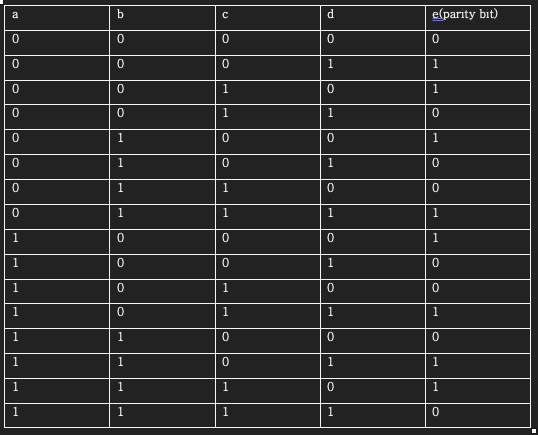
7주차 결과보고서

전공: 심리학과 학년: 3학년 학번: 20190345 이름: 김동현

**1.**

Even parity bit는 입력값 중 1의 개수가 홀수이면 parity bit를 1로, 짝수이면 parity bit를 0으로 설정한다. 다음은 4개의 입력과 이에 따른 parity bit를 진리표로 나타낸 표이다.



진리표를 토대로 카르노 맵을 구성하였다.

테이블이(가) 표시된 사진

자동 생성된 설명

식을 정리하면 다음과 같다.

E=a’b’c’d+a’b’cd+a’bc’d’+a’bcd+abc’d+abcd’+ab’c’d’+ab’cd

=a’b’c⊕d+a’b(c⊕d)’+abc⊕d+ab’(c⊕d)

=c⊕d(a⊕b)’+a⊕b(c⊕d)’

=a⊕b⊕c⊕d

위의 식을 토대로, 베릴로그를 이용하여 구현하였다.

텍스트이(가) 표시된 사진

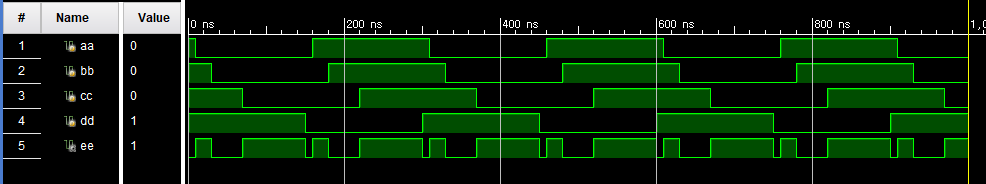
자동 생성된 설명

디자인 소스 코드이다. 입력 변수 a,b,c,d와 출력 변수 e를 선언하였다. E는 입력변수들에 대한 parity bit를 나타내며, 입력변수들의 xor 연산을 할당한다.

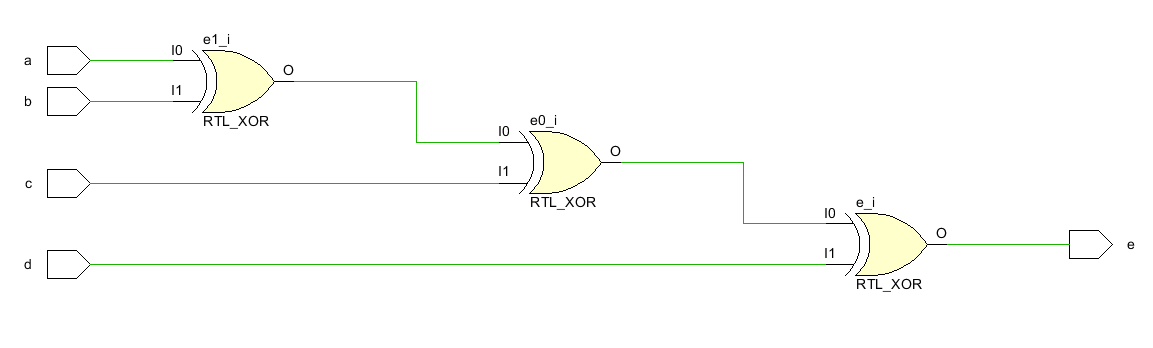
테이블이(가) 표시된 사진

자동 생성된 설명

시뮬레이션 소스 코드이다.

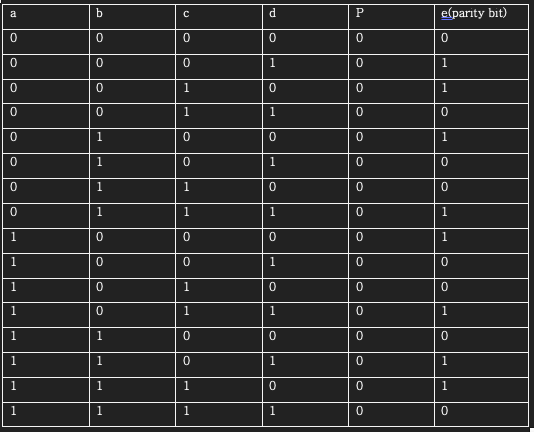


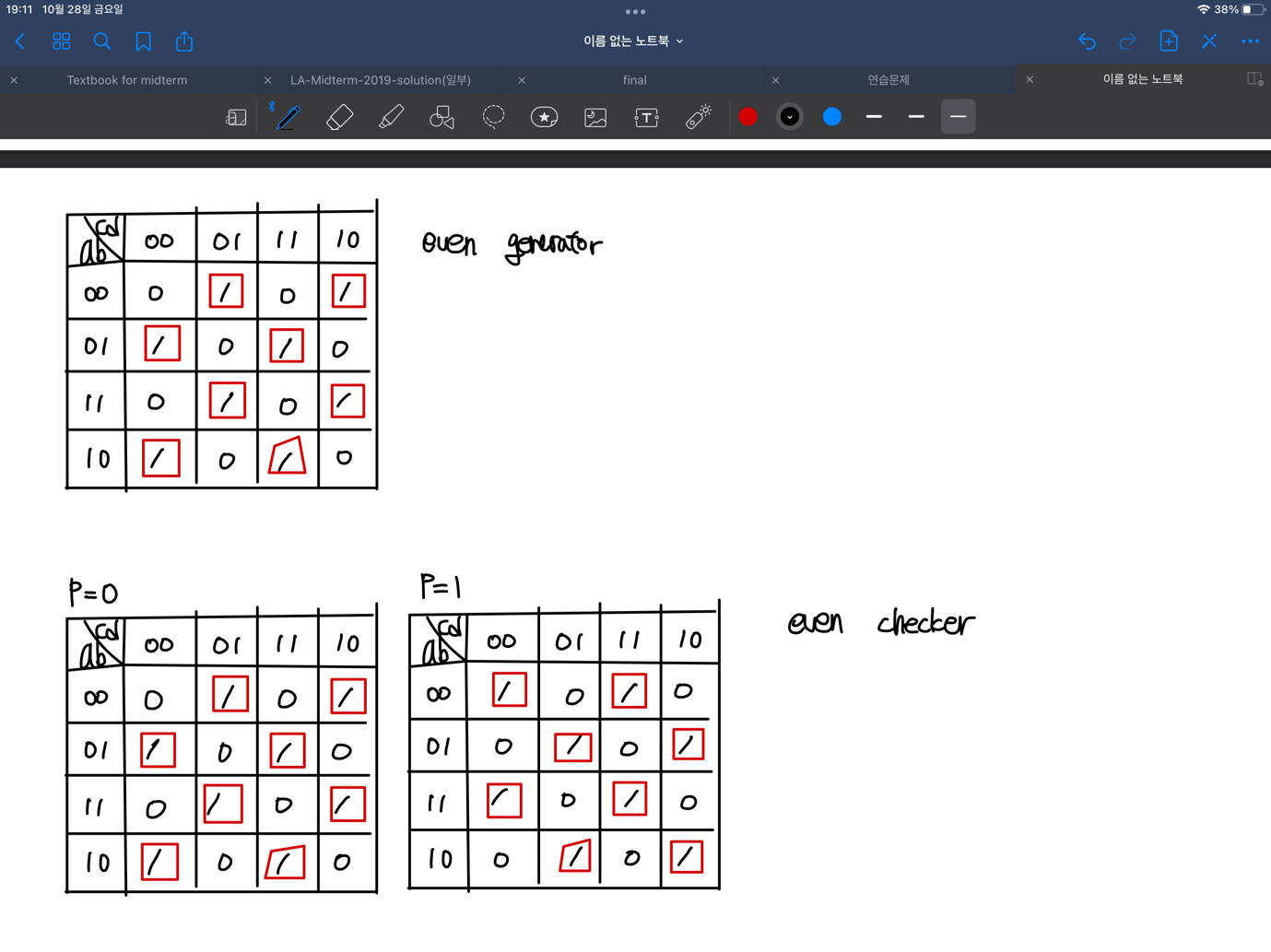
소스 코드를 통해 시뮬레이션을 구현하였다. 입력 변수의 개수가 홀수 일 때, parity bit를 나타내는 ee변수가 1이 됨을 알 수 있다.

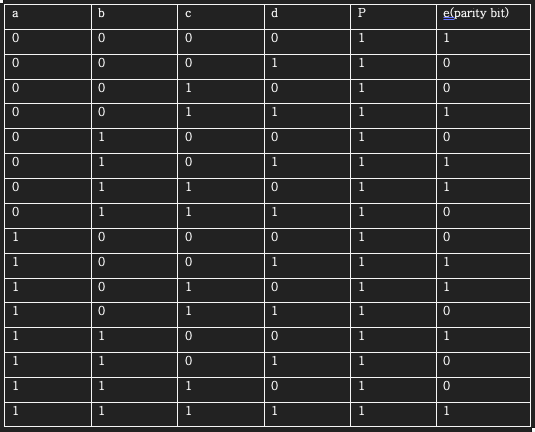


Even parity bit generator에 대한 schematic이다. 3개의 xor연산을 활용하였다.

다음은 even parity bit checker에 대한 내용이다. 5개의 입력(4개의 데이터 비트와 1개의 parity 비트)이 있고, 1개의 출력이 있다. 출력은 오류가 없을 때(데이터 비트 중 1의 개수가 홀수일때 parity bit가 1이거나 짝수일때 parity bit가 0인 경우) 0을, 오류가 있을 때(데이터 비트 중 1의 개수가 홀수일때 parity bit가 0이거나 짝수일때 parity bit가 1인 경우) 1을 출력한다. 다음은 입력과 출력에 대한 진리표와 카르노맵이다. Parity bit로 입력되는 변수 p에 따라 진리표와 카르노맵을 나누어 만들었다.







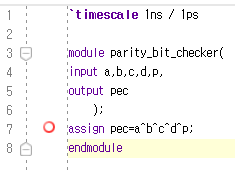
테이블이(가) 표시된 사진

자동 생성된 설명

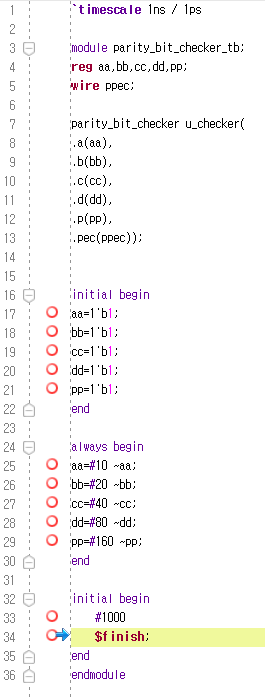
위의 식을 통해 pec=p’(a⊕b⊕c⊕d)+p(a⊕b⊕c⊕d)’

=a⊕b⊕c⊕d⊕p임을 알 수 있다.

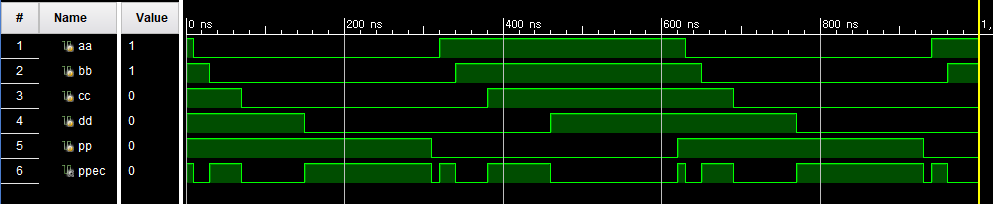
위 식을 통해, even parity bit check를 베릴로그로 구현하였다.



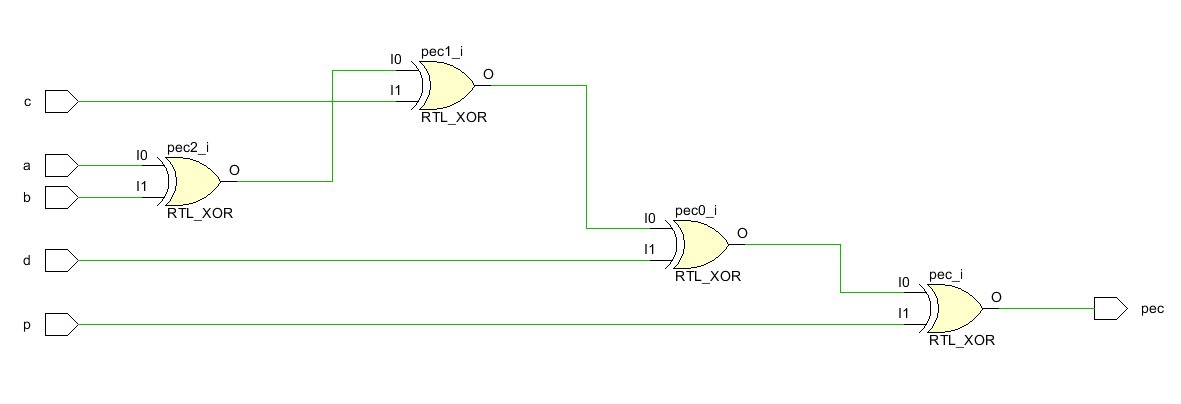
디자인 소스 코드이다. 입력 변수 a,b,c,d,p와 출력 변수 pec를 선언하였다. pec는 입력변수들에 대한 parity bit 오류여부를 나타내며, 입력변수들의 xor 연산을 할당한다.



시뮬레이션 소스 코드이다.



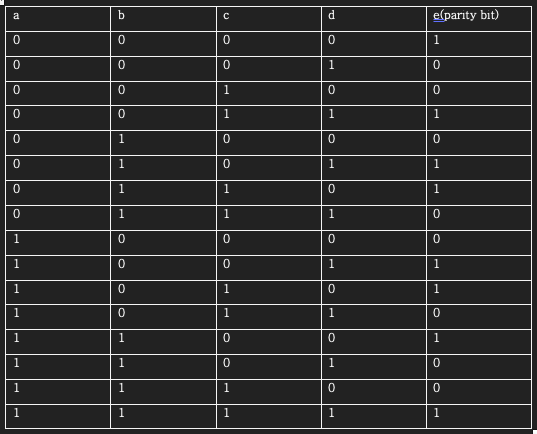
소스 코드를 통해 시뮬레이션을 구현하였다. 4개의 입력 변수의 개수가 홀수일 때 parity bit를 나타내는 pp의 값이 1이거나 짝수일 때 p의 값에 0일 때에는 ppec값이 0을, 4개의 입력 변수의 개수가 홀수일 때 parity bit를 나타내는 pp의 값이 0이거나 짝수일 때 p의 값에 1일 때에는 ppec값이 1을 나타낸다.



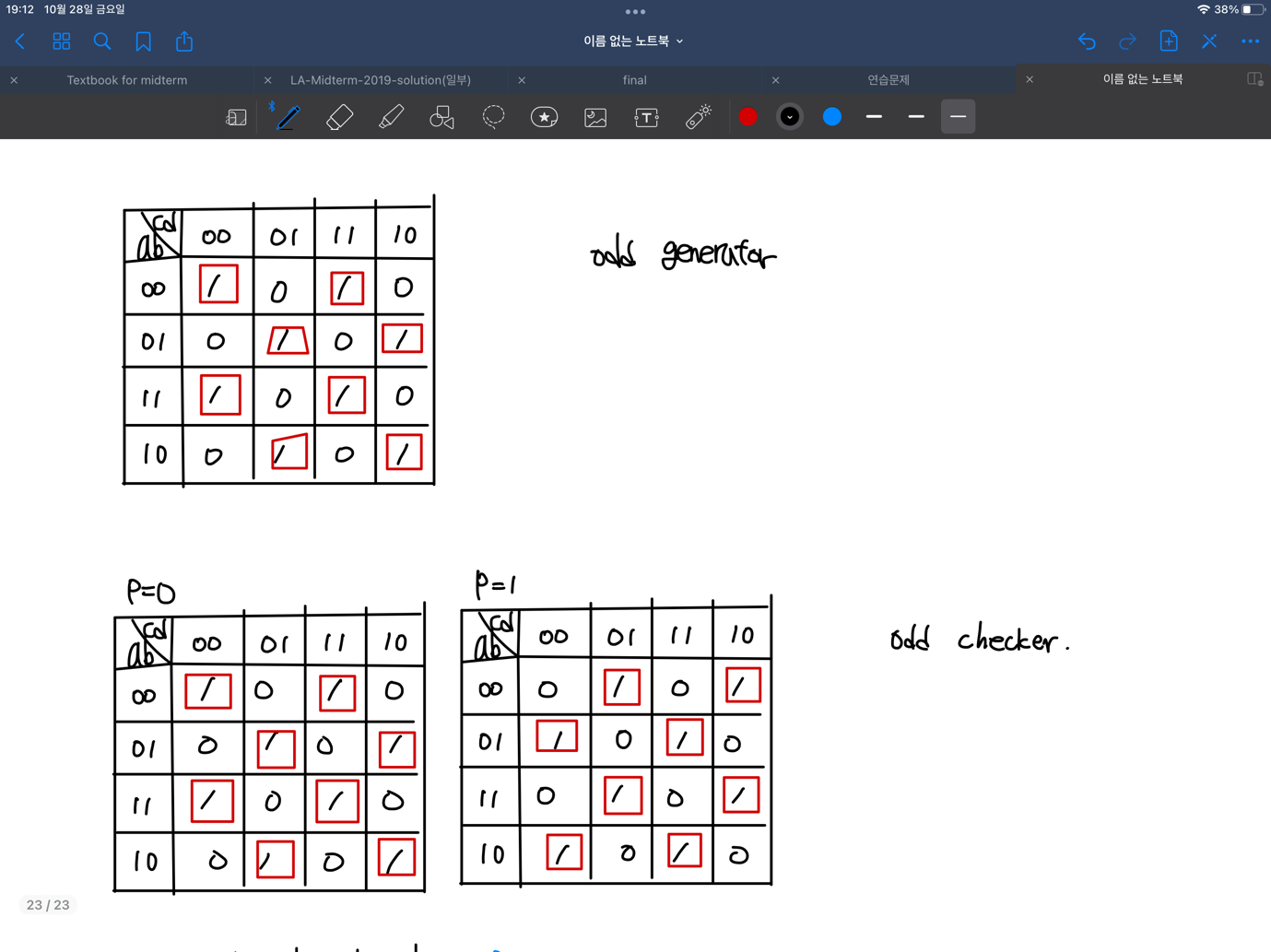
Even parity bit checker에 대한 schematic이다. 4개의 xor연산을 활용하였다.

2.

odd parity bit는 입력값 중 1의 개수가 홀수이면 parity bit를 0로, 짝수이면 parity bit를 1으로 설정한다. 다음은 4개의 입력과 이에 따른 parity bit를 진리표로 나타낸 표이다.



진리표를 토대로 카르노 맵을 구성하였다.



식을 정리하면 다음과 같다.

E=(a⊕b⊕c⊕d)’

위의 식을 토대로, 베릴로그를 이용하여 구현하였다.

텍스트이(가) 표시된 사진

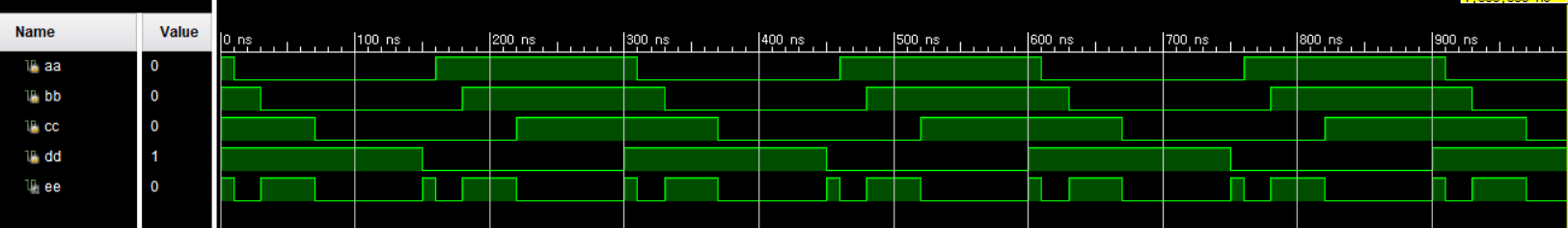
자동 생성된 설명

디자인 소스 코드이다. 입력 변수 a,b,c,d와 출력 변수 e를 선언하였다. E는 입력변수들에 대한 parity bit를 나타내며, 입력변수들의 xor 연산 후 not 연산을 하여 할당한다.

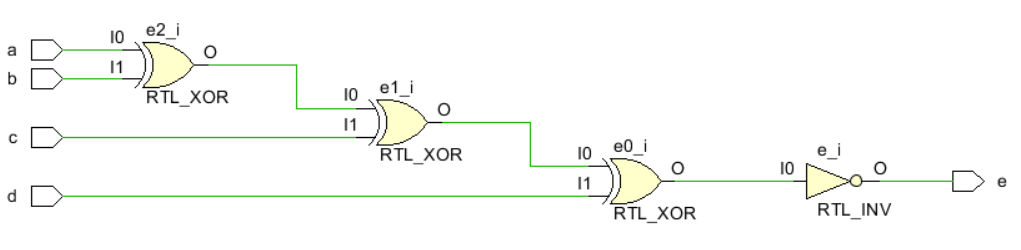
텍스트이(가) 표시된 사진

자동 생성된 설명

시뮬레이션 소스 코드이다.

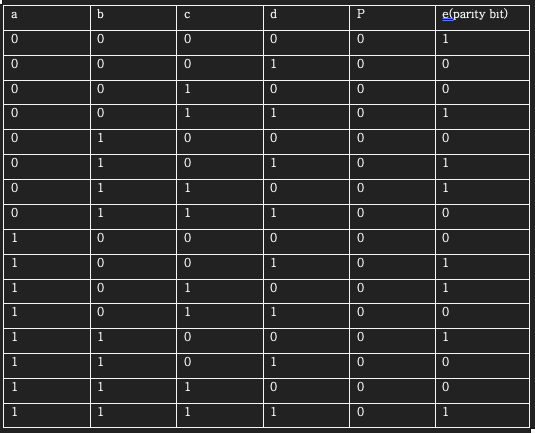


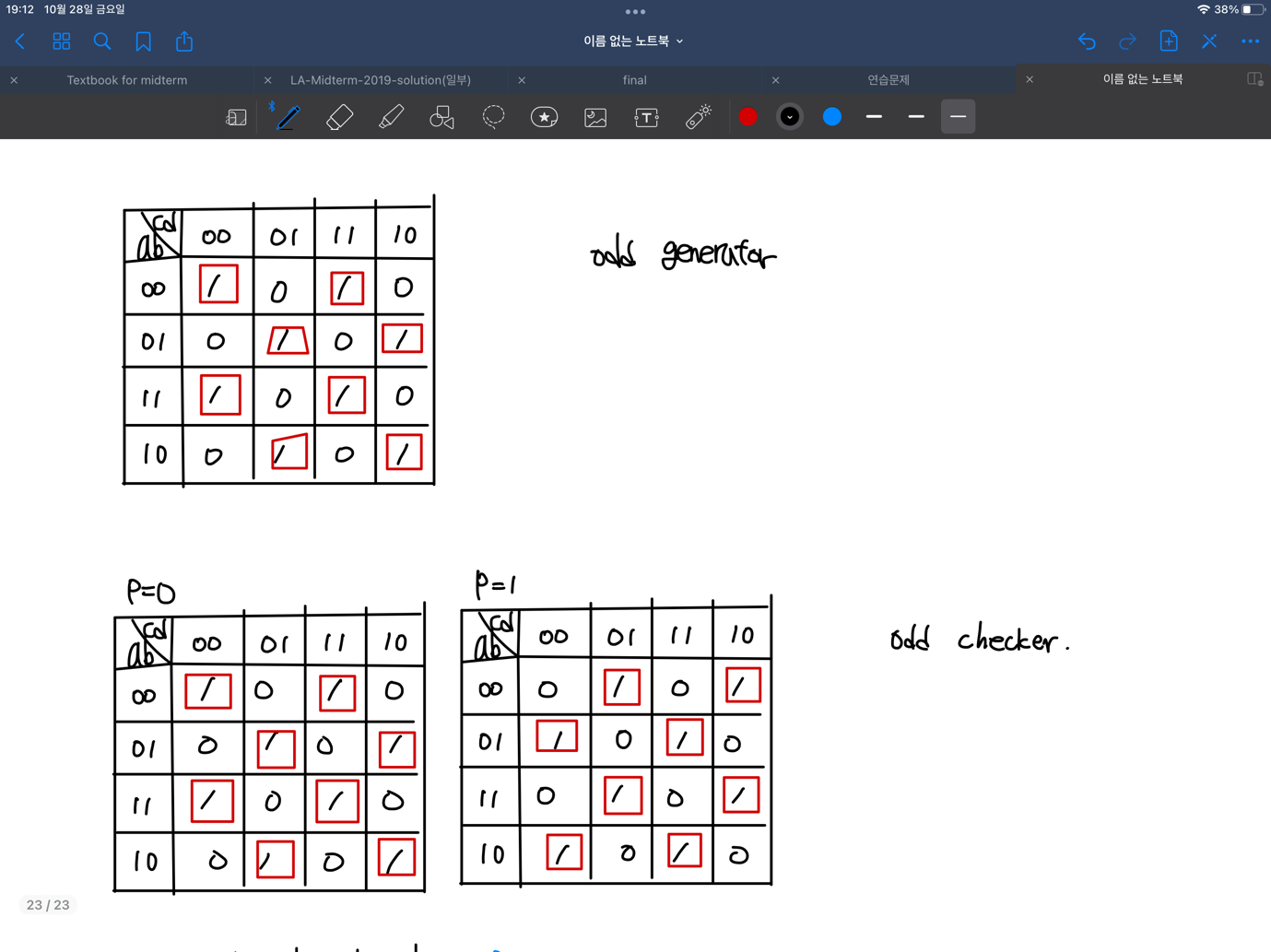
소스 코드를 통해 시뮬레이션을 구현하였다. 입력 변수의 개수가 짝수 일 때, parity bit를 나타내는 ee변수가 1이 됨을 알 수 있다.

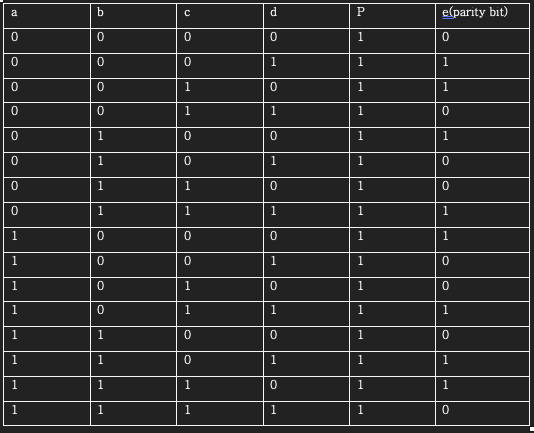


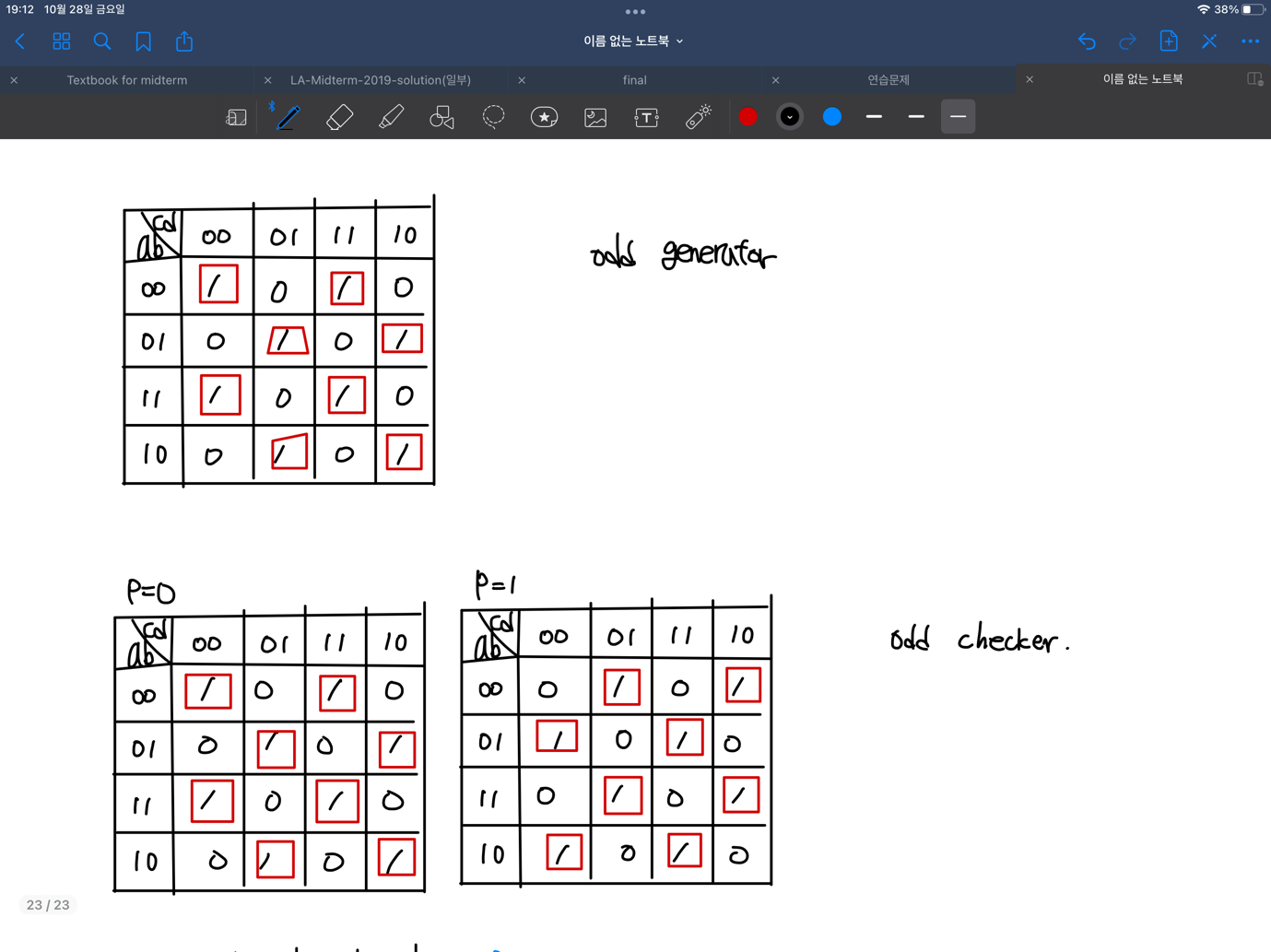
Even parity bit generator에 대한 schematic이다. 3개의 xor연산과 1개의 not연산을 활용하였다.

다음은 odd parity bit checker에 대한 내용이다. 5개의 입력(4개의 데이터 비트와 1개의 parity 비트)이 있고, 1개의 출력이 있다. 출력은 오류가 없을 때(데이터 비트 중 1의 개수가 짝수일때 parity bit가 1이거나 홀수일때 parity bit가 0인 경우) 0을, 오류가 있을 때(데이터 비트 중 1의 개수가 홀수일때 parity bit가 1이거나 짝수일때 parity bit가 0인 경우) 1을 출력한다. 다음은 입력과 출력에 대한 진리표와 카르노맵이다. Parity bit로 입력되는 변수 p에 따라 진리표와 카르노맵을 나누어 만들었다.









위의 식을 통해 pec=p’(a⊕b⊕c⊕d)’+p(a⊕b⊕c⊕d)

=(a⊕b⊕c⊕d⊕p)’임을 알 수 있다.

위 식을 통해, even parity bit check를 베릴로그로 구현하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

디자인 소스 코드이다. 입력 변수 a,b,c,d,p와 출력 변수 pec를 선언하였다. pec는 입력변수들에 대한 parity bit 오류여부를 나타내며, 입력변수들의 xor 연산과 not연산을 할당한다.

텍스트이(가) 표시된 사진

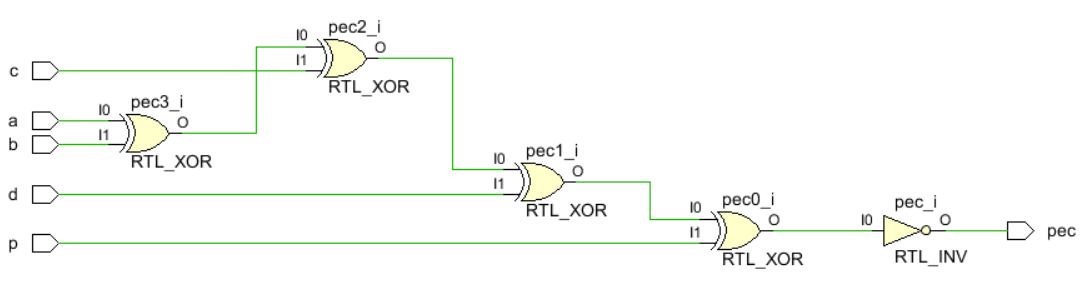
자동 생성된 설명

시뮬레이션 소스 코드이다.

녹색이(가) 표시된 사진

자동 생성된 설명

소스 코드를 통해 시뮬레이션을 구현하였다. 4개의 입력 변수의 개수가 짝수일 때 parity bit를 나타내는 pp의 값이 1이거나 홀수일 때 p의 값에 0일 때에는 ppec값이 0을, 4개의 입력 변수의 개수가 짝수일 때 parity bit를 나타내는 pp의 값이 0이거나 홀수일 때 p의 값에 1일 때에는 ppec값이 1을 나타낸다.



odd parity bit checker에 대한 schematic이다. 4개의 xor연산과 1개의 not연산을 활용하였다.

3. 2bit binary comparator simulation 결과 및 과정 (진리표ㅡ, 카르노맵, 스캐매틱)

2비트의 입력 2개에 대한 비교하고자 한다. 입력 a는 (a1)(a2)(2)로, 입력 b는 (b1)(b2)(2)로 표현한다. 만약 A>B인 경우에는 f1이 1로, A=B인 경우에는 f2가 1로, A<B인 경우에는 f3이 1이 된다. 다음 표는 이에 대한 진리표이다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A1 | A2 | B1 | B2 | F1 | F2 | F3 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

진리표를 토대로 f1,f2,f3에 대한 카르노 맵을 구성하였고, 카르노맵을 토대로 논리식을 간소화하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

F1에 대한 카르노 맵이고 이를 간소화하면 다음과 같다.

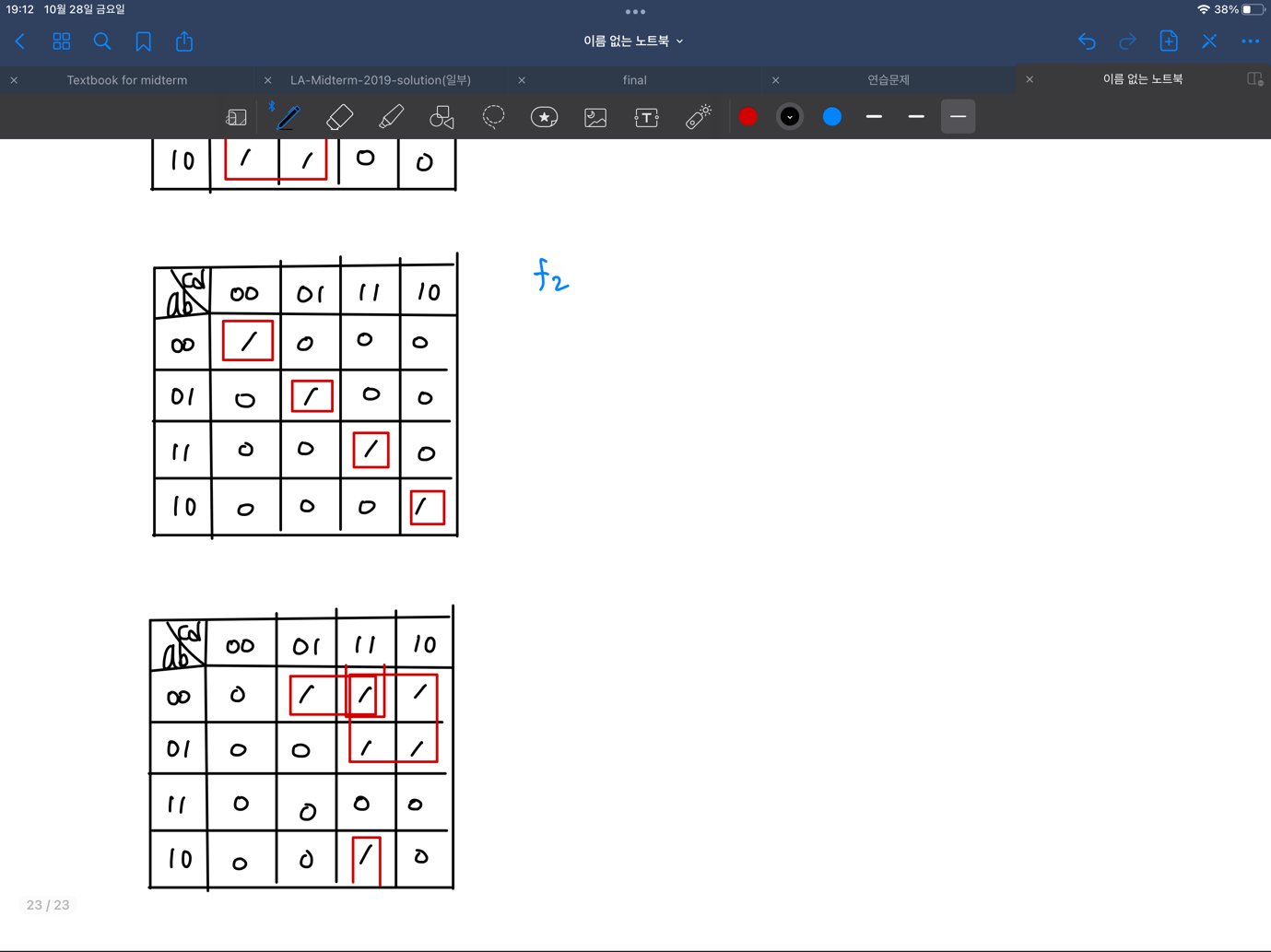
f1=(a1&(~b1))|(a2&(~b1)&(~b2))|(a1&a2&(~b2))

텍스트이(가) 표시된 사진

자동 생성된 설명

F2에 대한 카르노 맵이고 이를 간소화하면 다음과 같다.

f2=~(a1^b1)&~(a2^b2)



F3에 대한 카르노 맵이고 이를 간소화하면 다음과 같다.

f3=((~a1)&b1)|((~a1)&(~a2)&b2)|((~a2)&b1&b2)

다음과 같이 논리식을 간소화하였으며, 이를 베릴로그를 통해 구현했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Design source code를 구현했다. 입력 값으로, a1,a2,b1,b2를 출력값으로 f1,f2,f3을 선언하였으며, 카르노맵을 통해 간소화한 식을 각각 F1,f2,f3에 할당했다.

테이블이(가) 표시된 사진

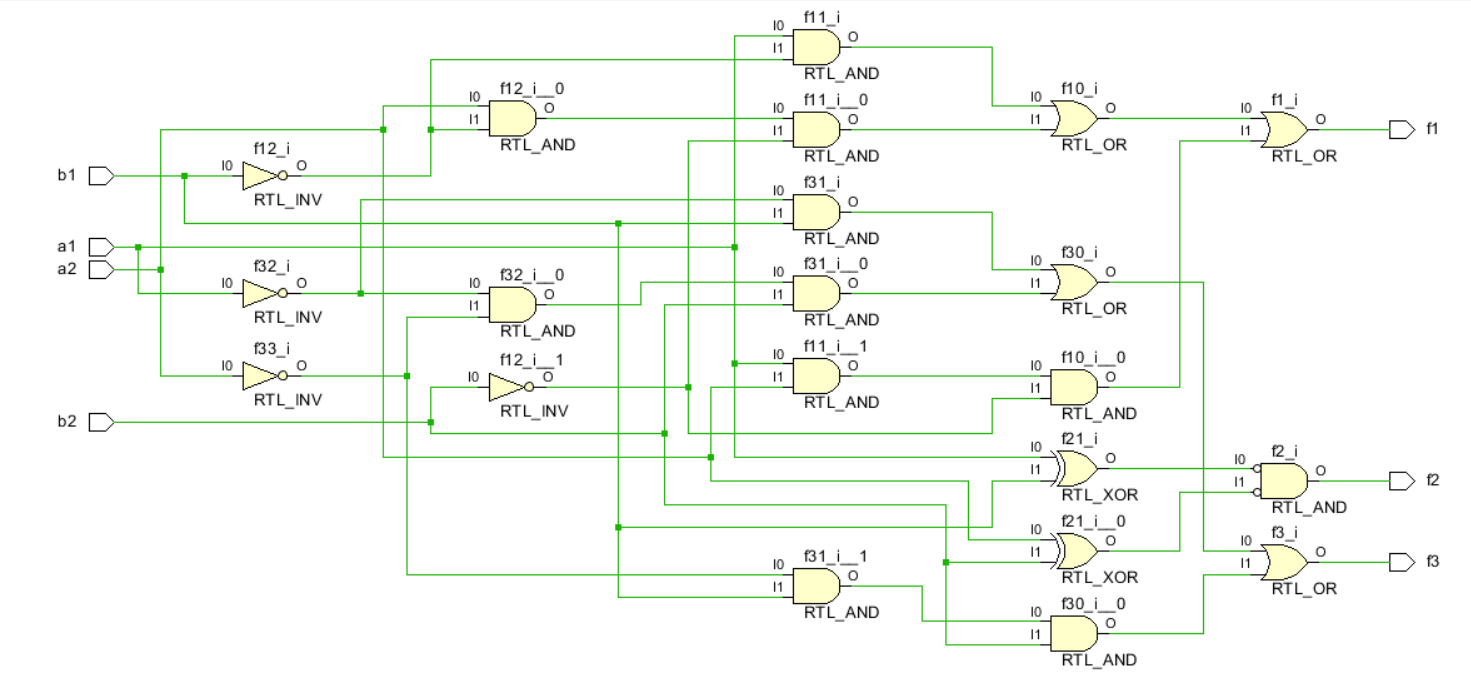
자동 생성된 설명

2bit binary comparator를 시뮬레이션하기 위해 시뮬레이션 코드로 구현하였다.

녹색, 흐린이(가) 표시된 사진

자동 생성된 설명

시뮬레이션 결과이며, 같은 비트 값이라면 f2이 1로, a1a2가 b1b2보다 크다면 f1이 1로, 작다면 f3가 1로 출력된다.



2 bit binary comparator에 대한 schematic이다. 간소화한 논리식에 대한 연산 결과가 diagram으로 표현되어 있다.

4.

Parity bit generator와 checker를 odd, even의 경우로 나누어 구현하였다. Odd인 경우, 입력 데이터에서 1의 개수가 홀수인 경우 parity bit를 1로 생성하고, 그렇지 않은 경우 0을 generator로 생성한다. Checker의 경우 입력 데이터에 따른 parity Bit가 정상인 경우, 즉 입력 데이터의 1의 개수가 홀수인데, parity bit가 1이거나 입력데이터의 개수가 짝수인데, parity bit가 0일 경우 오류가 없음을 의미하는 0을 출력하고, 나머지 경우 오류가 발생한 경우인 1을 출력한다. Even인 경우, 입력 데이터에서 1의 개수가 짝수인 경우 parity bit를 1로 생성하고, 그렇지 않은 경우 0을 generator로 생성한다. Checker의 경우 입력 데이터에 따른 parity Bit가 정상인 경우, 즉 입력 데이터의 1의 개수가 짝수인데, parity bit가 1이거나 입력데이터의 개수가 홀수인데, parity bit가 0일 경우 오류가 없음을 의미하는 0을 출력하고, 나머지 경우 오류가 발생한 경우인 1을 출력한다. 2bit binary comparator는 입력으로 입력된 두개의 입력 데이터를 비교하여 대소관계를 확인한다. 이는 비트 간의 논리 연산을 통해 대소 관계를 결정하고, 이에 맞는 결과를 출력한다. 출력을 결정하는 과정은 진리표를 통해 구현하고, 카르노맵을 통해 간소화한다.

5.

Xor는 변수 중 1의 개수가 홀수일때 1을, 짝수일 때 0으로 결정된다. 즉, 2개의 입력이 들어오는 상황에서는 두 변수의 값이 다른 경우 1이 된다. Xor의 연산자는 ^이며, a^b=a’b+ab’와 같이 표현할 수 있다. 이를 통해 다음과 같은 xor의 성질을 적용할 수 있다.

X^0=X

X^1=X’

X^X=0

X^X’=1

X^Y=Y^X

X^Y’=X’^Y=(X^Y)’

(X^Y)^Z=X^(Y^Z)=X^Y^Z

((X^Y)’^Z)’=(X^(Y^Z)’)’=X^Y^Z

와 같이 다른 연산자에 적용되는 associative, distributive, identity, complement 등의 laws가 적용된다.