9주차 결과보고서

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

**1. 2 to 4 Decoder 구현(+ truth table, K-map, simulation)**

2 to 4 decode는 2bit를 입력으로 받으니 4개의 output으로 나올 수 있다. Output의 논리식은 아래와 같다. (active high decoder, active low decoder을 모두 구현하였다. Active low decoder는 아예 처음부터 truth table을 작성 후 K-map을 통해 논리식을 구하거나, active high decoder의 논리식 마지막 과정에 모두 inversion을 넣어주면 된다.

1) active high decoder

가) truth table

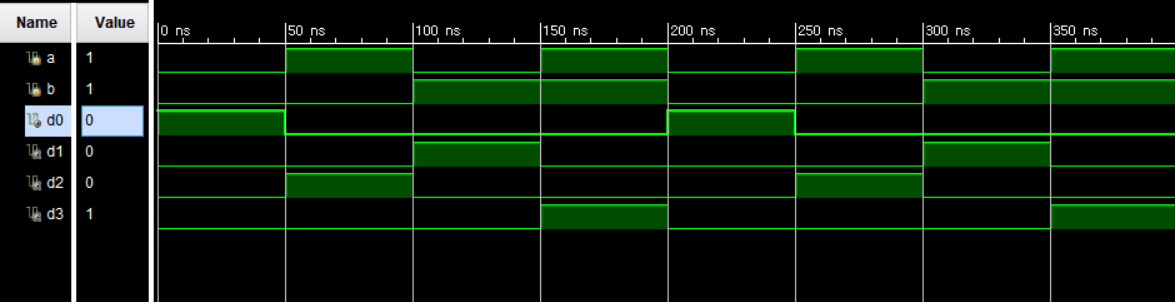
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | 0 | 1 | 2 | 3 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

나) K-map

도표, 텍스트, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

다) simulation



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 |

나) K-map

텍스트, 도표, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

다) simulation

스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

**2. 4 to 2 Encoder 구현(+ truth table, K-map, simulation)**

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 encoder를 truth table, k-map을 통해 논리식을 구하는 과정이다.

1) encoder truth table

가) truth table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | c | d | E0 | 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 |
| 0 | 0 | 0 | 0 | x | x |

나) K-map

위 truth table외의 진리 값은 모두 don’t care로 처리했다.

텍스트, 도표, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

다) simulation

스크린샷이(가) 표시된 사진

자동 생성된 설명

**3. 4 to 2 Encoder에서 4가지를 제외한 나머지 입력 형태**

Encoder의 input은 하나의 input이 1이고 나머지는 0이어야 한다. 그래서 4 to 2 encoder에서 4가지의 input만을 고려하고 나머지 12가지의 input은 나머지 입력 형태로 설정하여 보통은 don’t care로 처리한다. 하지만 이러한 don’t care input들은 목표하는 input과 output이 똑같아 설계자에게 혼동을 줄 수 있다. 그래서 이를 해결하기 위해 우선순위를 정하는 priority encoder을 설계한다. 이에 대한 것은 다음 챕터4에서 설명하겠다.

**4. priority encoder**

Priority encoder도 truth table과 K-map을 통해 이를 설계할 수 있다. 우선순위는 d, c, b, a 순으로 정했다.

가) truth table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | c | d | E0 | E1 |
| 0 | 0 | 0 | 0 | X | X |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

나) K-map

텍스트, 도표, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

다) simulation

스크린샷, 사각형이(가) 표시된 사진

자동 생성된 설명**5. BCD to Decimal decoder (+ truth table, K-map, simulation)**

BCD to Decimal decoder는 input 0~9값을 특정 output에 도출하는 장치이다. 이 또한 truth table, k-map을 통해 논리식을 구해볼 수 있다.

가) truth table

0~9 외의 값은 고려를 안 해도 되기에 don’t care로 둔다. X가 don’t care 표시이다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

나) K-map

텍스트, 도표, 평면도, 개략도이(가) 표시된 사진

자동 생성된 설명

다) simulation

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**6. decoder와 encoder 주요 응용**

Encoder는 여러 수의 입력을 하나로 더 적은 output으로 도출할 수 있는 장치이다. 이를 메모리 크기 관점에서 보면 원본파일을 더 작은 용량의 파일로 변환할 수 있음을 의미한다. 그러므로 파일 압축에서 사용될 수 있다 또한 통신보안, 음성 신호 처리 등에서도 사용된다.

Decoder는 하나의 input을 주면 수많은 output중에서 하나의 output을 고르는 장치이다. 이는 주소를 선택하는 방식으로 나타낼 수 있음을 의미한다. 그래서 ROM과 같은 주소를 찾는 장치에서 주로 사용된다.

**7. 4 to 1 line MUX (+code, simulation)**

Decoder는 하나의 input을 주면 수많은 output중에서 하나의 output을 고르는 장치이다. 이는 주소를 선택하는 방식으로 나타낼 수 있음을 의미한다. 그래서 ROM과 같은 주소를 찾는 장치에서 주로 사용된다. 4 to 1 line Mux는 4개의 input이 있으니 2개의 selection line이 필요하다. ( 2^n = 4 -> n = 2 ) 아래는 MUX의 truth table, code, simulation 결과이다.

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

가) code

|  |  |
| --- | --- |
| Design code | Testbench code |
|  | endmodule |

나) simulation

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**8. 4 to 16 decoder by 1 to 4 line deMUX (+ truth table, K-map, simulation)**

가) truth table(s0, s1은 selection code, input은 입력 데이터이다.)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S0 | S1 | Input | Out1 | Out2 | Out3 | Out4 |
| 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 |

나) code(+ k-map)

라인, 텍스트, 그래프, 도표이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| Design code | Testbench code |
|  |  |

다) simulation

**스크린샷, 텍스트, 사각형, 라인이(가) 표시된 사진

자동 생성된 설명**

**9. 결과 검토 및 논의 사항**

실습을 통해 encoder와 decoder 그리고 MUX, deMUX를 구현하면서 데이터의 크기를 줄이거나 늘릴 수 있는 방식을 탐구해보았다. 또한 encoder에서 don’t care의 문제점을 파악하고 이를 해결할 수 있는 방식인 priority encoder에 대해 고안했다.

**10. 추가 이론 조사 및 작성.(MUX, deMUX의 사용사례)**

MUX는 selection input을 통해 data input을 일부 선택할 수 있다. 이러한 특성을 통해 플레이어나 채널을 선택할 수 있는 CD 플레이어, DVD, 텔레비전 리모콘, 오디오와 같은 소스 제제어기에 사용된다.

deMUX는 selection input을 통해 데이터를 분산시킬 수 있다. 이러한 특성을 통해 서버나 모니터의 정보를 공유할 수 있는 장치나, 모니터 분배기에 사용된다.