7주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

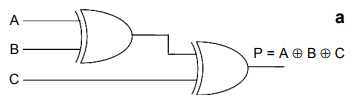
**1. Parity Bit 생성기**

1960년대 초반, R.Landauer’s 는 비트의 손실이 곧 정보의 손실임을 알게되었다. Parity Bit는 무선 통신 환경에서 여러 요인으로 인해 송수신하는 정보에 오류가 생기기 떄문에 고안되었다. 다른 오류 정정 기법이 많지만, 그 중 대표적인 기법이 Parity bit라 할 수 있다.

기본 아이디어는 1의 개수를 세는 것이다. 따라서, 송신자와 수신자들 끼리 사전에 기본적인 약속을 해야한다. 약속으로 보내려 하는 비트의 1의 개수를 항상 짝수나 홀수로 할지 정해야한다. 이렇게 정해놓는다면, 어떤 노이즈에 의해 0 -> 1로 바뀌거나, 1->0으로 바뀐다면 총 1의 개수가 홀수 -> 짝수 혹은 짝수 -> 홀수로 변하기 때문에 오류의 유무를 알 수 있다.

예로 이런 짝수 패리티 생성기는 XOR게이트의 특성을 생각하면 손쉽게 구현할 수 있다. XOR 게이트는 전체 입력에서 1의 개수가 홀수 일 때 1을 출력하며, 짝수일 때 0을 출력하기 때문이다. 예를 들어 3개의 input에 대해 짝수 패리티를 생각한다면, 아래 표와 같이 패리티 비트가 생성될 것이다. 오른쪽은 XOR게이트를 사용해 구현한 회로이다.

테이블이(가) 표시된 사진

자동 생성된 설명 

따라서, n개의 input에 대해 짝수 패리티 비트를 생성하기 위해서는 n-1개의 xor게이트가 필요하다. ( P= A\_0 xor A\_1… xor A\_n-1 ) 진리표를 보고 k-map을 이용해 불 함수식을 구할 수도 있다.

또한, 짝수와 홀수 패리티의 관계는 NOT의 관계에 있음을 쉽게 예측할 수 있다. 홀수 패리티의 경우에는 P = (A XOR B XOR C)’ 이 되며, A XNOR B XNOR C 와 같다.

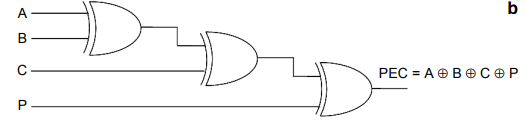
**2. Parity Bit 검사기**

가장 간단하다고 불리는 단일 패리티 검사 코드가 있다. 이는 앞서 말했듯이 비트 문자열에서 1의 개수를 감지하고, 만약 1의 비트 수가 짝수라면 짝수 패리티를 갖는다고 말한다. 마찬가지로, 홀수라면 홀수 패리티를 갖는다 말한다. 생성기가 짝수와 홀수로 나뉘는 것 처럼, 검사기 역시 짝수와 홀수가 있다.

예로 앞서 짝수 패리티 생성기로 3-bit의 메시지를 송신한다고 생각하면, 총 4-bit를 송신하게 된다. 이제, 짝수 패리티 검사기로 검사를 할 땐 4-bit 검사기로 확인하게 된다. 그런데, 검사기 역시 생성기와 똑같은 방식임을 알 수 있다. 1의 개수가 짝수여야 하므로, XOR게이트를 활용할 수 있다. 다음은 짝수 PEC의 진리표와 회로이다. 홀수 검사기 역시 XNOR로 구현가능하며 짝수와 NOT의 관계가 있다. 1은 에러를 의미한다.

테이블이(가) 표시된 사진

자동 생성된 설명



이런 점에서 생각해보면, 생성기와 검사기는 하나의 회로에서 구현 가능함을 알 수 있다. 위와 같은 회로에서 P = 0 으로 놓는다면, A XOR 0 = A 이기 때문에, 3-bit 패리티 비트 생성기의 역할이 가능하다.

하지만, 이러한 접근 방식은 당연하게도 오류의 수가 짝수일 때 감지할 수 없으며, 어떤 비트에서 오류가 생겨났는지 모른다는 것이다. 이러한 2개이상의 비트가 변경되는 오류를 폭주오류라 부른다.

이런 단일 1차원 패리티 비트의 문제점을 보완하고자 2차원 패리티 검사가 등장하였다. 2차원 패리티의 경우 송신하는 비트를 2차원으로 만든다. 예를 들어, 1011 1101 1000 0110 을 송신한다고 한다면, 다음과 같은 표를 만들 수 있다. 수평(LRC), 수직(VRC)으로 패리티를 추가하면 된다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | P |

따라서 전송하게 되는 데이터는 위의 표와 같다. 만약, 다음 빨간색에서 오류가 났다 생각해보면 첫 번째 세로줄에서 10101로 에러를 발견할 수 있고, 2번째 가로줄에서 01011로 마찬가지로 에러를 발견할 수 있다. 이에 따라, 에러의 위치가 둘이 겹치는 부분임을 알 수 있다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | P |

**텍스트, 낱말맞추기게임이(가) 표시된 사진

자동 생성된 설명**또한, 비슷한 예로 다음과 같이 2개의 에러 즉 폭주오류가 발생했을 때 역시 에러의 검출이 가능하다. 첫번째 줄 10101에서 에러의 검출이 가능하며, 4번째 줄 역시 10000으로 에러가 검출된다. 하지만, 이럴경우 2번째 가로줄은 에러를 검출하지 못하므로 에러의 정확한 위치는 알 수 없다. 오른쪽 그림은 에러 정정 불가의 예시이다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | P |

**3. 오류 검출기 및 정정기**

패리티 비트외에도 해밍코드라 부르는 오류 검출기 및 정정기가 존재한다. R.W Hamming이 개발했다. 일반적인 에러검출기와 달리 해밍코드는 스스로 1-bit 에러를 검출하고 수정할 수 있도록 개선되었다.

우선 해밍코드는 패리티 비트의 위치가 정해져있다. 전송할 데이터의 비트 수가 n일 때, 추가할 패리티 비트의 수 p는 다음 2^p >= p + n + 1 식을 만족하는 값이다.

예를들어 짝수 패리티이며, 4-bit를 송신한다고 하면, 패리티 비트의 위치는 p>=3 이다. 또한, 이때 추가해야하는 패리티 비트의 위치는 2^n 번째 (n = 0 ,1…) 위치이다. 따라서, 총 송신해야하는 bit는 7개가 된다.

만약, 1010이라는 데이터를 송신하기 위해서는 다음과 같다.

위치 : A7 A6 A5 A4 A3 A2 A1

값 : 0 1 0 p3 1 p2 p1 이 될 것이다. 이때, p의 값을 구하기 위해서 다음의 규칙을 따라야 한다.

1. 첫 번째 패리티 비트는 최하위 비트에 1이 있는 위치를 포함한다. ( 1= 1(2), 3 = 11(2), 5=101(2), 7 = 111(2) )
2. 두 번째 패리티 비트는 두 번째로 최하위 비트에 1이 있는 위치를 포함한다. ( 2 = 10(2), 3 = 11(2), 6 = 110(2), 7 = 111(2))
3. 이런식으로 패리티 비트에 포함되는 세트를 찾는다.

따라서, 이 규칙에 따라 먼저 p3 을 구해보면 p3 = A7,A6,A5 에 대한 패리티 비트이다. 따라서, p3 = 1이 된다. ( A7 XOR A6 XOR A5 = 1 = p3 ) 같은 방법으로 p2 = 0와 p1 =1 임을 알 수 있다. 최종적으로 송신하게 되는 비트는 1011010이다.

그러나 만약 1011011을 수신했다고 생각해보면,

위치 : A7 A6 A5 A4 A3 A2 A1

값 : 1 1 0 1 1 0 1 이때, p1을 확인해보면, A1,A3,A5,A7의 XOR 에서 오류가 났으므로, 1을 기록한다. 또한, p2 역시 A2, A3, A6,A7의 XOR에서 오류가 났으므로 1을 기록한다. 마지막 p3도 1이 기록된다. 111(2) = 7 이므로 최종적으로 7번째 자리에서 오류가 났음을 확인할 수 있다.

하지만, 해밍코드 역시 2-bit 에러가 발생할 경우 정확한 에러의 위치를 찾을 수 없다.

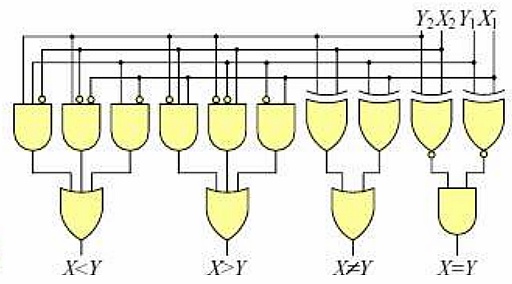
**4. N bit 비교기**

기존 1-bit 비교기는 앞선 주차에서 공부한 적이 있다.

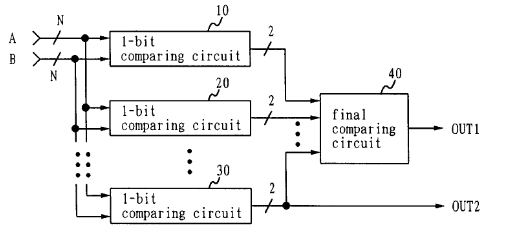
이를 바탕으로 n-bit 비교기의 구현이 가능하다. N-bit 비교기의 경우 1-bit 비교기를 직렬로 연결하여 확징이 가능하다. 앞선 주차에서 전가산기를 직렬로 연결하여 Ripple Carry Adder를 구현한 방식과 비슷하다. 또한, 2-bit의 경우 다음과 같이 진리표를 작성한 후 k-map을 사용해 손쉽게 구할 수 있다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 입력 | | 출력 | | | |
| A | B | A=B | A!=B | A>B | A<B |
| A1A2 | B1B2 | F1 | F2 | F3 | F4 |
| 00 | 00 | 1 | 0 | 0 | 0 |
| 01 | 0 | 1 | 0 | 1 |
| 10 | 0 | 1 | 0 | 1 |
| 11 | 0 | 1 | 0 | 1 |
| 01 | 00 | 0 | 1 | 1 | 0 |
| 01 | 1 | 0 | 0 | 0 |
| 10 | 0 | 1 | 0 | 1 |
| 11 | 0 | 1 | 0 | 1 |
| 10 | 00 | 0 | 1 | 1 | 0 |
| 01 | 0 | 1 | 1 | 0 |
| 10 | 1 | 0 | 0 | 0 |
| 11 | 0 | 1 | 0 | 1 |
| 11 | 00 | 0 | 1 | 1 | 0 |
| 01 | 0 | 1 | 1 | 0 |
| 10 | 0 | 1 | 1 | 0 |
| 11 | 1 | 0 | 0 | 0 |

텍스트, 낱말맞추기게임, 영수증이(가) 표시된 사진

자동 생성된 설명

4-bit 비교기의 대표적인 것이 IC 7485 비교기 이다. 특허청에 등록되어있는 N-bit comparator의 사진을 참고하면 아래 그림과 같다.



**5. IC 7485**

특이하게, 이 비교기는 4-bit 비교기 이지만, 확장을 위한 추가 입력이 존재한다. 이를 Cascading input으로 부른다. 아래는 진리표와 논리 회로이다.

**테이블이(가) 표시된 사진

자동 생성된 설명**

마찬가지로, IC 7485 비교기를 직렬로 연결하게 되면 자유롭게 n-bit 비교기 회로의 구현이 가능하다.

이제, 불함수를 생각해보면 A = B 이기 위해서는 서로 4개의 input이 모두 동일하여야 한다. 이는 즉, XNOR을 사용해 구현이 가능함을 말한다. XNOR의 경우 서로 같을 때 1을 출력하므로 Output(A=B) = (A3 ⊙ B3) (A2 ⊙ B2) (A1 ⊙ B1) (A0 ⊙ B0) 이다. 식의 간편성을 위해서 X\_i = A\_i ⊙ B\_i 로 표현한다.

이제 A>B 의 경우 다음 4가지 경우에 해당한다.

1. A3 = 1 이고 B3 = 0
2. A3 = B3 이고 A2=1 , B2 = 0
3. A3 = B3, A2=B2 이고 A1 =1, B1 = 0
4. A3=B3 , A2=B2, A1=B1 이고 A0 = 1 ,B0 = 0

비슷하게 A < B도 표현된다. 이를 바탕으로 Output(A>B) = A3B3’ + X3A2B2’ + X3X2A1B1’ + X3X2X1A0B0’ 이며, Output(A<B) = A3’B3 + X3A2’B2 + X3X2A1’B1+ X3X2X1A0’B0 이다.

**6. 기타이론**

부호화된 신호를 전송할 때 주로 발생하는 에러 유형은 다음 4가지로 나눌 수 있다.

1. 부호에 의해서 정정되어 오류가 발생하지 않은 것 처럼 보이는 경우
2. 검출을 할 수 있지만, 수정하지 못하는 경우
3. 검출을 못하는 경우
4. 검출은 되지만, 수정할 경우 잘못된 결과를 야기하는 경우

이 외에도 패리티 비트를 3차원까지 증가시킨 후 기존 2차원 패리티비트에 비해 정보율이나 오류정정불가확률의 면에서 효율적이라는 사실이 알려져있다.

오류 정정기/검출기에는 Checksum 이나 CRC 등이 있다. 또한 해밍코드의 확장 SECDED도 있다.

**7. 참고문헌**

All-optical parity generator and checker circuit employing semiconductor optical amplifier-based Mach–Zehnder interferometers

이주빈, 황철현, 전호진, 배의빈.(2014).패리티 부호의 N차원 확장에 관한 연구.한국컴퓨터교육학회 학술발표대회논문집,18(1),125-129.

S. S. Gayathri and A. V. Ananthalakshmi, "Design and implementation of efficient reversible even parity checker and generator," 2014 International Conference on Science Engineering and Management Research (ICSEMR), 2014, pp. 1-4, doi: 10.1109/ICSEMR.2014.7043605.

고형철, 김종우. (2018). 데이터 에러 검출과 수정에 대한 초등교육자료 개발. 정보교육학회논문지, 22(1), 169-176.

한국기술교육대학교 전기전자통신공학부. 7장 조합논리회로

<https://scienceon.kisti.re.kr/srch/selectPORSrchPatent.do?cn=USP1999115986538>

<https://ece-classes.usc.edu/ee459/library/datasheets/DM74LS85.pdf>