8주차 결과보고서

전공: 수학과 학년: 4학년 학번: 20171273 이름: 심현우

1. **실험 목적**

7-Segment Display에 대한 개념을 이해하고, 각 7segment output에 대한 회로를 구성한다. 7segment display는 7개의 획을 통해 0부터 9와 a, b, c, d, e, f를 display로 출력하는 것을 의미한다. 이때 output 16개를 구성하기 위해 input으로 4개의 bit가 들어간다. 이를 직접 truth table로 구성하고 카르노 맵으로 boolean식으로 나타낸다. 이를 Verilog 코드를 통해 simulation 결과를 확인하고 FPGA와 연동하여 회로의 동작도 확인한다.

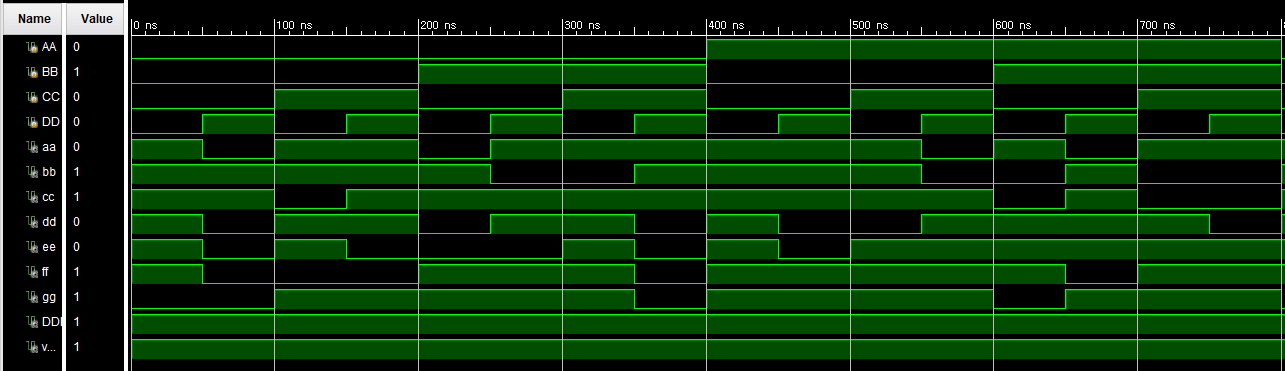
1. **7-Segment Display의 결과 및 simulation 과정에 대해서 설명하시오. (Truth table 작성 및 k-map 포함, 0~F,DP) (최대한 상세히 기술 할 것) simulation 등의 실험 결과는 본인이 직접 실험한 결과만 사용해야 함.**

텍스트이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명



우선 7segment의 동작을 먼저 분석해보자. Input 4개를 통해 16가지 0~F까지 출력을 만든다. 이때 출력은 7개의 segment로 직접 구성한다. 예를 들어, 0을 보자.

A

|  |
| --- |
|  |
|  |

이 7개의 segment에서 위부터 A로 시계방향으로 B, C, D, E, F 획으로 하고 가운데를 G로 한다. 0은 A, B, C, D, E, F의 출력을 1로 하여 display한다.

|  |
| --- |
|  |
|  |

위에 그림처럼 0을 표현할 수 있다. 이를 생각하여 각 input과 segment에 대한 truth table을 작성하면 다음과 같다.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | a | b | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

이외 DP라는 output도 7 segment를 구성할 때 사용해야 한다. DP는 segment의 display를 위해 항상 1을 출력하는 output이다. 따라서 DP의 Boolean 식은 DP=1이다.

나머지 segment 7개에 대한 카르노 맵을 작성해보자.

-a

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

Segment의 맨 윗 획, a에 대한 카르노 맵이다. 이를 prime implicant로 묶어 왼쪽 밑 4개의 1 (~A&C)과 그 오른쪽 4개의 1(B&C)을 묶는다. 또한 각 구석의 1 4개(~B&~D)를 묶는다. 오른쪽 3열 4열과 1행 4행의 1 4개(A&~D)를 묶는다. 이후 (2,2)의 1과 (2,4)의 1을 인접한 1과 묶는다. 이를 Boolean 식으로 구성하면 다음과 같다.

a = (~B&~D)|(A&~D)|(B&C)|(~A&C)|(A&~B&~C)|(~A&B&D)

-b

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

Segment의 오른쪽 윗 획, b에 대한 카르노 맵이다. 이를 prime implicant로 묶어 왼쪽 1열 4개의 1 (~A&~B)를 묶는다. 또한 각 구석의 1 4개(~B&~D)를 묶는다. 2행 3열 4열 두개의 1(A&~C&D)을 묶는다. 이후 (1,2)의 1과 (3,2)의 1을 인접한 1과 묶는다. 이를 Boolean 식으로 구성하면 다음과 같다.

b = (~A&~B)|(~B&~D)|(~A&~C&~D)|(A&~C&D)|(~A&C&D)

-c

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

Segment의 오른쪽 아랫 획, c에 대한 카르노 맵이다. 이를 prime implicant로 묶어 왼쪽 1, 2열 위 4개의 1 (~A&~C)를 묶고 아래 1 4개(~A&D)를 묶는다. 또한 2행의 1 4개(~C&D)를 묶는다. 2열의 1 4개(~A&B)를 묶고 4열의 1 4개(A&~B)를 묶는다. 이를 Boolean 식으로 구성하면 다음과 같다.

c = (~A&B)|(A&~B)|(~C&D)|(~A&~C)|(~A&D)

-d

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

Segment의 아래 획, d에 대한 카르노 맵이다. 이를 prime implicant로 묶어 왼쪽 1열 1, 4행 1 2개(~A&~B&~D)를 묶고 3행 1열 4열 1 2개(~B&C&D)를 묶는다. 4행 2, 3열 1 2개(B&C&~D)를 묶고 1행 3열 4열 1 2개(A&~C&~D)를 묶고 2행 2열 3열 1 2개(B&~C&D)를 묶는다. 이를 Boolean 식으로 구성하면 다음과 같다.

d = (~A&~B&~D)|(A&~C&~D)|(B&~C&D)|(~B&C&D)|(B&C&~D)

-e

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

Segment의 왼쪽 밑 획, e에 대한 카르노 맵이다. 이를 prime implicant로 묶어 3, 4열 3, 4행 1 4개(A&C)를 묶고 3열 1 4개(A&B)를 묶는다. 4행 1 4개(C&~D)를 묶고 각 구석 1 4개(~B&~D)를 묶는다. 이를 Boolean 식으로 구성하면 다음과 같다.

e = (~B&~D)|(A&B)|(C&~D)|(A&C)

-f

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

Segment의 왼쪽 위 획, f에 대한 카르노 맵이다. 이를 prime implicant로 묶어 3, 4열 3, 4행 1 4개(A&C)를 묶고 4열 1 4개(A&~B)를 묶는다. 1행 1 4개(~C&~D)를 묶고 1, 4행 2열, 3열 1 4개(B&~D)를 묶고 2열 1, 2행 1 2개(~A&B&~C)를 묶는다. 이를 Boolean 식으로 구성하면 다음과 같다.

f = (A&~B)|(~C&~D)|(A&C)|(B&~D)|(~A&B&~C)

-g

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

Segment의 가운데 획, g에 대한 카르노 맵이다. 이를 prime implicant로 묶어 3, 4열 3, 4행 1 4개(A&C)를 묶고 4열 1 4개(A&~B)를 묶는다. 2, 3행 3, 4열 1 4개(A&D)를 묶고 4행 1 4개(C&~D)를 묶고 1열 3, 4행 1 2개(~A&~B&C)를 묶는다. 2열 1, 2행 1 2개(~A&B&~C)를 묶는다. 이를 Boolean 식으로 구성하면 다음과 같다.

g = (A&~B)|(C&~D)|(~B&C)|(A&D)|(~A&B&~C)

-DP

DP=1

이를 Verilog 코드로 작성하여 simulation 결과와 truth table을 비교해보자. 둘의 결과가 일치하는 것을 확인할 수 있었다. 또한 이를 FPGA와 연결하여 Segment Display를 확인했다. SW버튼 4개를 사용하고 Digit 3의 7segment를 사용하여 확인했다. Digit3를 지정해주기 위해 view변수를 설정하여 digit3의 pin과 연결했다. View또한 항상 1로 설정하여 항상 digit과 연결할 수 있도록 했다. 결과를 확인했을 때 DP는 오른쪽 밑 점으로 항상 점등되어 있는 것을 확인할 수 있었고 SW의 버튼을 통해 0~F까지의 결과를 확인할 수 있었다.

1. **결과 검토 및 건의사항.**

작성한 truth table을 토대로 카르노 맵을 작성하여 각 segment에 대한 Boolean식을 구성했다. 또한 구성한 Boolean식에 대한 Verilog코드를 작성하여 simulation으로도 확인했다. 결과를 확인한 결과 둘이 일치하는 것을 확인할 수 있었고 카르노 맵을 통해 작성한 boolean식이 유효하다는 것도 확인할 수 있었다. 또한 각 segment에 대한 실제 display도 확인하기 위하여 FPGA와 연동하였다. 각 input pin에 대해서 16가지의 경우를 0000인 0부터 1111인 F까지 확인했다. 버튼을 각각 확인해 본 결과 잘 출력이 되었다.

1. **추가 이론 조사 및 작성.**

이번 실습에서 DP는 점등의 확인을 위해 항상 1을 출력하는 변수로 설정되었다. 하지만 이는 한자리의 7segment를 확장하여 자릿수를 증가시킨 7segment를 표현할 때 종종 소수점을 표현하기 위해 사용되기도 한다. 또한 7segment 이외에 표시의 다양성을 증가시키기 위해 14segment 또는 16segment가 쓰이기도 한다. 예를 들어 16segment를 보자.

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

위와 같이 총 16개의 획을 사용하여 표현의 다양성을 늘린다. 이는 각 16개의 segment를 각각 카르노 맵을 구성하여 회로를 만들어야 한다.

또한 7segment에서 input의 개수를 늘려 16가지보다 더 많은 경우를 표현할 수도 있다. 이를 이용하여 한글에서 자음, 모음이나 영어에서 알파벳을 표현할 수 있다.