7주차 예비보고서

전공 : 아트앤테크놀로지 학년 : 3학년 학번 : 20191098 이름 : 백승주

1

.

Parity Bit 생성기는 데이터를 전송할 때 오류를 검출하기 위해 사용하는 검사 부호 중 하나다. 이 생성기는 데이터를 전송할 때 parity 비트를 추가해서 전체 비트 수를 홀수 또는 짝수로 조절한다. 예를 들어, 전송할 데이터가 0110011이라면 이에 대한 parity 비트를 생성하고 싶다면 0을 붙여서 01100110를 전송하고 홀수 전송을 원한다면 마지막에 1을 붙여 01010111로 전송한다. 이 데이터를 받는 수신자는 전송받는 데이터를 전송받고 이 parity 비트를 확인해 데이터의 손상 여부를 확인할 수 있다.



Parity 비트 생성기는 xor 게이트를 이용한다. Xor 게이트는 1이 짝수 개면 0을 출려갛고 홀수면 1을 출력하기 때문에 짝수 parity 비트를 생성하고 싶을 때는 xor 게이트를 사용하고 홀수 parity 비트를 생성하고 싶을 때는 xor 게이트에 not을 붙여 사용한다.

2. Parity 검사기는 parity 비트 데이터를 전송받고 그 데이터에 오류가 발생했는지 검출하기 위해 사용하는 회로를 칭하는 말이다. 수신자가 사용하는 이 검사기는 전송된 데이터와 parity 비트를 비교해서 오류 여부를 확인한다.

검사 방법은 홀수 parity 검사와 짝수 parity 검사 두 가지가 존재하며 짝수 parity 검사기는 전체 비트에서 1의 개수가 홀수면 0, 1의 개수가 짝수면 1을 출력한다. 반대로 홀수 parity 검사기는 전체 비트에서 1의 개수가 홀수면 1을 출력하고 짝수면 0을 출력한다

또한, 검사 부호에는 패리티 비트 이외에도 CRC(Cyclic Redundancy Check)와 같은 다른 종류가 있습니다. CRC는 비트 시프트 레지스터(BSR)를 이용하여 다항식 계산을 통해 검사 부호를 생성하며, 전송된 데이터와 생성된 검사 부호를 비교하여 오류를 검출합니다.

3-1. 해밍 코드(Hamming Code)

해밍 코드 (Hamming Code)

Richard Hamming이 개발한 해밍 코드는 일반적으로 2^n-1 개의 비트로 이루어진 코드를 사용하고 데이터 전송 전에 특정한 방식으로 코드화해서 전송해, 수신자는 전송된 코드에 오류가 있는지 검사하고, 오류가 있으면 정정할 수 있는 코드다. 1 비트의 오류를 검출할 때 해밍 코드는 데이터가 전송되고 각 비트별로 그 비트를 포함한 세 비트의 합이 짝수인지 홀수인지 확인하며 오류를 검출한다.

3-2 . BCH 코드 (Bose-Chaudhuri-Hocquenghem Code)

BCH 코드는 해밍 코드보다 발전한 형태의 코드로 해밍 코드와 다르게 BCH 코드는 다중 오류를 비롯한 다양한 종류의 오류를 잡아내어 정정할 수 있다. BCH 코드 또한 해밍 코드와 마찬가지로 2^n-1 길이의 코드를 사용한다.

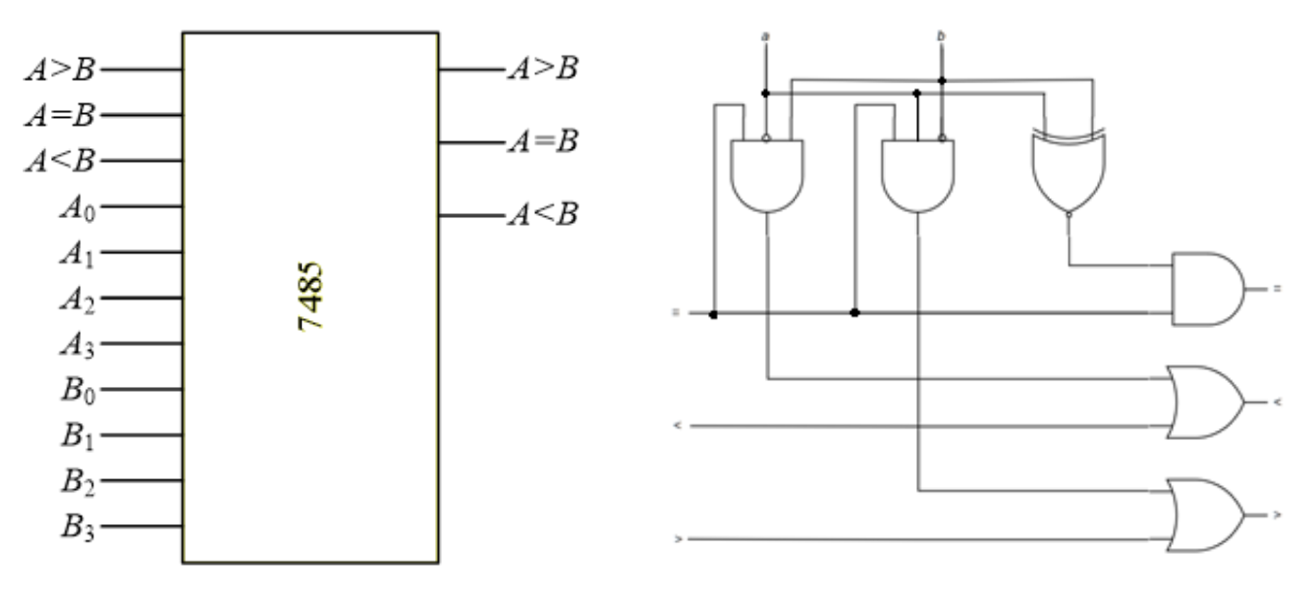
FEC (Forward Error Correction)

3-3. FEC는 전송 전에 오류 정정을 수행하여, 수신측에서 오류를 복구할 수 있도록 하는 기술입니다. FEC는 데이터 전송 시 오류가 발생할 가능성이 높은 무선 통신에서 많이 사용됩니다. FFC는 먼저 데이터에 맞게 파라미터를 조정하고 부호화해 오류를 검출하고 정정한다. 또한 데이터를 전송 후 수신한 데이터를 복호화해 오류를 발견하고 수정한다.

4. N bit 비교기는 두 개의 n 비트 input을 A와 B라는 두 개의 이진수를 입력받아서 , A=B, A!=B, A>B, A<B 연산의 결과값을 출력한는 회로다. n개의 비트 두 개를 입력 받을 때 Input의 개수는 2n개가 된다. n bit adder는의 종류로는 ripple carry adder, carry look-ahead adder, carry select adder 등이 있다. 먼저 Ripple carry adder는 두 개의 이진수를 각각의 비트별로 덧셈하여 carry를 구한다. 그리고 이 carry를 다음 비트에서 사용하는 방식으로 진행한다.

Carry look-ahead adder는 ripple carry adder의 성능을 향상시키기 위해 구현된 방식으로 각각의 비트별로 carry를 미리 계산하여 전파하기 대문에 연산 속도가 ripple carry adder 보다 빠르다. Carry select adder의 경우에는 여러 개의 ripple carry adder를 사용해서 그 때마다 제일효율적인 ripple carry adder를 선택하여 사용하는 방식이다.

5. IC 7485 비교기는 4 비트 크기의 이진수 두 개를 서로 비교해 한 결과를 출력하는 회로다 3개의 output 을 가지는 IC 7485 비교기의 output은 각각 A>B, A=B A< B를 의미한다. 각각의 OUTPUT들 그 output의 조건들을 A와 B가 만족할 때 출력 1을 가진다. 이 비교기는 이전 단계에서의 A>B, A=B, A<B도 입력값으로 받는데 현재 단계에서의 결과가 같았다면 하위 비트의 비교 결과에 따라 수를 비교한다.



6. Gray code : 데이터의 오류 판별에 사용하는 코드로 Grapy code가 존재한다. 이 그레이 코드는 4 비트의 코드이며 연속된 숫자들 사이에 단일 비트의 차이만 나도록 하는 방법으로 이진수에서 한 비트만 변경된다. 즉 4비트의 그레이 코드는 4비트 Gray code는 0, 1, 3, 2, 6, 7, 5, 4와 같은 순서를 가진다. 인접한 값들 사이에서는 항상 한 비트만의 차이를 가지기 때문에 신호가 변경될 때 발생할 수 있는 비트 오류를 방지할 수 있고 신호를 전송할 때 얻는 손실이나 에러를 최소화할 수 있다. 이 때문에 회로 설계에서 효과적일 수 있으나 표현 범위가 작다는 단점이 존재한다.