**9­­­주차 예비 보고서**

20150555 남민혁

**1. Encoder 에 대해 조사하시오.**

Encoder는 (2의 N승보다 작은) M 비트 입력으로부터 N개의 비트로 출력으로 구성되는 논리회로로서, 다양한 입력신호를 특정 코드로 변환하는 장치이다. 일반적 M to N Encoder로 구현되며, 2 to 1, 4 to 2, 8 to 3 encoder가 주로 활용된다. Encoder는 정보의 형태나 형식을 표준화하고, 보안과 처리 속도 향상 및 저장 공간 절약을 위해 형식을 변환하기 위해 사용된다.

다음은 4개의 입력을 받아 2개의 비트로 구성된 4개의 출력 값을 결정짓는4-to-2 encoder를 구현한 모습이다.

Table

Description automatically generated Diagram

Description automatically generated

좀 더 많은 8개의 입력을 받아 3개의 비트로 구성된 출력로 변환시키는 8-to-3 encoder는 다음과 같이 구현할 수 있다.

A picture containing table

Description automatically generatedDiagram

Description automatically generated

**2. Mux(Multiplexer) 에 대해 조사하시오.**

Multiplexer는 여러 제어 신호 (입력) 중에서 하나를 선택하여 선택된 입력을 전달하는 논리회로로서 스위치 박스의 집합으로 생각할 수 있다. 구체적으로 아래 그림과 같이 여러 개의 입력 (A-D)을 받아 control input (a, b)에 의해 하나의 입력이 선택되어Q로 그대로 출력된다. 이때 control bit를 이용하여 2의 N승개의 입력 중 고유한 하나의 결과만 선택해야 하므로 N개의 control input이 필요하다. 아래 예시에서는 4개의 입력 중 하나를 선택하기 위하여 2개의 control input이 사용되었다.

A picture containing clock, drawing, meter

Description automatically generated

가장 간단한 예로는 2개의 입력에 대해 하나를 선택하는 2-input multiplexer (74LS157)이 있으며, control input (S)의 상태에 따라 입력 2개 중 하나를 결정지어 출력한다. 즉, S가 HIGH일 때 I0의 입력 값에 상관없이 I1의 상태만을 출력으로 내보내고, S가 LOW 일 때는 I1의 입력 값에 상관없이 I0의 상태만을 출력으로 내보낸다. 구체적인 진리표와 구성된 논리회로는 다음과 같다.

Diagram

Description automatically generated Table

Description automatically generated

실제로 Multiplexer는 주 컴퓨터에 여러 단말기가 개별적인 통신회선으로 연결하면 비용과 관리측면에서 어려우므로, 각각의 단말기를 하나의 연결단자에 묶어서 하나의 통신회선으로 주컴퓨터에 연결해주는 장치로서 활용된다. 이를 통하여 하나의 통신 회선을 다양한 단말기와 공유함으로써 전송 효율을 높이고, 필요한 통신 회선의 수와 설치 비용을 줄일 수 있다.

**3. Priority Encoder 에 대해 조사하시오.**

Priority Encoder는 기존 Encoder에서 입력에 우선순위를 정하여 여러 개의 입력이 있을 때, 우선순위가 높은 입력값에 해당되는 출력 신호를 만드는 논리회로로 변형한 것이다. 이를 통하여 기존에 두개 이상의 비트에서 HIGH 입력이 들어왔을 때, 정확하지 않은 결과를 출력하였던 문제를 해결한 것이다.

구체적으로 기존의 Encoder는 입력에서는 (2의 N승 보다 적은) M개의 입력 비트에서 하나의 비트만 HIGH인 경우에 대해서만 출력 값을 결정하였다. 따라서 두개 이상의 비트가 1을 가질 경우에는 Don’t care로 구현되어 어떤 입력에도 정확하지 않은 결과가 출력되는 문제가 있었다. 그러나 Priority Encoder는 각 입력 비트에 우선순위를 부여함으로써, 두개 이상의 비트에서 HIGH 신호가 입력되었을 때 우선순위가 낮은 비트를 무시하는 방식으로 출력 값을 정하는 Encoder이다. 이를 통하여 기존의 Encoder에서 문제가 되었던 두개 이상의 입력에서 HIGH 입력을 받은 경우에 대한 ambiguity를 해결하였다.

다음은 8 to 3 Priority Encoder에 대한 진리표이다. 표현의 용이를 위하여 높은 숫자 비트가 높은 우선순위를 가지고 있다고 가정하자. Dn이 Dn-1보다 높은 우선순위를 가지기 때문에, HIGH 입력 중 가장 우선순위가 높은 것을 바탕으로 결과가 정해지는 모습을 확인할 수 있다. 즉, D7 비트에서 HIGH 입력이 있는 한 이진 코드는 111(2)로 7이 출력됨을 확인할 수 있다.

Table

Description automatically generated

**4. DeMux와 Decoder의 차이를 설명하시오.**

Demultiplexer와 Decoder는 크게 아래와 같은 차이가 존재한다.

|  |  |
| --- | --- |
| Demultiplexer | Decoder |
| * 1개의 입력과 N개의 스위치 입력 * 2의 N승 개의 출력단 * 1개의 입력 데이터가 N개의 스위치에 의해 결정된 출력단으로 입력 값이 그대로 전해진다. | * N개의 입력 * 2의 N승 개의 출력 * N개의 입력으로 구성된 코드를 바탕으로 본래의 입력 값으로 변환하여 출력한다. |

구체적으로 Demultiplexer와 Decoder는 다음과 같이 구현된다. 먼저 Demultiplexer는 Multiplex가 반대로 작동하는 논리회로로서, 하나의 입력을 N개의 control input 에 따라 어느 출력단으로 보낼지를 결정짓는 논리회로이다. 역시 2의 N승 개의 출력 단 중 하나를 결정짓기 위해서는 N개의 control input이 필요하다. 구체적으로 아래 그림과 같이 하나의 입력 (I)를 여러 개 출력단 (A-D)에 보낼지를 결정시키는 control input (a, b)이 추가적으로 존재한다. 이떄 4개의 출력단을 표현하기 위해서 2개의control input이 사용되었다.

A picture containing diagram

Description automatically generated

한편, Decoder 역시 Encoder가 반대로 작동하는 논리회로로서, N개 비트의 이진 코드를 입력에 대해서 최대 2의 N승 개 출력 중 하나로 결정하여 변환해주는 논리회로이다. 보통은 컴퓨터 내부에서 처리된 이진 코드를 사용자가 쉽게 이해할 수 있는 십진법으로 변환해주는 곳에 자주 활용된다. 구체적으로 다음은 2개의 입력을 받아 4개의 서로 다른 결과를 출력하는 가장 간단한 2-to-4 Decoder를 구현한 모습이다. A0과 A1으로 표현되어 있는 이진수를 십진법으로 변환하여 0부터 3중에 하나의 출력으로 내보내는 회로이다. (아래 회로는 low-active 방식으로 LOW 입력에 활성화되는 방식으로 구현되었다.)

A picture containing diagram

Description automatically generated

Diagram, engineering drawing

Description automatically generated

**5. 기타 이론**

**- Decoder를 활용한 다양한 조합회로 구현**

Decoder는 N개의 입력에 대해 가능한 모든 논리조합 2의 N승 개의 출력 종류에 대해 출력을 하는 Minterm (AND 게이트에 대해서), 또는 Maxterm (NAND 게이트에 대해) 로서, 각 출력에 대해 다음과 같이 고려할 수 있다.

A picture containing diagram

Description automatically generated

따라서 특정 출력이 mi, mj, mk의 논리곱인 경우 해당 출력을 AND게이트로, 반대로 논리합인 경우에는 OR 게이트로 decoder의 출력 결과를 활용함으로써, 동일 출력을 구현할 수 있다. 이 같은 방식은 decoder가 필요없는 min/max-term까지 만듦으로써 필요없는 연산을 진행하는 등 논리 회로를 최소화하여 최적화하지는 못하지만, 원하는 출력의 논리 식을 가지고 있는 경우 빠르게 만들 수 있다는 장점이 있다.

예를 들어 두개의 입력 (x, y)와 하나의 carry bit (z)를 활용한 Full adder에서 Sum은 1, 2, 4, 7 miniterm의 논리합, Carry는 3, 5, 6, 7 miniterm의 논리합이므로, 다음과 같이 구현할 수 있다.

Diagram

Description automatically generated

**- 7 Segment Display using Decimal to BCD Priority Encoder**

지난 주에 구현한 7 segment display 역시 decoder의 일종이라고 생각할 수 있다. 즉, 스위치 4개를 활용하여 4비트 입력을 제공하면, 각 디스플레이 세크먼트에 아리비아 숫자를 표현하기 위하여 정해진 출력 값을 가지도록 변환하기 때문이다.

더 나아가 BCD-to-7 decoder 앞에 Decimal-to-BCD Encoder를 구현할 경우 사용자로부터 스위치로부터 이진수로 구현된 입력이 아닌, 십진수를 그대로 입력으로 받을 수 있으므로, 좀더 사용자 친화적인 장치를 구현할 수 있다. 이 때 Decimal-to-BCD Encoder는 다음과 같이 구현할 수 있다.. 구체적으로 다음은 높은 값에 대해 우선순위를 가지는 Priority Encoder이며Low 입력값을 가질 때, 활성화되는 Low active로 구현하였다.

Table

Description automatically generated

**6. 참고문헌**

엠에스리, “엔코더 (Encoder), 디코더 (Decoder)”, <https://m.blog.naver.com/lagrange0115/220717414549>.

엠에스리, “멀티플렉서 (Multiplexer), 디멀티플렉서 (Demultiplexer)”, <https://m.blog.naver.com/lagrange0115/220716574603>

위키피디아, “Priority Encoder”, <https://en.wikipedia.org/wiki/Priority_encoder>.

위키피디아, “Encoder (digital)”, <https://en.wikipedia.org/wiki/Simple_encoder>.

위키피디아, “Multiplexer”, <https://en.wikipedia.org/wiki/Multiplexer>.

장영조, “디지털공학 및 실습 7장 강의자료: 조합 논리 회로”, <https://cms3.koreatech.ac.kr/sites/yjjang/down/dig_new/chap7.pdf>.