8주차 결과보고서

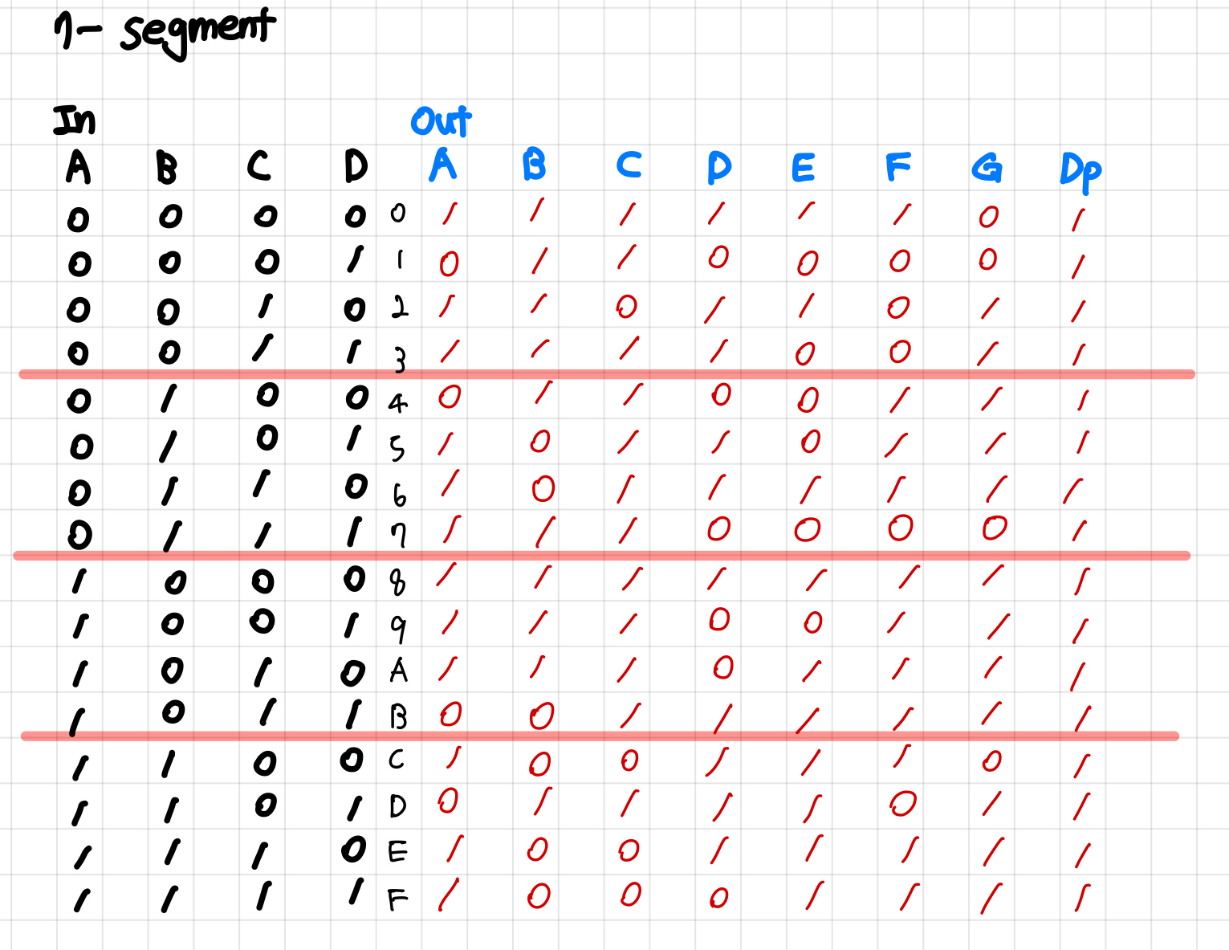
전공: 컴퓨터공학과 학년: 2학년 학번: 20231609 이름: 정희선

**1.**

7-segment display의 개념을 이해한 후, verilog를 이용해 직접 7-segment display를 구현한다. 이후 simualtion으로 각 gate가 올바르게 동작하는지 확인한다. 마지막으로 FPGA에 연결해 verilog로 구현한 회로의 동작을 display를 직접 보며 확인한다.

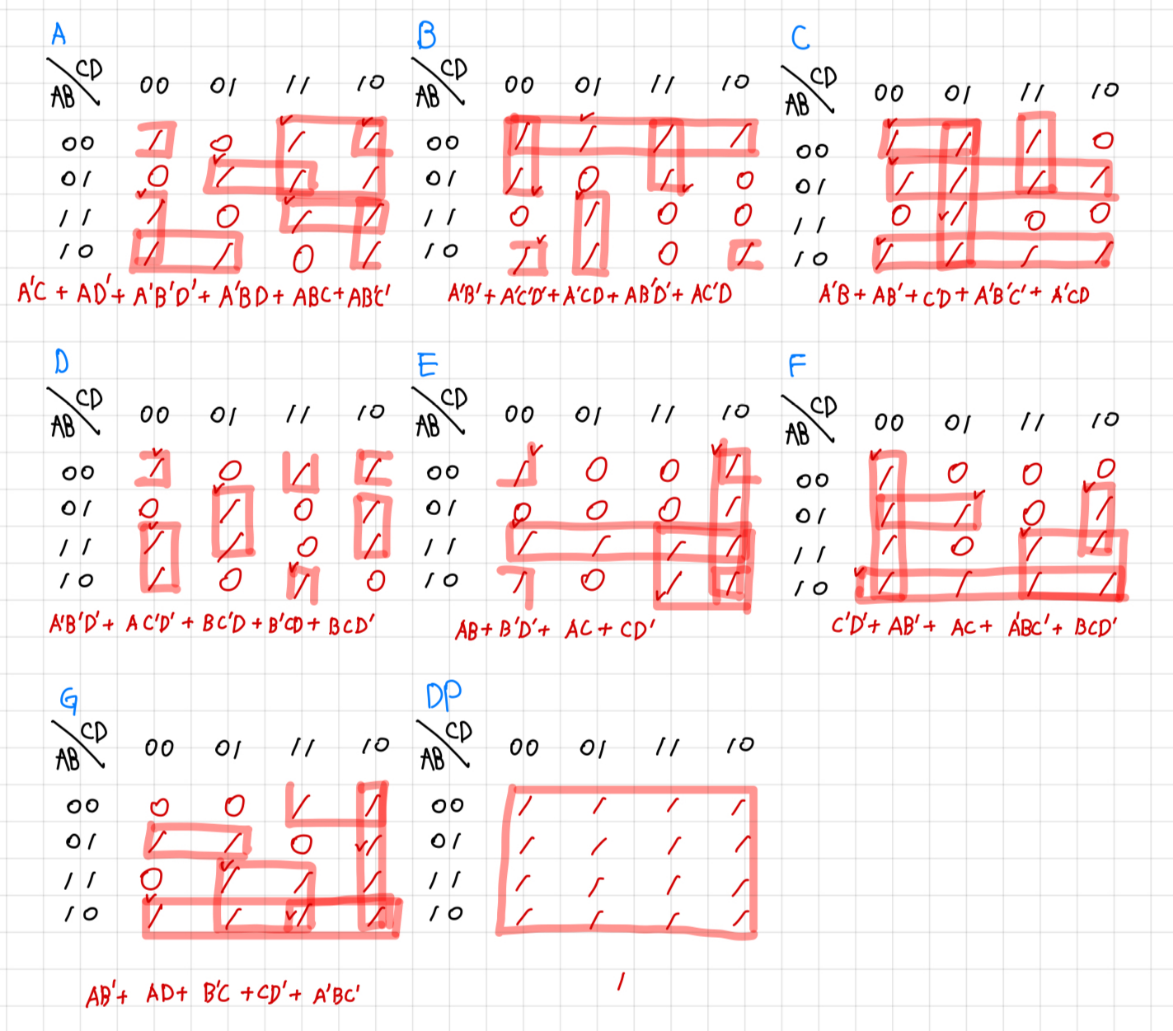
**2.**

**1) Truth table**

****

7 segment display의 각 segment에 대해 켜지고, 꺼져야 하는 상태를 truth table에 정리한다.

**2) K-map**

****

Truth table에서 얻은 데이터를 이용해 각 segment에 대한 k-map을 작성한다. 이러한 방식으로 k-map을 사용하면 각 segment를 켜고 끄는 최적의 논리식을 얻을 수 있다.

**3) Verilog**

*`timescale 1ns / 1ps*

*module boolean(*

*input aa,*

*input bb,*

*input cc,*

*input dd,*

*output a,*

*output b,*

*output c,*

*output d,*

*output e,*

*output f,*

*output g,*

*output dp,*

*output digit*

*);*

*assign a = ((~aa)&&(cc)) || (aa&&(~dd)) || ((~aa)&&(~bb)&&(~dd)) || ((~aa)&&bb&&dd) || (aa&&bb&&cc) || (aa&&(~bb)&&(~cc));*

*assign b = ((~aa)&&(~bb)) || ((~aa)&&(~cc)&&(~dd)) || ((~aa)&&cc&&dd) || (aa&&(~bb)&&(~dd)) || (aa&&(~cc)&&dd);*

*assign c = ((~aa)&&bb) || (aa&&(~bb)) || ((~cc)&&dd) || ((~aa)&&(~bb)&&(~cc)) || ((~aa)&&cc&&dd);*

*assign d = ((~aa)&&(~bb)&&(~dd)) || (aa&&(~cc)&(~dd)) || (bb&&(~cc)&&dd) || ((~bb)&&cc&&dd) || (bb&&cc&&(~dd));*

*assign e = (aa&&bb) || ((~bb)&&(~dd)) || (aa&&cc) || (cc&&(~dd));*

*assign f = ((~cc)&&(~dd)) || (aa&&(~bb)) || (aa&&cc) || ((~aa)&&bb&&(~cc)) || (bb&&cc&&(~dd));*

*assign g = (aa&&(~bb)) || (aa&&dd) || ((~bb)&&cc) || (cc&&(~dd)) || ((~aa)&&bb&&(~cc));*

*assign dp = 1;*

*assign digit = a || b || c || d || e || f || g;*

*endmodule*

k-map을 통해 얻은 논리식을 Verilog 코드로 구현하여 7 segment display가 0-F까지의 숫자와 문자를 표시하도록 한다. 이 때 digit 출력은 모든 segment 중 하나라도 켜져 있을 때 활성화된다.

**4) simulation**

**스크린샷, 소프트웨어, 멀티미디어 소프트웨어, 컴퓨터이(가) 표시된 사진

자동 생성된 설명**

Simulation 결과 올바르게 Verilog 코드를 작성했음을 확인할 수 있다. aa-dd까지가 입력한 이진코드를 나타내고, a-g까지가 7-segment display의 각 segment의 점등을 의미한다. 이후 fpga 보드에서 스위치를 조절하며 실제로 7 segment display에서 숫자와 문자를 표시하는지 확인한다.

**5) 7 segment display 결과**

모든 입력 조합에 대해 숫자와 문자가 올바르게 표시되는 것을 확인할 수 있다.

**3.**

실험에서 truth table과 k-map을 이용하여 7 segment display의 각 segment에 대한 논리식을 최적화하였다. 하지만 여전히 논리식이 복잡하므로 추가적인 회로 최적화가 가능한지 검토해볼 수 있다. 이때 회로 최적화가 중요한 이유는 회로에서 게이트의 수가 적을수록 회로의 크기가 작아지고 전력 소비와 gate delay가 줄어들기 때문이다. 논리식을 단순화하기 위해 중복된 조건들(ex. aa && bb, aa && ~bb, ~aa && bb, ~aa && ~bb)을 미리 계산하여 이를 재사용함으로써 효율을 높일 수 있다.

**4.**

**Multiplexing** : 7 segment display로 여러 자리 숫자를 표시하는 기법이다. 이는 다중 7 segment display를 사용하는 디지털 시계 또는 전자 계산기에 활용되는 방법이다. 특히 time division multiplexing을 통해 각 7 segment display에 빠르게 전환 신호를 보내면서 각 자리의 숫자를 순서대로 표시한다. 하지만 인간의 눈은 이 속도가 매우 빠른 전환을 감지하지 못하므로 여러 자리의 숫자가 동시에 표시되는 것처럼 보이는 것이다.