8주차 결과보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번 : 20201597 이름 : 신동준

1. **실험 목적**

7-Segment Display는 10진수, HEX 등을 시각적으로 출력해주는 역할을 한다. 7개의 LED를 연결하고, 각각의 segment에 전기를 흘려 빛이 들어오게 하여 원하는 문자를 표현한다. 여러 개의 7-Segment Display를 병렬로 연결해서 여러 개의 문자를 표현할 수도 있다. 경우에 따라서는 Dp를 이용해서 소수점을 표현할 수도 있음

**2. 7-Segment Display의 결과 및 Simulation 과정에 대해서 설명하시오.**

**(Truth table 작성 및 k-map 포함,0~F,DP)(최대한 상세히 기술 할것)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **In A** | **In B** | **In C** | **In D** | **Out A** | **Out B** | **Out C** | **Out D** | **Out E** | **Out F** | **Out G** | **Out 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 | 1 | 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 | 0 | 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 |

텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명

텍스트, 화이트보드이(가) 표시된 사진

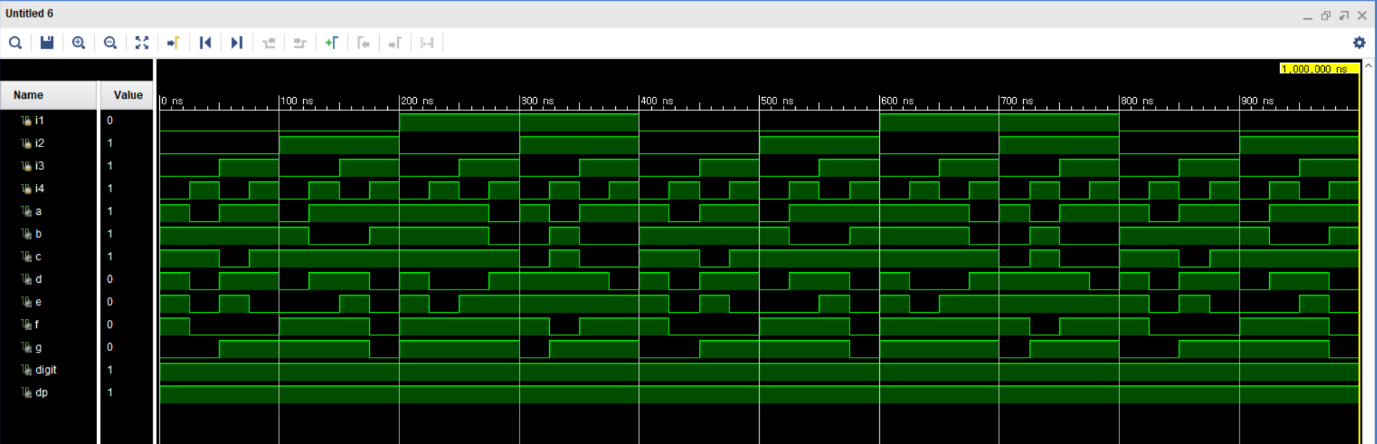
자동 생성된 설명

텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명

텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명



Truth table을 채우고 그를 바탕으로 카르노 맵을 작성해서 아래와 같은 식들을 얻었다.

a = (i1 & ~i2 & ~i3) | (~i1 & i2 & i4) | (i1 & ~i4) | (~i1 & i3) | (i2 & i3) | (~i2 & ~i4)

b = (~i1 & ~i3 & ~i4) | (~i1 & i3 & i4) | (i1 & ~i3 & i4) | (~i2 & ~i3) | (~i2 & ~i4)

c = (~i1 & ~i3) | (~i1 & i4) | (~i3 & i4) | (~i1 & i2) | (i1 & ~i2)

d = (~i1 & ~i2 & ~i4) | (~i2 & i3 & i4) | (i2 & ~i3 & i4) | (i2 & i3 & ~i4) | (i1 & ~i3 & ~i4)

e = (~i2 & ~i4) | (i3 & ~i4) | (i1 & i3) | (i1 & i2)

f = (~i1 & i2 & ~i3) | (~i3 & ~i4) | (i2 & ~i4) | (i1 & ~i2) | (i1 & i3)

g = (~i1 & i2 & ~i3) | (~i2 & i3) | (i3 & ~i4) | (i1 & ~i2) | (i1 & i4)

여기서 i1, i2, i3, i4는 in A, B, C, D이다. 각각 카르노 맵의 결과로 SOP 형태로 나타내고 그대로 적용하였다. Dp의 경우 이번에는 1로 고정을 한다.

그리고 나중에 여러가지 segment 중에 한가지를 고정해서 전체에 전류를 흘러줘야 하기 때문에 digit이라는 변수를 만들어서 키고자하는 모든 segment를 or 연산으로 묶어서 키는 segment를 특정해준다.

[digit = a | b | c | d | e | f | g]

1. **결과 검토 및 논의 사항.**

텍스트, 전자기기이(가) 표시된 사진

자동 생성된 설명

정상적으로 FPGA에 올렸을 때도 스위치를 올린 2진법에 따라서 정상적으로 segment가 보이는 것을 알 수 있다. 각각 switch 1,2,3,4에 i1, i2, i3 , i4를 할당했고, 그린 카르노 맵을 바탕으로 a~g의 segment를 각 상황별로 encoding 해줬다.

1. **추가 이론 조사 및 작성.**

Segment는 실제로는 동시에 켜져있는 것처럼 보이지만 위의 사진과 같이 병렬로 연결되어 있다면, 사람 눈으로 식별하기 힘들만큼 빠르게 회로를 순서대로 지나가면서 모두 동시간대에 켜져있는 것처럼 보이게 한다.