

## 컴퓨터공학실험 2 8 주차 결과보고서

전공: 컴퓨터공학과

학년: 2

학번: 20191559

이름: 강상원

### 1. 실험 목적

7-Segment Display의 실제 수행을 확인해본다. Verilog 언어를 이용해 7-Segment Display를 구현하는 방법을 익힌다. Schematic 구조, Simulation을 통해 gate의 구현을 확인한다. 경우의 수를 나누어 각 입력에 대한 gate의 출력값을 확인해 본다.

### 2. 7-Segment Display의 결과 및 Simulation 과정에 대해서 설명하시오. (Truth table 작성 및 k-map 포함, 0~F)

진리표를 먼저 작성하여 출력마다의 규칙(논리식)을 찾아야 한다. 다음은 7-Segment Display의 진리표이다.

In A	In B	In C	In D	Decimal (Hex)	Out A	Out B	Out C	Out D	Out E	Out F	Out G	
0	0	0	0	0	1	1	1	1	1	1	0	
0	0	0	1	1	0	1	1	0	0	0	0	
0	0	1	0	2	1	1	0	1	1	0	1	
0	0	1	1	3	1	1	1	1	0	0	1	
0	1	0	0	4	0	1	1	0	0	1	1	
0	1	0	1	5	1	0	1	1	0	1	1	
0	1	1	0	6	1	0	1	1	1	1	1	
0	1	1	1	7	1	1	1	0	0	0	0	
1	0	0	0	8	1	1	1	1	1	1	1	
1	0	0	1	9	1	1	1	0	0	1	1	
1	0	1	0	10 (A)	1	1	1	0	1	1	1	
1	0	1	1	11 (b)	0	0	1	1	1	1	1	
1	1	0	0	12 (c)	1	0	0	1	1	1	0	
1	1	0	1	13 (d)	0	1	1	1	1	0	1	
1	1	1	0	14 (E)	1	0	0	1	1	1	1	
1	1	1	1	15 (F)	1	0	0	0	1	1	1	

위 진리표를 참고하여, 카르노 맵을 작성한다.

각 Output (A ~ G)에 대한 카르노 맵은 다음과 같다.

- A

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

$$a = A'C + BC + B'D' + A'BD + AB'C' + AC'D'$$

- B

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

$$b = A'B' + B'C' + B'D' + A'C'D' + A'CD + AC'D$$

- C

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

$$c = C'D + A'B + AB' + A'C' + A'D$$

- D

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

$$d = B'C'D' + BC'D + ABD' + B'CD + A'CD'$$

- E

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

$$e = CD' + AB + AC + B'D'$$

- F

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

$$f = AB' + C'D' + AC + A'BC' + BCD'$$

- G

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

$$g = AB' + CD' + AD + B'C + A'BC'$$

카르노 맵을 통해 간소화된 각 논리식을 Verilog로 표현하면 다음처럼도 나타낼 수 있다.

```
`timescale 1ns / 1ps

module segment(

input a,b,c,d, output A,B,C,D,E,F,G,seg);

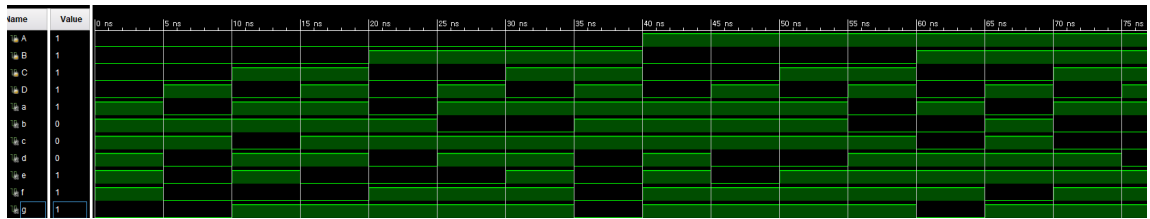
assign A = (b&c) | (!a&c) | (a&!d) | (!b&!d) | (!a&b&d) | (a&!b&!c);
assign B = (!b&!d) | (!b&!c) | (!a&c&d) | (a&!c&d) | (!a & !c & !d);
assign C = (!c & d) | (!a & d) | (a & !b) | ( !a & b) | (!b & !c);
assign D = (!a & c & !d) | (!b & c & d) | (b & !c & d) | ( a & b&!d) |
(!b&!c&!d);
assign E = (c &!d) | (a &c) | (a&b) | (!b & !d);
assign F = (a & c ) | (a&!b) | (b&!d) | (!c&!d) + (!a&b&!c);
assign G = (a&c) | (!b&c) | (a&!b) | (b&!c&d) | (!a&b&!d);
assign seg =1;

endmodule
```

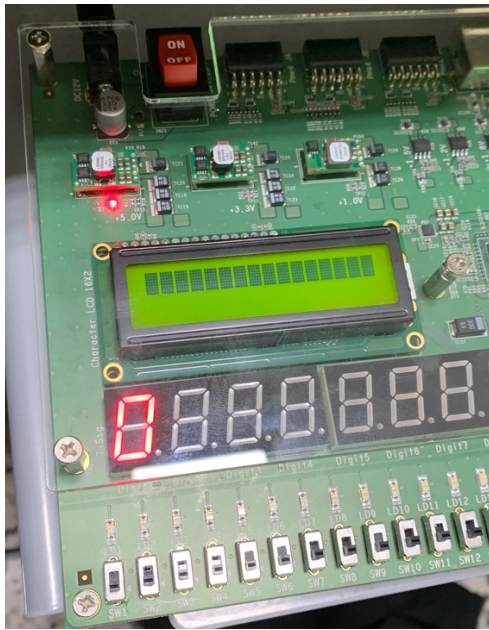
다음은 constraints 파일 segment\_c.xdc의 내용이다. constraints 파일은 친, 포트 연결을 설정해준다.

```
set_property IOSTANDARD LVCMOS18 [get_ports a]
set_property IOSTANDARD LVCMOS18 [get_ports b]
set_property IOSTANDARD LVCMOS18 [get_ports c]
set_property IOSTANDARD LVCMOS18 [get_ports d]
set_property IOSTANDARD LVCMOS18 [get_ports A]
set_property IOSTANDARD LVCMOS18 [get_ports B]
set_property IOSTANDARD LVCMOS18 [get_ports C]
set_property IOSTANDARD LVCMOS18 [get_ports D]
set_property IOSTANDARD LVCMOS18 [get_ports E]
set_property IOSTANDARD LVCMOS18 [get_ports F]
set_property IOSTANDARD LVCMOS18 [get_ports G]
set_property IOSTANDARD LVCMOS18 [get_ports seg]
set_property PACKAGE_PIN J4 [get_ports a]
set_property PACKAGE_PIN L3 [get_ports b]
set_property PACKAGE_PIN K3 [get_ports c]
set_property PACKAGE_PIN M2 [get_ports d]
set_property PACKAGE_PIN D20 [get_ports A]
set_property PACKAGE_PIN C20 [get_ports B]
set_property PACKAGE_PIN C22 [get_ports C]
set_property PACKAGE_PIN B22 [get_ports D]
set_property PACKAGE_PIN B21 [get_ports E]
set_property PACKAGE_PIN A21 [get_ports F]
set_property PACKAGE_PIN E22 [get_ports G]
set_property PACKAGE_PIN E14 [get_ports seg]
```

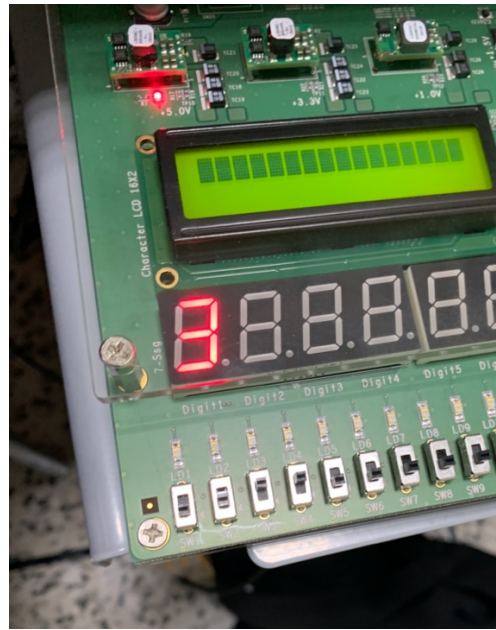
Simulation 결과는 다음과 같다.



아래는 FPGA의 7-Segment Display에 실제로 16진수 숫자를 출력해본 모습이다.



↑ 16진수 0을 출력한 모습



↑ 16진수 3을 출력한 모습