**컴퓨터공학 설계 및 실험Ⅱ**

7주차 예비보고서

서강대학교 공학부 컴퓨터공학 전공

20171646 박태윤

**1. Parity Bit 생성기에 대해 조사하시오.**

패리티 비트(Parity Bit)란 정보의 전달 과정에서 오류의 발생여부를 검사하기 위해 전송하고자 하는 데이터 끝에 추가한 1비트를 말한다. 홀수(Odd Parity), 짝수(Even Parity) 2가지 종류의 패리티 비트가 존재한다.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **0** | **0** | **1** | **0** | **1** | **1** | **1** | **1** |
|  |  |  |  |  |  |  |  |  |
| **1** | **0** | **0** | **1** | **0** | **1** | **1** | **1** | **0** |

위와 같이 8-bit짜리 데이터 10010111을 전달한다고 가정했을 때, 빨간색 칸처럼 데이터의 총 1의 개수가 짝수(6개)가 되도록 하는 패리티 비트가 짝수 패리티 비트이고, 반대로 밑에 파란색 칸처럼 데이터의 총 1의 개수가 홀수(5개)가 되도록 하는 패리티 비트가 홀수 패리티 비트이다. 즉 정리하면 짝수 패리티 비트는 데이터에서 1의 개수가 짝수, 홀수 패리티 비트는 데이터에서 1의 개수가 홀수가 되도록 만드는 비트를 의미한다. 이를 통해 수신자는 전체 비트를 계산하고 패리티 비트를 다시 계산함으로써 데이터에 오류가 발생했는지 아닌지를 검사할 수 있다. 이러한 역할을 하는 패리티 비트를 생성하는 것이 Parity Bit 생성기이다.

3-bit의 데이터를 전송할 때 필요한 Parity Bit 생성기의 진리표는 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x** | **y** | **z** | **(짝수)** | **(홀수)** |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

3-bit Parity bit 생성기의 식은 다음과 같이 정할 수 있다.

|  |  |
| --- | --- |
| **짝수 패리티 비트 생성기** | **홀수 패리티 비트 생성기** |
| = x’y’z + x’yz’ + xy’z’ + xyz  = z’(x’y + xy’) + z(x’y’ + xy)  = z’(x) + z(x)’  = z(x) | = x’y’z’ + x’yz + xy’z + xyz’  = z’(x’y’ + xy) + z(x’y + xy’)  = z’(x)’ + z(x)  = (z(x))’ |

따라서 3-bit Parity bit 생성기의 논리회로는 다음과 같다.

|  |  |
| --- | --- |
| **짝수 패리티 비트 생성기** | **홀수 패리티 비트 생성기** |
|  |  |

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

Parity Bit 생성기를 통해 패리티 비트를 생성하고 기존에 수신하고자 했던 데이터와 생성한 패리티 비트를 같이 전송한다. 수신자는 전송받은 데이터와 패리티 비트에 대해 홀수함수를 취한다. 3-bit 데이터에 대한 Parity Bit 검사기를 예시로 들면 다음과 같다.

xyzP = C

이 때, 짝수 패리티 비트를 사용했으면 패리티 비트를 포함한 전달받은 비트에서 1의 총 개수가 짝수이기 때문에 C의 값은 0이어야 정상이며 1이면 오류가 발생한 것이다. 반대로 홀수 패리티 비트를 사용했다면 C의 값은 1이어야 정상이다. 위의 Parity Bit 검사기의 논리회로는 다음과 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명

오류가 발생을 할 시 C의 값이 바뀌기 때문에 오류를 검출할 수 있다. 그러나 홀수 개 비트에서 오류가 발생하면 오류 발생 여부를 알 수 있지만 짝수 개 비트에서 오류가 발생하면 오류 발생 여부를 검출할 수 없다는 단점을 지닌다.

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

- 해밍 코드(Hamming Code)

수신 측에서 오류가 발생한 비트를 검출 후 직접 수정을 하는 방식이다. 1-bit의 오류만 수정이 가능하며 정보 비트 이외에 잉여 비트가 많이 필요하다. 생성, 검출, 수정 3단계를 거친다. 8-bit 데이터를 예시로 들면 다음과 같다.

텍스트이(가) 표시된 사진

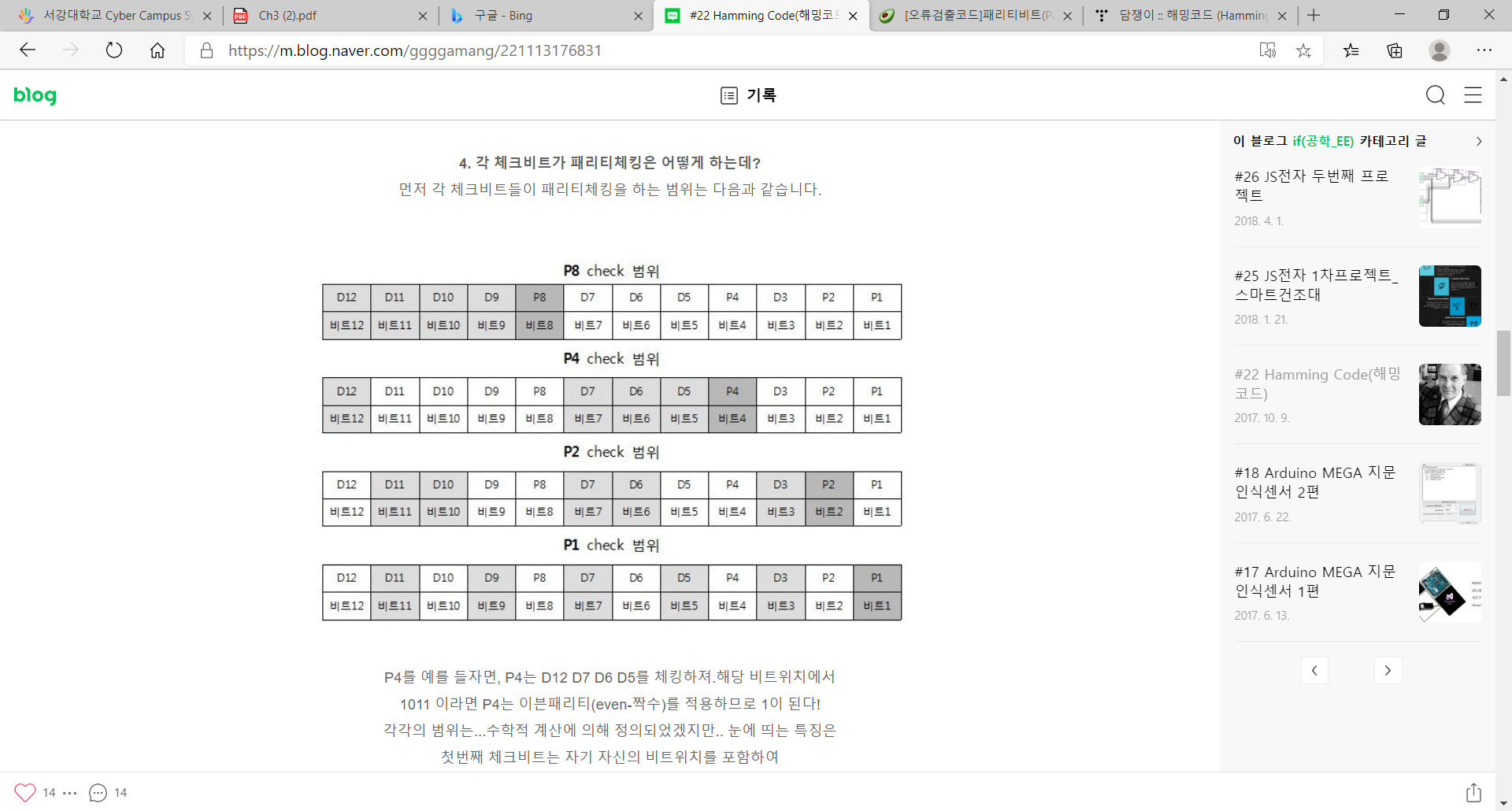
자동 생성된 설명

그림과 같이 자리마다 체커비트를 삽입한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

체커비트가 들어가지 않은 빈 칸에 기존에 전송하고자 했던 데이터 비트들을 삽입한다.



각 체커비트들은 그림과 같이 해당하는 범위의 비트를 체크할 수 있으며 짝수 패리티 비트이다. 이 때, 해당 범위는 체커비트가 삽입된 위치인 에 따라 나뉘어지는 것을 확인할 수 있다. 즉, P2같은 경우 처음 체커비트는 자신을 포함하는 개의 비트와 개를 건너 뛴 이후 개의 비트를 검사하며 이 패턴을 반복한다. 만약 8-bit 01010101을 전송하려고 하는데 D9에서 에러가 발생하여 01000101이 전송된 경우가 있다 가정해보자. 기존의 01010101에서 체커비트 P1,P2,P4,P8은 짝수 패리티 비트이기 때문에 (0,1,1,1)이 나오는 것을 확인할 수 있다. 하지만 01000101로 잘못 전송된 경우 체커비트는 (1,1,1,0)이 나오게 된다. P8과 P1를 통해 오류가 발생했음을 확인할 수 있고 두 체커 비트가 검사하는 공통적인 한 개의 비트는 D9가 유일하므로 D9에서 오류가 발생했음을 알 수 있다.

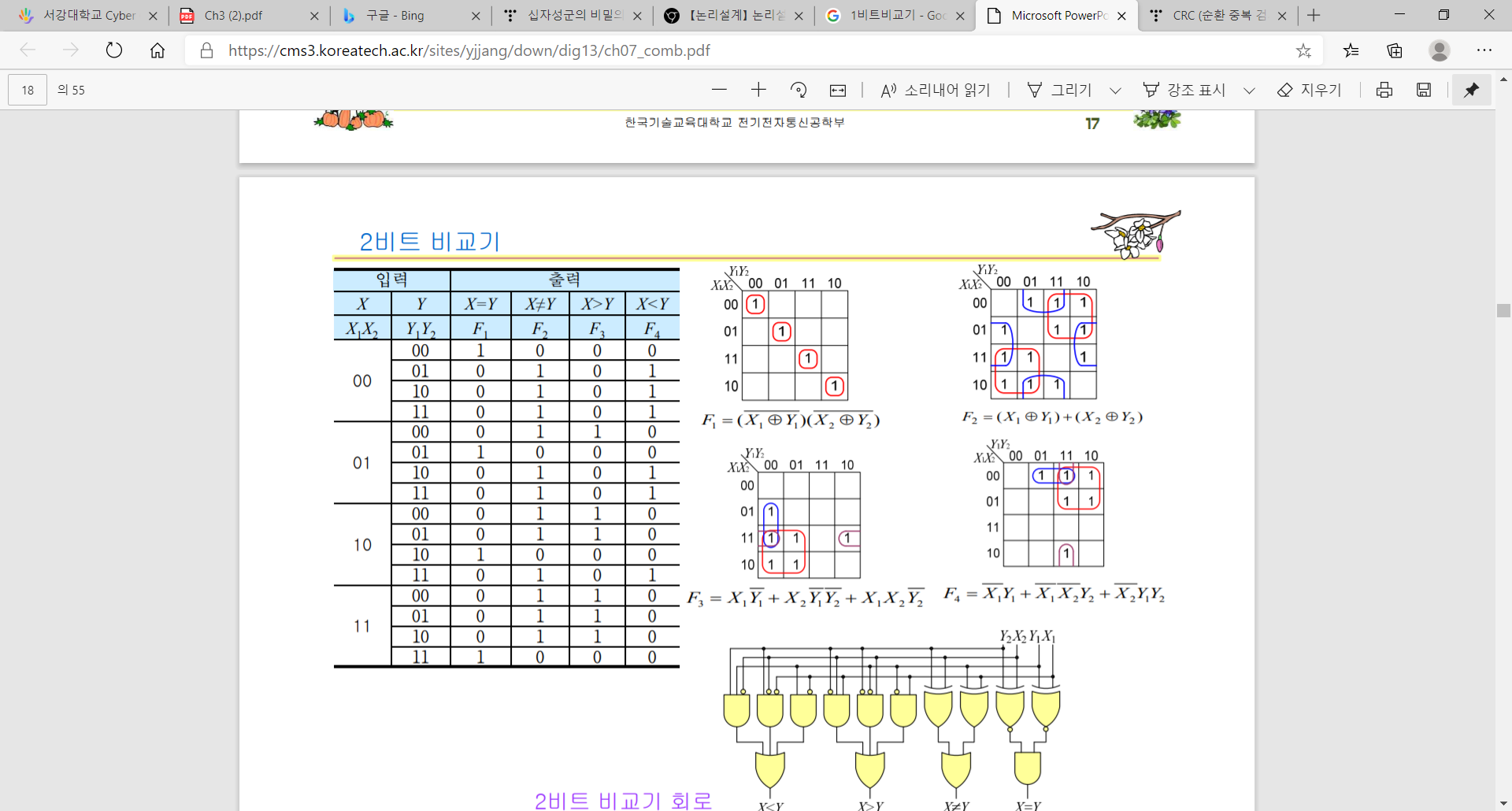
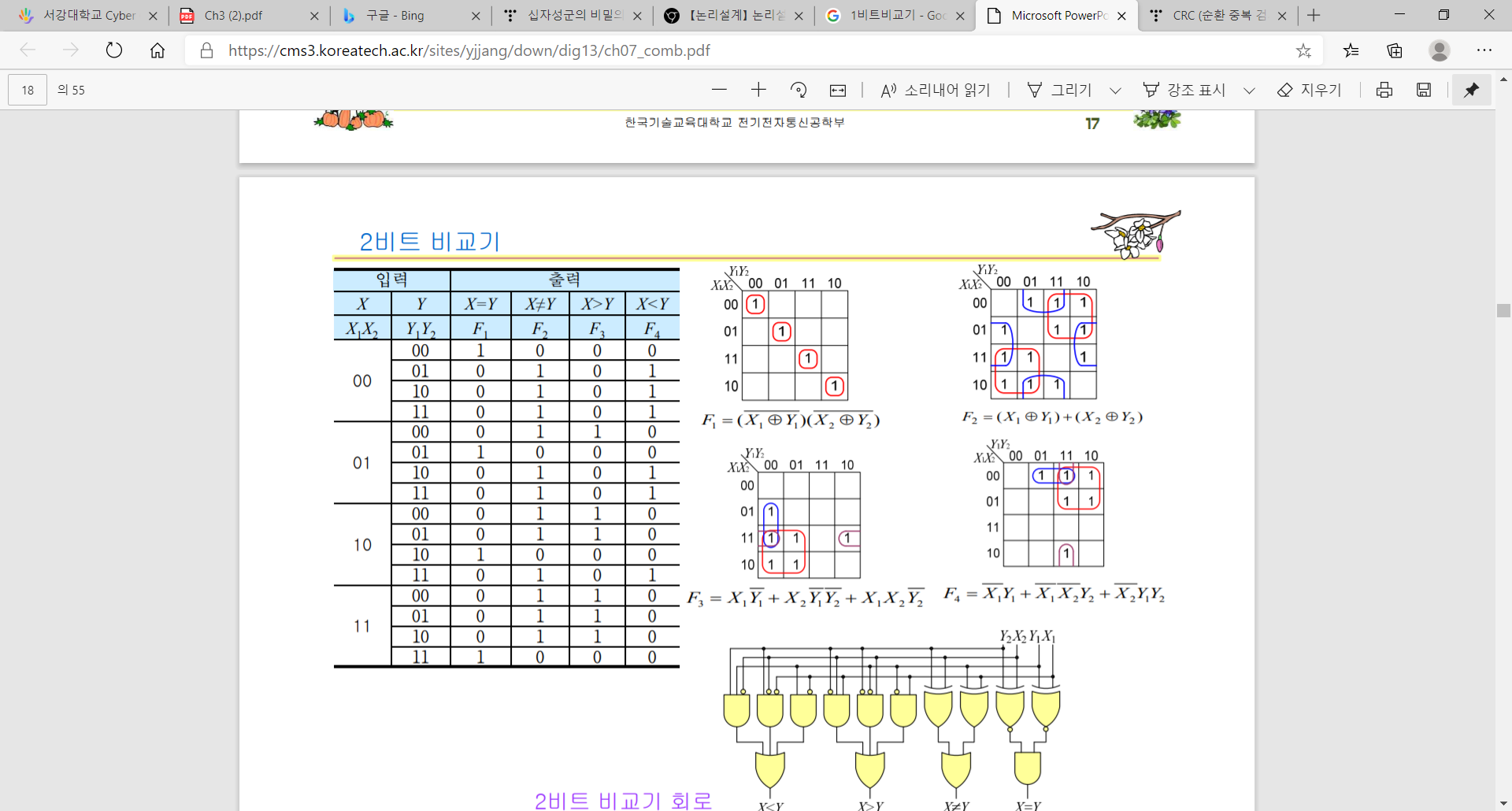
**4. N bit 비교기에 대해서 조사하시오.**

N개의 비트로 이루어진 두 수 A, B를 비교하여 A와 B의 일치여부와 대소관계를 출력한다. 가장 작은 1-bit비교기의 진리표와 논리회로는 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

이어서 이보다 큰 2-bit의 수를 비교하는 2-bit비교기의 진리표와 논리회로는 다음과 같다.



위의 1-bit 비교기, 2-bit 비교기와 마찬가지로 다른 N-bit 비교기 또한 진리표를 통해 논리회로로 구현할 수 있다.

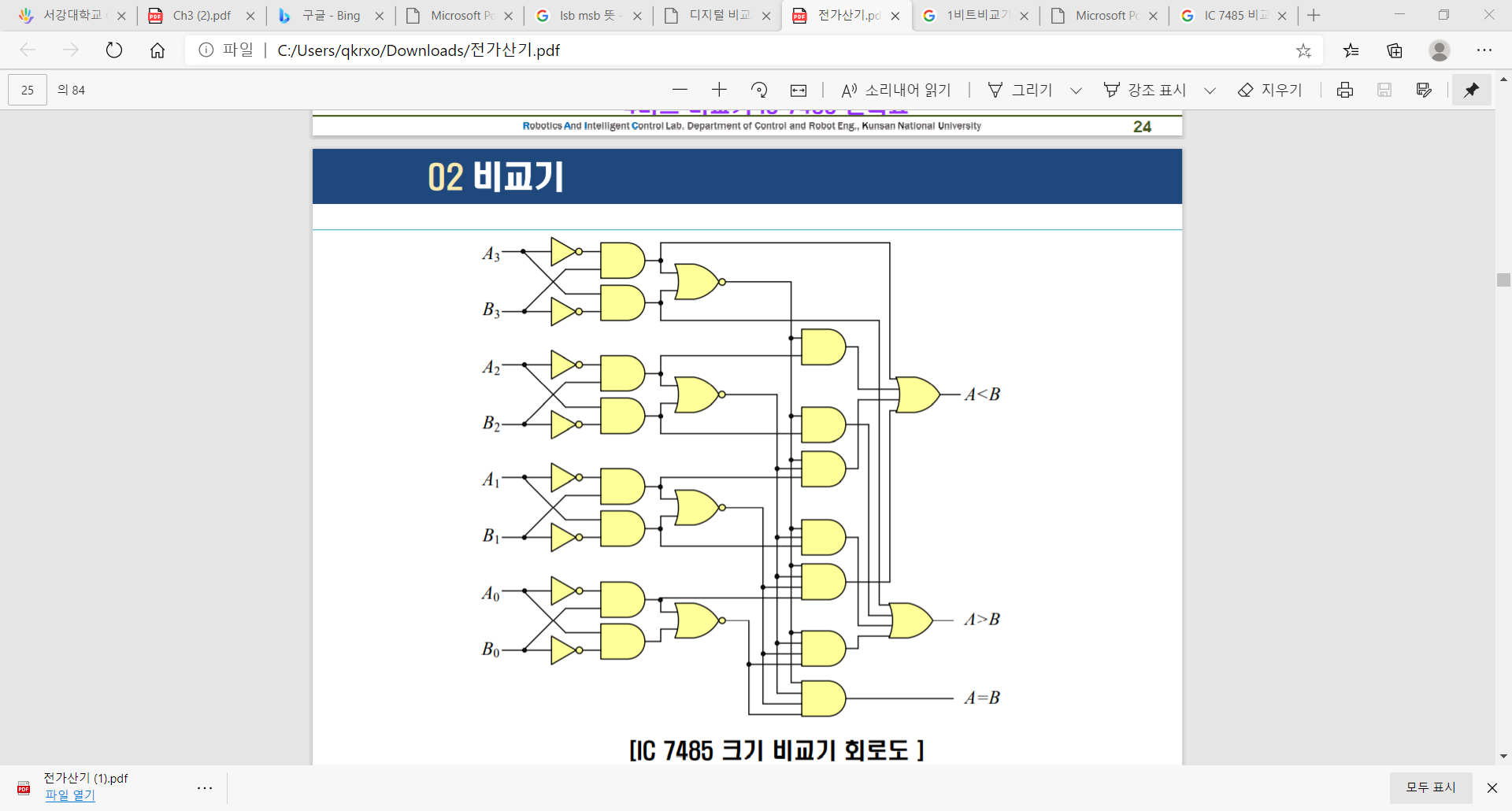
**5. IC 7485 비교기에 대하여 조사하시오.**

IC 7485는 4-bit의 두 수를 비교하는 비교기이다. 4-bit인 두 수 A,B가 있으면 A와B가 일치하는지, 일치하지 않으면 A와 B중 어느 수가 큰지 출력한다. 진리표는 다음과 같다.

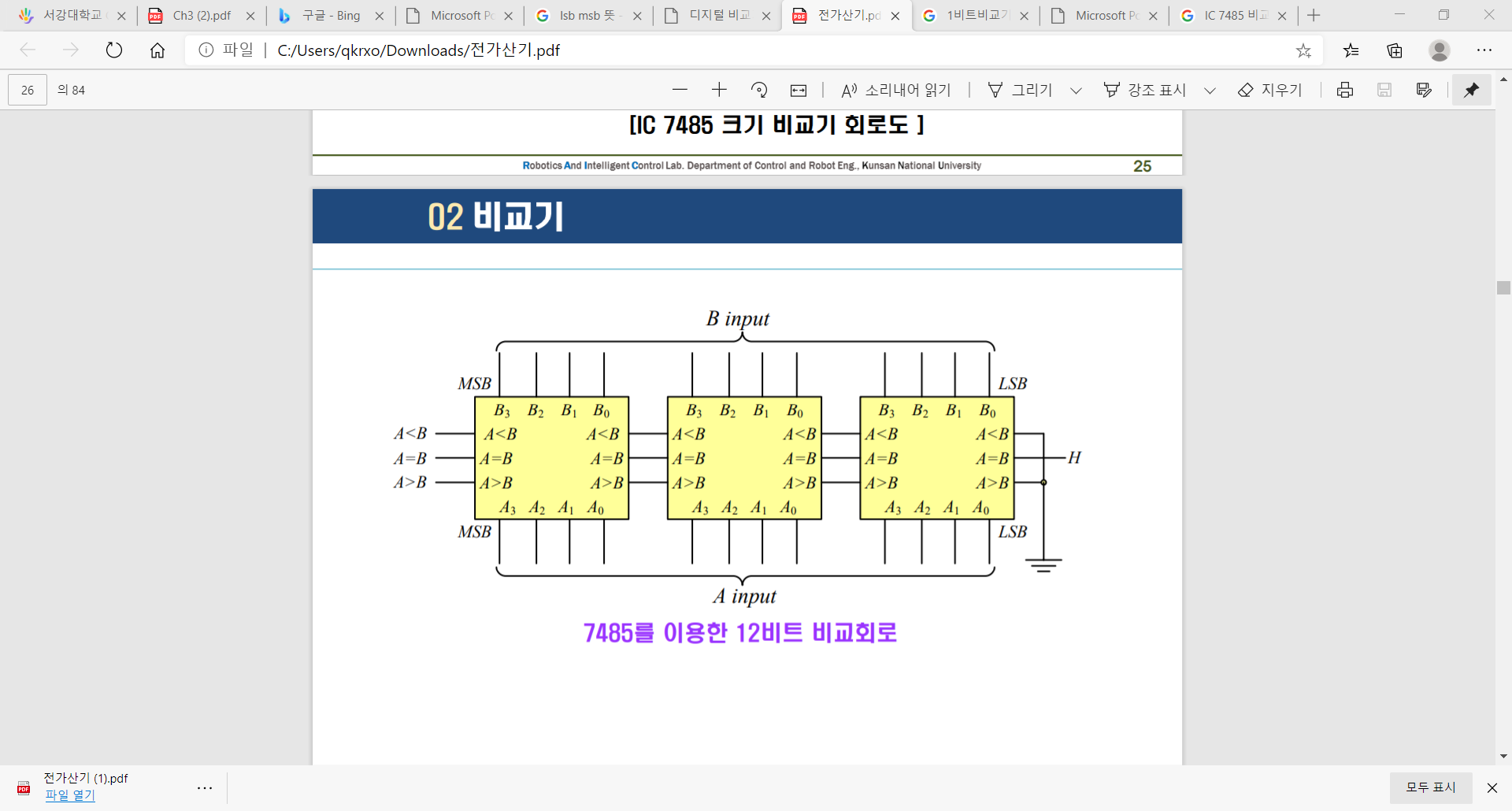
테이블이(가) 표시된 사진

자동 생성된 설명

A = , B = 이며 AGBI, ALBI, AEBI는 8-bit 비교기나 16-bit 비교기와 같이 4-bit비교기인 IC7485비교기를 여러 개 연결했을 때, 하위 4개의 bit 크기 비교 결과로 LSB(Least Significant Bit)으로 입력이 된다. IC7485 비교기의 논리회로는 다음과 같다.



이 IC7485 비교기를 직렬로 연결하여 8-bit, 12-bit, 16-bit….비교기를 만들어 낼 수 있으며 다음 그림은 12-bit 비교기의 논리회로를 나타낸 것이다.



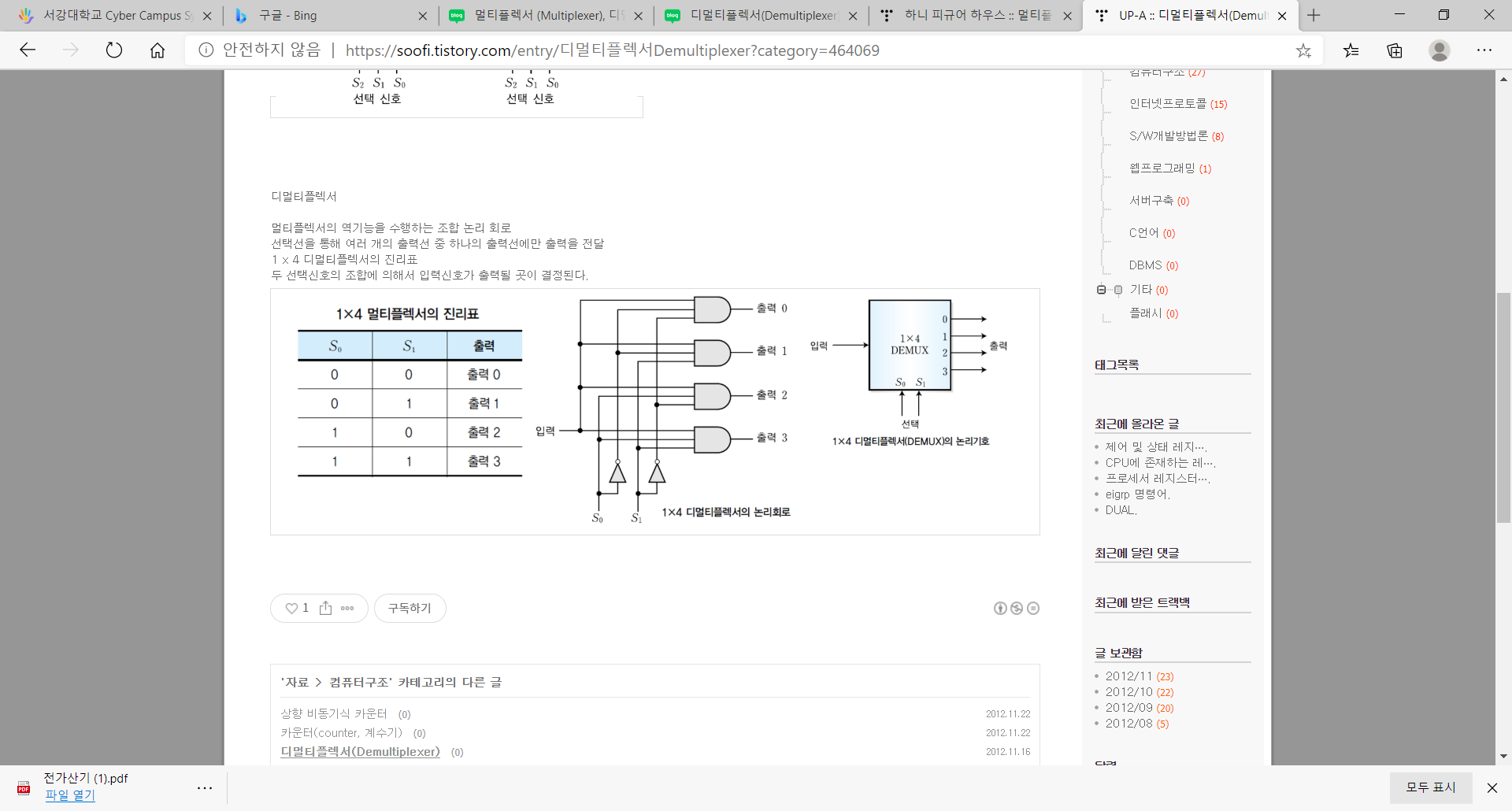
**6. 기타이론**

멀티플렉서는 MUX라고도 불리며 n개의 입력 중 하나를 선택하여 출력으로 내보내는 논리 회로이다. 4-input 멀티플렉서의 진리표와 논리회로는 다음과 같다.

|  |  |
| --- | --- |
|  |  |

MUX는 다수의 호스트 컴퓨터 또는 단말기에서 작성된 데이터를 하나의 통신회선을 이용하여 전송할 수 있게 해주는 회로로 이를 통해 회선 비용을 절감하고 전송 효율을 높일 수 있다.

멀티플렉서와 반대 개념인 디멀티플렉서 또한 존재한다. 디멀티플렉서는 한 개의 입력을 받아들여 여러 개의 출력으로 내보내는 논리 회로이다. 4-output 디멀티플렉서의 진리표와 논리회로는 다음과 같다. 개의 입력을 받아 n개의 출력으로 내보내는 인코더와는 출력 개수에서 차이를 보인다.



디멀티플렉서는 단일 소스를 여러 목적지에 연결할 필요가 있는 분야에서 활용된다. n개의 입력을 받아 개의 출력으로 내보내는 디코더와는 입력 개수에서 차이를 보인다.