7주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231523 이름: 김민정

**1. Parity bit generator**

Parity bit는 오류를 detection할 수 있게 도입된 bit이다. 이 parity bit의 위치는 원래 data bit의 앞, 뒤, 중간 어디에 위치시켜도 괜찮다. Parity bit generator는 input data bit를 검사하여 이에 대한 parity bit를 추가한다.

이는 even parity bit generator와 odd parity bit generator으로 나뉜다. 먼저 even parity bit generator는 parity bit와 기존의 input data bit에서의 1의 개수가 짝수가 될 수 있게 만드는 generator이다. 이와 반대로 odd parity bit generator는 parity bit와 기존의 input data bit에서의 1의 개수가 홀수가 될 수 있게 만드는 generator이다. 이에 대한 예시를 들어보겠다. 0111이라는 input data bit가 있다고 가정해보자. 만약 이를 even parity bit generator 입력한다면. Parity bit는 1의 개수를 짝수로 만들기 위해서 1을 output(parity bit)로 도출할 것이다. 이와 반대로 odd parity bit generator는 0111을 입력한다면 1의 개수를 홀수로 만들기 위해서 0을 output(parity bit)로 도출할 것이다.

이에 대한 함수식은 XOR과 XNOR을 이용해서 제작할 수 있다. 1의 개수가 홀수 일 때 1을 출력하는 XOR logic gate의 특성과, 1의 개수가 짝수일 때 1을 출력하는 XNOR을 살펴보면, 전자로는 even parity bit generator, 후자로는 odd parity bit generator을 제작할 수 있다.

**2. Parity bit checker**

송신기에서 Parity bit generator에서 parity bit를 제작하고, 이러한 신호를 수신기로 보낸다. 수신기에서 이 신호를 받으면 parity bit checker을 통해 데이터를 오류없이 받았는지 확인하는 과정을 거친다. Parity checker의 output은 PCE(parity error checker)으로 표시된다. 만약 오류일 경우에는 PCE bit는 1을 output으로 가지고 만약 오류가 없고 정상적으로 신호가 전달되었다면 0을 output으로 가진다.

**3. parity bit 검사기 외 오류 검출기 및 오류 정정기**

Bit의 error detection만을 할 수 있는 parity bit와 달리 error를 정정할 수 있는 hamming code에 대해 설명해보겠다. Hamming code는 Richard Hamming이 제작한 코드로 correction을 진행할 수 있다. 예시를 들어서 설명해보겠다. 4개의 비트를 correction할 때는 2~4개의 parity bit를 사용할 수 있다. (이번 예시에는 3개의 detection bit를 사용한다.) a1, a2, a4를 detection bit를 사용해보자. (Hamming code에서 detection bit는 2의 지수 승만 가능하다). 이들의 논리식을 구해보면 a1 = a3^a5^a7, a2 = a3^a6^a7, a4=a5^a6^a7이다. 자 이제 오류를 correction 해보자. 만약 수신기에서 0010011의 신호를 받았다고 하자. 여기서 e1=0, e2=1, e4=0이라고 할 때, 오류 난 비트의 위치는 1\*0 + 2\*1 + 4\*0 = 2가 된다. 그러면 두번째 비트를 고쳐주면 0110011이 된다. 그러면 최종 input bit는 1011이 된다.

**4. N bit comparator**

N bit comparator는 N자리수의 bit를 비교하는 기계이다. 여기서는 right big, left big, equal이 존재한다. 여기서 대표적인 2-bit comparator에 대해 소개하고자 한다. 진리표->카르노맵->논리식으로 전개해보고자 한다.

1) Truth table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Truth table | | | | | | |
| Input | | | | Output | | |
| a1 | a2 | b1 | b2 | a<b | a=b | a>b |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

2) 카르노맵과 논리식(SOP)

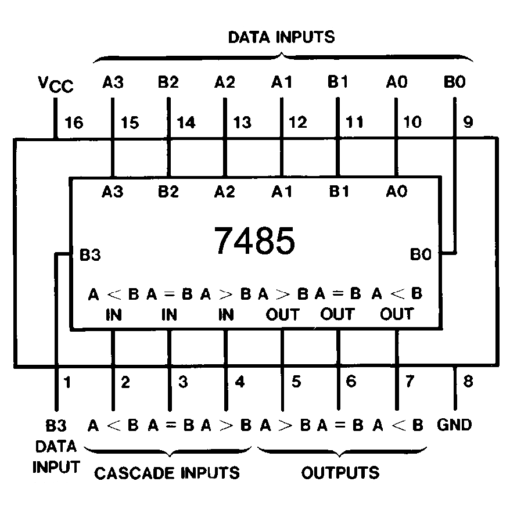
텍스트, 폰트, 도표, 번호이(가) 표시된 사진

자동 생성된 설명

**5. IC 7485 comparator**

N bit comparator는 N자리수의 bit를 비교하는 기계이다. 여기서는 right big, left big, equal이 존재한다. IC 7485는 4bit 자리수의 입력값 A, B를 비교하는 IC이다. 7485는 A0, A1, A2, A3, B0, B1, B2, B3를 입력값으로 받는다. 이후 A를 기준으로 E(equality), GT(greater than), LT(Less than) 을 output들이 될 수 있다. 이 각각 output pin의 출력값은 1(True)이나 0(False)로 표현된다.

IC 7485 역시 comparator이기에 비교 연산을 위해 디지털 시스템에서 사용된다. 숫자비교뿐만 아니라 제어 신호 생성, 우선순위 선정 등의 용도로 사용된다. 이는 Dual In-line Package로 제공되며, TTL logic family에 속한다. 아래는 IC 7485를 표현한 그림이다.



**6. 기타이론**

8421 code, 2421 code외의 다른 code에 대해 설명해보겠다. 바로 gray code이다. Gray code는 자료를 표현하는 code 중 하나로 I/O장치, A/D translator 등에서 숫자를 표현하기 위해 사용된다. 이 코드의 특징은 인접한 수의 code가 한자리 수밖에 차이나지 않는 특징이 있다. 그러므로 이러한 데이터를 송신할 때 1자리수의 bit만 변화하므로 오류가 발생할 확률을 줄인다.