9주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211522 이름: 김정환

**1.**

.................

먼저 2 to 4 Decoder에서 Active High인 경우에 대해서 진리표를 쟉성하면 아래와 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 바탕으로 카르노 맵을 작성하면,

텍스트, 친필, 도표, 디자인이(가) 표시된 사진

자동 생성된 설명

위와 같이 정리할 수 있고 이를 바탕으로 코드를 구현한 것은 다음과 같다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

이를 simulation한 결과는 아래와 같다.

스크린샷, 텍스트, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

이는 진리표에서 작성한 결과와 같은 결과를 얻는 것을 볼 수 있다.

다음으로 2 to 4 Decoder에서 Active Low인 경우에 대해서 진리표를 작성하면 아래와 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| In A | In B | Out D0 | Out D1 | Out D2 | Out 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 |

이를 바탕으로 카르노 맵을 작성하면,

텍스트, 도표, 스케치, 친필이(가) 표시된 사진

자동 생성된 설명

위와 같이 NAND Gate로 나타낼 수 있고 이를 코드로 작성하면,

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

위와 같다. 이를 Simulation하면,

스크린샷, 멀티미디어 소프트웨어, 사각형, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

위와 같은 결과로 진리표와 같은 결과를 얻는 것을 볼 수 있다.

................

**2.**

.......................

4 to 2 Encoder의 진리표를 작성하면 아래와 같다.

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

이를 이용하여 카르노 맵을 작성하면 아래와 같다.

텍스트, 친필, 스케치, 도표이(가) 표시된 사진

자동 생성된 설명

이를 이용하여 작성한 코드는 아래와 같다.

텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명

이를 Simulation한 결과는 아래와 같다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

Simulation을 확인하면 진리표와 결과가 동일한 것을 볼 수 있다.

.........................

**3.**

.......................

본 실험에서 4 to 2 Encoder에 대해서는 한 번에 하나의 핀만 1인 경우에 대해서만 유효하거나 하나의 핀만 0인 경우에 대해서만 유효하도록 회로를 구현하였다. 따라서 각각 4가지로 나눌 수 있는데 남은 12가지의 경우의 입력에 대해서는 무관항이라고 할 수 있다. 위에서 카르노 맵을 사용하여 식을 정리할 때도 해당 입력들은 무관항으로 처리하여 간소화 시에 함께 사용하였다. 해당 입력들에 대해서도 출력을 정상적으로 반환할 수 있도록 하려면 Priority Encoder를 사용하여 핀에 우선순위를 부여하여 출력하도록 하는 방법이 있다.

.........................

**4.**

.......................

우선 진리표로 정리해보면 아래와 같다.

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

이를 바탕으로 카르노 맵을 작성하면,

텍스트, 친필, 스케치, 도표이(가) 표시된 사진

자동 생성된 설명

위와 같이 간소화할 수 있다. 이를 Verilog 코드로 나타내면,

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

위와 같이 나타낼 수 있고, Simulation을 돌려보면

스크린샷, 사각형이(가) 표시된 사진

자동 생성된 설명

위와 같이 나오며 진리표와 같은 결과를 보이는 것을 볼 수 있다.

.........................

**5.**

.......................

BCD to Decimal decoder의 진리표를 작성하면 아래와 같다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | | Output | | | | | | | | |
| A3 | A2 | A1 | A0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 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 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

이를 바탕으로 카르노 맵을 작성하면 아래와 같다.

텍스트, 친필, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

텍스트, 친필, 폰트, 서예이(가) 표시된 사진

자동 생성된 설명

이를 이용하여 코드를 작성하면 아래와 같이 나타낼 수 있다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

이를 Simulation하면 아래와 같다.

스크린샷, 그래픽 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

이를 확인해보면 진리표와 동일한 결과가 나오는 것을 확인할 수 있다.

.........................

**6.**

.......................

Encoder는 주로 압축되지 않은 파일을 압축하거나, 암호화되지 않은 정보를 암호화하는데 주로 사용된다. Decoder는 그 반대의 역할을 수행하여 압축된 파일을 원래대로 돌리거나 암호화된 정보를 읽을 수 있도록 변환하는 역할을 수행한다. 대표적인 예시로는 컴퓨터에서 문자를 표기할 때 나타내는 ASCII 코드가 있다. 이는 각각의 문자를 대응하는 수로 변환하여 표기하는데 문자에서 숫자로 바꿔줄 때를 Encoder를 활용한다고 볼 수 있고, 그 숫자를 문자로 바꿔주는 것이 Decoder가 활용된다고 볼 수 있다. 이외에는 Base 64처럼 데이터를 모두 ASCII 영역의 문자들로만 이루어진 문자열로 바꿔주는 방식도 존재한다. 물론 이와 같은 인코딩/디코딩은 키를 이용한 것이 아닌 규칙에 따른 변환이므로 완전한 암호화라고 볼 수는 없다.

.........................

**7.**

.......................

4 to 1 line MUX는 아래와 같은 주어진 회로도를 바탕으로 코드를 구현하였다.

도표, 기술 도면, 평면도, 개략도이(가) 표시된 사진

자동 생성된 설명

이를 구현한 Verilog 코드는 아래와 같다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

이를 Simulation한 결과는 아래와 같다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

이 결과를 이용하여 진리표를 작성하였다. 이는 a, b값에 따라서 각각 나누어서 작성하였다. 우선 a = 0, b = 0인 경우의 진리표는 아래와 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| In a = 0  In b = 0 | In A | In B | In C | In D | Out X |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

다음으로 a = 0, b = 1인 경우의 진리표는 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| In a = 0  In b = 1 | In A | In B | In C | In D | Out X |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

다음으로 a = 1, b = 0인 경우의 진리표는 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| In a = 1  In b = 0 | In A | In B | In C | In D | Out X |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

마지막으로 a = 1, b = 1인 경우의 진리표는 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| In a = 1  In b = 1 | In A | In B | In C | In D | Out X |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

각각 a = 0, b = 0인 경우에는 A의 값과, a = 0, b = 1인 경우는 C의 값과, a = 1, b = 0인 경우에는 B의 값과, a = 1, b = 1인 경우에는 D의 값과 X가 동일한 것을 확인할 수 있다.

.........................

**8.**

.......................

우선 4 to 16 decoder의 진리표를 작성하면 다음과 같다. 소문자가 입력으로 주어진 핀, 대문자가 출력되는 핀들을 의미한다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| 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 |

실습에서 구현한 1 to 4 line deMUX의 구조를 확인하면 이와 같다.

도표, 기술 도면, 평면도, 라인이(가) 표시된 사진

자동 생성된 설명

이를 구현한 코드는 아래와 같다.

텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명

이를 이용하여 4 to 16 decoder를 구현하려면, 1 to 4 line deMUX에서 F로 들어오는 input에 대해서 각각 c, d에 대한 1 to 4 line deMUX가 연결되는 형태로 구현되도록 하는 것이라 할 수 있다. 이를 코드로 구현하면,

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

괄호 안에 묶인 a, b가 기존의 1 to 4 line deMUX, 그 뒷 부분이 각각에 묶인 c, d가 입력인 1 to 4 line deMUX라 할 수 있다.

이를 Simulation한 결과는 아래와 같다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

이 결과는 진리표와 동일하게 나오는 것을 확인할 수 있다.

.........................

**9.**

.......................

이번 실습에서는 decoder와 encoder를 구현할 때는 진리표를 작성 후 카르노 맵을 통해서 정리하는 과정으로 구현하였고, MUX와 deMUX는 강의자료에 주어진 회로도를 바탕으로 구현한 후에 나온 결과를 바탕으로 진리표를 작성하는 방식으로 진행하였다. 또한 BCD to decimal decoder의 경우에는 실습에서는 입력이 1001보다 큰 경우에 대해서는 모두 0으로 처리하여 진행하였으나 해당 항을 무관항으로 처리하여 식을 간소화할 때 이용할 수 있는 점에 대해서도 알 수 있었다. 그리고 1 to 4 line deMUX를 이용하여 4 to 16 decoder를 구현하면서 각각의 회로를 블록으로 생각하며 구현하는 과정에 대해서도 사고해볼 수 있었다.

.........................

**10.**

.......................

추가적으로 NAND Gate를 이용하여 MUX와 deMUX를 구현하는 방법에 대해서 조사해보았다. NAND로 다른 모든 Gate를 나타낼 수 있으므로 이 역시도 똑같이 나타낼 수 있는데, 우선 2 to 1 MUX를 NAND Gate의 회로도로 나타내면 아래와 같다.

도표, 라인, 폰트, 평면도이(가) 표시된 사진

자동 생성된 설명

이를 Verilog로 나타내보면 위에서의 In 1과 In 2를 각각 A, B, Sel을 a라 하면 아래와 같다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

deMUX의 경우에는 아래와 같은 회로도로 나타낼 수 있다.

도표, 라인, 기술 도면, 평면도이(가) 표시된 사진

자동 생성된 설명

이를 Verilog 코드로 구현해보면 아래와 같은 코드로 구현할 수 있다. 위의 회로도에서 선택에 영향을 주는 A는 a로, 입력인 D는 F로 나타내었고, 출력인 Y0, Y1은 각각 A, B로 나타내었다.

텍스트, 폰트, 친필, 서예이(가) 표시된 사진

자동 생성된 설명

.........................