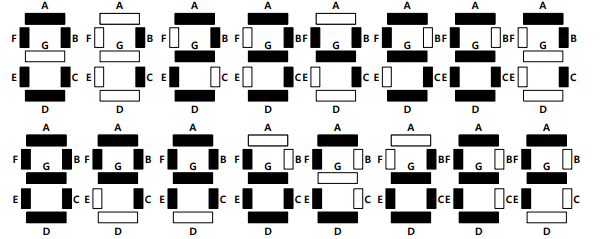
8주차 결과보고서

전공: 물리학과 학년: 3학년 학번: 20191286 이름: 김나현

1. 실험 목적

본 실험의 목적은 숫자 0부터 9까지와 영어 A부터 F까지, 즉 16진수를 나타낼 수 있는 7-segment display를 Verilog를 통해 구현하고 FPGA를 통해 구현된 결과를 확인해보는 것입니다.

1. 7-Segment display의 결과 및 simulation 과정에 대해서 설명하시오(진리표 작성 및 k-map 포함, 0~F, DP) (최대한 상세히 기술할 것).

<Figure 1> 7-segment display의 모습

텍스트, 화면이(가) 표시된 사진

자동 생성된 설명

<Figure 2> 7-segment display simulation 결과

<Table 1> 7-segment display의 진리표

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

<Table 2> a의 논리 함수를 구하기 위한 카르노맵

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

a=(B’C’D’)+(A’BD)+(AC’D’)+(AB’C’)+(ACD’)+(A’C)+(BC)

<Table 3> b의 논리 함수를 구하기 위한 카르노맵

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

b=(A’C’D’)+(A’CD)+(B’CD’)+(AC’D)+(AB’C’)+(A’B’)

<Table 4> c의 논리 함수를 구하기 위한 카르노맵

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

c=(A’C’)+(A’D)+(C’D)+(AB’)+(A’B)

<Table 5> d의 논리 함수를 구하기 위한 카르노맵

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

d=(B’C’D’)+(AC’D’)+(BC’D)+(B’CD)+(A’B’C)+(BCD’)

<Table 6> e의 논리 함수를 구하기 위한 카르노맵

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

e=(B’C’D’)+(AB)+(CD’)+(AC)

<Table 7> f의 논리 함수를 구하기 위한 카르노맵

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

f=(C’D’)+(A’BC’)+(AB’)+(ACD)+(BCD’)

<Table 8> g의 논리 함수를 구하기 위한 카르노맵

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

g=(A’BC’)+(AB’)+(AD)+(A’B’C)+(CD’)

dp=a+b+c+d+e+f+g

<Figure 1>과 같이 4비트를 이용해 7-segment 상에서 0부터 F까지 16진수를 표현하기 위해 a, b, c, d, e, f, g의 7개의 획의 논리 함수를 구하기 위해 위의<Table 2>부터 <Table 8>까지 카르노맵을 그렸습니다. 각 획과 dp에 대한 논리 함수를 구해보면 a는 a=(B’C’D’)+(A’BD)+(AC’D’)+(AB’C’)+ (ACD’)+(A’C)+(BC)이고, b는 b=(A’C’D’)+(A’CD)+(B’CD’)+(AC’D)+(AB’C’)+(A’B’)이고, c는 c=(A’C’)+(A’D) +(C’D)+(AB’)+(A’B)이고, d는 d=(B’C’D’)+(AC’D’)+(BC’D)+(B’CD)+(A’B’C)+(BCD’)이고, e는 e=(B’C’D’)+ (AB)+(CD’)+(AC)이고, f는 f=(C’D’)+(A’BC’)+(AB’)+(ACD)+(BCD’)이고, g는 g=(A’BC’)+(AB’)+(AD)+ (A’B’C)+(CD’)이고 dp는 dp=a+b+c+d+e+f+g가 됩니다.

1. 결과 검토 및 논의 사항

Simulation 결과를 보고 진리표를 작성하여 Verilog 코딩이 잘 되었는지 확인하는 것보다 FPGA에 연결하여 0부터 15까지 switch를 조정하여 7-Segment로 나타나는 것을 보고 Verilog 코딩이 잘 되었는지 확인하는 것이 훨씬 편리하였습니다. 카르노맵으로 논리 함수를 작성할 때, 모든 prime implicant과 essential prime implicant을 작성해야 7-segment 상에 오류가 없이 나타났는데 논리 함수를 작성할 때 몇 개의 implicant를 빼고 작성하였더니 7-segment로 숫자를 확인할 때 불이 들어오지 않는 획이 존재하기도 하였습니다. 이를 통해 카르노맵을 사용하여 논리 함수를 작성할 때는 반드시 essential prime implicant가 커버하지 못하는 minterm들을 커버할 수 있는 non-essential prime implicant를 모두 써줘야 한다는 것을 알 수 있었습니다.

1. 추가 이론 조사 및 작성

4비트 출력이 있는 논리 회로에서 verilog 코딩이 잘 되었는지 확인하기 위해 simulation을 사용하여 진리표로 작성하여 올바른 결과가 나오는지를 확인하는 것보다 디코더를 이용해 A, B, C, D의 출력을 a, b, c, d, e, f, g의 출력으로 변환한 후, 7-segment를 통해 육안으로 출력을 바로 확인하는 방법이 편리할 것 같다는 생각이 들었습니다. 따라서, 4비트의 출력을 얻는 논리 회로에서는 디코더와 7-segment를 이용하여 FPGA를 통해 회로 구현이 제대로 되었는지 확인하는 것이 효과적일 거 같다는 결론을 내릴 수 있었습니다.