주차 보고서

전공: 수학과 학년: 4학년 학번: 20171273 이름: 심현우

1. **Decoder에 대해 조사하시오.**

Decoder는 encoder를 통해 encoding된 데이터를 encoding 전으로 돌리기 위해 이에 맞게 구성한 회로이다. 여기서 encoding에 대해서 알아야 하는데 encoding이란 보안이나 데이터의 크기를 축소하기 위해 데이터를 변환하는 작업을 의미한다. 이 encoding은 데이터 크기를 축소하는 것에 중점을 두어 영상이나 이미지의 크기를 줄이는 데 많이 쓰인다. 따라서 이렇게 encoding을 통해 변환된 데이터를 다시 원래의 데이터로 변환시켜 주는 회로를 decoder라고 한다. Decoder의 가장 기본적인 구성은 2 to 4 decoder이다. 이 decoder는 2개로 압축되어 온 데이터를 원래의 4개의 데이터로 변환시켜주는 회로이다. 따라서 input은 2개의 bit가 들어오고 output으로는 4개의 bit가 나온다. 2개의 input으로 총 4가지의 경우를 만들 수 있는데 이 4가지의 경우를 각각 output하나씩에 배정하는 것이다. Output을 Decoding0, D1, D2, D3로 구성한다. 예를 들어 input으로 00이 들어오는 경우를 생각해보자. 이는 D0만 1을 출력하도록 하여 input의 크기를 통해 output의 하나의 1출력을 배정한다. 이를 truth table로 작성하며 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| In A | In B | Out D0 | Out D1 | Out D2 | Out D3 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

이를 회로로 구성하기 위해 카르노 맵을 작성하면 다음과 같다.

-D0

|  |  |  |
| --- | --- | --- |
| A  B | **0** | **1** |
| **0** | **1** | **0** |
| **1** | **0** | **0** |

카르노 맵을 통해 확인했을 때 D0의 boolean식은 ~A&~B인 것을 알 수 있다.

2변수 카르노 맵은 간단하므로 나머지를 boolean식으로 구성하면 D1=(~A&B), D2=(A&~B), D3=(A&B)이다.

1. **Encoder에 대해 조사하시오.**

Encoder는 위에서 설명한 decoder의 이전에 encoding을 진행하는 회로이다. decoder에서 예를 들었던 2 to 4 decoder를 진행하기 전 encoder는 4 to 2 encoder이다. 이는 4개의 데이터를 압축하여 2개의 output으로 만들어주는 회로이다. 이 encoder를 priority encoder라 부르며 4번에서 설명하도록 하겠다.

1. **Mux(Multiplexer)에 대해 조사하시오.**

Multiplexer란 여러 개의 input중에 하나만 선택하여 출력을 연결시켜주는 회로이다. 이는 원하는 조건이 있을 때 input 중 한 개만 선택하여 원하는 출력을 내보내기 위해 사용한다. Input을 선택하기 때문에 데이터 선택회로라고 불리기도 한다. Mux의 기능을 위해 input을 선택하기 위한 controller가 필요하며 회로로 구성한다. 예를 들어, 9주차 실습에서 구현할 4 to 1 line multiplexer를 보자. 이는 4개의 input 중 하나를 선택하기 위해 2개의 input을 더 설정한다. 2개의 input으로 발생하는 4가지 경우를 이용하여 4개의 input 중 선택할 하나를 고른다. Input 4개를 A, B, C, D로 하고 선택을 위한 input 2개를 Selector0, S1로 구성한다. Output을 O라고 하자. 이를 진리표로 작성하면 다음과 같다.

|  |  |  |
| --- | --- | --- |
| In S0 | In S1 | Output O |
| 0 | 0 | A |
| 0 | 1 | B |
| 1 | 0 | C |
| 1 | 1 | D |

이를 회로로 구성하면 다음과 같다.

우선 A가 선택되는 경우를 보자. A가 선택되기 위해 (~S0&~S1)의 회로가 필요하다. 이 때, A가 (~S0&~S1)은 AND게이트로 연결하여 A가 1이고 (~S0&~S1)의 출력이 1일 때 O에서 1을 출력할 수 있도록 한다. 따라서 A&~S0&~S1이 된다. B와 C와 D에 대해서도 S0와 S1의 회로를 구성하면 B&~S0&S1, C&S0&~S1, D&S0&S1이다. 이를 각각 OR게이트로 연결하여 Output을 나타낸다.

1. **Priority Encoder에 대해 조사하시오.**

2번에서 설명한 priority encoder를 진리표로 작성해보자. Input으로 4개, A, B, C, D가 있고 output으로는 encoding을 한 압축된 2개의 bit, Encoding0, E1이 있다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| In A | In B | In C | In D | Out E0 | Out E1 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |

이는 위의 1번의 2 to 4 decoder와 반대되는 진리표를 가지고 있는 것을 확인할 수 있다. 이 4 to 2 encoder에 대한 카르노 맵을 작성해보자.

-E0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB  CD | **00** | **01** | **11** | **10** |
| **00** | **X** | **0** | **1** | **1** |
| **01** | **0** | **0** | **1** | **1** |
| **11** | **1** | **0** | **1** | **1** |
| **10** | **1** | **0** | **1** | **1** |

이를 prime implicant로 묶어 boolean식을 작성해보자. 3, 4열 1 8개(A)를 묶고, 1,4열 3, 4행 1 4개(~B&C)를 묶는다. 이를 Boolean식으로 나타내면, E0=A | (~B&C)이다.

-E1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AB  CD | **00** | **01** | **11** | **10** |
| **00** | **X** | **1** | **1** | **1** |
| **01** | **0** | **1** | **1** | **1** |
| **11** | **0** | **1** | **1** | **1** |
| **10** | **0** | **1** | **1** | **1** |

이를 prime implicant로 묶어 boolean식을 작성해보자. 3, 4열 1 8개(A)를 묶고, 2, 3열 1 8개(B)를 묶는다. 이를 Boolean식으로 나타내면, E1=A | B이다.

1. **DeMux와 Decoder에 대해 조사하시오.**

DeMux는 Demultiplexer로 multiplexer와 반대의 역할을 하는 한 input에 대해 다수의 output 중 하나와 연결시켜주는 역할을 한다. 따라서 여기서도 output 중 select를 수행해주기 위한 input이 있다. 위 3번의 예와 정반대 기능을 수행하는 이번 실습에서 구성할 1 to 4 line demux에 대해서 보자. 여기서도 output 4개 중 선택을 진행하기 위해 selector0, S1의 input을 두개 추가하여 4가지의 경우의 수를 만들 수 있도록 한다. 따라서 input으로 A, 하나의 데이터와, S0, S1 두개의 bit로 구성된다. output으로는 O1, O2, O3, O4가 있다. 이를 진리표로 작성하며 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| In S0 | In S1 | In A | Out O0 | Out O1 | Out O2 | Out O3 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

S0와 S1으로 O0부터 O3까지 어떤 output과 연결시킬지 선택하고 A의 데이터를 그대로 가져와 output으로 한다.

이를 회로로 구성하면 위의 multiplexer와 비슷하다. A의 데이터와 S0, S1을 구성하므로 O0의 경우 A&~S0&~S1인 것을 쉽게 알 수 있다. 나머지 O1=A&~S0&S1, O2=A&S0&~S1, O3=A&S0&S1인 것을 알 수 있다.

1. **기타 이론.**

위의 예시들 모두 4가지의 input또는 output을 다뤘다. 이를 더 다수의 input과 output을 다루도록 바꿀 수 있다. Decoder의 경우 input의 가짓수가 3가지가 되면 총 8가지의 경우의 수를 decode해야하므로 8가지의 output으로 구성해야 한다. 따라서 encoder의 경우도 총 8가지의 데이터를 3개의 output으로 압축하여 encoding할 수 있다. Encoder와 decoder를 보면 input과 output의 관계가 2의 n제곱의 관계를 갖는 것을 알 수 있다. Mux와 demux는 선택을 하기 위한 회로이므로 위의 4가지의 input과 output중 선택을 더 다수의 경우로 확장할 수 있다. 예를 들어, 8가지의 input과 output 중 선택을 하는 회로를 구성하고 싶으면 3개의 selector변수를 설정하여 3개의 input으로 8가지의 경우를 표현할 수 있으므로 8개 중 선택을 진행할 수 있다. 이는 8 to 1 line multiplexer, 1 to 8 line demux로 불린다.