컴퓨터공학실험 2 7 주차 예비보고서

전공: 컴퓨터공학과 학년: 2 학번: 20191559 이름: 강상원

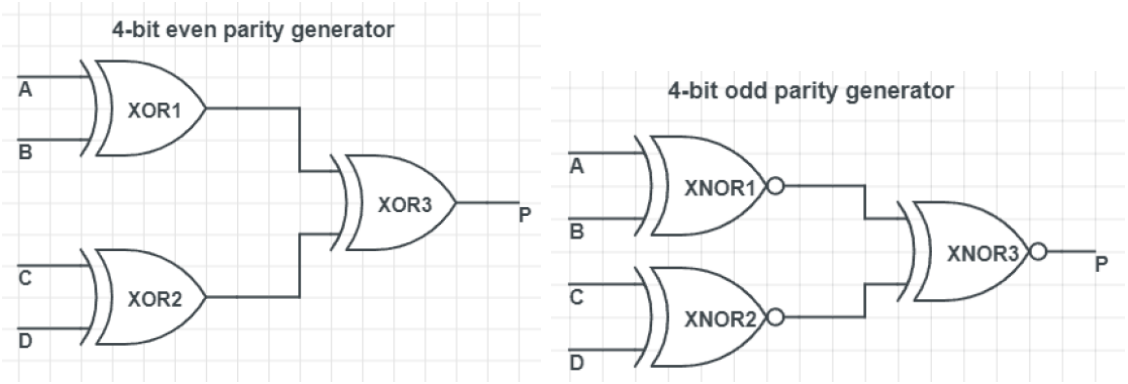
1. Parity Bit 생성기에 대해 조사하시오.

2진수 정보를 주고받으며 중간에 오류가 있는지 확인하기 위하여 parity bit을 사용한다. Binary bit에서 비트가 하나만 바뀌어도 전혀 다른 정보를 나타내기 때문에 parity bit이 필요하다.

parity bit에는 두 가지 종류가 있는데, 짝수 parity와 홀수 parity이다.

짝수 parity는 전체 중 1의 개수가 짝수가 되도록 만들어주고, 홀수 parity는 1의 개수를 홀수로 맞춘다. 따라서 parity bit을 추가하기 위해서는 1의 개수가 짝수인지 홀수인지 확인하는 XOR 또는 XNOR gate를 사용한다. (짝수 parity: XOR gate, 홀수 parity: XNOR gate)

다음은 4비트 짝수/홀수 parity bit 생성기의 예시이다.

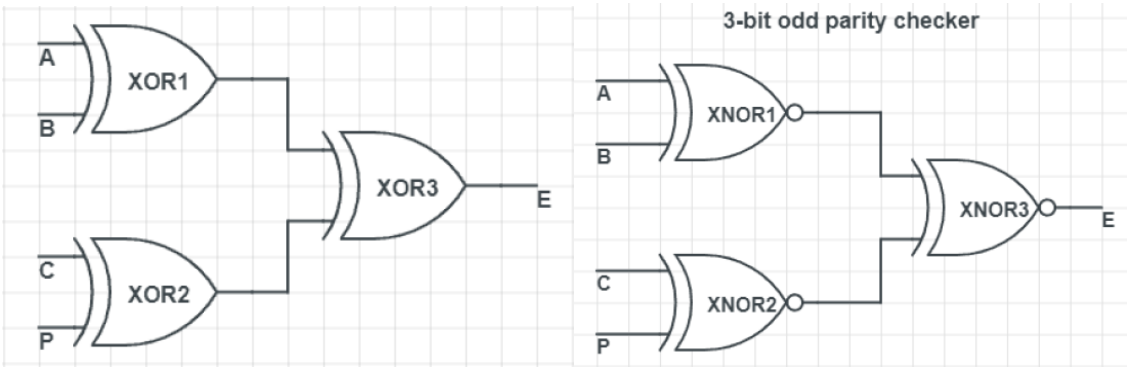


2. Parity Bit 검사기 에 대해 조사하시오.[검사 부호 종류 포함]

Parity bit 검사기도 parity bit 생성기와 마찬가지로 짝수/홀수 검사기로 나뉜다.

짝수 parity bit 검사기는 입력값을 모두 XOR 했을 때 1이 반환되면 오류를 감지하고, 홀수 parity bit 검사기는 입력값을 모두 XOR 했을 때 0이 반환되면 오류를 감지한다.

다음은 3비트 짝수/홀수 parity bit 검사기의 예시이다.



Parity bit 검사기는 2개 이상의 bit에서 문제가 생기면 오류를 감지하기 어렵다. 일례로 2개의 bit에서 문제가 발생하면 1의 개수 짝/홀에 변동이 없다.

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

Parity Bit 검사기 외에 Hamming Code 방식의 오류 검출기가 존재한다. Hamming Code는 Parity Bit 검사기와 달리 최대 2비트의 오류를 감지하고, 1비트 오류를 수정할 수 있다. 대표적인 구성 방법은 1부터 시작해 비트들의 번호를 매기고, 2의 거듭제곱의 위치에 해당하는 비트들을 검사 비트로 설정하는 것이다. 4비트의 정보를 검사하기 위해서는 다음과 같은 형태를 띤다.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7
BIT 1	X		X		X		X
BIT 2		X	X			X	X
BIT 3				X	X	X	X

$$a_1 = a_3 \oplus a_5 \oplus a_7, \quad a_2 = a_3 \oplus a_6 \oplus a_7, \quad a_4 = a_5 \oplus a_6 \oplus a_7$$

아래와 같이 비트들을 묶은 형태로 검사한다.

오류가 없다면 아래의 세 식 모두 결과값이 0이어야 한다.

$$e_1 = a_1 \oplus a_3 \oplus a_5 \oplus a_7, \quad e_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7, \quad e_4 = a_4 \oplus a_5 \oplus a_6 \oplus a_7$$

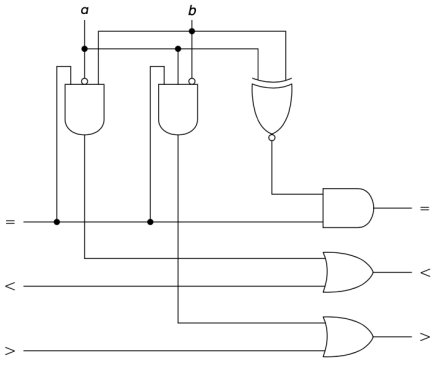
오류가 있다면, $4e_4 + 2e_2 + e_1$를 계산하여 오류가 난 위치를 확인할 수 있다.

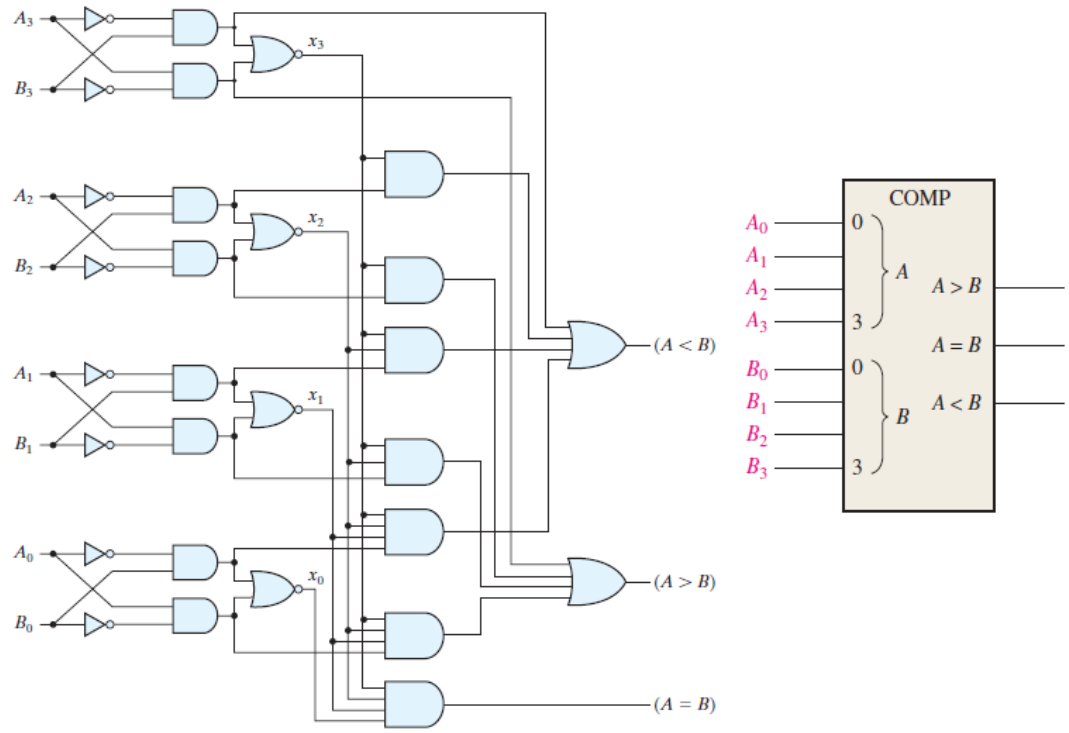
Check Sum 검사기는 bit 정보를 모두 더해 Check Sum 숫자를 구하고 bit 수의 modular를 이용, 재구성한다. 기존 데이터와 재구성된 데이터를 함께 전송하며 이를 수신자 측에서 Check Sum이 알맞은지 확인하여 오류를 확인하는 방식이다.

4. N bit 비교기에 대해서 조사하시오.

4비트 비교기에 대해서는, 6주차 결과보고서의 추가 이론 조사에서 한 차례 다룬 바 있다. 다시 정리하자면 비교기는 두 수를 비교해서, 크기가 같은지 어느 한 쪽이 더 큰지 알려준다. 다중 비트 수를 비교할 때는 어느 한 비트라도 일치하지 않으면 다른 수임을 나타내야 한다.

1 비트 비교기는 다음과 같은 형태이다.





4비트 비교기는 위와 같은 형태로 설계할 수 있다.

두 수의 대소관계까지 알려주는 4비트 비교기를 구현하기 위해서는 다음과 같은 관계를 알아야 한다.

$a > b$ if $a_4 > b_4$ or $(a_4 = b_4$ and $a_3 > b_3)$ or $(a_4 = b_4$ and $a_3 = b_3$ and $a_2 > b_2)$ or $(a_4 = b_4$ and $a_3 = b_3$ and $a_2 = b_2$ and $a_1 > b_1)$

$a < b$ if $a_4 < b_4$ or $(a_4 = b_4$ and $a_3 < b_3)$ or $(a_4 = b_4$ and $a_3 = b_3$ and $a_2 < b_2)$ or $(a_4 = b_4$ and $a_3 = b_3$ and $a_2 = b_2$ and $a_1 < b_1)$

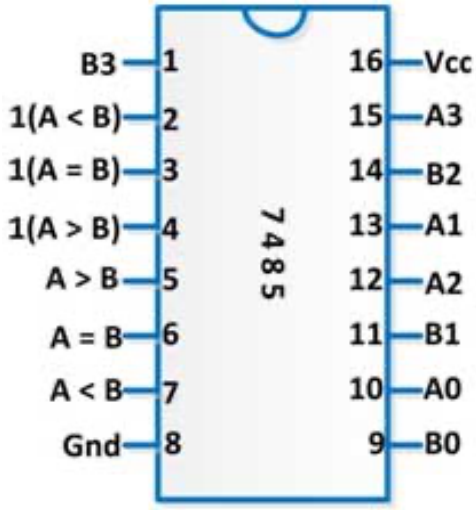
$a = b$ if $a_4 = b_4$ and $a_3 = b_3$ and $a_2 = b_2$ and $a_1 = b_1$

4비트 이상의 비트에 대해서는 위와 같은 4bit 비교기를 종속 연결하여 사용한다.

5. IC 7485 비교기에 대하여 조사하시오.

IC 7485 비교기는 입력과 출력이 종속 연결된 4비트 비교기이다,. 가산기처럼 종속 연결되어 있는 신호는 낮은 자리로부터 높은 자리로 전달된다.

a 입력이 b 입력보다 큰 경우 1 출력, 같은 경우에는 그 아래 단의 > 출력이 큰 경우 > 이 1이 된다. 아래 개형을 확인해 보면 기본 입력 a, b 이외에 3개의 입력이 추가로 있음을 알 수 있다. 이는 종속 연결 때문이다. 각 핀의 기능을 정리한 도표를 함께 첨부하였다.



Pin Number	Pin Name	Pin Function
1	B3	MSB of Binary Number B
2	A < Bin	Input for A < B on cascading
3	A = Bin	Input for A = B on cascading
4	A > Bin	Input for A > B on cascading
5	A > Bout	Comparator Output for A > B
6	A = Bout	Comparator Output for A = B
7	A < Bout	Comparator Output for A < B
8	GND	Ground
9	B0	LSB of Binary Number B
10	A0	LSB of Binary Number A
11	B1	Third MSB of Binary Number B
12	A1	Third MSB of Binary Number A
13	A2	Second MSB of Binary Number A
14	B2	Second MSB of Binary Number B
15	A3	MSB of Binary Number A
16	VCC	Supply Voltage

TRUTH TABLE

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A_3,B_3	A_2,B_2	A_1,B_1	A_0,B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$O_{A>B}$	$O_{A<B}$	$O_{A=B}$
$A_3>B_3$	X	X	X	X	X	X	H	L	L
$A_3<B_3$	X	X	X	X	X	X	L	H	L
$A_3=B_3$	$A_2>B_2$	X	X	X	X	X	H	L	L
$A_3=B_3$	$A_2<B_2$	X	X	X	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	X	X	X	X	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	X	X	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	A_1=B1	$A_0>B_0$	X	X	X	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	X	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	H	L	L	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	H	L	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	X	X	H	L	L	H
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	H	H	L	L	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	L	L	H	H	L

6. 기타이론.

Hamming Code에서 검사 비트 수가 n이면 (n≥2), $2^n - n - 1$만큼의 정보 비트를 지원할 수 있다.

CHECK BITS	DATA BITS
2	1
3	4
4	11
5	26

- 확장 Hamming Code

  다른 Parity Bit을 하나 추가해 단일 비트 오류와 2 비트 오류를 구분할 수 있다. 이 확장 Hamming Code는 SECDED(Single Error Correction Double Error Detection)에서 널리 사용된다.

Bit Position	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit Number	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data/Parity Bit	P5	D 15	D 14	D 13	D 12	D 11	P4	D 10	D9	D8	D7	D6	D5	D4	P3	D3	D2	D1	P2	D0	P1	P0

↑ Hamming Code Data & Parity Bits

$$P0 = D15 \oplus D13 \oplus D11 \oplus D10 \oplus D8 \oplus D6 \oplus D4 \oplus D3 \oplus D1 \oplus D0$$

$$P1 = D13 \oplus D12 \oplus D10 \oplus D9 \oplus D6 \oplus D5 \oplus D3 \oplus D2 \oplus D0$$

$$P2 = D15 \oplus D14 \oplus D10 \oplus D9 \oplus D8 \oplus D7 \oplus D3 \oplus D2 \oplus D1$$

$$P3 = D10 \oplus D9 \oplus D8 \oplus D7 \oplus D6 \oplus D5 \oplus D4$$

$$P4 = D15 \oplus D14 \oplus D13 \oplus D12 \oplus D11$$

P1 ~ P4가 위와 같을 때, 하나의 추가적인 Parity Bit인 P5는 전체적인 parity bit으로, 모든 데이터 bit과 D15 ~ D0, parity bit, P0 ~ P4를 XOR함으로서 구해진다.