

7주차 예비보고서

전공: 신문방송학과

학년: 4학년

학번: 20191150

이름: 전현길

1. Parity Bit 생성기에 대해 조사하시오.

binary bit 정보를 전송할 때, 오류가 발생했는지를 검출하기 위해 parity bit가 사용된다. 구체적 동작은 even parity인지 odd parity인지에 따라 달라지는데, 이는 정보를 전송하는 주체들이 결정하기(=약속하기) 나름이다.

even parity에서 parity bit는 binary data에서 '1'의 수가 홀수 개이면 1로, '1'의 수가 짝수 개이면 0으로 set된다. 결과적으로, 전체 binary data의 '1'은 항상 짝수 개가 된다. odd parity는 반대로 동작하여 전체 binary data의 '1' bit는 항상 홀수 개가 된다.

even parity bit generator의 경우 binary data의 각 bit들(input)을 모두 XOR 연산하면 parity bit(output)를 생성한다. 예로 3bit binary data를 전송할 경우 다음과 같다.

$$P = d_1 \oplus d_2 \oplus d_3$$

2. Parity Bit 검사기에 대해 조사하시오.

binary data를 수신받았을 때 parity가 적절한지 검출하기 위해서 parity checker가 사용된다. parity bit checker의 output은 PEC(parity error checker)로 표시되며, (even parity를 기준으로) 1이 홀수 개라면 1(=error)을 표시하고 1이 짝수 개라면 0을 표시한다.

even parity checker의 경우 binary data의 각 bit들과 parity bit(input)을 모두 XOR 연산하면 PEC를 출력한다. 예로 3bit binary data와 1bit parity bit에 대한 parity checker의 경우 다음과 같다.

$$PEC = P \oplus d_1 \oplus d_2 \oplus d_3$$

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

$$2^r \geq m + r + 1$$

해밍 코드(hamming code)는 1bit 이하의 오류가 발생했을 때 검출할 뿐만 아니라 정정까지 할 수 있는 코드이다. 해밍 코드에 따르면 m 이 data bit, r 이 check bit일 때 위의 공식을 만족하는 r 개의 check bit가 필요하다.

해밍 코드에 따라 인코딩된 총 bit data에서, check bit는 2^n 위치($n = 0, 1, 2, \dots$)에 순서대로 삽입된다. 예를 들어 4개의 check bit, 8개의 data bit를 갖는 data라면 아래와 같이 1, 2, 4, 8번째 bit가 check bit가 된다.

bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1
--------	--------	--------	-------	-------	-------	-------	-------	-------	-------	-------	-------

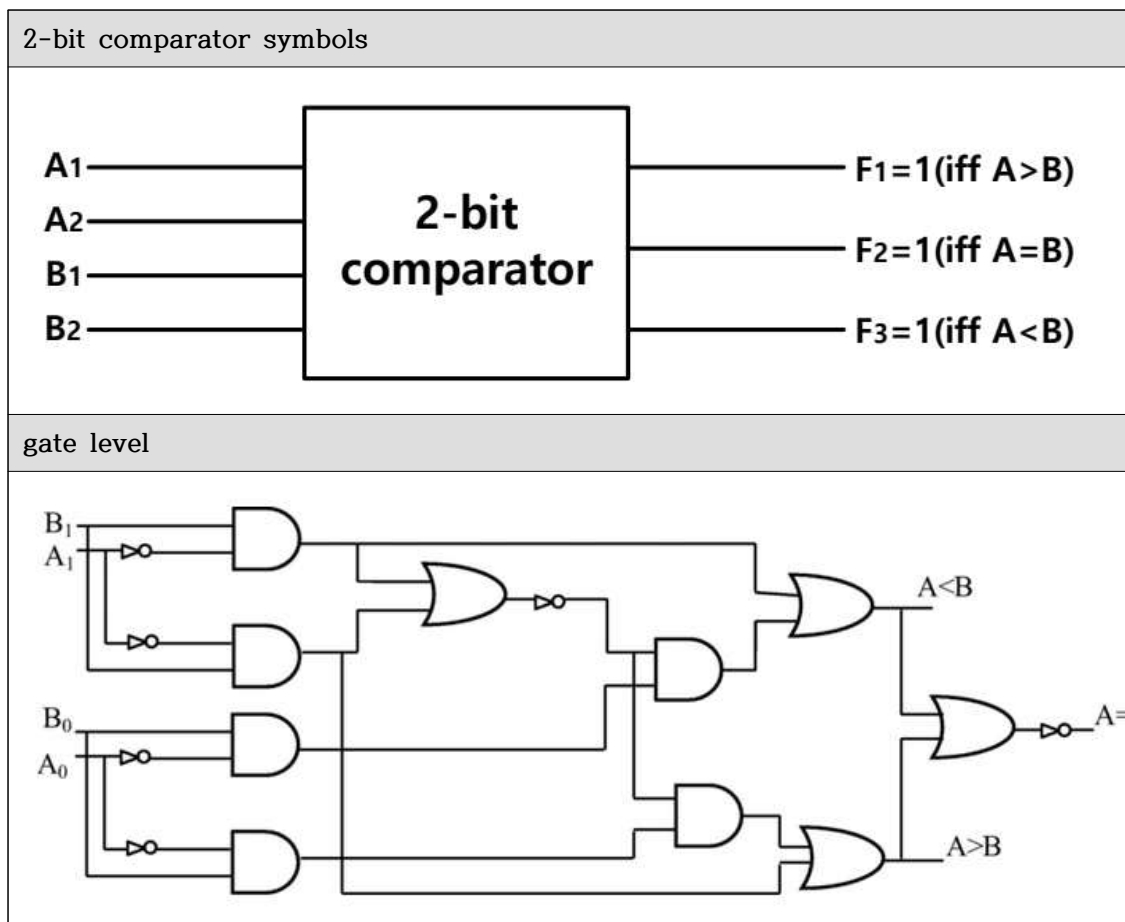
각 check bit들은 자기 자신부터 2^n 만큼의 범위를 번갈아가며 확인해 even parity check를 수행한 결과를 자신의 값으로 저장한다. 그림으로 표현하면 이해하기 쉬운데, 아래와 같이 even parity를 검사한다. (p16은 무시)

Bit position		1	2	3	4	5	6	7	8	9	10	11	12
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8
Parity bit coverage	p1	✓		✓		✓		✓		✓		✓	
	p2		✓	✓			✓	✓			✓	✓	
	p4				✓	✓	✓	✓					✓
	p8								✓	✓	✓	✓	✓
	p16												

위 원리에 따라 check bit의 값을 설정한 뒤, 수신된 측에서는 각 data bit를 바탕으로 다시 계산한 check bit와, 실제 송신된 check bit에 대해 XOR 연산을 수행한다. 1bit 이하의 오류가 발생했을 경우, XOR 연산의 결과값은 정확히 오류가 발생한 지점을 가리킨다. 오류가 발생한 지점의 bit를 뒤집으면 오류를 정정할 수 있다.

4. N-bit 비교기에 대해서 조사하시오.

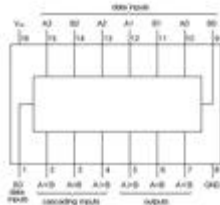
N-bit 비교기는 N-bit의 이진수 A, B 둘을 비교하여 $A > B$, $A = B$, $A < B$ 의 결과를 출력하는 조합 논리 회로이다. MSB부터 비교해 먼저 대소를 비교하고, 그 결과가 같다면 다음 bit를 비교하는 방식으로 연산을 수행한다. 아래는 2bit comparator의 동작을 표현한 symbol과 gate level circuit diagram이다.



5. IC-7485 비교기에 대해서 조사하시오.

조합 논리 회로 IC 74LS85(4비트 크기 비교기 IC)

두 개의 4비트를 입력 받아 크기를 비교하여 $A > B$, $A = B$, $A < B$ 등으로 출력하는 비교기 IC이다.



연결 핀 다이어그램



74LS85 실물

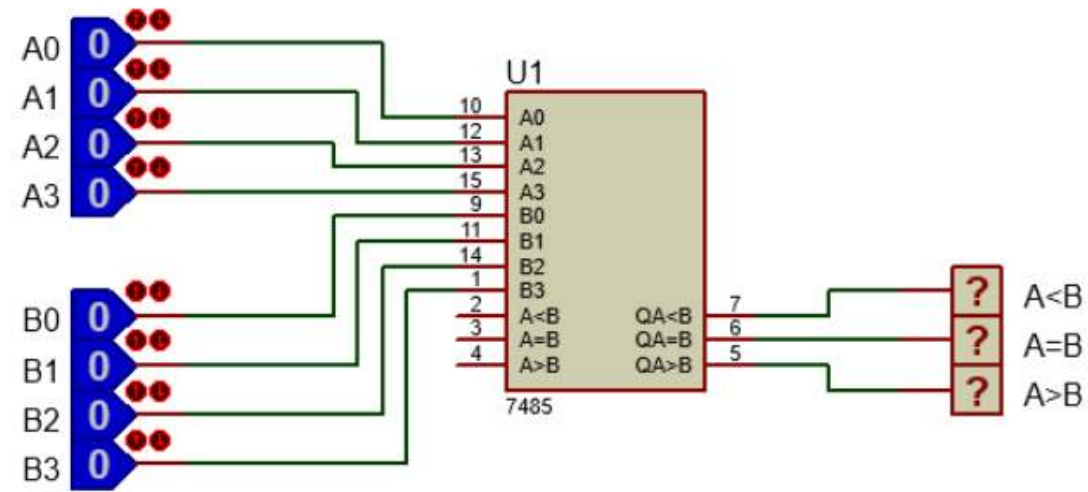
IC-7485 비교기는 4bit 전비교기 IC로, 두 수 A, B에 대해 A_0, A_1, A_2, A_3 과 B_1, B_2, B_3, B_4 를 4bit와 하위 자릿수의 비교 결과를 입력받아 3개의 입력을 비교한다. 따라서 IC-7485 비교기를 이용하면 4bit가 넘는 숫자에 대해서도 비교를 수행할 수 있다.

예를 들어 8bit 숫자 A, B를 비교한다면, 먼저 하위 4bit를 비교한 뒤 발생한 출력을 다시 입력한 뒤 상위 4bit를 비교한다. 예를 들어, 상위 4bit에서 $A=B$ 이고 하위 4bit에서 $A > B$ 였다면, 최종 결과는 $A > B$ 이다.

N-bit 비교기와 IC-7485 비교기의 관계는 반가산기와 전가산기의 관계와 유사하다. 가산기처럼 ripple carry adder처럼 구현할 수도, serial adder처럼 구현할 수도 있다.

입력							출력		
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	AGBI	ALBI	AEBI	AGBO $A > B$	ALBO $A < B$	AEBO $A = B$
$A_3 > B_3$	X	X	X	X	X	X	1	0	0
$A_3 < B_3$	X	X	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	1	0	0
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	0	0	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	0	1	1	0

IC 7485 comparator



6. 기타 이론

전가산기를 연결하여 ripple carry adder을 만들듯이, 4bit comparator를 연결해 8bit, 16bit comparator를 만들 수도 있다. 반대로 serial adder를 만들었던 것처럼 clock에 따라 4bit씩 비교를 수행하도록 할 수도 있다.

아래의 회로는 1bit 단위로 slice된 값을 바탕으로 비교 연산을 수행하는 serial comparator의 gate-level circuit diagram이다. 전가산기에서 학습한 내용을 비교기에도 적용할 수 있음을 확인할 수 있었다.

