은 input bit를 n개를 입력받으면 2^n개의 output을 제작하는 장치이다. 여기서 2^n개의 output중에 하나의 값만이 다른 결과를 보여준다. 만약 active high decoder라면 하나의 output만이 1이 되고 나머지 output은 0이 된다. 이와 반대로, active low decoder가 된다면 하나의 output만이 0이 되고 나머지 output은 1이 된다.

Decoder의 예시로 2-4 decoder에 대해 예시를 들어 설명해보겠다. 2bit를 입력으로 받으니 4개의 output으로 나올 수 있다. Output의 논리식은 아래와 같다. (active high decoder을 기준으로 논리식을 구하였다. 여기서 만약 active low decoder를 구하고 싶으면 모든 논리식 마지막 과정에 inversion을 넣어주면 된다.) f0 = a’b’, f1=a’b, f2=ab’, f3=ab으로 구할 수 있다. 아래는 active high decoder와 active low decoder의 truth table과 logic gate이다.

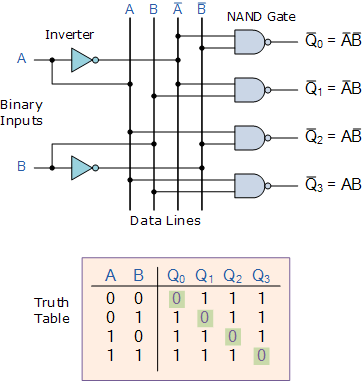
1) active high decoder truth table & logic gate

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | 0 | 1 | 2 | 3 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |



2) active low decoder truth table & logic gate

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | 0 | 1 | 2 | 3 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |



**2. Encoder**

Encoder은 input bit를 2^n개를 입력 받으면 n개의 output을 제작하는 장치이다. 즉 Decoder의 반대의 동작을 실행하는 장치임을 알 수 있다. Encoding은 일반적인 정의가 숫자로 표현한다 에서 알 수 있듯이 주소를 받으면 특정 값으로 표현하는 과정이다.

Encoder도 예시를 통해서 설명해보겠다. Decoder에서는 2-4 형식이었다. Encoder는 반대의 과정을 진행하니 4-2 형식이 된다. 4-2 encoder는 4개의 input을 받으면 0~3사이의 output을 도출하는 동작을 진행한다. 아래는 encoder의 truth table과 logic gate이다.

1) encoder truth table & logic gate

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3 | 2 | 1 | 0 | a | b |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | x | x |

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

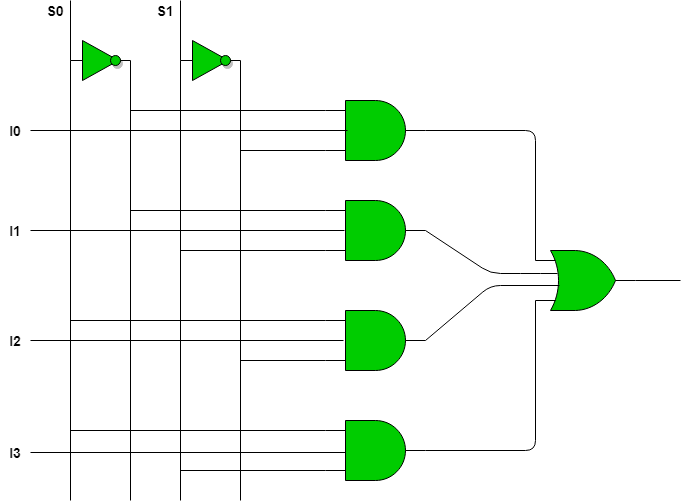
**3. Multiplexer (Mux)**

Mux는 여러 개의 input을 특정 신호에 따라 특정 output을 도출하는 장치이다. Mux는 2^n개 input이 있다면 n개의 특정 신호가 있다면, 모든 input을 output으로 도출할 수 있다. Multiplexer도 위의 encoder와 decoder와 같이 예시로 설명해보겠다.

4-1 multiplexer에 대해 설명해보자면 4개의 input이 존재하니 필요한 특정 신호의 수는 2개가 된다. 이를 S0, S1이라고 설정하면 00, 01, 10, 11일 때 input D0, D1, D2, D3들이 순서대로 output으로 나올 수 있게 된다. 이를 논리식으로 나타내면 f=S1’S0’D0 + S1’S0D1 + S1S0’D2 + S1S0D3이다. 아래는 4-1 multiplexer의 truth table과 logic gate이다.

1) multiplexer truth table & logic gate

|  |  |  |
| --- | --- | --- |
| S0 | S1 | Output |
| 0 | 0 | D0 |
| 0 | 1 | D1 |
| 1 | 0 | D2 |
| 1 | 1 | D3 |



**4.** **Priority Encoder**

출할 수 있다. Multiplexer도 위의 encoder와 decoder와 같이 예시로 설명해보겠다. Priority Encoder는 기존의 encoder에서 우선순위를 정하는 기능이 도입된 것이다. 만약 동시에 눌렀을 때 어떤 것을 먼저 반영할 것인지를 정하고 우선순위에 따라 입력을 적용하는 것이다. 만약 3>2>1>0 순이라면 3이 만약 1로 input이 들어오면 나머지 input은 고려를 하지 않고 3만을 반영하는 것이다. 아래는 기존의 4-2 encoder에 priority 기능을 추가한 것의 truth table이다

1) encoder truth table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3 | 2 | 1 | 0 | a | b |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | X | 0 | 1 |
| 0 | 1 | X | x | 1 | 0 |
| 1 | x | x | X | 1 | 1 |
| 0 | 0 | 0 | 0 | x | x |

**5. DeMux와 Decoder의 차이**

demultiplexer는 multiplexer의 반대의 동작을 수행하는 장치이다. Multiplexer가 여러 개의 input 중에서 하나만을 선택하여 output으로 도출하는 장치였다면. Demultiplexer는 하나의 input을 여러 개의 output으로 도출하는 장치이다. 이도 decoder와 같이 n개의 데이터가 input으로 들어가면 2^n개의 output을 만들어낸다. 이를 보면 Demultiplexer와 decoder의 차이를 알 수 없다. 둘 다 n개의 input이 들어오면 2^n개의 output을 만들어내는 것이 동일하다.

이 둘의 차이는 개념에서 나타난다. Decoder은 input이 암호가 들어오면 이를 해석한 결과를 output으로 도출하는 해독기이다. 예시를 들자면 상위 언어로 쓰인 언어를(코딩) 이진 코드로 출력시키는 과정을 도맡는다. Demultiplexer는 decoder의 상위 개념으로 input으로 하나의 데이터가 들어오면 이 데이터를 여러 데이터로 쪼개 output으로 도출하는 장치이다.

또한 이들의 데이터 선택 방식에 따라서도 차이가 발생하는데 decoder는 input 자체가 output을 결정한다. 구동입력이 존재하지만, 이는 그저 decoder의 동작을 수행할지 말지를 결정한다. 하지만 demultiplexer는 selection bit를 통해 input으로 들어온 데이터를 쪼개는 과정을 선택한다.

**6. 기타이론**

Decoder가 사용되는 예시 중 하나의 ROM에 대해 설명하겠다. ROM는 Read Only Memory의 약자로 오직 읽기만이 가능한 메모리이다. 즉 ROM은 해당 주소에 한번의 입력이 이뤄지고 나면, 이후 해당 주소의 내용은 바꿀 수 없게 된다. 내용을 바꿀 수 없다는 단점이 있지만, ROM에는 데이터가 어떠한 상황에서도 유지된다는 장점이 존재한다. 여기서 decoder가 어떻게 사용되는지 살펴보자.

Rom은 decoder와 memory array로 구성되어 있다. 만약 특정 숫자를 input으로 입력하면 이러한 input이 decoder를 통해 특정 output으로 도출되고, 이러한 output이 특정 memory array의 주소가 된다. ROM은 이 주소를 가지고 원하는 data를 읽고 output으로 도출한다. ROM에서 decoder는 특정 input에 해당하는 주소를 output으로 만들어내는 역할을 수행한다.