6주차 예비 보고서

20191621 이민영

1. 전 가산기 및 반 가산기에 대해 조사하시오. (예시 포함)

가산기는 여러 비트로 된 두 수를 더하는 덧셈 연산을 수행하는 논리회로를 말한다. 가산기의 종류에는 반 가산기(Half Adder)와 전 가산기(Full Adder)가 있다.

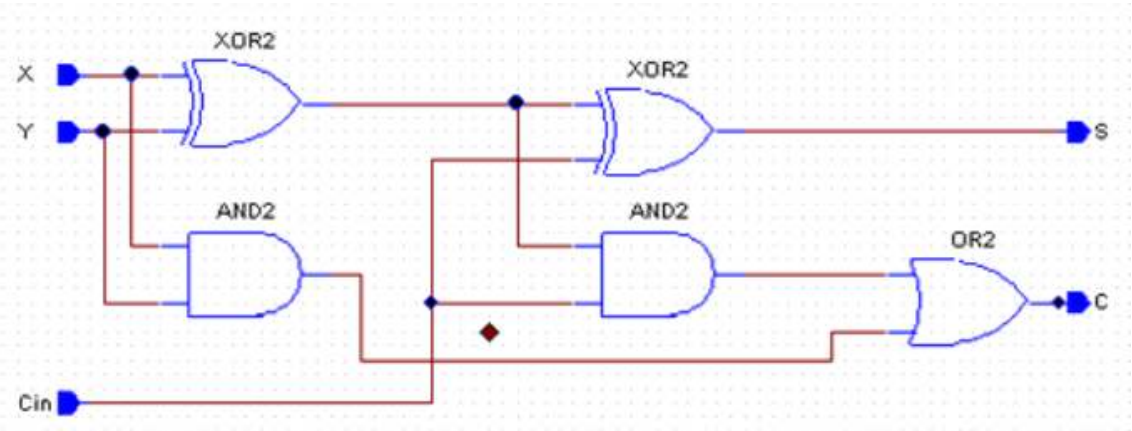
1) 전가산기

전가산기는 이진수 덧셈을 수행할 때 두 개의 한 자릿수 이진수 입력과 함께 하위 자리올림수를 포함하는 방식을 말한다.

전가산기는 반가산기와 달리 아래 자리수에서 발생한 Carry까지 포함해서 세 비트를 더하는 논리회로이다.

X	Y	Z	Sum	Carry
0	0	0	0	0
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	1

전가산기의 논리 회로는 회로를 XOR형식으로 바꿔서 더 간단하게 아래와 같은 논리회로로 나타낼 수 있다.



위 논리회로도는 반가산기 회로가 2개 겹쳐졌다는 것을 알 수 있다.

예시) 0011(3) + 1010(10)

1과 0을 더했을 때는 sum은 1, carry는 0이 된다.

따라서 다음 자리는 1과 1과 0을 더하는 것이 되고, 이는 sum은 0, carry는 1이 된다.

다음 자리는 0과 0과 1을 더하는 것이며 이는 sum은 1, carry는 0이 된다.

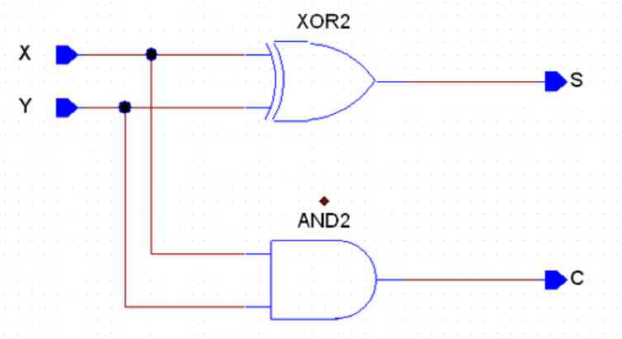
마지막 자리는 0과 1과 0을 더하는 것이며 이는 sum은 1, carry는 0이 된다.

따라서 결과는 1101(13)이 된다.

2) 반가산기

반 가산기는 2진수인 X,Y의 논리 변수를 더해서 합과 Carry를 나타내기 위한 조합 논리 회로를 말한다.

X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



반 가산기의 논리회로는 순수한 AND,OR,NOT 으로만 구성될 수도 있다. 이 결과를 XOR로 표현해서 나타내면 위와 같은 간단한 논리회로를 완성시킬 수 있다.

예시) 반 가산기는 1비트만 계산하고 carry 입력을 받지 않는다.

1+1 일 때, sum의 결과는 0으로 나오고, carry의 값은 1이 된다.

2. 전 감산기 및 반 감산기에 대해 조사하시오. (예시 포함)

감산기는 가산기에서 조금 변형된 형태로 2진수의 한자리 수 두 개의 비트의 차를 산출하는 회로이다.

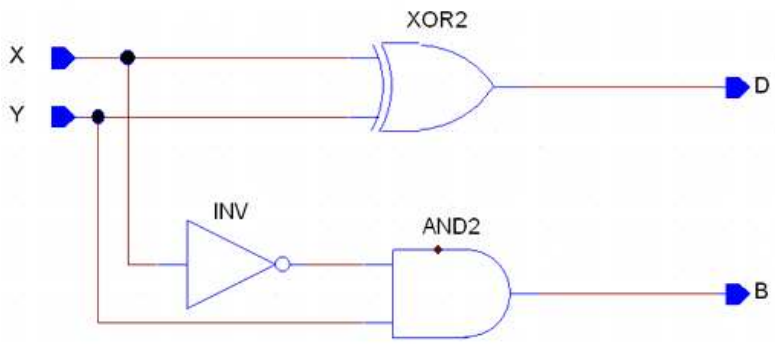
감산기의 종류는 가산기와 마찬가지로 전감산기와 반감산기가 있다.

1) 반감산기 (Half Subtracter)

X	Y	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

예시) 반 감산기는 1비트만 계산한다.

0-1 일 때, Difference 값은 1이 되고, Borrow 값은 1이 된다.

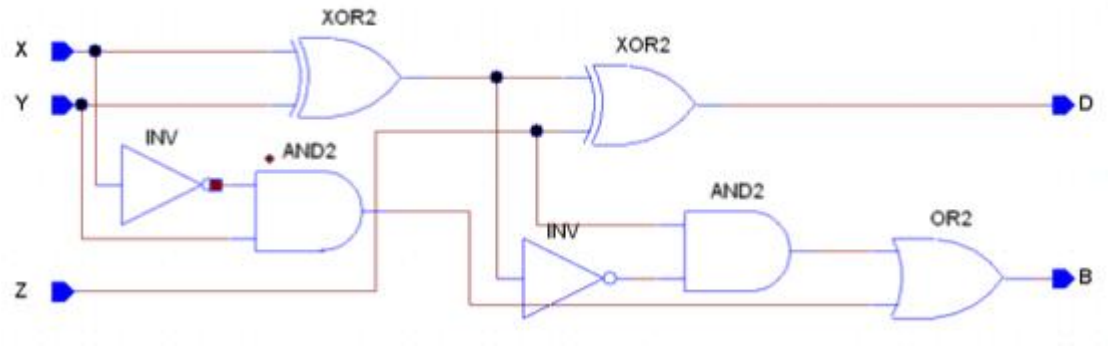


반감산기의 논리회로는 위와 같이 그려질 수 있다. 이는 반가산기 논리회로와 비교해보면 반가산기에 NOT 게이트가 추가된 것을 알 수 있다.

2) 전 감산기(Full Subtracter)

전 감산기는 입력 변수 3자리의 뺄셈에서 차이와 빌려오는 수를 구한다. 이는 반 감산기와는 달리 윗자리로부터 빌려온 값을 포함해서 세 비트의 뺄셈을 수행할 수 있다는 특징이 있다.

X	Y	Z	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



전감산기의 논리회로도는 위와 같다. 이 때 전가산기와 마찬가지로 전감산기의 논리회로도는 반감산기가 2개 그려진 것임을 확인할 수 있다.

예시) 1101(13) - 0010(2)

맨 끝자리의 경우 1-0 으로 값은 1이 되고 Borrow는 0이 된다.

두 번째의 경우 0-1로 값은 1이 되고 Borrow는 1이 된다.

세 번째의 경우 1-0으로 값은 0이 되고 Borrow는 0이 된다.

마지막의 경우 1-0으로 값은 1이 된다. 따라서 값은 1011(11)이 된다.

3. BCD 가산기에 대해 조사하시오.

BCD 가산기란 두 개의 BCD 수를 더해서 BCD로 결과를 출력하는 회로이다. 이진 가산기를 이용했을 때 합의 결과가 9이하일 때는 그대로 사용하고, 9보다 크면 보정을 해서 사용한다.

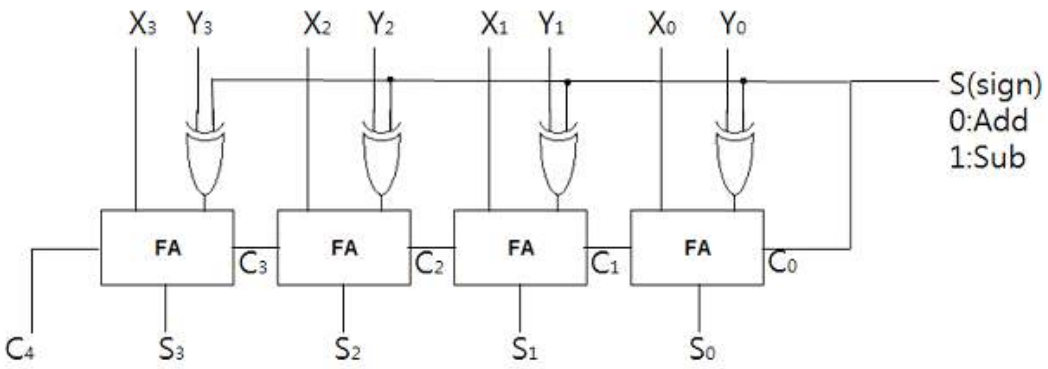
이진 병렬 가산기					BCD 가산기					십진수
C	S4	S3	S2	S1	X	Z4	Z3	Z2	Z1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6

0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	1	0	1	0	1	1	1	17
1	0	0	1	1	1	1	0	0	1	19

S는 이진 가산기의 출력에서 각 자릿수의 합을 의미하고 C는 carry를 의미한다. Z는 BCD에서 각 자릿수를 의미한다. 진리표에서도 9까지는 같은 이진가산기와 같은 값으로 나오고 이후부터 다른 값으로 변형된다는 것을 알 수 있다.

4. 병렬 가감산기에 대해 조사하시오.

전가산기를 병렬로 여러개를 연결할 경우 2비트 이상인 가산기를 만들 수 있으며 이를 병렬가산기라고 한다. 이때 병렬가산기의 입력 값에 부호처리를 하고 XOR를 할 경우에는 부호가 1일때는 Y의 값이 반전되어 1의 보수가 입력되기 때문에 덧셈과 뺄셈을 모두 할 수 있는 병렬가감산기가 만들어진다. 즉, 뺄셈일 때는 XOR이 출력이 1의 보수가 되어서 나가게된다.



5. Carry Look-Ahead Adder을 Ripple Carry Adder와 비교하여 설명하시오.

Carry Look_Ahead Adder는 디지털 논리에서 사용되는 가산기의 한 종류이다. Carry Look-Ahead Adder는 각 자리에서 자리올림 연산을 수행한다.
Ripple Carry Adder에서 가산기의 각 비트는 이전 비트로부터 자리올림수 출력을 기다려야 실행될 수 있는 반면에 Carry Look-Ahead Adder는 기다리지 않고 한 번에 계산된다.
이로 인해서 Carry Look-Ahead Adder는 지연이 적게 일어날 수 있다.
또한, Ripple Carry Adder는 많은 양의 논리 게이트가 필요한 반면에 Carry Look-Ahead

Adder는 필요한 논리 게이트의 수가 일정하기 때문에 더 빠르게 수행될 수 있다는 장점이 있다. 또한 입력이 많은 논리 게이트는 느려질 수 있다. 이로 인해서 Carry Look-Ahead Adder는 시간적인 측면에서 많은 이점을 가진다.

Carry Look-Ahead Adder는 A,B가 입력으로 들어왔을 때 연산을 수행한다.

$G(A,B) = A \cdot B$
$P(A,B) = A \oplus B$

이를 통해서 G와 P를 구한다. 이때 G는 자리올림수 생성이다. P는 자리올림수 전파로 추가적인 Carry가 생길경우를 알린다. G와 P를 이용해서 가산기의 Sum, Carry를 구할 수 있다. 이를 통해서 기존 연산과 상관없이 Carry가 처리되기 때문에 시간적으로 큰 장점이 있다.

6. 기타이론

맨체스터 자리올림수 회로

맨체스터 자리올림수 회로는 트렌지스터의 수를 줄이기 위해서 Carry Look-Ahaed Adder를 변형한 것이다. 맨체스터 자리올림수 회로는 최상위 자리올림수 값을 계산하는 게이트에서 값을 꺼내어 중간 자리올림수를 생성한다. 일반적으로 4비트를 초과해서 만들지 않으며, 트래지스터의 저항이 다른 회로보다 더 빠르게 증가된다는 단점이 있다.