8주차 결과보고서

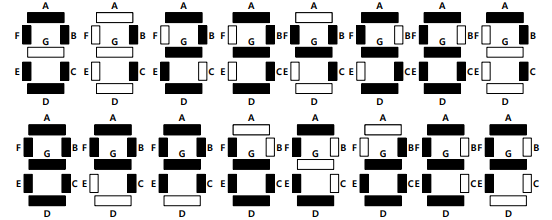
전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

**1. 목적**

우선, 7-Segment Display가 무엇인지 개념을 이해한다. 이해한 바를 바탕으로 진리표를 작성하고, k-map을 통해 함수를 얻는다. 함수를 바탕으로 Verilog를 사용해 직접 구현해 본다. 이후 시뮬레이션으로 회로의 구현이 생각한 대로 동작하는 확인한다. FPGA보드를 통해 최종적으로 구현된 회로의 동작을 시뮬레이션 및 진리표와 동일하게 동작하는지 확인해 본다.

**2. 실험결과 및 과정**

7-Segment Display는 7개의 LED를 이용해 숫자 또는 영문자를 화면에 표시해주는 FND 장치이다. 이때 실습에서 할 7-Segment는 아래 그림처럼 4개의 input으로 7개의 output을 출력한다. 또한, 추가적으로 소수점을 표현하는 Dp(Dot point)가 있지만, 이는 항상 켜지도록 하여 값을 1로 하였다.



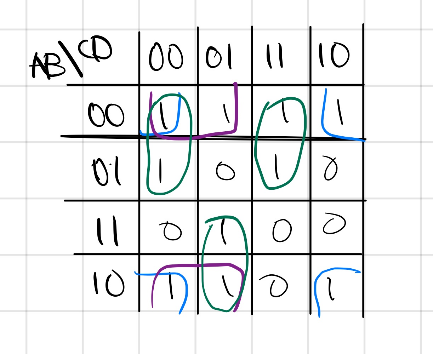
그림을 참고하여 아래의 진리표를 작성했다. 이때, 4-bit의 입력값은 10진수로 0~15를 나태내며, 출력값 A~G는 LED가 켜지는 경우를 1로 하여 작성했다.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **In A** | **In B** | **In C** | **In 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 | 1 | 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 |

위 진리표를 이용해 k-map을 작성할 수 있다.

a의 k-map은 왼쪽그림과 같이 작성된다. 우선 파란색은 4모퉁이 부분과 묶이며 B’D’이다. 빨갠색 묶음은 A’C가 되고 주황색 묶음은 BC이다. 2개의 초록색 묶음은 각각 A’BD와 AB’C’이다. 보라색 묶음은 AD’이 된다.

최종적으로 a = B’D’ + A’C + BC + A’BD + AB’C’ + AD’이다.



b의 k-map은 왼쪽그림과 같이 작성된다. 파란색은 a와 동일하게 B’D’이다. 보라색 묶음은 B’C’이다. 3개의 초록색 묶음은, 각각 A’C’D , AC’D, A’C’D’ 이다.

최종적으로 b = B’D’ + B’C’ + A’C’D + AC’D + AC’D’이다.

이런 방식으로 나머지 c~g에 대한 k-map과 식은 다음과 같다.

텍스트, 낱말맞추기게임이(가) 표시된 사진

자동 생성된 설명

c = AB’ + C’D + A’B + A’D + B’C’

d = BCD’ + ABC’ + B’C’D’ + BC’D + A’B’C + B’CD

e = B’D’ + CD’ + AB + AC

f = AB’ + BD’ + AC + C’D’ + A’BC’

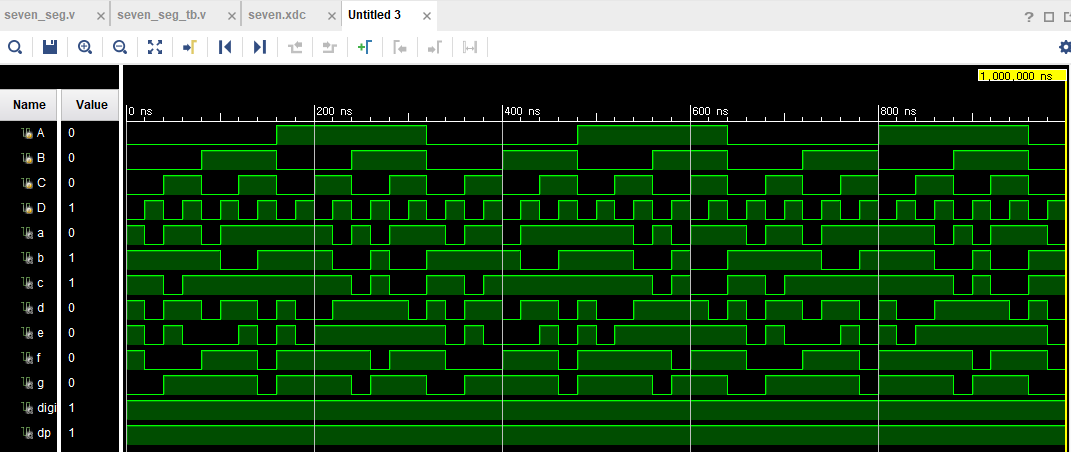
g = CD’ + AB’ + B’C + AD + A’BC’

K-map으로 얻은 함수를 Verilog로 코딩을 하면 아래와 같다. 이때 Digit 변수는 FPGA보드판에 있는 여러 개의 7-Segment중 하나를 선택하는 변수이며 아래와 같은 과정을 따른다.

1. Digit변수를 a|b|c|d|e|f|g|dp 로 하여 어떤 상황에서라도 1이 되도록 한다. ( a~dp 중 하나는 어떤 경우더라도 반드시 켜진다.)
2. xdc파일에서 변수 digit에 Signal Name이 DIGIT인 pin을 할당해 준다.
3. 회로 동작 시 선택한 DIGIT 번호의 LED에 불이 들어온다.

**텍스트이(가) 표시된 사진

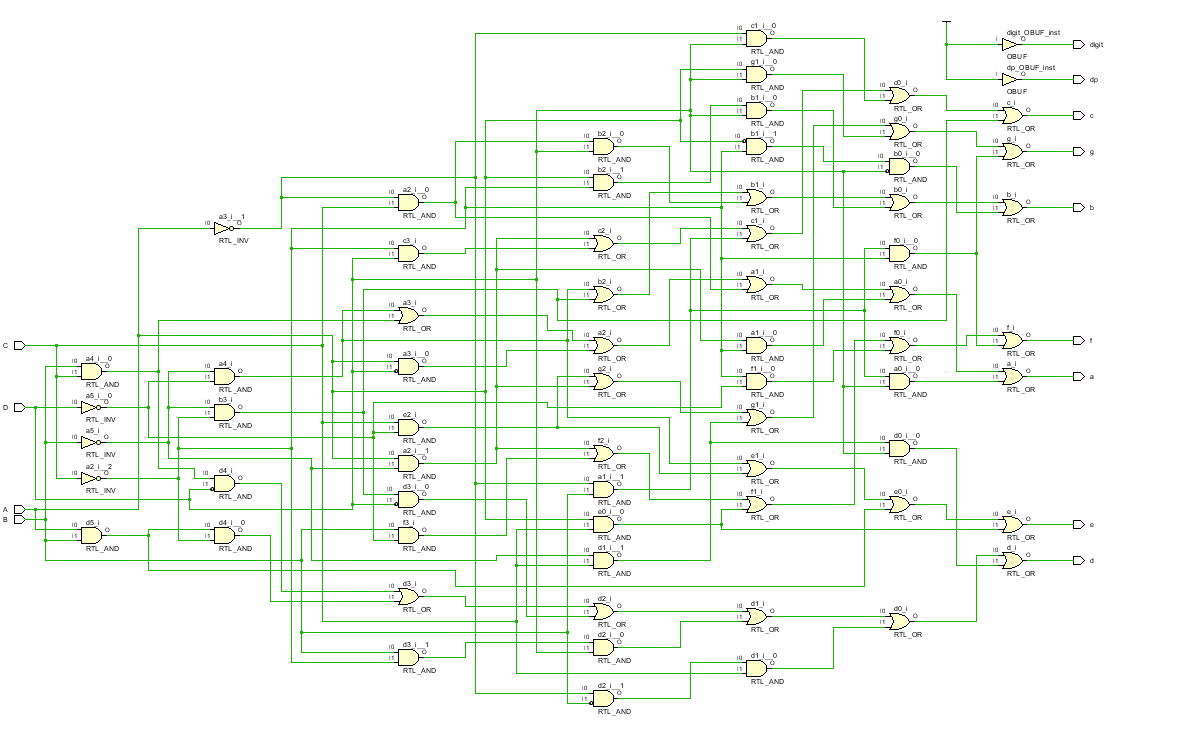
자동 생성된 설명**

****

시뮬레이션 결과와 참고했던 그림을 비교해보자.

1. 0000 입력으로 들어올 때 시뮬레이션 상에서 a,b,c,d,e,f는 1이고 g는 0이다. 이때, 0을 디스플레이한다.
2. 0001이 들어올 때 a,d,e,f,g는 0이고 b,c는 1의 값을 가지고 1을 디스플레이 한다.
3. 0010일 땐 a,b,d,e,g가 1로 2를 디스플레이 한다.
4. 0011 일 땐, f,e가 0이고 나머지가 모두 1이므로 3을 디스플레이한다. 0100일 때 a,d,e가 0이고 나머지가 1로 4를 디스플레이한다.
5. 0101일 때 b,e는 0으로 5를 잘 디스플레이한다.
6. 0110 일땐 b만 0으로 6을 디스플레이한다.
7. 0111은 a,b,c만 1로 7을 디스플레이 한다.
8. 1000은 모두 1이 되어 8을 디스플레이 한다.
9. 1001은 e,d가 0으로 9를 표시한다.
10. 1010부터는 문자를 디스플레이 하는데, d만 0으로 A를 표시한다.
11. 1011은 a,b가 0으로 소문자 b를 표현한다. 이는 B가 8과 동일한 모습이기 때문이다.
12. 1100은 대문자 C를 표시하며, b,c,g가 0이다.
13. 1101은 a,f가 0이다. 1011때와 마찬가지로 대문자 D를 표현하기 애매하기 때문에 소문자 d를 디스플레이 한다.
14. 1110일 때 b,c는 0이고 나머지는 모두 1이다.
15. 1111은 b,c,d가 0이고 나머지가 모두 1로 각자 E와 F를 잘 표시한다.

결론적으로, 입력값 0000~1111에 따른 실습에서 구현하고자 한 7-Segment Display를 완벽하게 구현했음을 알 수 있다. 아래는 구현한 함수의 스케메틱이다.

****

**3. 결과 검토 및 논의**

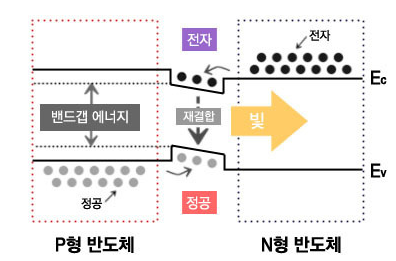
기존에 배웠던 7-Segment는 10이후는 모두 don’t care를 사용해 나름 손쉽게 구현할 수 있었다. 하지만, 이번엔 HEX를 display해야 했으므로 조금 더 복잡했다. 특히, 조금 신기했던 점은 영문자가 숫자와 비슷하게 표시되어 몇 개를 소문자로 표시했다는 점이다. K-map을 작성 후 시뮬레이션을 통해 확인하는 작업이 번거로워 FPGA의 보드를 직접 켜보면서 확인했다. 다행히도 어떤 LED에 문제가 있는지는 쉽게 판단할 수 있었다. 각 출력 a~g가 담당하고 있는 LED의 위치가 명확했기 때문이다. 4와5의 디스플레이에서 g부분이 출력되지 않는 것을 확인하고 다행히 금방 수정하였다.

또한, 이번에 새롭게 digit이라는 변수를 사용해 실습을 진행했다. Digit은 앞서 설명한 바와 같이 표시할 디스플레이의 위치를 나타내는 변수이며 항상 1이 되도록하기위해 a + b + c + d + e + g 로 코딩했다.

**4. 추가 이론**

7-Segment display는 LED의 등장으로 많이 상용화 되었다. 발광 다이오드로 부르며, 순방향으로의 전압을 가했을 때의 전기신호를 반도체 단자에 흘려 발광한다. 반도체 소자로 Light Emitting Diode의 이름을 가지게 되었다. LED는 백열전구의 광효율보다 우수하며, 전력 소모도 적고 수명도 더 길다. 초기에는 특정 색의 발광 다이오드를 만들 수 있었지만, 과학의 발전으로 (표면 실장 기술 등) 고휘도의 LED가 나오게 되었다.

아래그림처럼 PN접합의 구조로 되어있다. n층의 전자들이 p층으로 이동하면서 정공과 결합 후 에너지가 빛의 형태로 발산한다.



**5. 참고문헌**

<https://www.samsungsemiconstory.com/kr/led%EB%9E%80-%EB%AC%B4%EC%97%87%EC%9D%BC%EA%B9%8C-led%EC%9D%98-%EA%B8%B0%EB%B3%B8-%EC%9B%90%EB%A6%AC%EC%99%80-%EC%A2%85%EB%A5%98-%EC%9E%A5%EC%A0%90/>