8주차 결과보고서

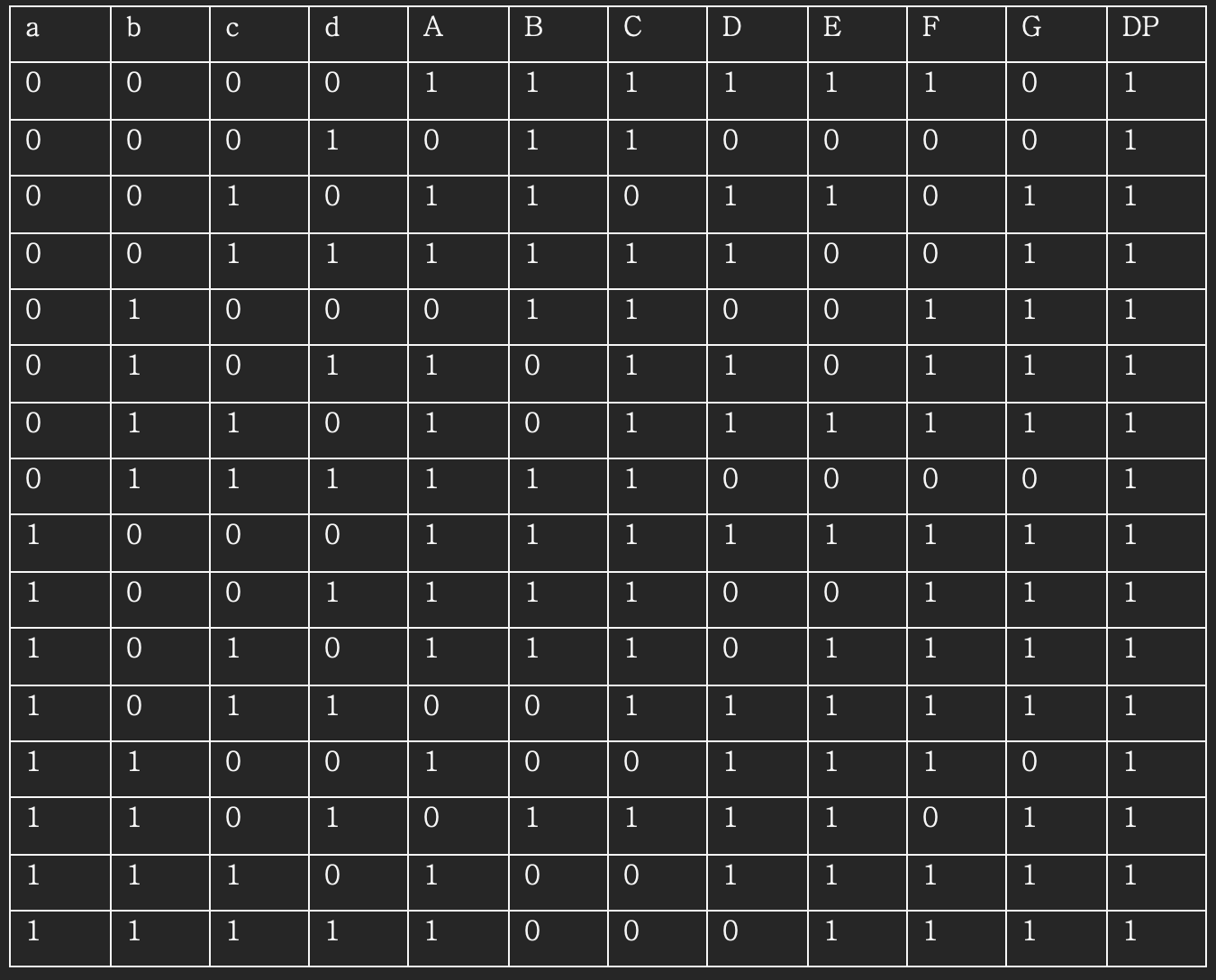
전공: 심리학과 학년: 3학년 학번: 20190345 이름: 김동현

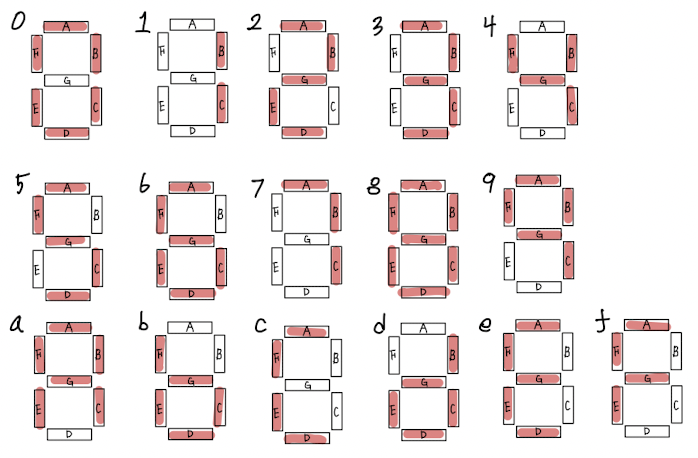
**1.**

7-segment display를 베릴로그를 통해 구현한다. 진리표와 카르노맵을 통해 논리식을 간소화하여 구현하며, 시뮬레이션과 FPGA를 비교하며 회로의 동작을 검증한다.

2.

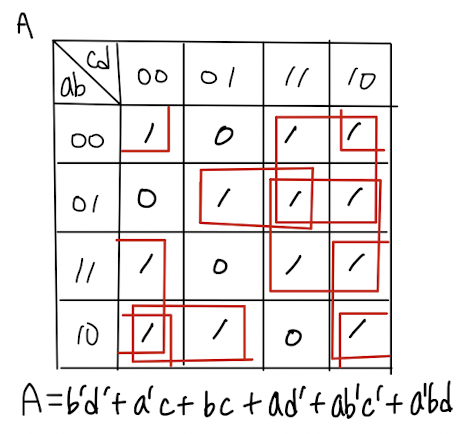
7 segment display에 대한 진리표이다. 4개의 입력 변수 a,b,c,d의 값에 따라 표현해야 하는 숫자의 segment 부분을 표시한다. 불이 들어와야 하는 부분은 1을, 불이 들어오지 않아야 하는 부분은 0의 값을 가진다. Dp값은 소수점을 나타내는 값이지만, 본 실험에서는 항상 불이 들어오도록 하기 위해 1로 설정한다.



ㅌ

진리표에 따라 불이 들어온다면, 위 그림과 같이 FPGA의 7-segment display에 표현될 것이다.

각 출력 변수를 카르노맵을 통해 Sum of product 형식으로 논리식을 간소화 하면 다음과 같다.

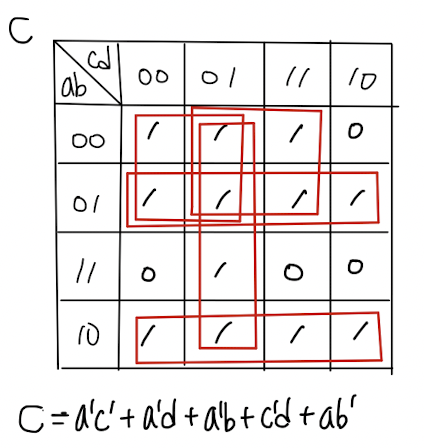


A=(~b&~d)|(~a&c)|(b&c)|(a&~d)|(a&~b&~c)|(~a&b&d)

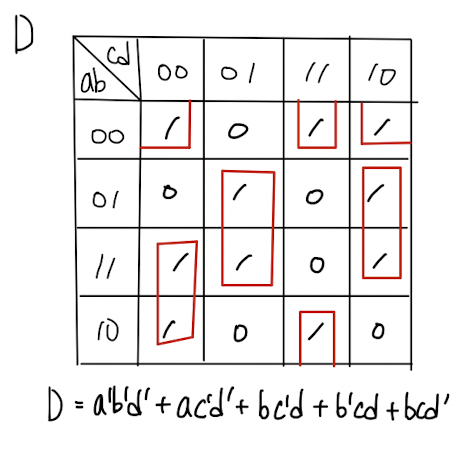
광장이(가) 표시된 사진

자동 생성된 설명

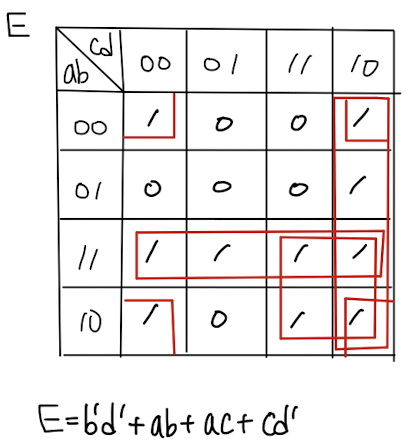
B=(~b&~d)|(~a&~b)|(~a&~c&~d)|(a&~c&d)|(~a&c&d)



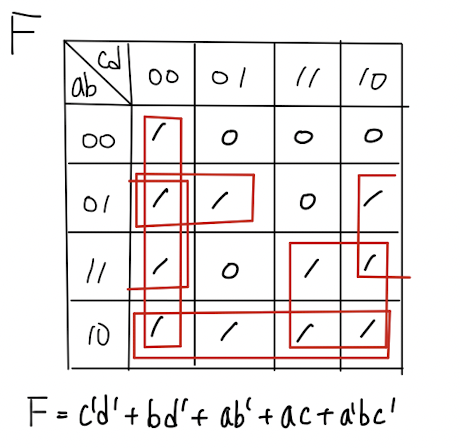
C=(~a&~c)|(~a&d)|(~a&b)|(~c&d)|(a&~b)



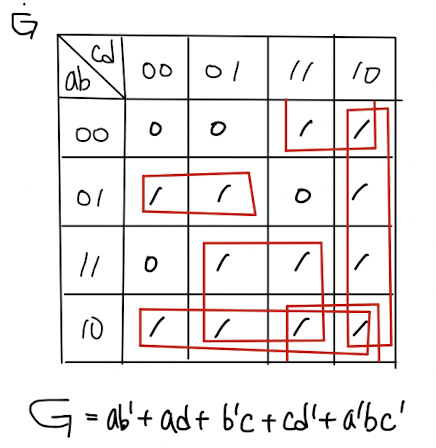
D=(~a&~b&~d)|(a&~c&~d)|(b&~c&d)|(~b&c&d)|(b&c&~d)



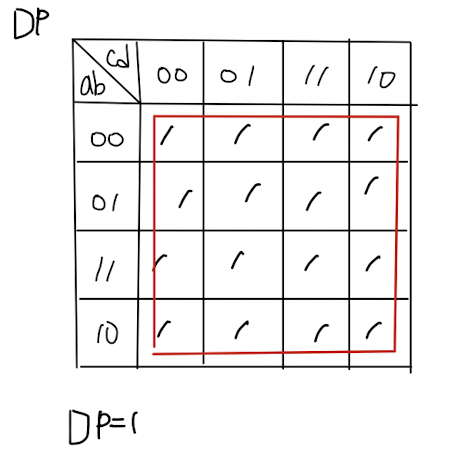
E=(~b&~d)|(a&b)|(a&c)|(c&~d)



F=(~c&~d)|(b&~d)|(a&~b)|(a&c)|(~a&b&~c)



G=(a&~b)|(a&d)|(~b&c)|(c&~d)|(~a&b&~c)



DP=1

카르노 맵을 통해 간소화 한 논리식을 바탕으로 베릴로그 내에서 7-segment display를 구현했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

7-segment display의 디자인 소스 파일이다. 입력변수 4개(a,b,c,d)와 출력변수 9개(A,B,C,D,E,F,G,DP,digit)을 선언하였다. A,B,C,D,E,F,G에는 입력변수 a,b,c,d를 활용하여 간소화한 논리식을 할당하며, DP변수에는 소수점이 항상 켜지도록 하기 위해 1로 설정한다. Digit변수는 7개의 segment를 or연산하여 할당한다. Digit 변수를 통해 FPGA의 7 segment display가 표현될 자리수를 지정한다.

테이블이(가) 표시된 사진

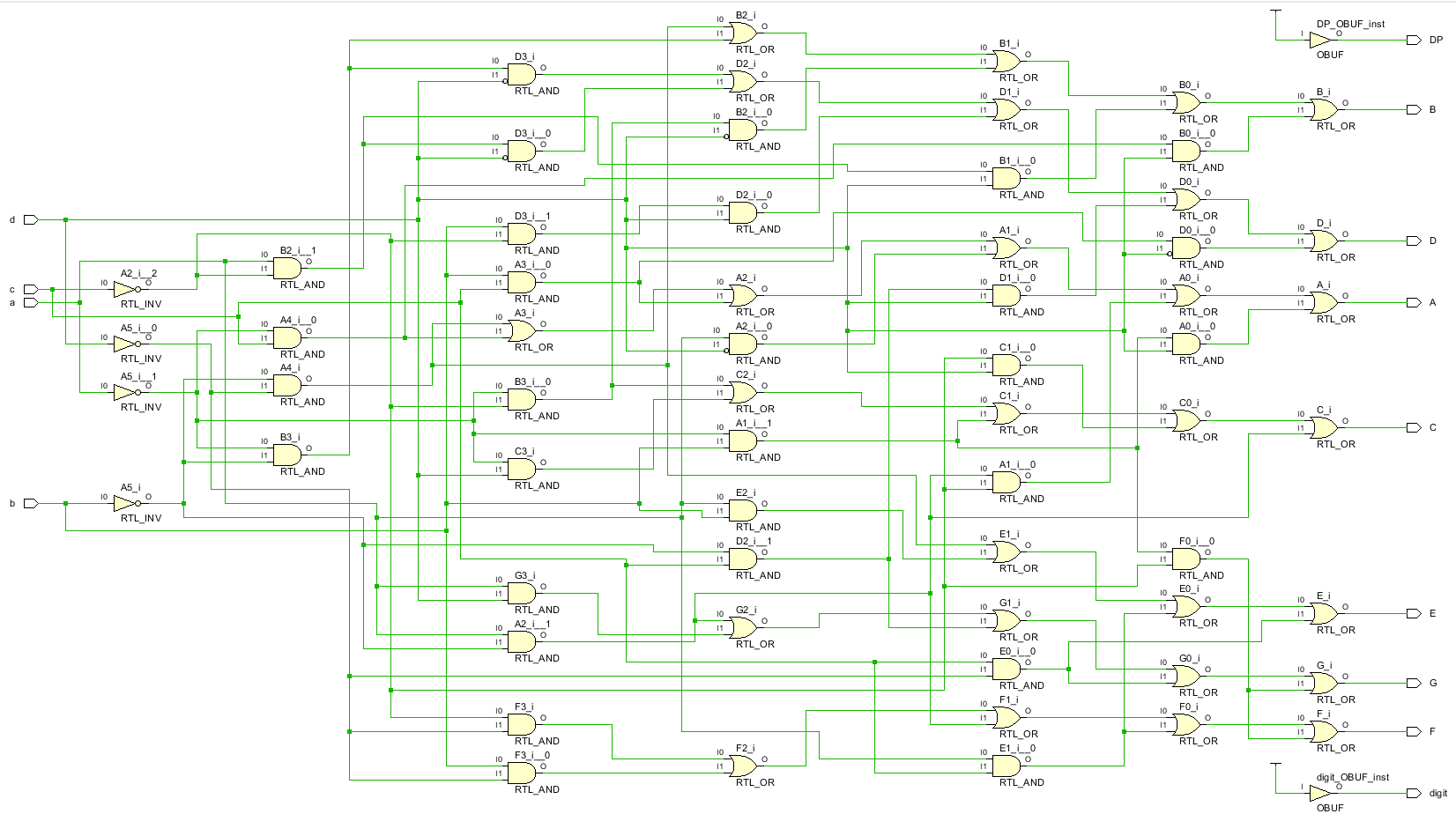
자동 생성된 설명

7 segment display의 시뮬레이션 소스 파일이다. Reg변수와 wire변수를 각각 할당하여 입력변수 4개와 출력변수 9개를 각각 할당한다. 이후 입력변수를 클럭에 따라 변화하도록 설정한다.

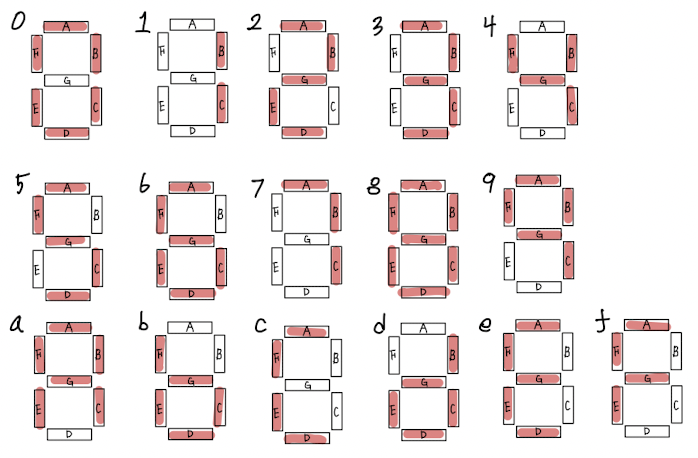
녹색, 화물컨테이너이(가) 표시된 사진

자동 생성된 설명

시뮬레이션 코드에 따른 시뮬레이션 결과이다. Reg변수 aa,bb,cc,dd에 따라 표현하고자 하는 숫자(0~f)가 표현되기 위해 켜져야 하는 wire변수의 값을 보여주고 있다.



7 segment display의 schematic이다. And, or, not 연산자를 활용하여 논리식을 간소화하였다.



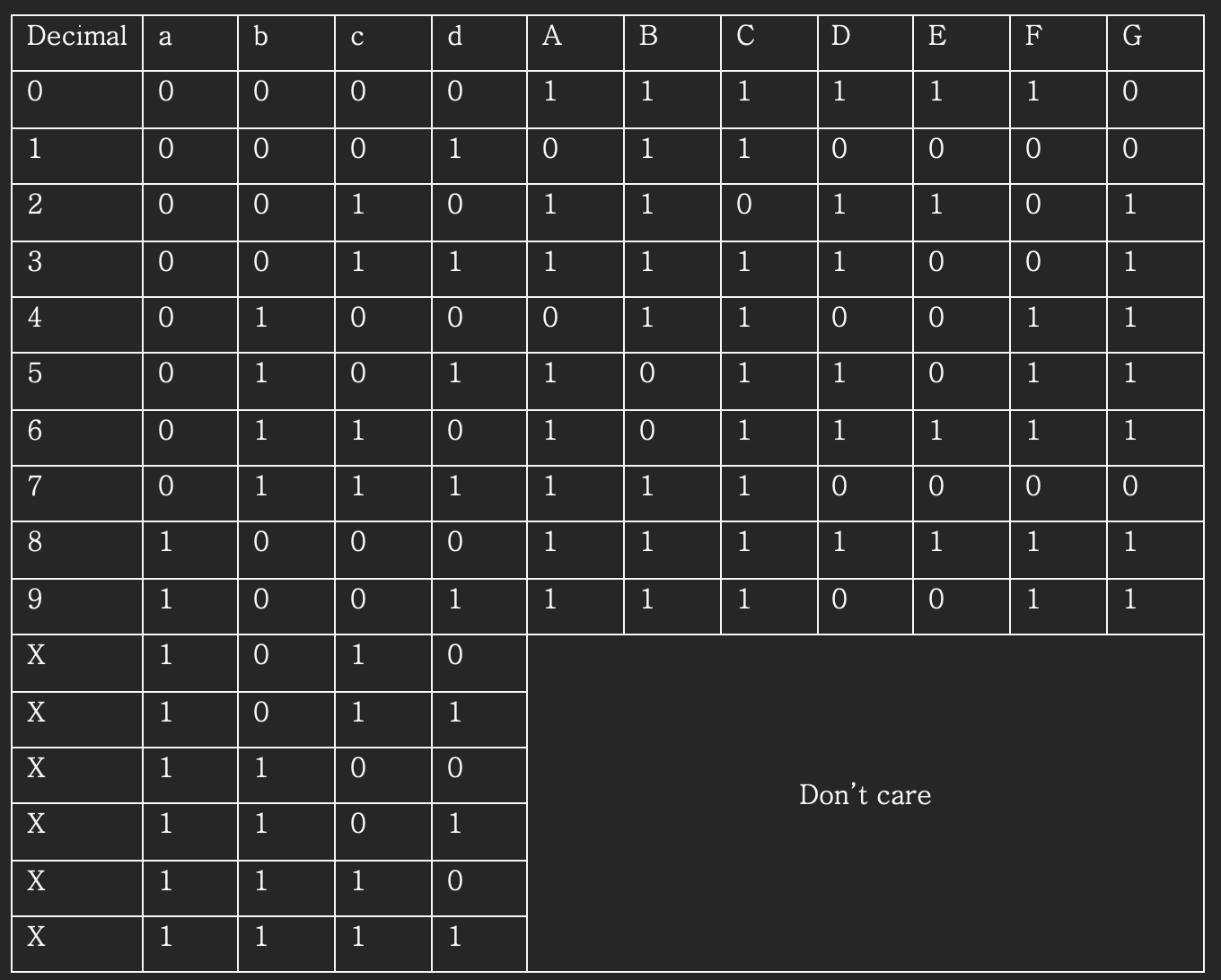
이후 FPGA를 통해 결과를 확인하였을 때, 위 그림과 같이 입력에 맞는 문자의 모양이 출력되는 것을 확인하였다. i/o port를 설정하는 과정에서 기존과 다르게 segment를 나타내는 핀에 포트 번호를 설정하였다. A-D20, B-C20, C-C22, D-B22, E-B21, F-A21, G-E22, DP-D22, digit-D16에 설정하였다.

3.

7 segment display를 구현하였다. 4개의 입력변수를 통해 16개의 문자(0~9,A~F)를 segment의 점등 여부에 따라 문자를 표현했다. 입력변수에 대한 segment의 식을 진리표와 카르노맵을 이용하여 구성하고 간소화했다. 논리식을 활용하여 디자인 소스 파일과, 시뮬레이션 소스 파일을 작성하였으며, 시뮬레이션 결과와 schematic를 통해 구현 결과를 확인하였다. 이후, synthesis과정과, implementation과정을 통해 포트를 할당하였으며, bitstream을 생성하였다. FPGA내 7 segment display를 활용하여 시뮬레이션 결과와 비교하엿으며, 입력값에 따른 알맞은 segment가 점등되는 것을 확인하였다.

4.

위 실험에서는 0부터 F까지 총 16개의 문자를 표현하였다. 만약 10진수만을 표기하고자 한다면, 0부터 9까지 총 10개만을 표기하게 된다. 이로 인해 6개 케이스가 낭비될 수 있지만, 카르노맵을 활용한 논리식 구현에서는 don’t care를 활용하여 보다 간편하게 구현할 수 있다.



위와 같은 진리표를 통해 카르노맵을 구성하면, 다음과 같이 논리식이 간소화 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

A에 대한 카르노맵을 수식화하면 A=a+c+b’d’+bd로 표현할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

B에 대한 카르노맵을 수식화하면 B=a+b’d’+c’d’+b’d+cd로 표현할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

C에 대한 카르노맵을 수식화하면 C=c’+a’b+cd로 표현할 수 있다.

텍스트이(가) 표시된 사진

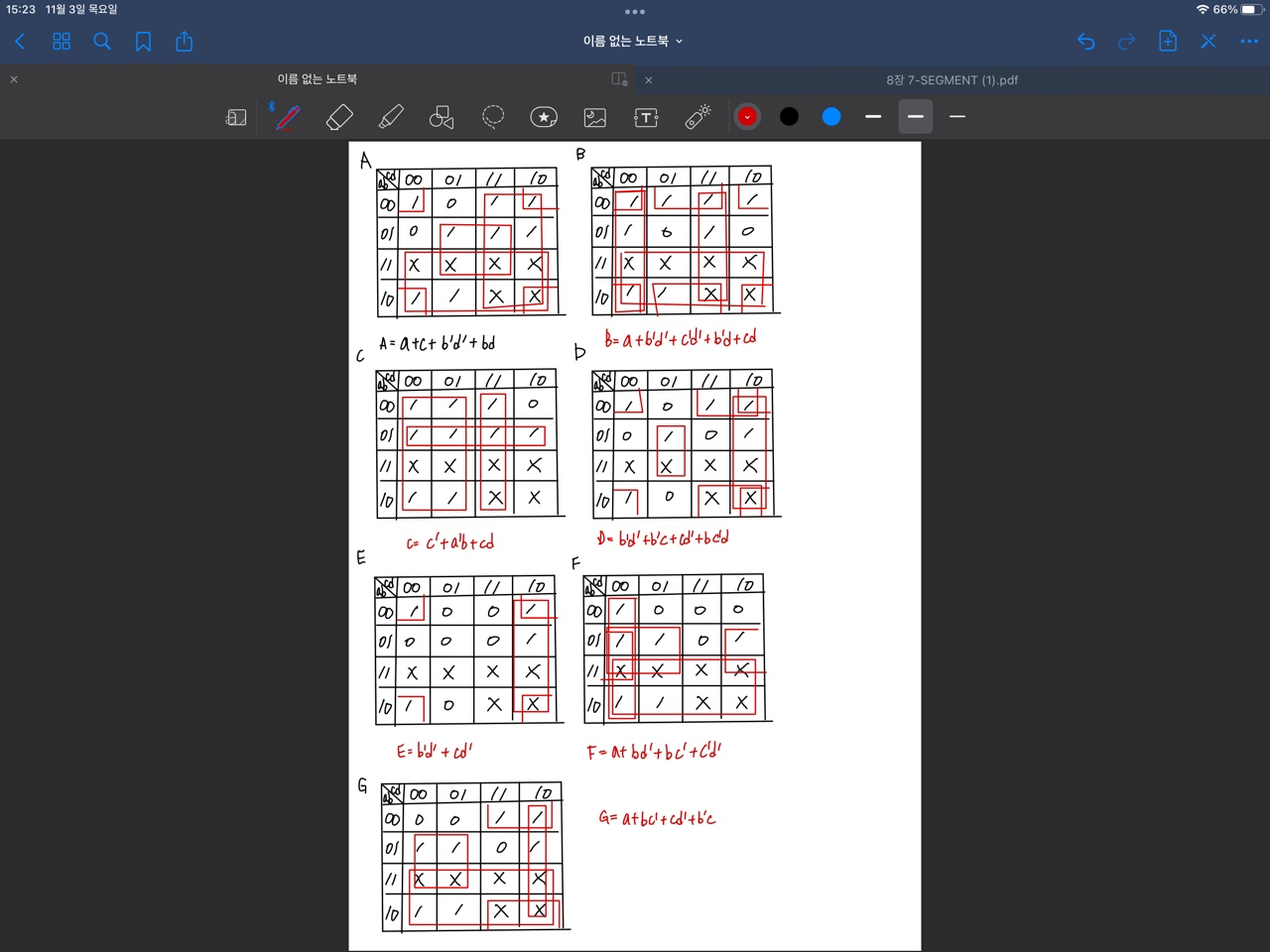
자동 생성된 설명

D에 대한 카르노맵을 수식화하면 D=b’d’+b’c+cd’+bc’d로 표현할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

E에 대한 카르노맵을 수식화하면 E=b’d’+cd’로 표현할 수 있다.



F에 대한 카르노맵을 수식화하면 F=a+bd’+bc’+c’d’로 표현할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

G에 대한 카르노맵을 수식화하면 G=a+bc’+cd’+b’c로 표현할 수 있다.

16개의 변수를 나타내는 경우보다 10개의 변수를 나타내는 경우 더 간소화된 논리식으로 표현할수 있음을 알 수 있다. 이는 don’t care의 활용으로 인해 보다 더 간소화하여 카르노맵을 구성할 수 있기 때문이다.