9주차 결과보고서

전공 : 컴퓨터공학과 학년 : 3학년 학번 : 20211558 이름 : 윤준서

**1. 2 to 4 Decoder**

1-1. Active high

Active high 2 to 4 Decoder의 진리표와 카르노 맵은 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | d0 | d1 | d2 | 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 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a  b | 0 | 1 | a  b | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |

d0 d1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a  b | 0 | 1 | a  b | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |

d2 d3

이를 통해 논리식을 구하면 다음과 같다. (AND gate로 표현)

**d0 = a'b' d1 = a'b d2 = ab' d3 = ab**

논리식을 통한 schematic과 simulation의 결과는 다음과 같다.

도표, 평면도, 라인이(가) 표시된 사진

자동 생성된 설명

스크린샷, 사각형, 다채로움이(가) 표시된 사진

자동 생성된 설명

2-2. Active low

Active low는 Active high와 반대로 하나의 출력 비트만 값이 0이고 나머지 값은 모두 1이다.

Active low 2 to 4 Decoder의 진리표와 카르노 맵은 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | d0 | d1 | d2 | d3 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a  b | 0 | 1 | a  b | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |

d0 d1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a  b | 0 | 1 | a  b | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

d2 d3

이를 통해 논리식을 구하면 다음과 같다. (NAND gate로 표현)

**d0 = (a'b')' d1 = (a'b)' d2 = (ab')' d3 = (ab)'**

논리식을 통한 schematic과 simulation의 결과는 다음과 같다.

도표, 라인, 평면도이(가) 표시된 사진

자동 생성된 설명

스크린샷, 사각형, 다채로움이(가) 표시된 사진

자동 생성된 설명

**2. 4 to 2 Encoder**

4 to 2 Encoder의 진리표와 카르노 맵은 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | c | d | e0 | 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 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cd  ab | 00 | 01 | 11 | 10 | cd  ab | 00 | 01 | 11 | 10 |
| 00 | X | 0 | X | 1 | 00 | X | 0 | X | 0 |
| 01 | 0 | X | X | X | 01 | 1 | X | X | X |
| 11 | X | X | X | X | 11 | X | X | X | X |
| 10 | 1 | X | X | X | 10 | 1 | X | X | X |

e0 e1

이를 통해 논리식을 구하면 다음과 같다. (기본 식 -> AND gate로 표현)

**e0 = ab'c'd' + a'b'cd' -> b'd' e1 = ab'c'd' + a'bc'd' -> c'd'**

논리식을 통한 schematic과 simulation의 결과는 다음과 같다.

도표, 라인, 평면도이(가) 표시된 사진

자동 생성된 설명

스크린샷, 사각형이(가) 표시된 사진

자동 생성된 설명

**3. Encoder의 나머지 입력 형태**

Encoder의 입력 신호는 하나의 bit만 값이 1(혹은 0)이고 나머지 bit는 모두 0(혹은 1)이어야 한다. 이외의 값들이 특정한 값을 가질 경우 데이터 처리, 전송 과정에서 오류가 발생할 수 있기에 이외의 입력 신호들은 모두 don't care 값으로 처리된다.

Priority Encoder를 사용하면 이외의 입력 신호에 대해서도 출력 값을 생성할 수 있다. Priority Encoder는 말 그대로 우선 순위를 사용하기에, 사용하려는 입력 신호보다 우선 순위가 낮은 신호들은 값을 가지더라도 출력 값에 영향을 끼치지 않는다.

**4. Encoder 모든 입력 형태**

4 to 2 Priority Encoder의 진리표와 카르노 맵은 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E0 | E1 | NR |
| 0 | 0 | 0 | 0 | X | X | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 |

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

E0 E1

이를 통해 논리식을 구하면 다음과 같다.

**E0 = C + D E1 = BC' + D NR = A'B'C'D'**

NR은 입력 신호의 값이 모두 0일 때만 값이 1이다.

논리식을 통한 schematic과 simulation의 결과는 다음과 같다.

도표, 라인, 스크린샷, 그래프이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

해당 Priority Encoder의 입력 우선 순위는 D C B A 순으로 높다. 우선 순위가 높은 입력의 값이 1이면, 다른 우선 순위가 낮은 입력의 값이 1이더라도 출력에 영향을 끼치지 않는다. 즉 입력이 0110인 경우, C가 B보다 우선 순위가 높으므로 0010과 같은 입력으로 취급한다.

**5. BCD to Decimal Decoder**

BCD to Decimal Decoder의 진리표와 카르노 맵은 다음과 같다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a3 | a2 | a1 | a0 | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 |
| 0 | 0 | 0 | 0 | X | X | X | X | X | X | X | X | X |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | X | X | X | X | X | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X | X | X | X | X | X |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1a0  a3a2 | 00 | 01 | 11 | 10 | a1a0  a3a2 | 00 | 01 | 11 | 10 | a1a0  a3a2 | 00 | 01 | 11 | 10 |
| 00 | X | 1 | 0 | 0 | 00 | X | 0 | 0 | 1 | 00 | X | 0 | 1 | 0 |
| 01 | 0 | 0 | 0 | 0 | 01 | 0 | 0 | 0 | 0 | 01 | 0 | 0 | 0 | 0 |
| 11 | X | X | X | X | 11 | X | X | X | X | 11 | X | X | X | X |
| 10 | 0 | X | X | X | 10 | 0 | 0 | X | X | 10 | 0 | 0 | X | X |

b1 b2 b3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1a0  a3a2 | 00 | 01 | 11 | 10 | a1a0  a3a2 | 00 | 01 | 11 | 10 | a1a0  a3a2 | 00 | 01 | 11 | 10 |
| 00 | X | 0 | 0 | 0 | 00 | X | 0 | 0 | 0 | 00 | X | 0 | 0 | 0 |
| 01 | 1 | 0 | 0 | 0 | 01 | 0 | 1 | 0 | 0 | 01 | 0 | 0 | 0 | 1 |
| 11 | X | X | X | X | 11 | X | X | X | X | 11 | X | X | X | X |
| 10 | 0 | 0 | X | X | 10 | 0 | 0 | X | X | 10 | 0 | 0 | X | X |

b4 b5 b6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1a0  a3a2 | 00 | 01 | 11 | 10 | a1a0  a3a2 | 00 | 01 | 11 | 10 | a1a0  a3a2 | 00 | 01 | 11 | 10 |
| 00 | X | 0 | 0 | 0 | 00 | X | 0 | 0 | 0 | 00 | X | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 0 | 01 | 0 | 0 | 0 | 0 | 01 | 0 | 0 | 0 | 0 |
| 11 | X | X | X | X | 11 | X | X | X | X | 11 | X | X | X | X |
| 10 | 0 | X | X | X | 10 | 1 | 0 | X | X | 10 | 0 | 1 | X | X |

b7 b8 b9

이를 통해 논리식을 구하면 다음과 같다.

**b1 = a3'a2'a1' b2 = a2'a1a0' b3 = a2'a1a0**

**b4 = a3'a1'a0' b5 = a2a1'a0 b6 = a2a1a0'**

**b7 = a2a1a0 b8 = a3a0' b9 = a3a0**

논리식을 통한 schematic과 simulation의 결과는 다음과 같다.

텍스트, 도표, 평면도이(가) 표시된 사진

자동 생성된 설명

스크린샷, 사각형이(가) 표시된 사진

자동 생성된 설명

**6. Encoder / Decoder 의 응용**

다음 두 예시를 통해 Encoder와 Decoder의 응용의 차이점을 알아볼 수 있다.

1. 키패드

- Encoder : 사용자가 누른 숫자 키를 이진 코드로 변환한다. 예시로 패드에 숫자 키 6을 누르면, Encoder는 이를 0110으로 변환하여 전송한다.

- Decoder : 입력된 이진 코드를 사람이 읽을 수 있는 숫자로 변환해서 출력한다. 예시로 0110을 전달받으면 Decoder는 이를 숫자 6으로 변환하여 화면 상에 출력한다.

2. 메모리 시스템

- Encoder : 메모리 저장소가 여러 개일 때, Encoder는 각 메모리 블록에 대한 주소 신호를 이진 코드로 변환하여 메모리의 특정 위치를 가리킨다. 예시로 6번 메모리 블록을 선택하면 Encoder가 주소를 0110의 이진 코드로 변환하여 CPU에 전송한다.

- Decoder : 이진 코드로 변환된 주소 신호를 받아 특정 메모리 블록을 활성화하고 데이터를 읽는다. 예시로 CPU를 통해 0110이라는 이진 코드 주소를 전달 받으면 6번 메모리 블록을 활성화하여 데이터를 읽고 쓸 수 있게 한다.

**7. 4 to 1 line Mux**

4 to 1 line Mux의 진리표는 다음과 같다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | x | y | e1 | e2 | e3 | e4 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

이를 통해 논리식을 구하면 다음과 같다.

**e1 = ax'y' e2 = bxy'**

**e3 = cx'y e4 = dxy**

논리식을 통한 schematic과 simulation의 결과는 다음과 같다**.**

**도표, 텍스트, 라인, 평면도이(가) 표시된 사진

자동 생성된 설명**

**스크린샷, 컴퓨터, 사각형, 키보드이(가) 표시된 사진

자동 생성된 설명**

**8. 1 to 4 line DeMux**

4 to 1 line DeMux의 진리표는 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | y | f | e1 | e2 | e3 | e4 |
| 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 |

이를 통해 논리식을 구하면 다음과 같다.

**e1 = fx'y' e2 = fx'y**

**e3 = fxy' e4 = fxy**

논리식을 통한 schematic과 simulation의 결과는 다음과 같다**.**

도표, 평면도이(가) 표시된 사진

자동 생성된 설명

스크린샷, 텍스트, 사각형, 도표이(가) 표시된 사진

자동 생성된 설명

**1 to 4 line DeMux**를 이용하여 **4 to 16 Decoder**를 표현하는 과정은 다음과 같다.

• 진리표

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a3 | a2 | a1 | a0 | e0 | e1 | e2 | e3 | e4 | e5 | e6 | e7 | e8 | e9 | e10 | e11 | e12 | e13 | e14 | e15 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

• 코드

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

두 입력 a3와 a2를 사용하여 4개의 DeMux중 하나를 결정한다. 그리고 a1과 a0은 DeMux의 출력 결과를 저장한다. 4개의 DeMux 출력 결과를 e0 - e15에 각각 저장하면 모든 입력 조합에 대해 하나의 출력 값만 도출된다.

**9. 결과 검토 및 논의 사항**

2 to 4 Decoder는 2-bit의 입력이 주어졌을 때 4-bit 출력을 반환한다. Active high의 경우 4개의 출력 bit들 중 하나만 1이고 나머지는 모두 0이다. 반대로 Active low의 경우 4개의 출력 bit들 중 하나만 0이고 나머지는 모두 1이다. 실습에선 입력 변수 a가 MSB일 때 입력 값이 10진수로 0(00), 1(01), 2(10), 3(11)이면 출력 값은 d3가 MSB일 때 각각 0001, 0010, 0100, 1000 이다.

4 to 2 Encoder는 4-bit의 입력이 주어졌을 때 2-bit 출력을 반환한다. 변수 a가 MSB일 때 입력 값이 0001, 0010, 0100, 1000 이면 출력 값은 e1이 MSB일 때 각각 00, 01, 10, 11 이다.

위의 Priority Encoder의 입력 우선 순위는 D C B A 순으로 높다. 즉 A가 MSB일 때 입력이 1111인 경우, D가 제일 우선 순위가 높으므로 0001과 같은 입력으로 취급한다. 이를 통해 don't care 값을 사용하는 Encoder와 같은 출력 값을 얻는다.

BCD to Decimal Decoder는 8421 BCD 코드를 입력 값으로 받고, 9-bit 출력을 반환한다. Decoder와 마찬가지로 출력 값이 하나만 1이고 나머지가 0을 가지는데, BCD 값에 따라 해당 자릿수의 출력 bit가 1의 값을 가진다. 즉 9번째 bit인 b9가 MSB일 때 0111을 입력한 경우 7번째 bit인 b7 1의 값을 가지므로 001000000을 출력한다.

4 to 1 line Mux는 select signal x, y와 4-bit의 입력이 주어졌을 때 4-bit 출력을 반환한다. select signal은 입력 값에 따라 대응되는 값이 달라진다. 실습에서는 a와 대응되는 xy의 값은 00, b는 01, c는 10, d는 11임을 알 수 있다. 그리고 해당 입력 값에 대응되는 출력 값이 출력된다. 즉 e1이 MSB일 때 a와 대응되는 출력 값은 1000, b는 0100, c는 0010, d는 0001이다. 예시로 xy의 값이 01인 경우, 그와 대응되는 입력 b의 값이 1이어야 0100이 출력된다. 입력과 select signal, 출력이 모두 대응하지 않는 경우 0000을 출력한다,

1 to 4 line DeMux는 제어 신호에 따라 2-bit 입력과 4-bit 출력이 이루어진다. 실습의 제어 신호 f의 값이 0이라면 출력 값은 항상 0000이다. f의 값이 1인 경우에 Decoder와 마찬가지로 출력 비트 중 하나만 값이 1이고 나머지는 모두 0을 출력한다. f가 1일 때 01이 입력으로 주어진 경우, 그에 맞는 0100이 출력된다.

**10. 추가 이론**

앞서 다룬 Encoder와 Decoder 이외에 BCD, Mux, DeMux의 활용 분야는 다음과 같다.

**BCD**

- 디지털 시계, 계산기 : 7-Segment Display 등의 출력 장치에 BCD로 변환된 값을 Decoder를 통해 출력한다.

- 임베디드 시스템 : 센서가 인식하는 온도, 습도 등의 외부 데이터를 10진수 형태로 저장하고 처리한다.

- 데이터 통신 : 각 자릿수를 독립적으로 처리하는 BCD의 특성은 각 데이터 통신에서 각 자릿수의 오류를 효과적으로 검출할 수 있다.

**Mux**

- 메모리 선택 : CPU가 메모리에 접근할 때 Mux를 통해 원하는 메모리 블록을 선택해 데이터를 읽을 수 있다.

- 신호 선택 : 비디오 또는 오디오 등에서 여러 카메라 신호 중 하나를 선택해 출력할 수 있다.

- 데이터 선택 : 연산을 수행할 때 필요한 데이터를 선택하여 연산 장치에 전달할 수 있다.

**DeMux**

- 데이터 분배 : 네트워크 시스템에서 데이터를 여러 경로로 분배하거나 필요한 위치에 전달할 때 사용한다.

- 메모리 주소 선택 : CPU에서 데이터를 쓰거나 읽기 위해 특정 메모리 주소를 선택할 때 사용한다.

- 통신 시스템 : 특정 데이터를 여러 특정 채널로 분배할 때 사용한다.