

7주차 예비 보고서

20191621 이민영

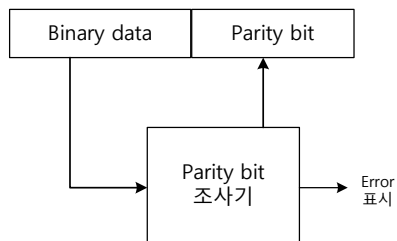
1. Parity Bit 생성기에 대해 조사하시오.

Parity Bit이란, 정보가 전달될 때 오류가 생겼는지를 판단하기 위해서 만들어진 bit이다. Parity Bit은 오류 검출 부호에서 가장 간단한 형태이다.

이진으로 된 데이터의 1의 개수가 홀수 개 일 때 Parity bit는 1로 설정되고, 1의 개수가 짝수 개이면, Parity bit는 0으로 설정된다. 결과적으로 1의 개수가 홀수 개 일 때 Parity bit가 1로 설정되기 때문에 데이터는 항상 짝수개의 1을 가지게 된다.

이와 같은 방식으로, 전송되는 데이터가 항상 홀수개의 1을 가지도록 Parity Bit을 이용할 수 있다.

7비트 데이터의 경우 0000000, 1101000, 1111001 일 때, 위의 짝수 Parity Bit 생성기에 의하면, 전송되는 데이터는 00000000, 01101000, 11111001 이 된다.



홀수 Parity Bit 생성기를 이용할 경우, 아래와 같이 Parity Bit이 생성된다.

Three-Bit Message			Parity Bit
X	Y	Z	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

데이터를 받았을 때, 받은 데이터의 비트를 계산해서 Parity Bit를 이용해서 데이터 오류 발생 여부를 판단할 수 있다. 그러나, 오류를 수정할 수는 없다.

2. Parity Bit 검사기에 대해 조사하시오. [검사 부호 종류 포함]

Parity Bit 검사기는 Parity Bit를 이용해서 전달된 정보가 정확한 정보인지 확인한다. 짝수 Parity 부호와 홀수 Parity 부호가 존재한다. Parity Bit 검사기는 1의 개수가 홀수개인지 짝수개인지에 따라서 값을 판단하기 때문에 XOR를 이용한다.

짝수 Parity 부호의 경우, Parity Bit를 포함한 모든 데이터를 XOR 연산을 했을 때, 결과값이 0일 경우 오류가 없는 데이터가 되며, 홀수 Parity 부호의 경우, Parity Bit를 포함한 모든 데이터를 XOR 연산을 했을 때, 결과값이 1인 경우 오류가 없는 데이터가 된다.

3. Parity Bit 검사기 외의 다른 오류 검출기 및 오류 정정기를 조사하시오.

순환 중복 검사는 네트워크 등을 통해서 데이터를 전송할 때 데이터에 오류가 있는지 없는지 확인해 줄 수 있는 체크 값을 결정하는 검사이다. 데이터를 전송하기 전에 데이터에 맞는 CRC 값을 계산해서 이 값을 데이터와 함께 전송한다. 데이터를 받은 이후에는 데이터를 이용해서 다시 CRC 값을 계산하고, 이를 전송된 CRC 값과 비교해서 데이터의 정확도 여부를 판단할 수 있다. 다항식 코드는 CRC를 위해서 미리 정한 다항식을 이용해서 얻은 코드이다.

해밍코드 방식은 받는 곳에서 오류가 발생한 비트를 검출하고 수정하는 방식의 오류 정정기이다. Parity Bit 검사기는 오류 검출만 수행하고 스스로 수정하지는 못하는 반면, 해밍코드 방식은 스스로 수정까지 할 수 있기 때문에 자기 정정 부호라고 부르기도 한다. 그러나, 1비트의 오류만 수정이 가능하고, 기본 정보 데이터 이외에 많은 비트가 추가적으로 필요하다는 단점이 있다.

블록 합 검사는 문자를 블록으로 전송할 경우에 오류를 검사하는 방법이다. 문자 블록에 대해서 수평, 수직인 2차원 적으로 패리티를 검사하는 방법이다. 블록 합 검사 또한 Parity 검사와 마찬가지로 오류를 수정하지는 못하고 검출만을 시행한다.

4. N bit 비교기에 대해서 조사하시오.

N bit 비교기는 N개의 비트에 대해서 어떤 비트가 더 큰지 값을 비교하는 회로이다. N비트 수 2개가 입력되었을 때 회로의 입력 수는 적어도 2N개가 된다. 따라서 회로를 설계하기 위해서는 2^{2N} 가지의 조합이 이루어지게 된다.

가장 단순한 1bit 비교기를 살펴보면, 진리표를 이용해서 출력 값을 판단할 수 있다.

Input		Output			
X	Y	$X=Y$	$X \neq Y$	$X > Y$	$X < Y$
0	0	1	0	0	0

0	1	0	1	0	1
1	0	0	1	1	0
1	1	1	0	0	0

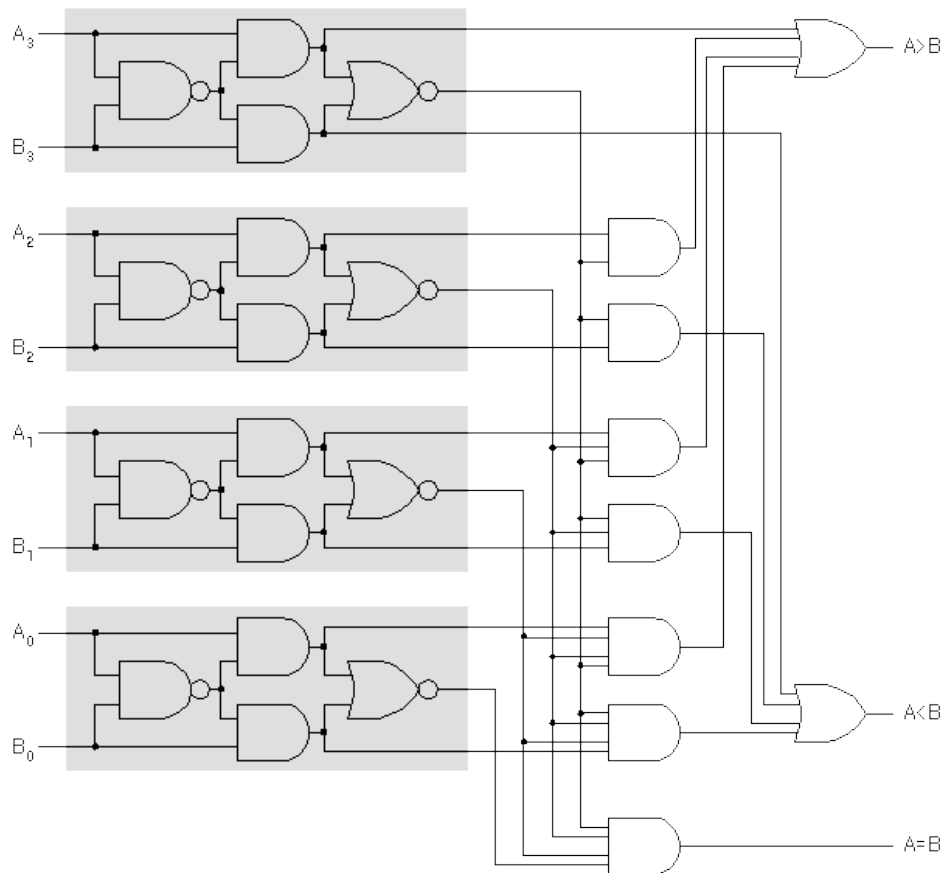
두 수를 비교하기 위해서는 각 자리의 숫자를 비교해야 한다. 4비트를 예시로 판단해보면, $A = A_3A_2A_1A_0$, $B = B_3B_2B_1B_0$ 로 표현할 수 있다. 이때 각 자리 숫자는 이진 숫자인 0 또는 1이다. 이 때 A와 B가 같기 위해서는, $A_3=B_3$, $A_2=B_2$, $A_1=B_1$, $A_0=B_0$ 이다.

크기 비교를 위해서 $X_i = A_iB_i + A_i'B_i'$ 인 논리식을 생각해볼 수 있다. 이 식에서 A_iB_i 는 두 숫자가 모두 1일 때이며, $A_i'B_i'$ 는 두 숫자가 모두 0인 경우를 나타낸다.

따라서,

$(A=B) = X_3X_2X_1X_0$, $(A>B) = A_3B_3' + X_3A_2B_2' + X_3X_2X_1A_0B_0'$, $(A<B) = A_3'B_3 + X_3A_2'B_2 + X_3X_2A_1'B_1 + X_3X_2X_1A_0'B_0$ 이다.

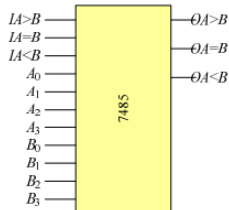
이를 N-bit 로 확장해서 생각해 보면, $(A=B) = X_nX_{n-1}...X_1$, $(A>B) = A_nB_n' + X_nA_{n-1}B_{n-1}' + ... + X_nX_{n-1}...X_1A_0B_0'$, $(A<B) = A_n'B_n + X_nA_{n-1}'B_{n-1} + ... + X_nX_{n-1}...X_1A_0'B_0$ 이다.



<4비트 비교기 회로>

5. IC 7485 비교기에 대해서 조사하시오.

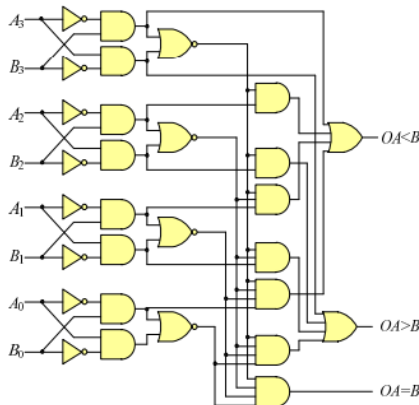
IC 7485 비교기는 4비트 비교기이다. 입력 값으로는 $A_3, A_2, A_1, A_0, B_3, B_2, B_1, B_0, I_{A>B}, I_{A<B}, I_{A=B}$ 가 들어오며, 출력 값은 $O_{A>B}, O_{A<B}, O_{A=B}$ 이다.



진리표를 아래와 같다.

입력							출력		
A3,B3	A2,B2	A1,B1	A0,B0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$O_{A>B}$	$O_{A<B}$	$O_{A=B}$
A3>B3	X	X	X	X	X	X	1	0	0
A3<B3	X	X	X	X	X	X	0	1	0
A3=B3	A2>B2	X	X	X	X	X	1	0	0
A3=B3	A2<B2	X	X	X	X	X	0	1	0
A3=B3	A2=B2	A1>B1	X	X	X	X	1	0	0
A3=B3	A2=B2	A1<B1	X	X	X	X	0	1	0
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	1	0	0
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	1	0	0	1	0	0
A3=B3	A2=B2	A1=B1	A0=B0	0	1	0	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	1	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	0	1	1	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	1	0	1	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	1	1	1	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	1	1	0	0	0	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	0	1	1	0

IC 7485 비교기의 회로도는 아래와 같다.



IC 7485 비교기 회로에서 ,

$$O_{A=B} = X3X2X1X0,$$

$$O_{A>B} = A3B3' + X3A2B2' + X3X2A1B1' + X3X2X1A0B0'$$

$$O_{A<B} = A3'B3 + X3A2'B2 + X3X2A1'B1 + X3X2X1A0'B0$$

이다.

6. 기타이론

데이터를 전송할 때, 수신기에서 오류가 발견되었을 때 재전송을 요구하는 경우와 수신기에서 직접 오류 데이터를 정정하는 방법이 있다. 재전송을 요구하는 경우를 역방향 오류 정정이라고 하며, 수신기에서 직접 정정하는 경우를 순방향 오류 정정이라고 한다.

역방향 오류 정정(BEC)은 패리티 검사, CRC, 블록합 방식 등을 이용해서 오류를 검출하고, 오류 제거는 자동 반복 요청을 통해서 이루어진다. 이 때 오류 검출 방식은 오류를 직접 수정하지는 못하고 오류의 발생 여부만을 알아낸다.

순방향 오류 정정(FEC)은 해밍코드 상승 코드 방식 등을 이용해서 오류를 검출하고 수정한다. 이 때 스스로 오류를 수정까지 하기 위해서는 기본 데이터에 오류 검출과 수정을 위한 비트들이 추가로 많이 전송되어야 하기 때문에 전송 효율이 떨어진다는 단점이 있다.