7주차 예비보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20211547 이름: 신지원

**1.**

Parity Bit 는 정보의 전달 과정에서 오류가 생겼는지를 검사하기 위해 추가된 비트를 의미한다. 문자열 내 1비트의 모든 숫자가 짝수인지 홀수인지를 보증하기 위해 데이터의 각 문자에 1비트를 더해 전송하는 방법으로 오류를 검사한다. 이는 전체 전달되는 비트의 1의 개수에 따라 odd Parity Bit, even Parity Bit로 구분된다. 만약 ‘11110000’ 수가 있을 때 even Parity Bit 라면, 마지막에 붙는 Parity Bit 는 0이 될 것이다. 중간에 오류로 인해 1이 소실된다고 하더라도 소실된 데이터가 1이었음을 유추할 수 있다. 따라서 정보 전달 과정에서 odd Parity Bit, even Parity Bit 둘 중 어떤 Parity Bit를 사용할건지 사전에 맞춰야 한다.

Parity Bit 생성기는 앞서 말한 Parity Bit를 생성한다.

스케치, 도표, 라인, 그림이(가) 표시된 사진

자동 생성된 설명

위는 3bit - even Parity Bit의 logic digagram 이다. P는 x, y, z의 exclusive OR 의 출력을 갖는다. 따라서 1의 개수가 홀수 일때만 P는 1을 가질 것이다.

도표, 라인, 폰트, 그래프이(가) 표시된 사진

자동 생성된 설명

위는 3bit - odd Parity Bit의 logic digagram 이다. (짝수와 같은 맥락의 diagram 을 첨부하고 싶었지만 구하기 어려워 대략적인 논리식만 유추하기 위해 위 사진을 첨부하겠다.) CP는 b와c의 ~ exclusive OR 과 a의 exclusive OR 의 출력을 갖는다. 따라서 1의 개수가 짝수 일때만 CP는 0을 가질 것이다.

**2.**

1번을 통해 Parity Bit를 생성하였다면, Parity Bit 확인은 어떻게 하게될까 ?

도표, 스케치, 라인 아트, 디자인이(가) 표시된 사진

자동 생성된 설명

위는 even- Parity Bit 검사기의 diagram 이다. Parity Bit 생성기에서 구한 p를 다시 출력과 함께 3번의 exclusive OR를 통해 Check 할 수 있다. 따라서 위는 even- Parity Bit 이기 때문에 C는 입력의 1의 개수가 홀수일 때만 1이라는 에러를 출력할 것이다.

도표, 라인, 그래프, 평면도이(가) 표시된 사진

자동 생성된 설명

반면 odd- Parity Bit 검사기는 세개의 not exclusive OR 를 사용하여 에러를 검출한다. 이렇게 되면 CP는 입력의 1의 개수가 짝수일 때만 1이라는 에러를 출력할 것이다.

**3.**

오류코드에는 대표적으로 해밍 부호, 해밍 코드를 말할 수 있다. 해밍부호란 데이터를 전송할 때 1비트의 에러를 정정할 수 있는 오류정정부호 중 하나다. Hamming 이라는 미국의 연구자에 의해 고안되었다. 해밍코드는 parity bit 와 달리 에러비트의 위치까지 알 수 있으며 발생한 에러를 정정할 수 있는 해밍코드다.

폰트, 텍스트, 화이트, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

데이터비트수가 8인 경우에 추가되는 ParityBit 의 수는 위의 공식과 같다.

텍스트, 폰트, 번호, 라인이(가) 표시된 사진

자동 생성된 설명

추가되는 Parity Bit 를 구하였다면 어디 추가되는지 비트 위치를 정해야 한다. 위는 데이터비트 수가 8일 때를 기준으로 잡고 구현한 표다.

많은 장점이 있지만, 많은 수의 체크비트를 요구하기 때문에 비효율적이라는 단점이 존재하기도 하다. 이처럼 해밍코드는 길이에 구애받지 않고 사용할 수 있으며 N개의 데이터에 K개의 parity bit를 더하여 새로운 cord word를 생성해낸다.

**4.**

n 비트 비교기는 n 비트 수를 가진 A, B가 같은지, 어떤 것이 더 큰지, 더 작은지를 결정하는 회로이다. 따라서 논리식은 A < B, A > B, A == B, A != B 4가지로 나눌 수 있다. 지난 실습에서 1bit비교기를 구현한 바가 있다.

텍스트, 폰트, 화이트, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

위는 5주차에서 구현한 1bit 비교기의 Verilog 코드 일부다.

이렇듯 n 비트 비교기는 2개의 수를 비교하는데 사용될 수 있다. 또한 서로 다른 비교기를 조합하여 새로운 비교기를 설계할 수 있다는 장점도 있는데, 예를 들어 N bit 비교기와 M bit 비교기를 결합시켜 (N+M) bit 비교기를 만들 수 있다. 추가적으로 ASCⅡ나 EBCDIC 코드와 같이 알파벳의 순서와 숫자의 오름차순으로 결정되어 있는 코드를 사용할 수도 있는데 이 경우에는 2개의 문자열의 크기를 비교하는 데에도 사용될 수 있다.

**5.**

IC 7485 비교기는 4비트의 비교기 IC이다. IC 7485 비교기는 확장입력을 통해 LSB에 입력되는다. 각각 AGBI, ALBI, AEBI 형식으로 이루어졌다. A가 B보다 클 때, A가 B보다 작을 때, A와 B가 같을 때 각각의 값은 AGBO의 출력이 1, ALBO의 출력이 1, AEBO의 출력이 1이 된다.

도표, 라인, 평면도, 패턴이(가) 표시된 사진

자동 생성된 설명

위는 7485 크기의 비교기 회로다.

**6.**

ECC 램에 대하여 알아보고자 한다. ECC는 일반 PC와 달리 방대한 정보를 처리하는 PC다. 일반 PC와 램에서부터 차이를 보이는데 ECC램은 에러를 정정하는 특수한 램이다. 실습에서 진행할 패리티 메모리가 ECC의 구성요소 중 하나인데 다른 램보다 많은 패리티 메모리가 있어 ECC 램은 오류 정정에 효과적인 메모리로 만들어진다.

**7. 참고문헌**

<https://www.electronicshub.org/parity-generator-and-parity-check/>