7주차 예비보고서

전공: 기계공학과 학년: 3학년 학번: 20191820 이름: 김형준

**1.**

Parity Bit는 전송된 비트에서 오류가 생긴 비트가 있는지를 검사하기 위해 데이터의

맨 끝에 추가된 1비트를 의미한다. Parity Bit는 크게 두 가지 종류로 구분할 수 있는데,

전송되는 데이터의 비트들 중 1의 개수가 홀수가 되도록 맞추는 비트를 Odd Parity 라고

하고, 짝수가 되도록 맞추는 비트를 Even Parity라고 한다.

Odd Parity의 경우 데이터 비트에 포함된 1의 개수가 홀수면 Parity bit를 0으로 설정하

고, 1의 개수가 짝수면 Parity bit를 1로 설정해서 전체 1의 개수를 홀수로 유지한다.

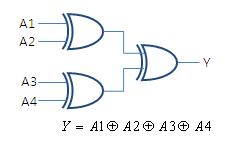
마찬가지로, Even Parity의 경우 데이터 비트에 포함된 1의 개수가 홀수면 Parity bit를 1

로 설정하고, 1의 개수가 짝수면 Parity bit를 0으로 설정해서 전체 1의 개수를 짝수로 유

지한다.

Parity Bit 생성기는 위와 같이 n개의 이진 비트를 입력받아, 홀수 또는 짝수 Parity Bit를 출력하는 회로이다. Even parity bit 생성기의 경우, 입력 비트들 중 1의 개수가 홀수일 때는 1을, 짝수일 때는 0을 출력한다. 이는 n-input XOR과 동일하므로

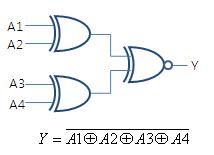
로 나타낼 수 있다. 따라서 even parity bit 생성기는 n-1개의 XOR 게이트를 사용하여 구현할 수 있다.



[4bit even parity bit 생성기]

또한, Odd parity bit 생성기의 경우 입력 비트들 중 1의 개수가 홀수일 때는 0을, 짝수일 때는 1을 출력한다. 이는 n-input XNOR과 동일하므로

로 나타낼 수 있다. 따라서 odd parity bit 생성기는 n-2개의 XOR 게이트와 1개의 XNOR게이트를 사용하여 구현할 수 있다.



[4bit odd parity bit 생성기]

**2.**

parity bit 검사기는 parity bit 생성기를 통해 만들어진 데이터(Parity Bit가 추가된)를 수신했을 때, 데이터에 오류가 없으면 0을 출력하고, 이상이 있으면 1을 출력하는 회로이다.

even parity bit 검사기의 경우, 입력 비트들(parity bit를 포함) 중 1의 개수가 짝수일 때 0을, 1의 개수가 홀수일 때 1을 출력한다. 이는 n-input XOR에 해당하므로 로 나타낼 수 있다.

또한, odd parity bit 검사기의 경우, 입력 비트들(parity bit를 포함) 중 1의 개수가 홀수일 때 0을, 1의 개수가 짝수일 때 1을 출력한다. 이는 n-input XNOR에 해당하므로 이는

로 나타낼 수 있다.

parity bit 검사기는 짝수개의 비트의 값이 바뀌는 경우 에러가 발생했다는 것을 알 수 없고, 홀수개의 비트의 값이 바뀌는 경우라도 오류가 발생했다는 사실만 알 수 있고 어디서 오류가 난 지에 대해서는 알 수 없다는 단점이 있다.

**3.**

해밍 코드(Hamming code)는 단일오류정정 (single-error-correction) 코드이다. 정보 비트에 검사 비트를 추가하여 전송 또는 저장과정에서 최대 1비트만 변경된 경우에는 원래의 값으로 복원할 수 있다. 검사 비트의 수가 n(이면 만큼의 정보 비트를 검사할 수 있다.

가장 단순한 구성방법은 1로부터 시작해서 비트들의 번호를 매기고, 번째의 비트를 검사비트로 간주하는 것이다.

예를들어, 4비트 해밍코드의 경우, 2의 제곱수인 1, 2, 4번째 비트는 검사 비트(even parity bit)가 되고, 나머지 비트들은 정보 비트가 된다. 비트들을 순서대로 MSB부터 부터 로 표기하면, 은 , 는 , 은 에 대해 even parity 검사를 수행하므로 이를 수식으로 나타내면, 각각 , , 이 된다.

이렇게 정보 비트에 검사 비트를 추가한 송신된 7비트의 데이터를 수신 받으면, 검사비트를 포함하여 even parity bit 검사를 수행한다.

, ,

각각의 검사로 위와 같은 연산을 수행해 결과값 , , 를 얻는데, 오류가 없으면 모두 0이 된다. 만약 한 개의 오류가 존재한다면 를 사용해 어디서 오류가 났는지를 계산할 수 있다. 이런 방법을 통해 수신자가 별도의 추가 정보를 수신하지 않고도 최대 한 개의 오류까지는 원래의 값으로 복원 할 수 있다.

그러나, 두 개 이상의 오류의 경우에는 위의 오류 발생 비트의 위치를 계산하는 공식이 잘못된 결과를 내기 때문에 어디서 오류가 났는지 정확하게 알 수 없다는 단점이 있다.

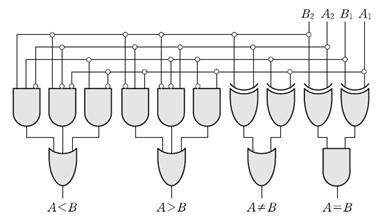
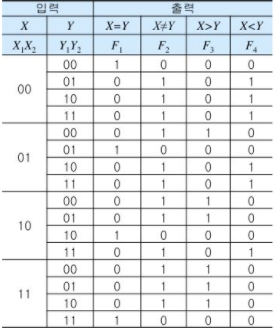
**4.**

N bit 비교기는 두 개의 N bit 이진수의 대소관계를 구하기 위한 논리 회로로써, 상위 자릿수부터 하위 자리수의 순서로 비교를 수행한다.

1비트 비교기와 마찬가지로, N비트 비교기도 두 이진수에 대해 를 수행한다. 비교의 결과가 참이면 1을, 거짓이면 0을 반환한다.

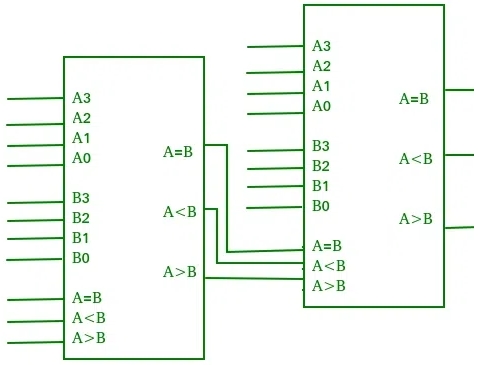
1비트 비교기를 확장하여 2비트의 비교기를 구현하면, 두 이진수의 각 자리수의 비트 (A1,A2,B1,B2)를 입력으로 받고, 각각의 출력을 구하는 4-input, 4-output 회로가 된다.

2비트 비교기의 진리표와 회로도는 다음과 같다.

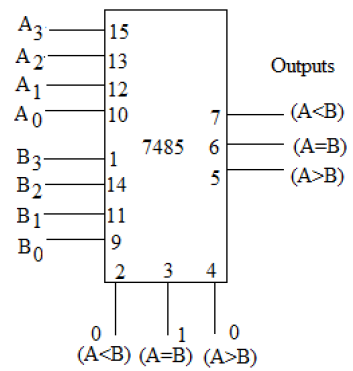


이와 같은 방식으로 n비트의 비교기를 만들 수 있으나, n의 값이 점점 커져서 비교할 비트의 개수가 많아지면 입력값의 개수도 많아져 출력값의 논리 함수가 점점 복잡해지기 때문에, N bit () 비교기의 경우 4비트 비교기 다수를 엮는 방식으로 설계한다.

이런 방식의 비교기를 Cascading Comparator라 하며, 아래의 회로도는 8bit 비교기를 4bit 비교기 두 개를 엮어서 구현한 회로이다.



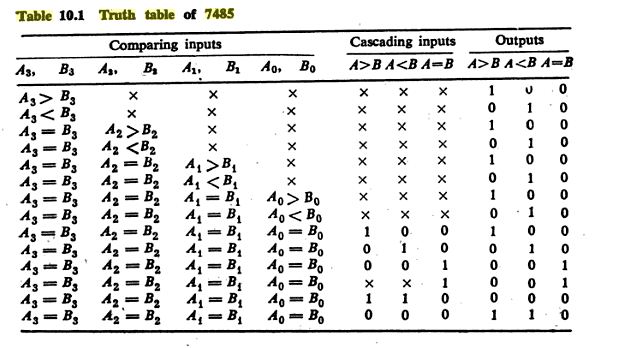
**5.**

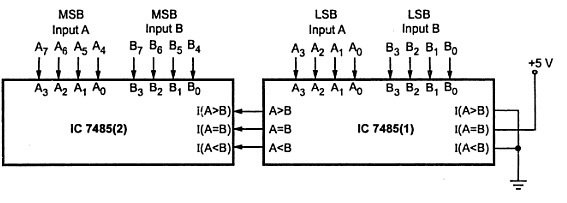


IC 7485 비교기는 4비트의 두 이진수의 대소관계를 구하는 비교기로, 이진수 A, B의 각 자리수의 비트를 입력 A3, A2, A1, A0과 B3, B2, B1, B0으로 받고, 을 각각 출력한다.

또한, IC 7485 비교기는 이전 단계의 출력값을 입력으로 각각 받는데, 이 입력(cascading input)을 통해 IC 7485 비교기를 여러 개 연결하여 4n bit 비교기를 만들 수 있다.

아래는 IC 7485의 진리표와 IC 7485 비교기 2개를 엮어서 8bit 비교기를 구현한 회로도이다.

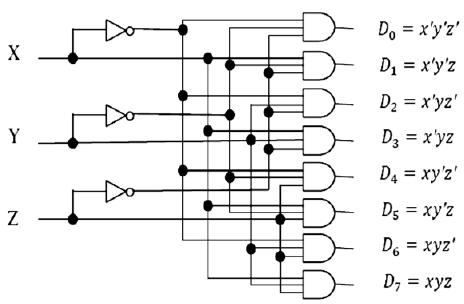
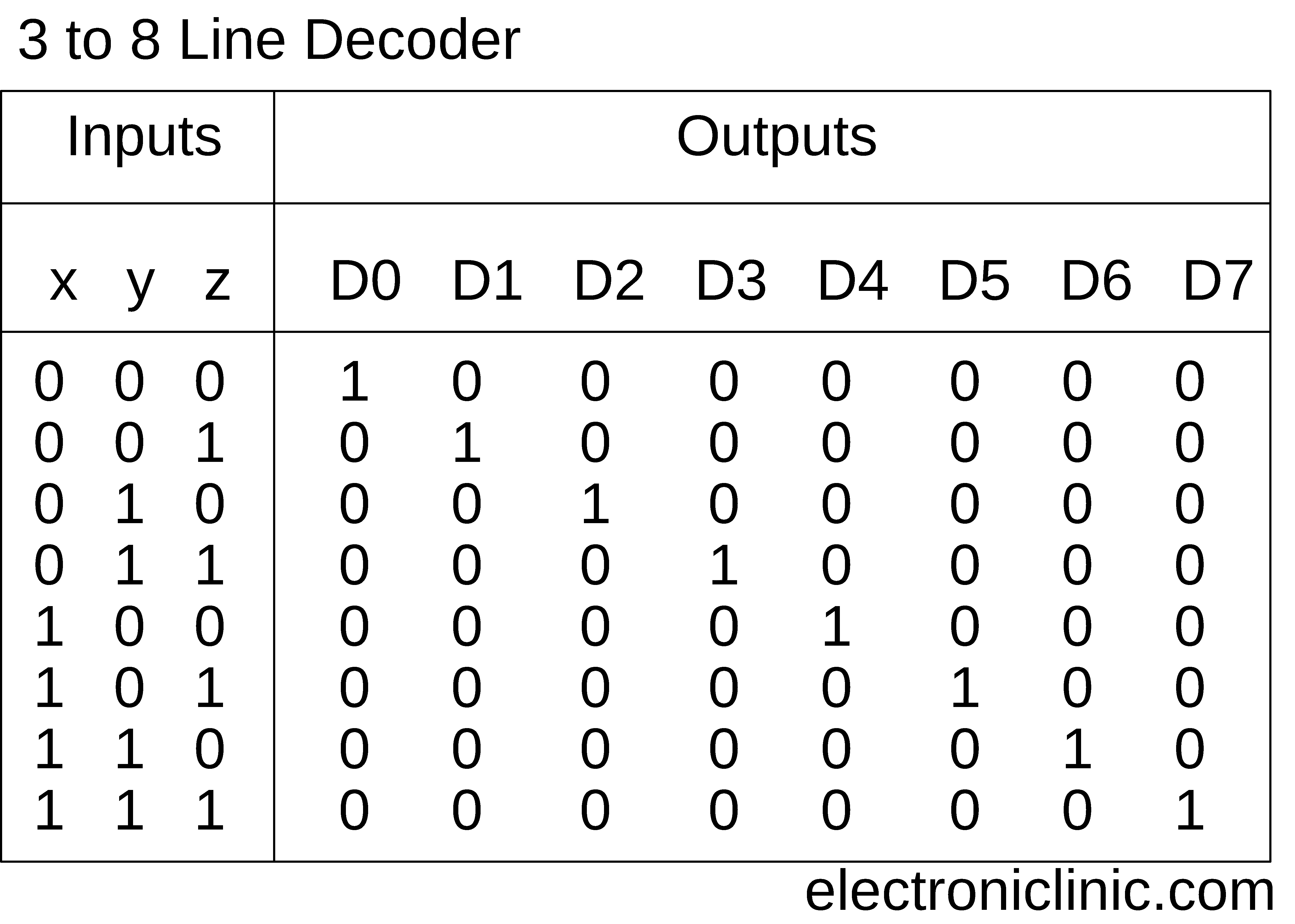




**6.**

이진 디코더(Decoder)는 활성화 되었을 때, 입력신호에 기초하여 여러 개의 출력 중 하나를 선택하는 논리 회로이다. 디코더는 여러 장치 중에서 하나를 선택하는 용도로 많이 사용되며, 인코더와 정 반대의 기능을 수행한다.

n개의 비트를 입력으로 받으면 개의 출력이 존재하는데, 하나의 출력값이 1이면 나머지 출력값 들은 모두 0이라는 특징이 있다. 이러한 디코더의 가장 대표적인 예시는 3x8 디코더가 있는데, 진리표와 회로도는 다음과 같다.



대부분의 디코더는 하나 이상의 enable 입력을 가지고 있는데, 이 입력이 활성화 되는 경우 디코더는 그대로 동작하게 되고, 이 입력이 비활성화 되는 경우 디코더의 모든 출력이 0이 된다.