컴퓨터공학실험2 8주차 결과 보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20201635 이름: 전찬

**0. 목차**

1. 실험 목적

2. 7-Segment Display의 결과 및 simulation 과정 설명

3. 결과 검토 및 논의 사항

4. 추가 이론 조사 및 작성

**1. 실험 목적**

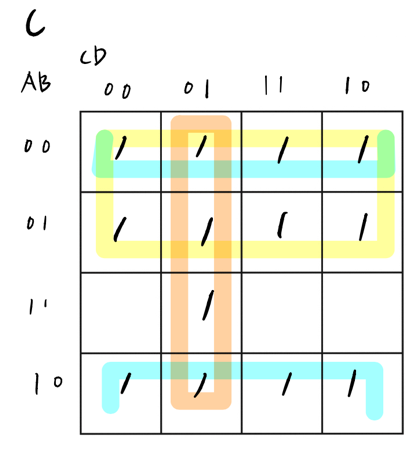
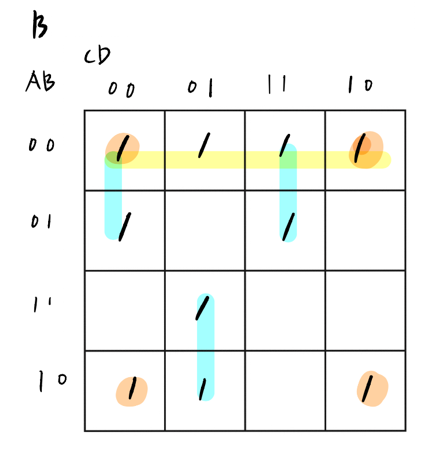
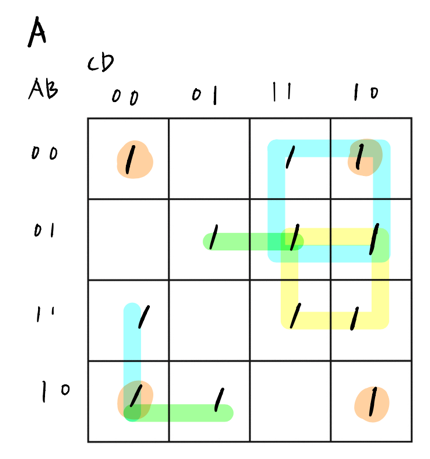
k-map을 통해서 7-segment display의 각 output(A, B, …, Dp)에 대한 식을 직접 표현해보며, simulation, FPGA 동작 검증을 통해서 input에 따라 올바른 output을 만들어냈는지 파악한다. FPGA와 구현한 module에 대한 연결을 직접 수행하며, pin을 알맞게 연결해서 동작을 검증한다.

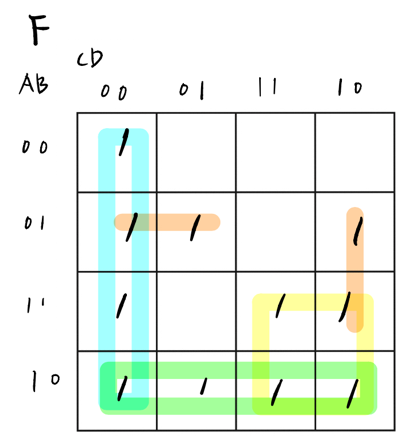
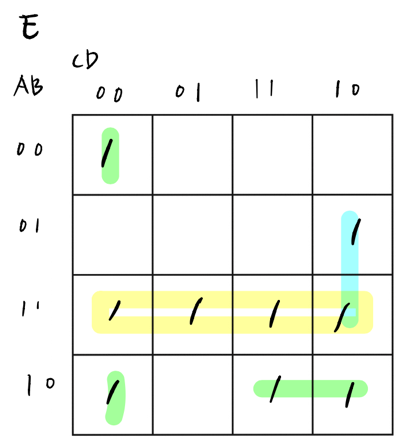
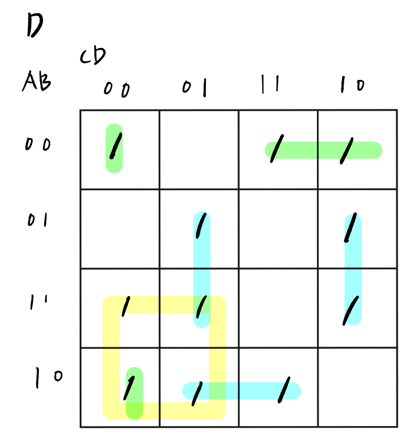
**2. 7-Segment Display의 결과 및 simulation 과정 설명**

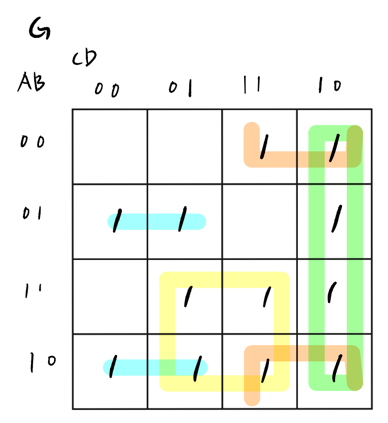
이번 실습에서는 4 bit input에 맞춰서, 각 input이 나타내는 수를 Hexadecimal 형태의 output으로 7-Segement display에 표시하는 실습을 수행했다.(input 0 ~ 15에 따라서 output으로 0~,,9, A, b, C, d, E, F를 표현하는 형태) 이에 따라 존재하는 4개의 input에 대해 존재하는 모든 경우에 대한 8가지 output(7-Segment display의 screen A, B, C, D, …, Dp)을 각각 k-map을 통해서 4bit input에 대한 SOP 형태로 표현하는 과정을 거쳐서 output을 표현했다. 이에 대한 Truth table과 k-map은 아래와 같다.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7-Segment Display | | | | | | | | | | |
| input | | | | output | | | | | | |
| in1 | in2 | in3 | in4 | 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 | 1 | 1 | 0 | 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 | 1 | 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 |

<7-Segment Display의 truth table>







<7-Segment Display의 k-map>

여기에서 Dp는 항상 1이 되게 설정하면 되므로 생략했다. 또한 아래 simulation에서 screen을 켜주는 변수 sc도 항상 1이면 되기 때문에 생략했다. 또한 논리식은 아래와 같다.(아래 실습 시간에 작성한 것과 위 k-map에서의 표현이 약간 다를 수 있다.)

assign A = (in2&in3)|(~in1&in3)|(in1&~in4)|(~in2&~in4)|(~in1&in2&in4)|(in1&~in2&~in3);

assign B = (~in2&~in4)|(~in2&~in3)|(~in1&in3&in4)|(in1&~in3&in4)|(~in1&~in3&~in4);

assign C = (~in3&in4)|(~in1&in4)|(in1&~in2)|(~in1&in2)|(~in2&~in3);

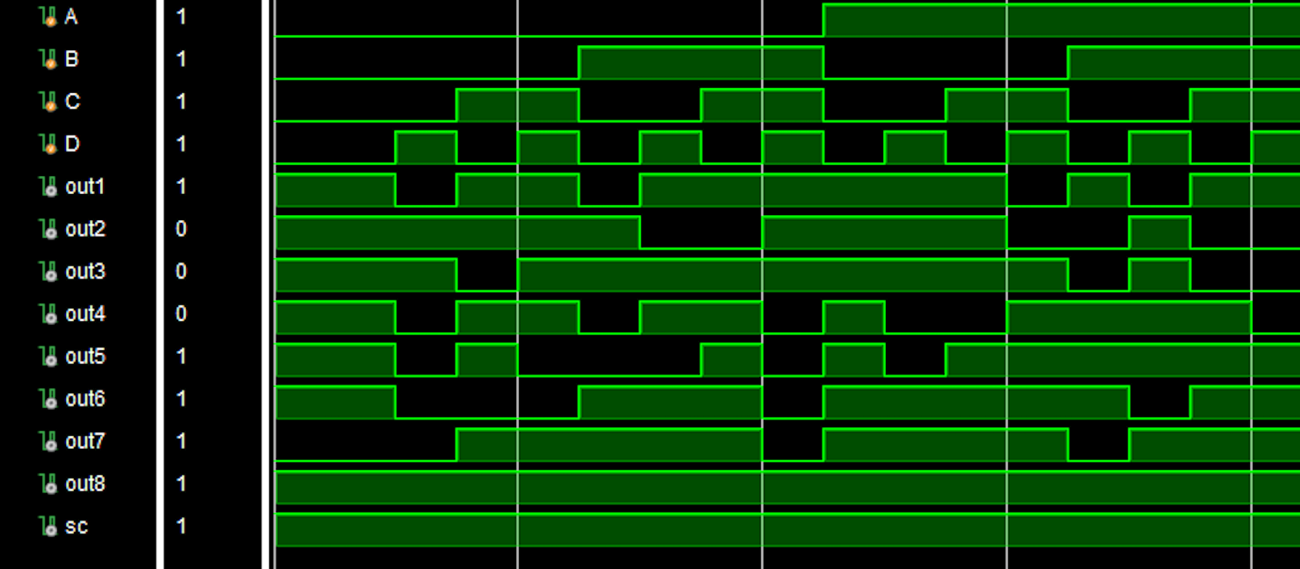
assign D = (~in1&in3&~in4)|(~in2&in3&in4)|(in2&~in3&in4)|(in1&in2&~in4)|(~in2&~in3&~in4);

assign E = (in3&~in4)|(in1&in3)|(in1&in2)|(~in2&~in4);

assign F = (in1&in3)|(in1&~in2)|(in2&~in4)|(~in3&~in4)|(~in1&in2&~in3);

assign G = (in1&in3)|(~in2&in3)|(in1&~in2)|(in2&~in3&in4)|(~in1&in2&~in4);

또한 simulation 결과는 아래와 같다.



<7-Segment display의 simulation 결과>

추가로 FPGA를 통한 동작 검증을 수행할 때, 각 input(4개의 1bit input)을 DIP\_SW 0~3에 연결했으며, output A, B, …, DP를 SEG\_A, …., SEG\_DP에 연결했다. 이렇게만 연결했을 때, 만든 module이 실행은 되지만, LED가 켜지지 않는 형태가 발생했다. 따라서 DIGIT1 (E14) pin을 항상 High인 output에 연결했는데 이때 정상적으로 작동이 되었다. 따라서 Segment 뿐만이 아닌 알맞은 Digit number에도 입력을 주어야 화면이 켜진다는 점을 실습을 진행하며 파악할 수 있었다. 따라서 위 simulation 결과에서 sc라는 변수가 항상 출력이 1인, 이를 수행해주는 변수이다.

**3. 결과 검토 및 논의 사항**

이번 실습에서는 k-map을 이용한 7-Segment display를 직접 구현해보며, FPGA에 연결해 input에 따라 알맞은 output을 출력해내는 형태를 수행했다. 이를 진행하며 FPGA의 화면을 키기 위해서는 Segment 뿐만 아니라 Digit Pin 또한 input을 주어야 적절하게 동작 검증을 수행할 수 있음을 파악할 수 있었다.

따라서 실습을 수행하면서 갖게 된 의문점이 존재하는데, Segment pin은 1개씩밖에 존재하지 않는데, FPGA의 Digit은 1~8, 7-Segment display는 총 8개의 화면이 존재한다. 따라서 만약 각 8개의 display를 독립적으로 작용하게 하려면 어떻게 수행해야 되는지에 대한 의문점이 생겼다.

**4. 추가 이론 조사 및 작성**

위의 의문을 해결할 수 있는 방법이 7-Segment display의 dynamic 구동방식이다. 우선 이를 이해하기 위해 LED의 잔상 효과에 대해서 먼저 알아보자. LED 잔상 효과란, LED가 계속해서 켜지고 꺼짐을 반복하지만, 우리 눈에서는 잔상이 남아 계속해서 켜져 있는 것처럼 보이는 효과이다. 이를 통해서 7-Segment 4개를 통해 12:31의 시간/분을 표시한다고 해 보자. 이는 아래 과정을 통해서 표현될 수 있다.

(1) 첫 번째 display만 High로 만들며 7-Segment display input에 0001을 대입한다.

(2) 두 번째 display만 High로 만들며 7-Segment display input에 0010을 대입한다.

(3) 세 번째 display만 High로 만들며 7-Segment display input에 0011을 대입한다.

(4) 네 번째 display만 High로 만들며 7-Segment display input에 0001을 대입한다.

(5) (1)~(4)의 과정을 분이 바뀌기 전까지 계속해서 반복한다.

이를 통해서 여러 화면에서 각각 다른 숫자를 표시할 수 있다. 하지만 이를 설계하기 위해서는 위 simulation에서 설정했던 변수 sc를 sc1, sc2, sc3, sc4까지 4개로 확장해서 각각 1000, 0100, 0010, 0001 형태를 계속해서 반복해주어야 하기도 한다.