7주차 예비보고서

전공: 심리학과 학년: 3학년 학번: 20190345 이름: 김동현

**1.**

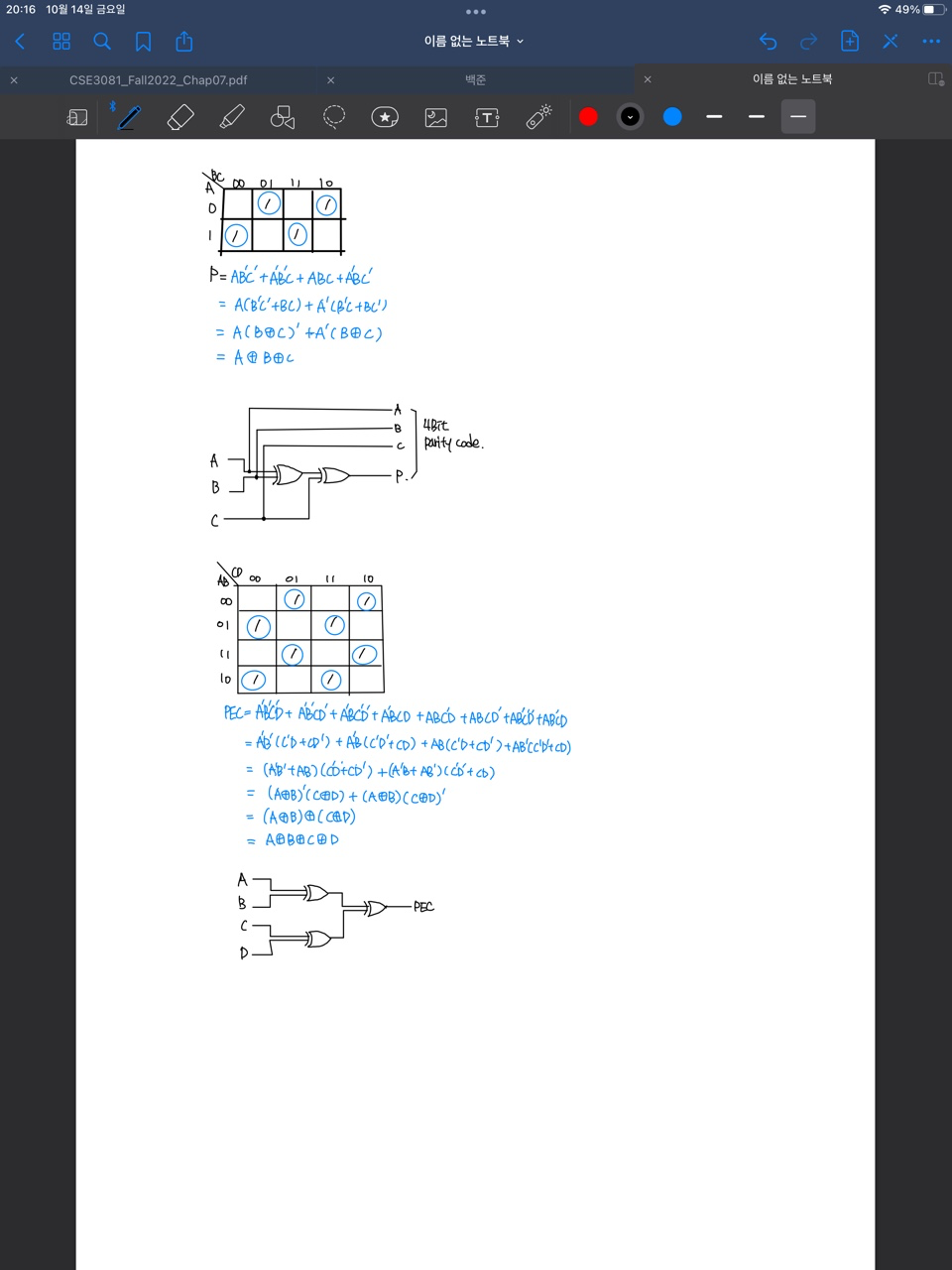
Parity bit는 비트에서의 오류를 감지하기 위해 사용되는 비트이다. Parity bit 생성기는 이러한 parity bit를 생성하는 조합 논리 회로이다. 비트들 중 1의 개수에 따라 parity bit를 결정하여 parity bit를 포함한 정보에서 1의 개수가 항상 짝수가 되도록 한다. 1인 비트가 홀수 개이면 parity bit는 1이, 짝수 개이면 parity bit는 0이 된다. 이러한 parity bit 생성기는 xor 연산의 특징인 1의 개수가 짝수인지 홀수인지에 따른 특징과 유사하며, 실제로 xor연산을 사용하여 parity bit를 생성한다. 하지만 홀수개의 비트에 대한 오류 여부만 확인할 수 있고, 짝수개의 비트가 바뀌더라도 parity bit는 정상이므로, 오류여부를 확인하지 못한다.

실제 3bit의 정보에 대해 parity bit를 생성하면 다음과 같다.

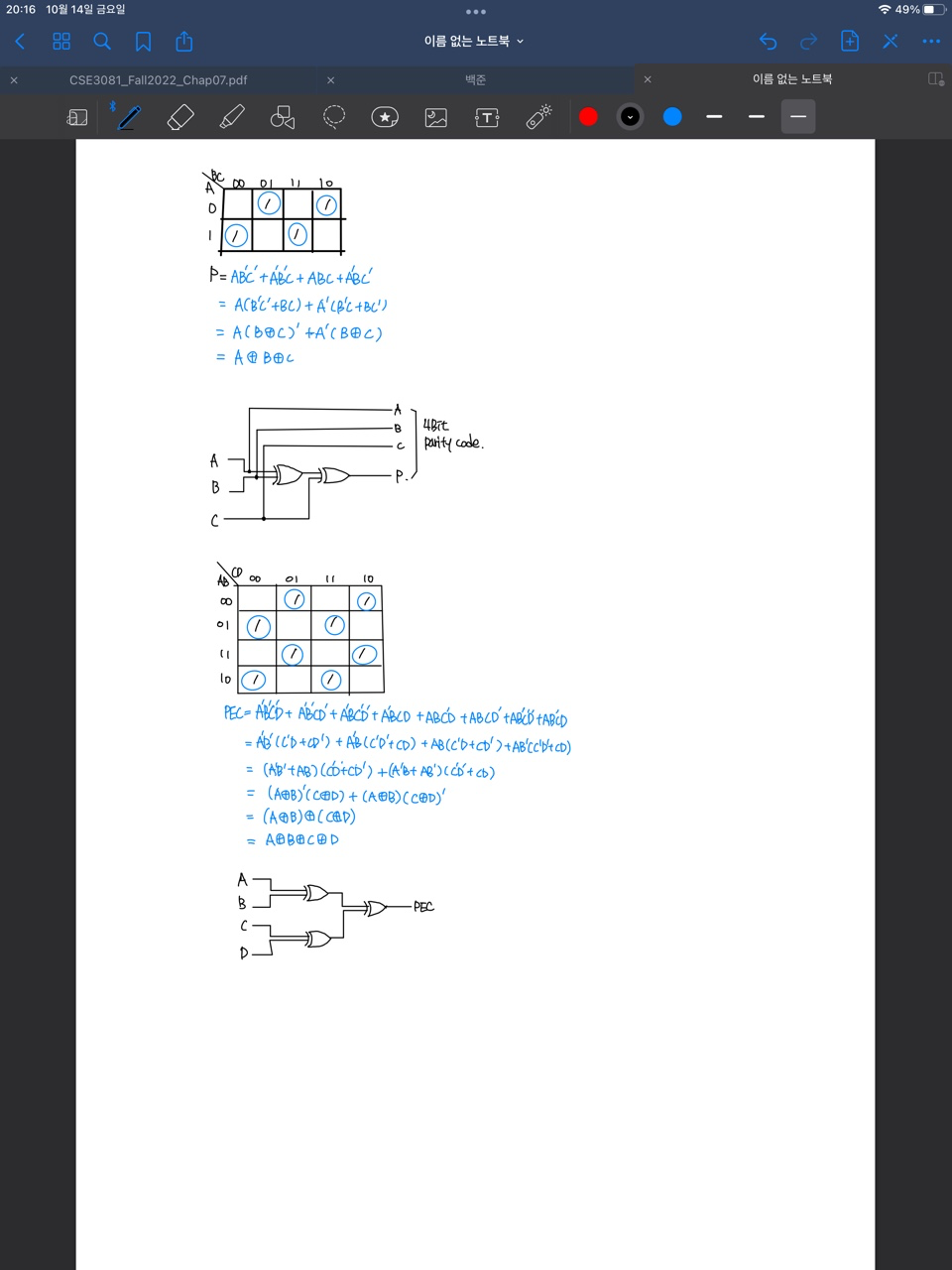
테이블이(가) 표시된 사진

자동 생성된 설명

진리표를 바탕으로 3비트 parity bit에 대한 카르노맵을 그리면 다음과 같다.



카르노맵을 바탕으로 얻은 논리식을 단순화하면, parity bit(p)=A^B^C로, 3개의 비트에 대한 xor 연산임을 알 수 있다.

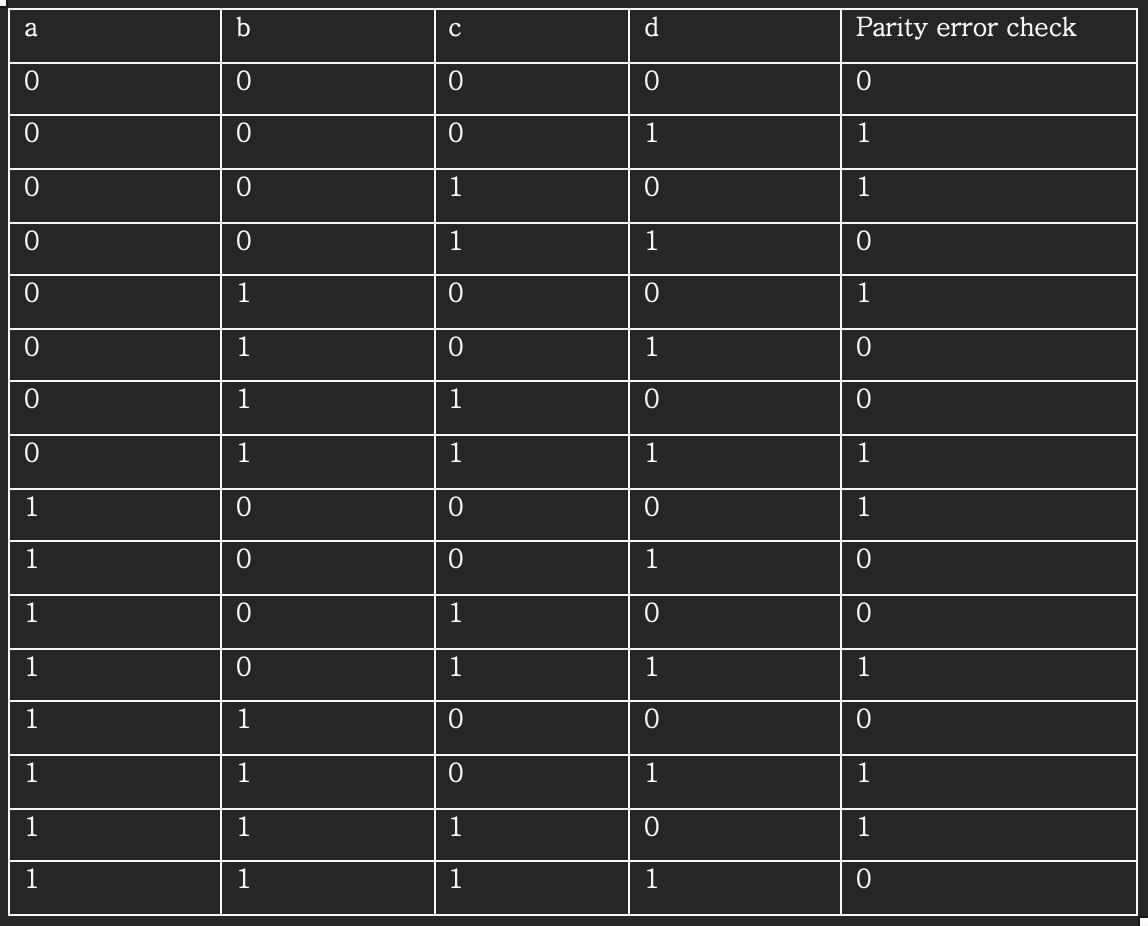


3비트 parity bit 생성기의 diagram이다. 3개의 입력 비트와 parity bit가 더해져, 4비트 parity code가 생성되었다.

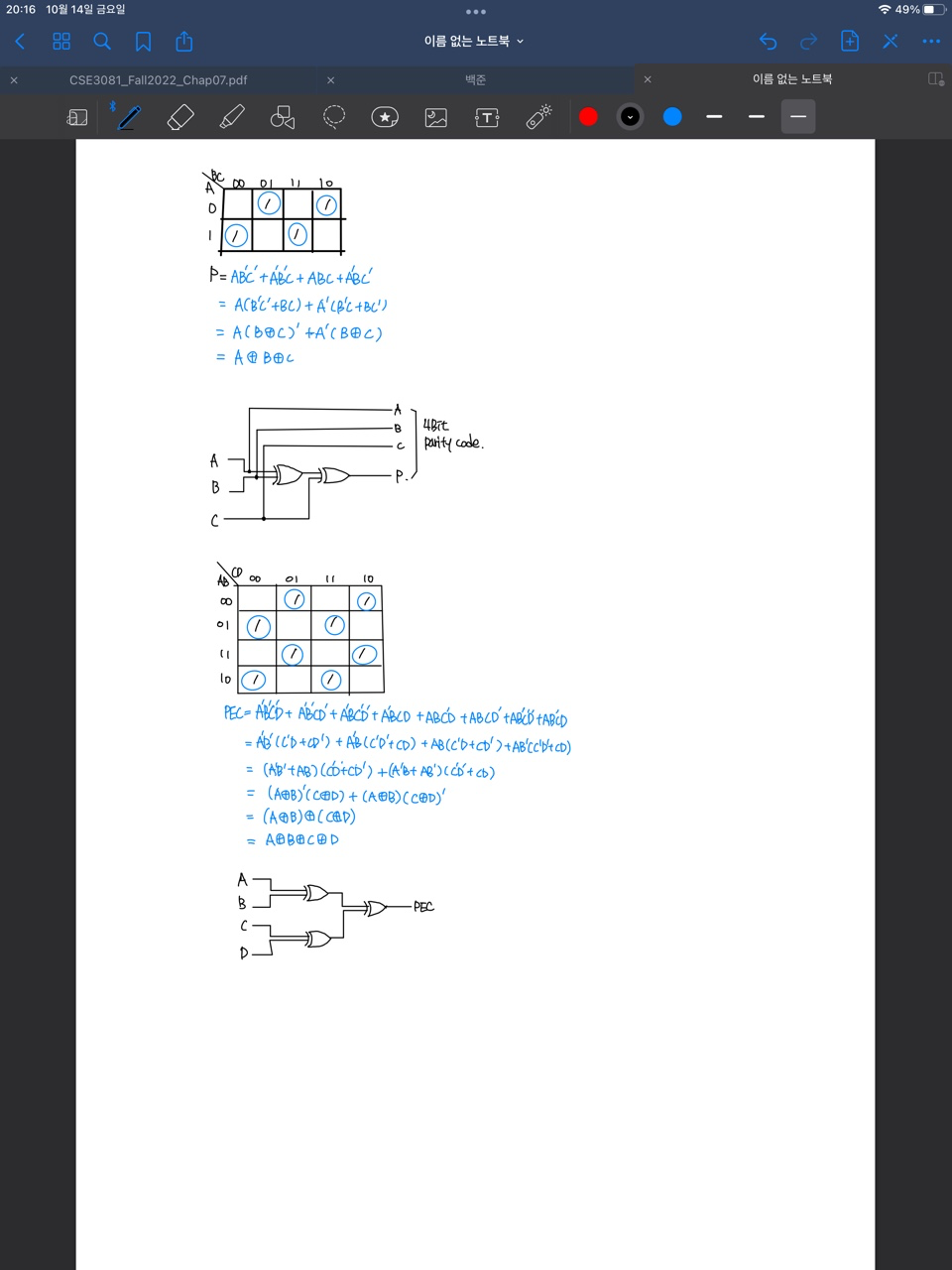
2.

Parity bit 검사기는 수신기를 통해 받은 정보에서 parity에 따른 오류 여부를 확인하는 회로이다. 검사 부호의 종류에 따라 오류 여부가 결정된다. 짝수 parity bit 검사기의 경우 점검하는 비트들의 1의 개수가 짝수이면 정상으로, 홀수이면 오류로 결정된다. 반대로 홀수 parity bit 검사기의 경우 점검하는 비트의 1의 개수가 홀수이면 정상으로, 짝수이면 오류로 결정된다.

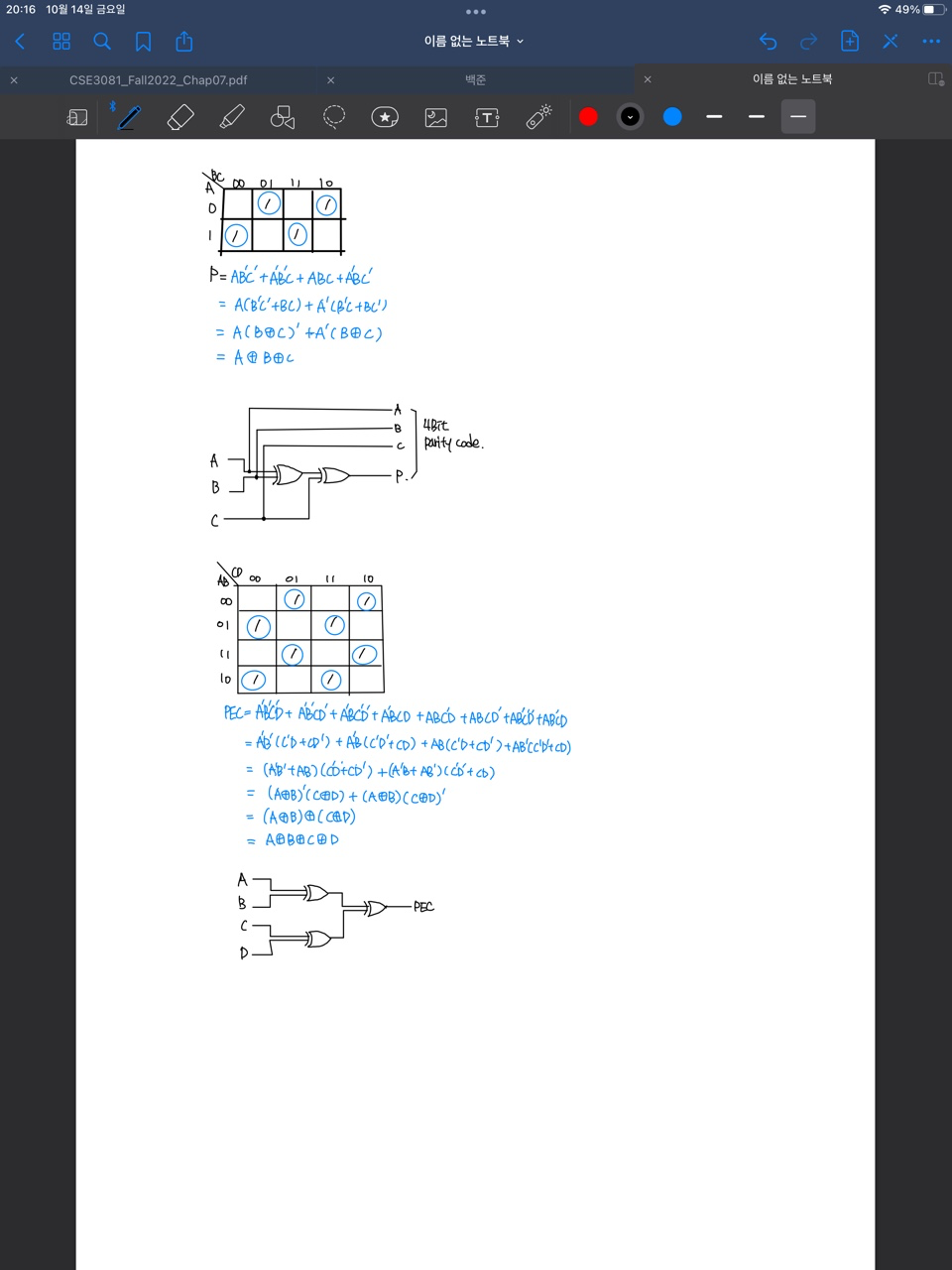
3비트 정보와 1비트의 parity bit를 포함하는 4비트의 정보에 대한 parity bit 검사기는 다음과 같이 작동한다. 짝수 parity bit 검사기의 경우, 4비트의 1의 개수가 짝수이면 parity error check 값으로 0을, 홀수이면 1을 반환한다.



위는 짝수 parity bit 검사기의 진리표이다. Bit가 1인 값들의 개수가 짝수이면 parity error check값으로 0을, 홀수이면 1을 반환하는 것을 알 수 있다.

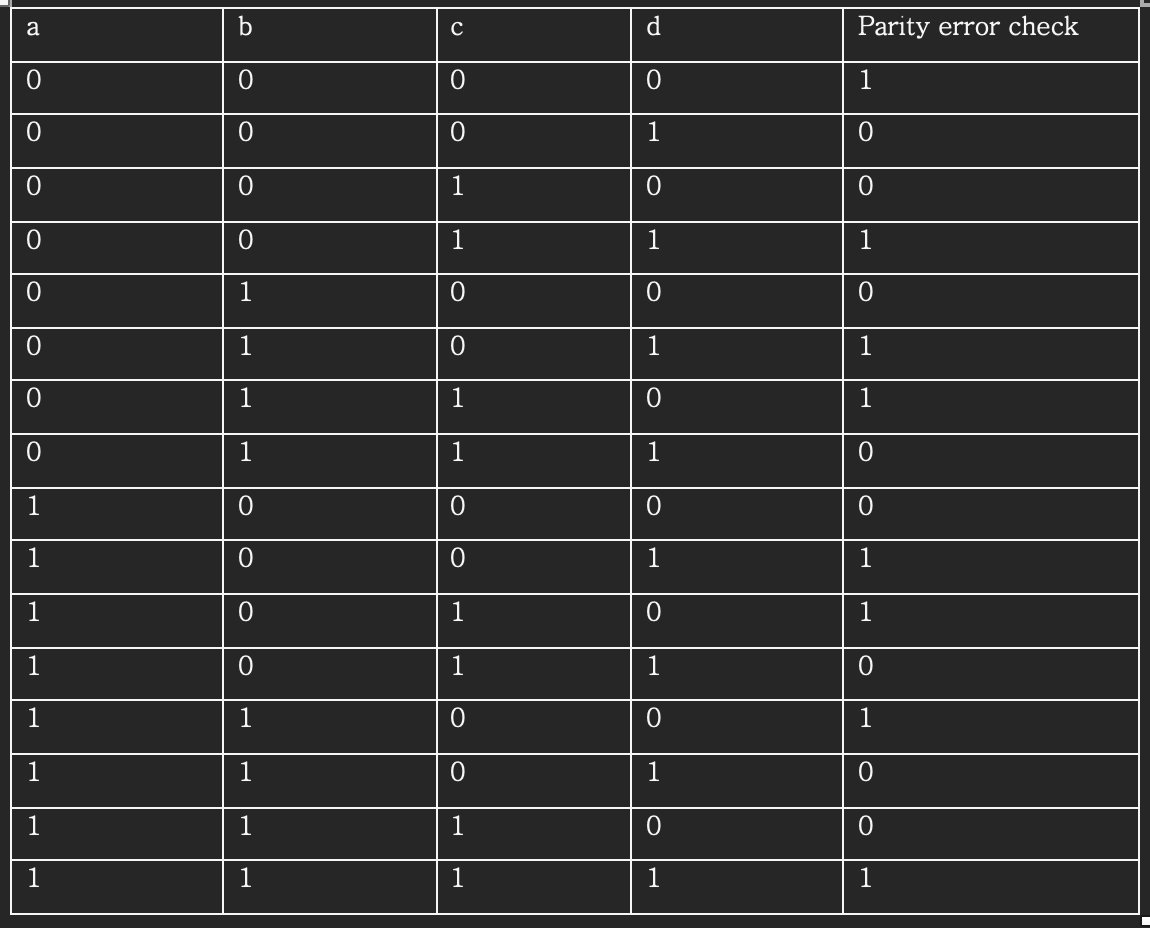


진리표를 바탕으로 짝수 parity bit 검사기의 카르노맵을 그리면 다음과 같다.



카르노맵을 통해 얻은 논리식을 단순화하면, Parity Error Checker(PEC)=A^B^C^D로, 4비트의 xor연산임을 알 수 있다.

홀수 parity bit 검사기의 경우, 4비트의 1의 개수가 홀수이면 parity error check 값으로 0을, 짝수이면 1을 반환한다.



위는 홀수 parity bit 검사기의 진리표이다. Bit가 1인 값들의 개수가 홀수이면 parity error check값으로 0을, 짝수이면 1을 반환하는 것을 알 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

진리표를 바탕으로 홀수 parity bit 검사기의 카르노맵을 그리면 다음과 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명

카르노맵을 통해 얻은 논리식을 단순화하면, Parity Error Checker(PEC)=A xnor B xnor C xnor D로, 4비트의 xnor연산임을 알 수 있다.

3.

Parity bit 이외에 오류 검출기 및 정정기로 hamming code가 있다. Parity code가 오류를 검출할 수만 있다면, hamming code는 오류 검출은 물론 오류 정정까지 가능하다.

4비트의 데이터를 hamming code와 함께 전송받는 예시는 다음과 같다. 4비트의 데이터와 3비트의 해밍 코드로 총 7비트를 수신한다. 0010011과 같은 7개의 비트를 수신받으면, 다음과 같은 식을 연산한다.

A1=a3^a5^a7

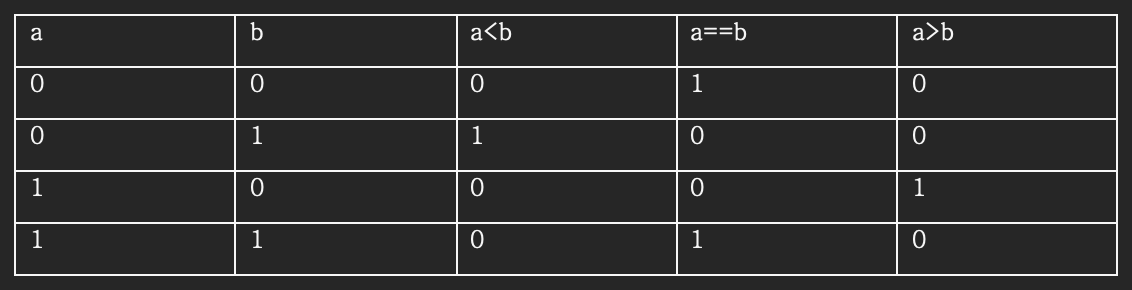
A2=a3^a6^a7

A4=a5^a6^a7

A1= 0, a2=1, a4=0이라는 결과가 나온다. 0으로 출력된 a1, a4가 오류이므로, 비트를 0110011로 수정한다. 데이터 비트인 a3,a5,a6,a7은 1011이므로 전송하고자 한 코드가 오류일 경우 다음과 같이 정정 가능하다.

4.

N비트 비교기는 두 수를 비교하여 두 수의 대소관계를 나타내는 논리 회로이다. Xor와 xnor 연산을 이용하여 구현할 수 있다.



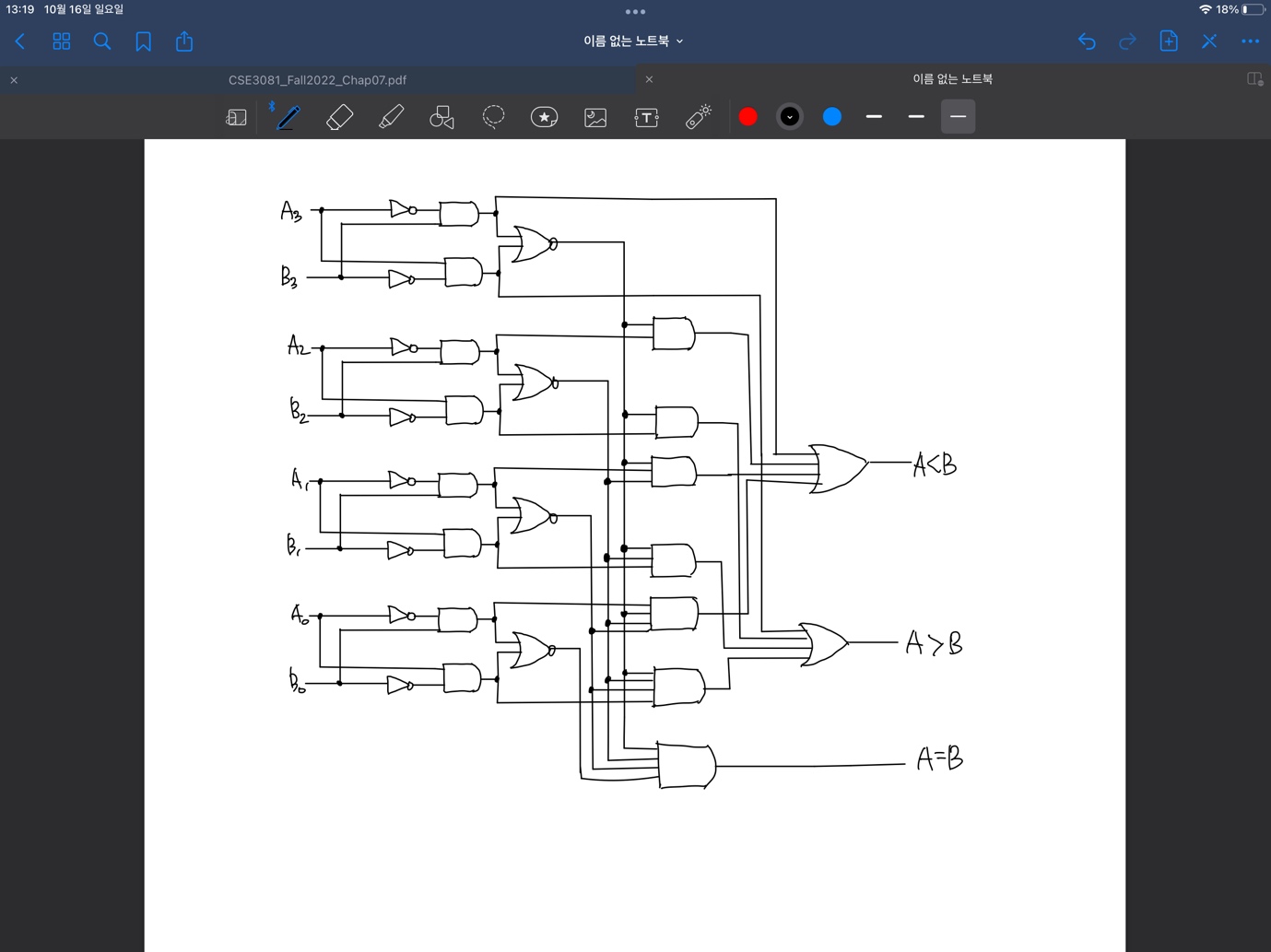
이는 1비트 비교기의 진리표이다. 참인 결과는 1을 거짓인 결과는 0을 반환한다. 진리표를 토대로 4비트의 수(a=a4a3a2a1과 b=b4b3b2b1)에 대해 다음과 같은 논리식을 구성할 수 있다.

a>b=a4>b4 | (a4==b4 & a3>b3) | (a4==b4 & a3==b3 & a2>b2) | (a4==b4 & a3==b3 & a2==b2 & a1>b1)

a<b= a4<b4 | (a4==b4 & a3<b3) | (a4==b4 & a3==b3 & a2<b2) | (a4==b4 & a3==b3 & a2==b2 & a1<b1)

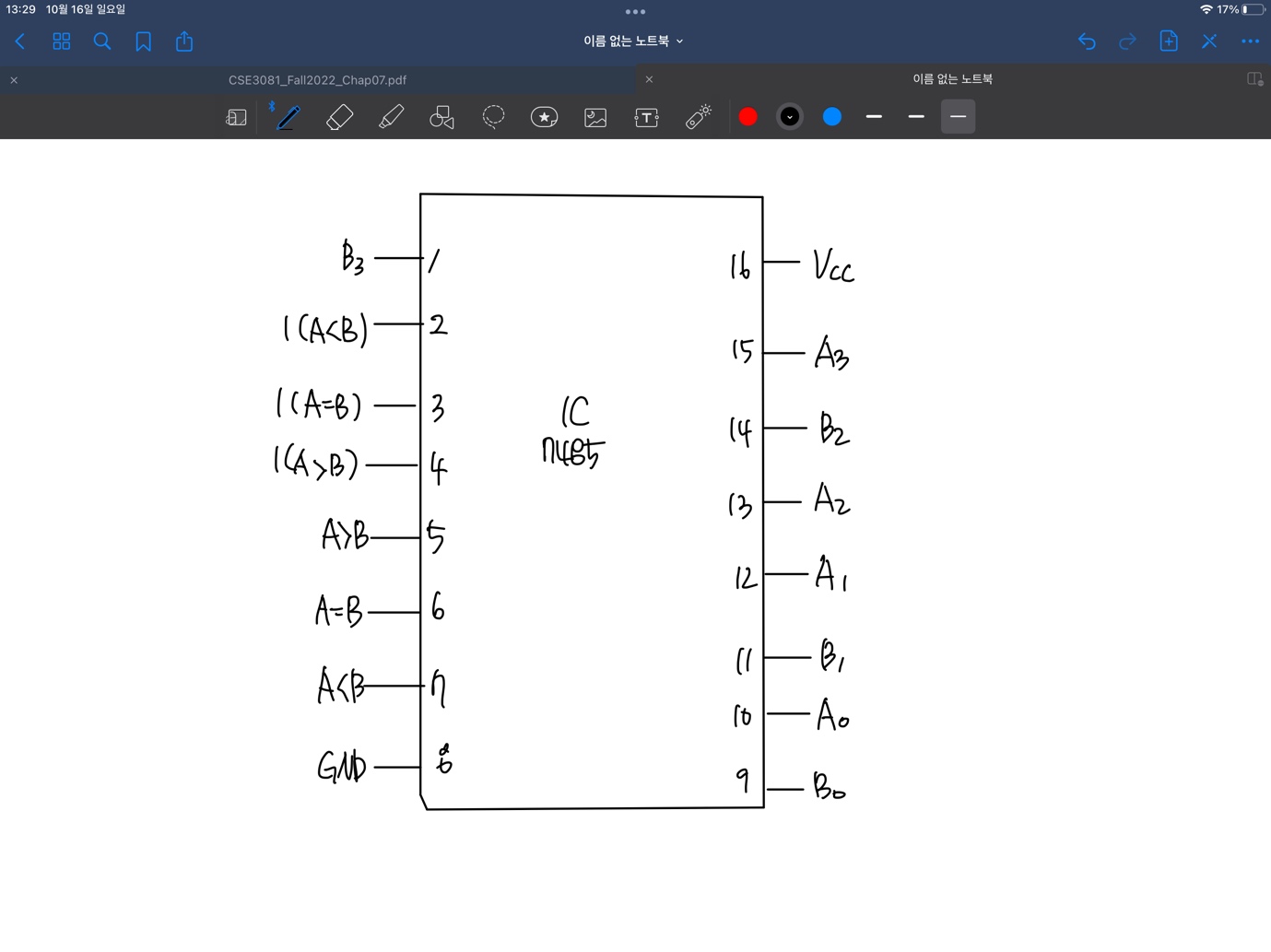
a==b = a4==b4 & a3==b3 & a2==b2 && a1==b1

이를 토대로 다음과 같은 회로를 구성할 수 있다.



회로를 확장 혹은 축소하여 n비트에 대한 비교기를 구성할 수 있다.

5.   
IC 7485는 입력과 출력이 종속적으로 연결된 4비트 비교기로서 입력에 알맞은 출력을 내보낸다. 아래 그림은 IC7485의 pin diagram을 보여준다.



다음 진리표는 IC7485의 입력에 따른 출력(5,6,7번 핀)의 값을 보여준다.

테이블이(가) 표시된 사진

자동 생성된 설명

이 회로를 cascading하게 연결하여 4\*n비트의 비교기를 구성할 수 있다.

6.

위에서 언급한 IC 7485를 cascading하여 8비트의 비교기를 구현할 수 있다. 하위 수준의 ic7485에서 나온 a<b, a>b, a=b의 값을 상위 수준의 ic 7485의 I(a<b), I(a>b),I(a=b)의 입력으로 입력한다. 하위수준의 회로에는 a0a1a2a3, b0b1b2b3의 비트 값에 해당하는 값을, 상위수준의 회로에는 a4a5a6a7, b4b5b6b7의 비트 값에 해당하는 값을 입력한다. 아래 그림은 IC 7485 2개를 이용하여 8비트 비교기를 구현한 모습이다.

텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명

7. 참고 문헌

Alan B. Marcvitz, Introduction to Logic Design, McGraw-Hill, 2010

M. Lentmaier and K. S. Zigangirov, "On generalized low-density parity-check codes based on Hamming component codes," in IEEE Communications Letters