

2018년 2학기 컴퓨터공학실험Ⅱ
CSE3016-05반 8주차 결과 보고서

학번: 20171665

이름: 이 선 호

2018. 11. 09

목 차

I | 7-Segment Display 결과 및 Simulation 과정

1. 7-Segment Display	3
----------------------	-------	---

II | 논의

1. 결과 검토 및 논의사항	7
2. 추가 이론 조사 및 작성	9

I 7-Segment Display 결과 및 Simulation 과정

1. 7-Segment Display

1) 카르노 맵

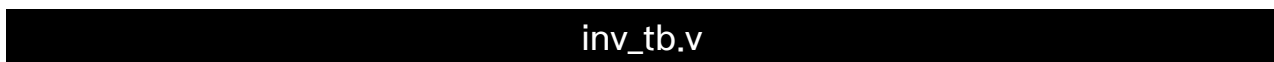
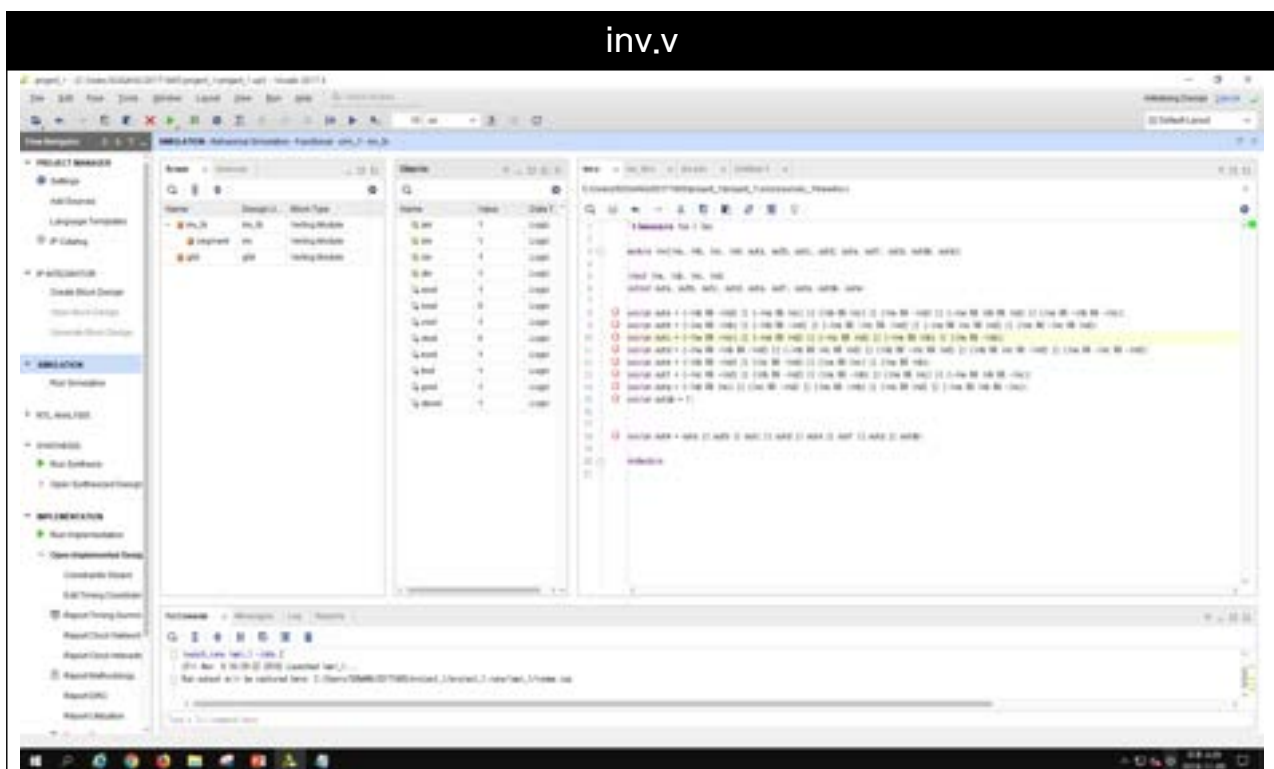
Karnaugh Map			
Output A		Output B	
Output C		Output D	
Output E		Output F	
Output G		Output DP	

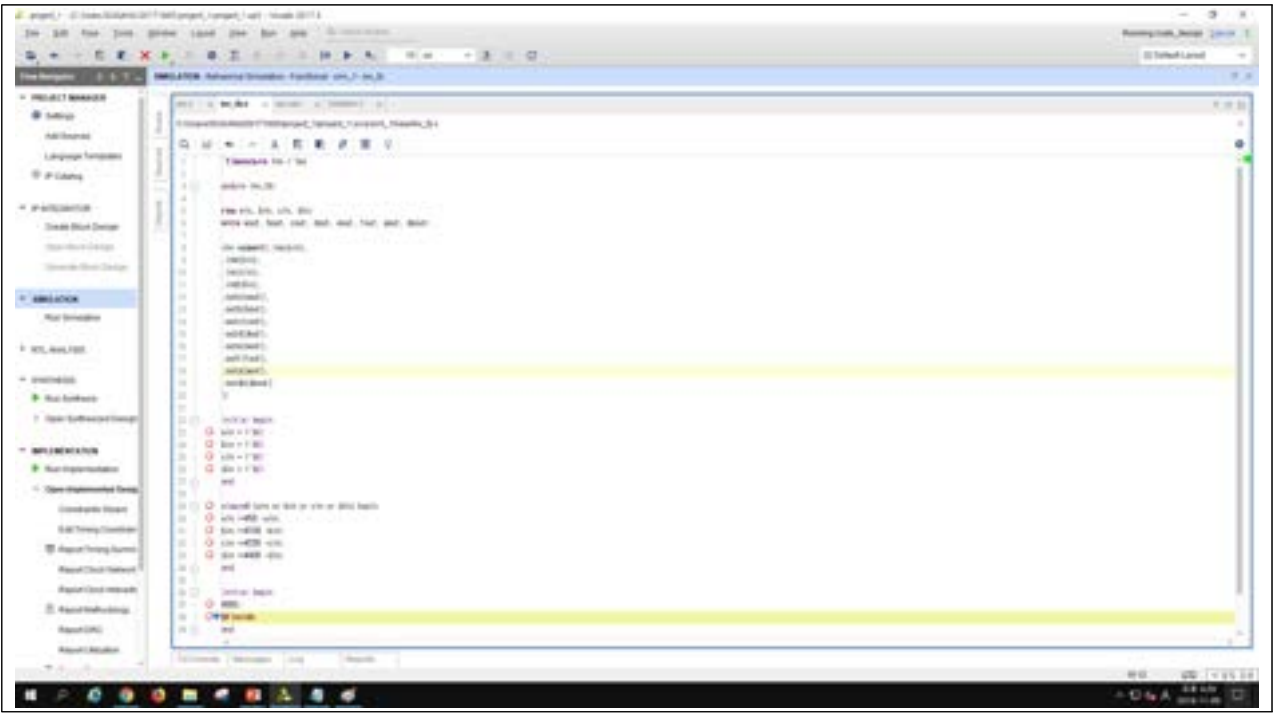
Out6

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

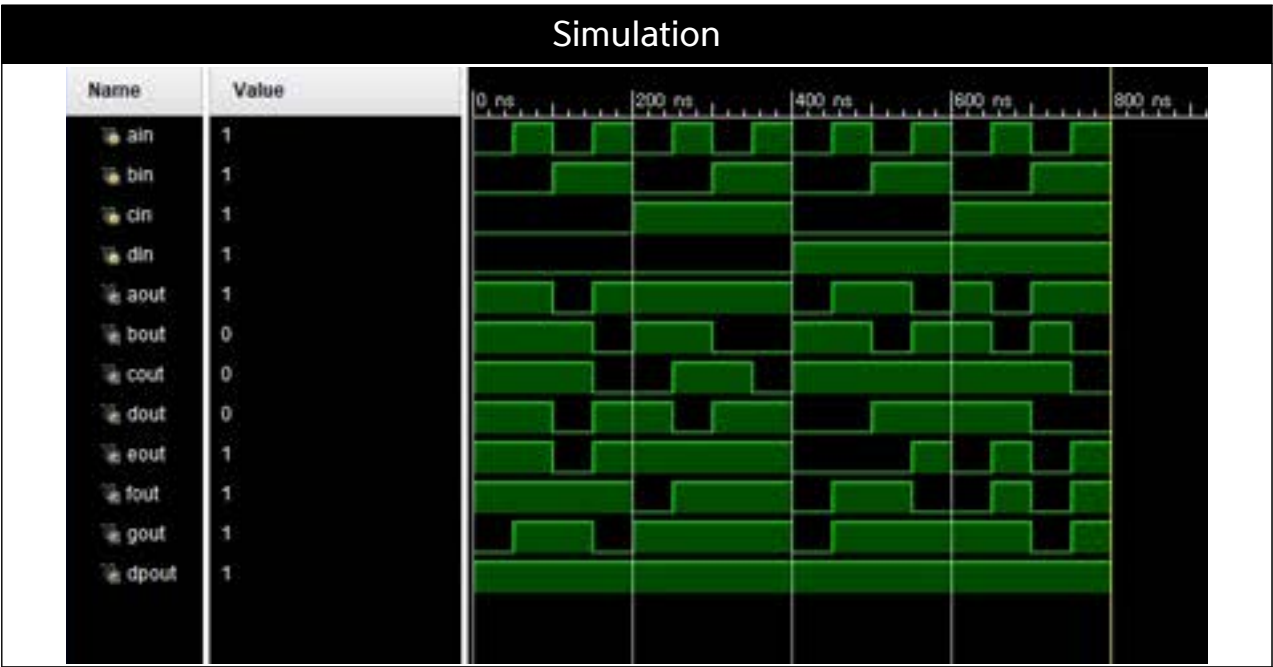
• outDP outDP=1

2) Verilog 코딩





3) Simulation 결과



4) 진리표

Truth Table

In A	In B	In C	In D	Out A	Out B	Out C	Out D	Out E	Out F	Out G	OutDP
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

6) 설명

입력하는 input 값을 7-Segment Display에 출력하는 논리 회로를 제작하기 위해서는 우선적으로 input 값이 0부터 F까지 그대로 입력되지 않는다는 점을 고려해야 된다. 전자 회로에서는 0과 1만을 사용하기 때문이다. 2의 4 제곱이 16이므로 0부터 F까지 16개의 input을 받기 위해서는 0 또는 1로 갖는 4개의 input을 필요로 한다. 즉, 0000 부터 1111까지를 입력으로 받아야 하며, 입력의 각 자리 비트를 MSB부터 LSB까지 차례대로 In A, In B, In C, In D로 정했다.

7-Segment Display에서 입력한 값을 출력할 때 장치의 각 LED에 불이 켜지거나 꺼 지도록 조절해야 한다. 이 LED들을 시계 방향으로 차례대로 Out A, Out B, Out C, Out D, Out E, Out F, Out G, 그리고 소수점을 출력하는 Out DP로 명명했다. 이러한 LED들이 In A, In B, In C, In D의 값에 따라 불이 들어오는지 아닌지(출력이 1이 되는지 아닌지)를 진리표로 그려서 표의 결과를 바탕으로 카르노 맵을 그렸다. 각 출력 들에 해당하는 카르노 맵을 그려서 Prime Implicant(주항)들을 묶은 후 얻은 SOP 식 들은 아래와 같다. 편의상 In A, In B, In C, In D는 각각 A, B, C, D로 작성했다.

$$Out A = B'D' + A'C + BC + AD' + A'BD + AB'C'$$

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

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

$$Out D = A'B'D' + B'CD + BC'D + BCD' + AC'D'$$

$$Out E = B'D' + CD' + AC + AB$$

$$Out F = C'D' + BD' + AB' + AC + A'BC'$$

$$Out G = B'C + CD' + AB' + AD + A'BC'$$

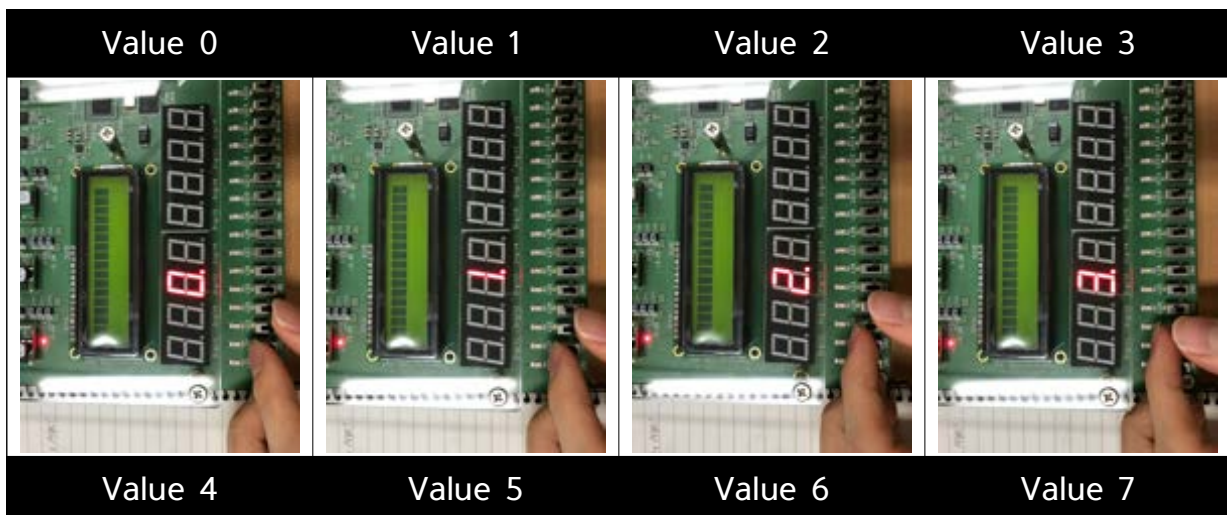
$$Out DP = 1$$

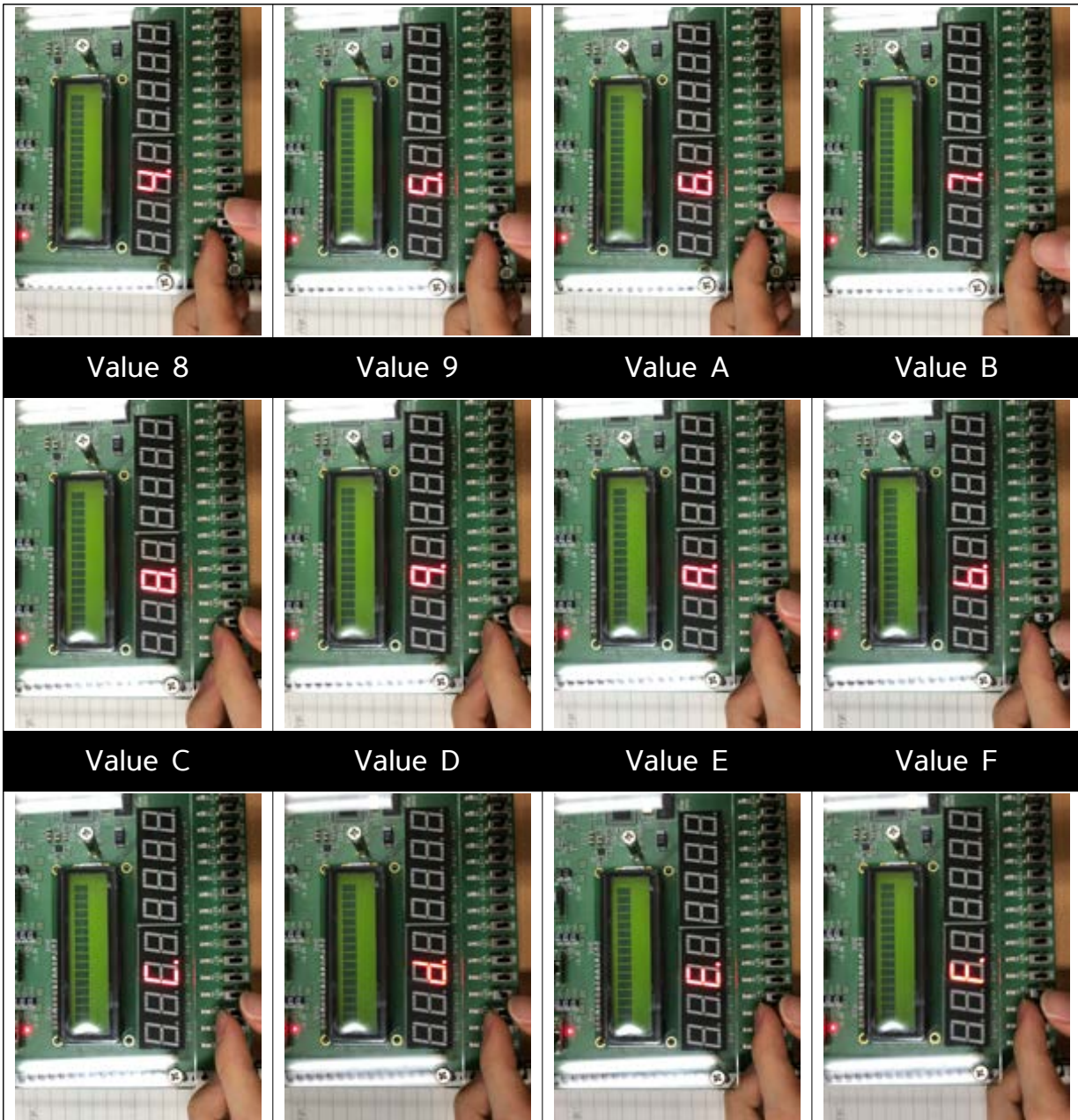
위의 SOP 식을 바탕으로 각 입력들에 대하여 AND 게이트와 OR 게이트를 조합하여 Verilog 코드를 작성하여 '3) 시뮬레이션 결과'와 같이 회로의 시뮬레이션 동작을 확인했다.

III 논의

1. 결과 검토 및 논의사항

실습 시간에 구현한 7-Segment Display 회로가 정상적으로 잘 제작되었는지 검토하기 위해 FPGA를 사용하여 스위치 4개를 In A부터 In D까지의 input port로 설정하고 LED를 output port로 설정하여 아래와 같이 직접 구동해 보았다.





이를 통해 실습에서의 7-Segment Display 회로가 정상적으로 구현되었다는 사실을 알 수 있다.

실습에서는 출력의 각 SOP 식의 SUM 항들을 그대로 OR 연산하여 출력했다. 그러나 이를 실제 회로로 제작해서 그대로 구현하는 것은 중복되는 항을 계속 계산하게 되므로 효율적이지 않다. 그래서 SOP 식마다 서로 겹치는 항들을 공통적으로 묶어서 연산과정의 중복을 줄이는 방법이 있을 것이다. 예를 들어, Out A의 SOP 식의 B'D'항은 Out B의 SOP 식의 B'D' 항과 겹친다. 그러므로 B'D'를 한 번 연산한 결과를 각각 Out A와 Out B에 OR 연산으로 바로 연결해주면 B'D'를 각 논리식마다 두 번 계산하는 번거로움이 줄어든다. 어차피 각 SOP 식들은 항들을 OR 연산하므로 결과에 영향을 미치지 않는다. 이러한 식으로 회로를 효율적으로 간단하게 구현하는 방법이 존재한다.

2. 추가 이론 조사 및 작성

이번 실습에서는 단순히 한 자리의 0부터 F까지의 값을 하나의 LED Display에 출력하는 방식으로 회로를 구현했다. 그러나 만일 input이 16진수의 네 자리 수라고 가정하면 이 input 값을 실습에서 진행했던 방식처럼 하나의 LED Display에서만 출력하는 것은 어렵다. 그래서 각 자릿수마다 병렬적으로 한 개씩 네 개의 LED Display를 연결하여 0부터 F까지의 값 중 하나를 출력하도록 제작하는 방법이 있다. 다시 말해서, 16개의 input을 네 개씩 각 하나의 LED Display에 출력하도록 하는 것이다. 또 다른 방법으로는 클럭과 BCD-to-7 Segment Display Decoder를 이용해서 모두 4개의 동일한 input과 연결되어 있는 네 개의 LED Display를 일정 시간에 따라 주기적으로 한 개씩 출력하도록 해서 사람의 눈으로 볼 때 네 개의 디스플레이가 모두 켜져 있는 착각을 주게끔 구현할 수 있다. 이는 앞서 예비 보고서에서 조사했던 Dynamic 구동방식과 비슷하며, 실제로 적지 않은 LED Display들이 회로에 들어가는 cost의 효율성을 고려하여 이와 같은 방식을 채택하고 있다. 아래는 7-Segment Display를 4개 사용하여 Dynamic 구동방식으로 16진수의 네 자리 수를 출력하는 회로도다.

