8주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231561 이름: 심소현

**1.**

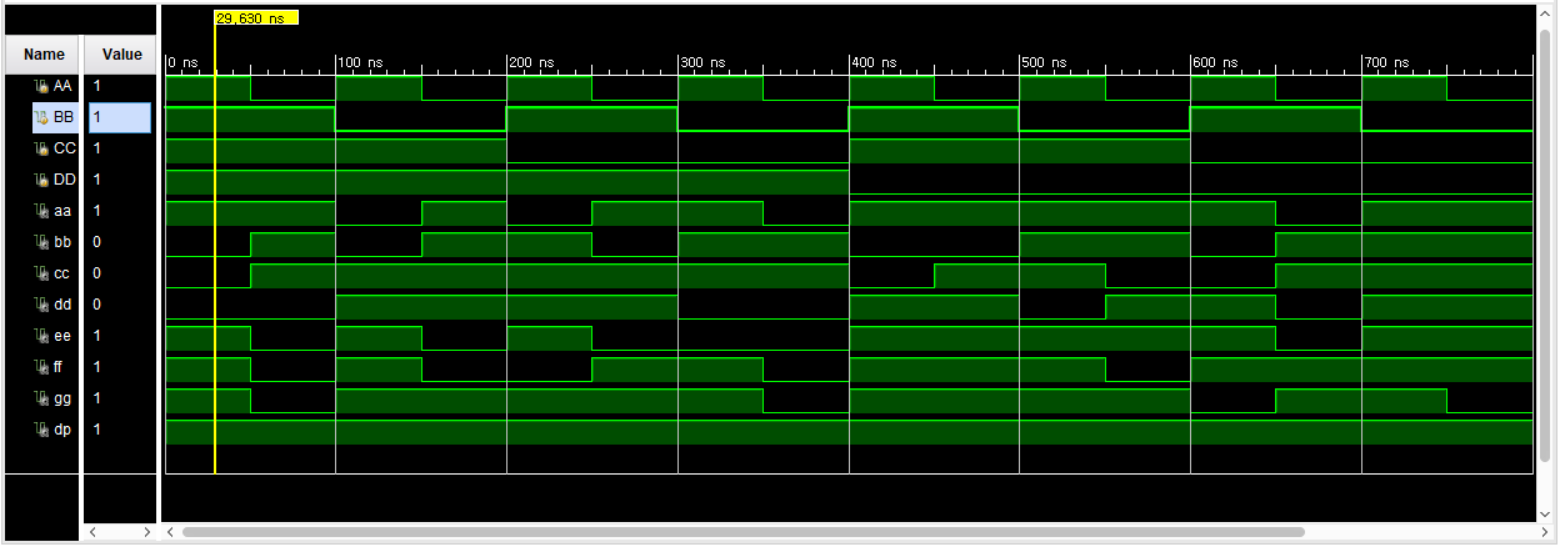
.................

이번 실험은 7-segment display을 구현해보는 것이었다. 우선 Verilog 언어를 사용해서 각 segment 마다 논리식을 구하여 이를 적용한 코드를 작성한다. 이를 simulation과 FPGA를 통해 눈으로 확인하여 7-segment display의 작동을 확인하는 것이 이번 실험의 목표라고 할 수 있다.

................

**2.**

.......................



위는 직접 구현한 7-segment display의 simulation 결과이다.

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

위는 7-segment display의 진리표이다. 대문자 알파벳은 입력 데이터, 소문자 알파벳과 DP는 출력 데이터이다.

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

위는 segment a의 카르노 맵이다.

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

위는 segment b의 카르노 맵이다.

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

위는 segment c의 카르노 맵이다.

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

위는 segment d의 카르노 맵이다.

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

위는 segment e의 카르노 맵이다.

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

위는 segment f의 카르노 맵이다.

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

위는 segment g의 카르노 맵이다.

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

위는 segment DP의 카르노 맵이다.

위의 카르노 맵들을 보고 각각 segment 마다 우선 논리식을 세워야 한다. 그에 따른 논리식은 아래와 같다.

a=B’D’+A’C+BC+AD’+A’BD+AB’C’

b=B’C’+B’D’+A’C’D’+AC’D+A’CD

c=A’C’+A’D+A’B+AB’+C’D

d=B’C’D’+ABD’+BC’D+B’CD+A’CD’

e=AB+AC+CD’+B’D’

f=C’D’+AB’+AC+BD’+A’BC’

g=CD’+AB’+AD+B’C+A’BC’

dp=1

위에 식에 맞게 Verilog 코드를 작성하면 16진수에 따른 결과가 출력된다. 9 이후의 숫자들에서는 알파벳으로 A, b, C, d, E, F와 같이 나타난다.

.........................

**3.**

.......................

위의 결과를 검토해보았을 때, 7-segment display에서는 어떤 진수를 사용해서 수를 표현할 것인지를 우선 정해야 한다. 위의 실험에서는 한 자릿수에서 16진수로 모두 표현을 하여 digit 3만 사용했지만, 10진수를 사용하는 등 두 자릿수 이상으로 표현하게 된다면 다른 digit도 사용해야 할 것이다. 또한 카르노 맵에 맞는 논리식을 알맞게 준비하는 과정도 중요하다. FPGA가 없는 상황에서는 simulation만으로 확인하는 것은 힘들기 때문이다.

.........................

**4.**

.......................

14-segment 혹은 16-segment display도 존재한다. 14-segment display는 추가적인 문자와 숫자를 더 잘 표현할 수 있도록 추가 segment가 배치되어 있다. 기본적으로 A에서 N까지의 문자 표현 또한 가능하게 되어 있다. 16-segment display는 14-segment display보다 2개 더 많은 segment를 가지고 있고, 덕분에 다양한 기호 표현까지 가능하다. 이 두 display는 7-segment display보다 더 많은 segment를 조절해야 하기 때문에 더 복잡한 회로를 가지고 있다. 이때 멀티플렉싱 방법을 사용하여 신호를 전달하는 것이 더 효율적인데, 멀티플렉싱은 여러 자릿수를 표시할 때 매우 빠른 속도로 각 display에 순차적으로 신호를 전달하는 방법이다. 한 번에 하나의 display에만 신호를 보내기 때문에 전력을 아낄 수 있다.

.........................