

실험 Verilog-5: 예비보고서

전공: 컴퓨터공학

학년: 2

학번: 20171645

이름 박찬우

1. 목적

드모르간의 정리와 논리회로의 간소화 등의 이론에 대해 이해한다.

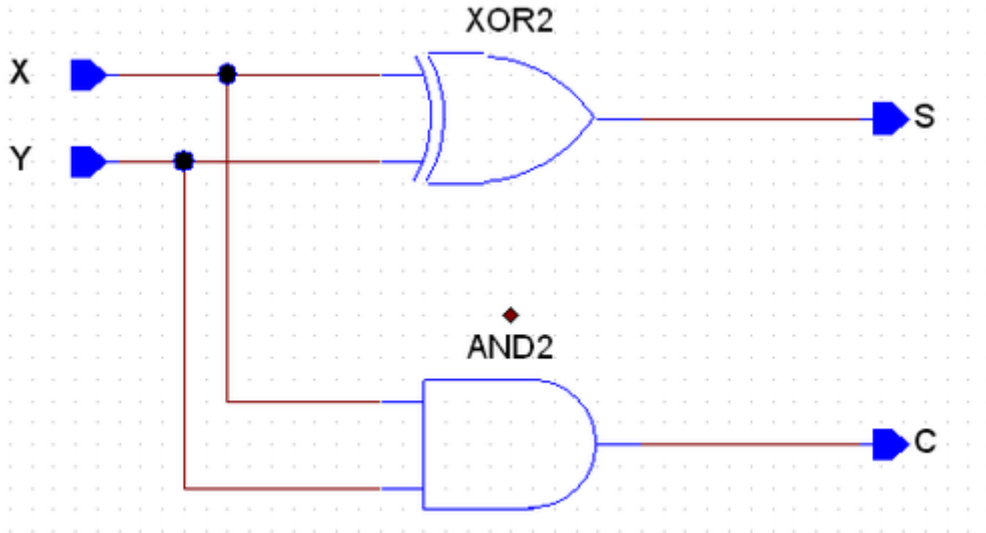
2. 요구 사항

진 가산기 및 반 가산기에 대해 조사하시오. (예시 포함)

반가산기는 한 비트의 두 수를 덧셈한 합과 carry를 구하는 조합논리회로로, 그 결과는 다음과 같은 진리표를 따른다.

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

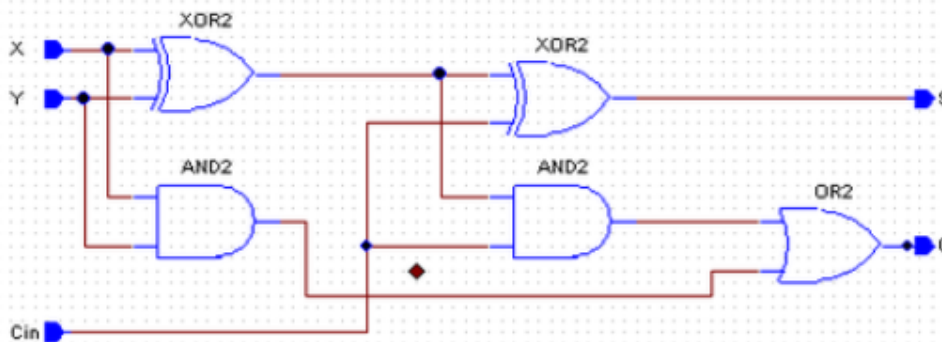
이를 논리회로로 구현하면 다음과 같다.



컴퓨터 내부에서 여러 비트의 두 수를 더하면 두 비트에서 생긴 carry는 더 높은 자리의 두 비트의 덧셈에 추가된다. 이때, 아래 자릿수에서 발생한 carry까지 포함하여 세 비트를 더하는 논리회로를 전가산기라고 한다.

즉, 전가산기는 carry를 포함한 3개의 입력을 받는다.

전가산기를 구현한 회로는 다음과 같다.



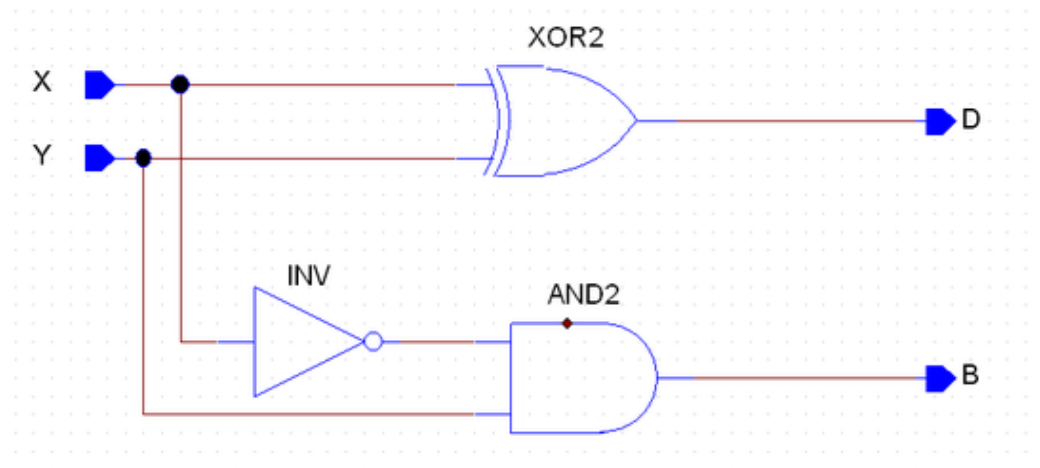
위 그림을 잘 보면, 전가산기에는 두개의 반가산기가 포함되어 있는 걸 확인할 수 있다.

전 감산기 및 반 감산기에 대해 조사하시오.

반감산기는 한 비트의 두 수를 뺄셈한 차와 빌려오는 값 Borrow를 구하는 조합논리회로로, 그 결과는 다음과 같은 진리표를 따른다.

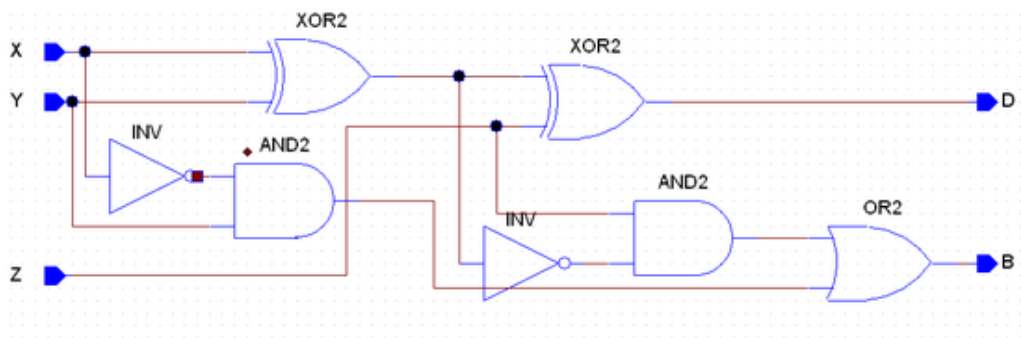
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

이를 논리회로로 구현하면 다음과 같다.



이는 반가산기의 회로에 NOT gate만 추가한것과 같음을 확인할 수 있다.

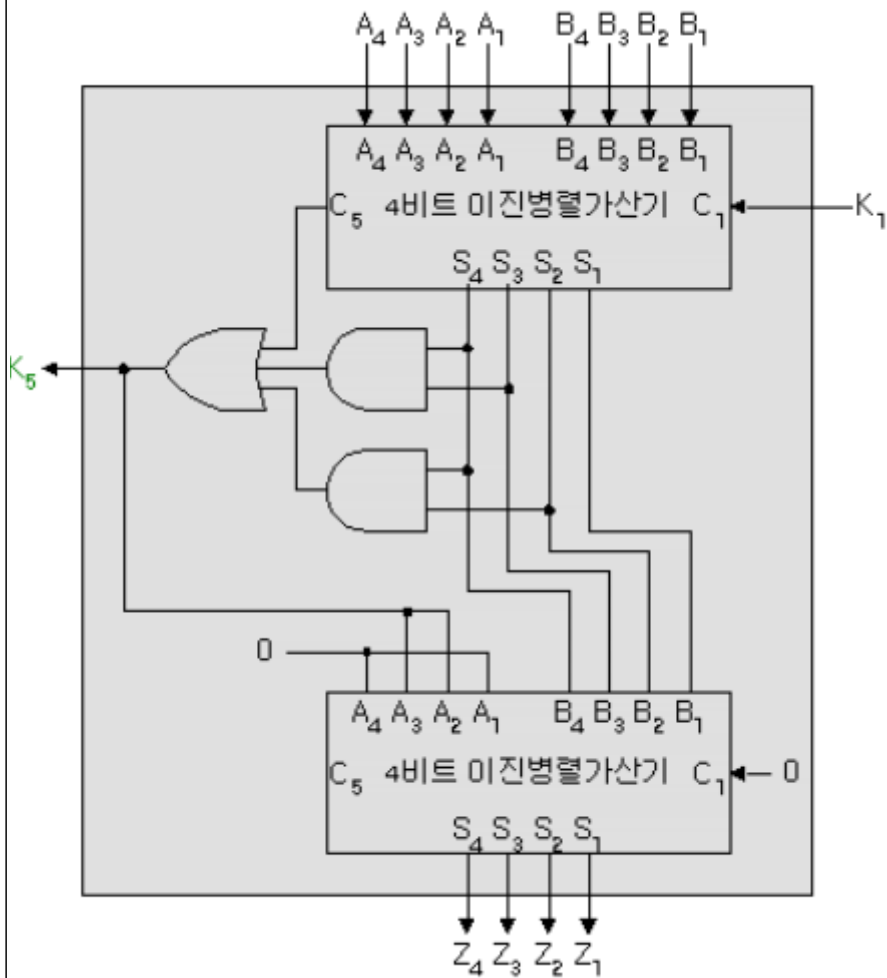
전감산기는 입력 변수 3개의 뺄셈에서 차와 Borrow를 구하는 조합논리회로로, 다음과 같은 논리회로를 따른다.



전가산기와 마찬가지로 두 개의 전감산기를 갖는다는 걸 확인할 수 있다.

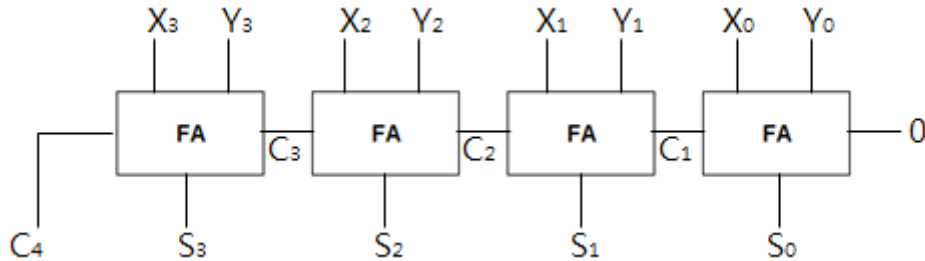
BCD 가산기에 대해 조사하시오.

BCD 가산기란 10진수의 덧셈을 수행할 수 있는 조합논리회로이다. 10진수의 계산을 하기 위해서는 0~9까지의 수 표현을 해야 하기 때문에 4bit를 사용하며, 0000부터 1001까지의 코드만 사용한다. 따라서, BCD 가산기는 carry를 포함해 입력 9개, 출력 5개가 된다. 이때, 입력값 0~9 두개와 carry를 포함한 19까지의 출력값을 이진수로 변환하면 00000 ~ 10011인데, 이러한 값을 4비트 이진병렬가산기를 통해 BCD 코드로 변환하여 BCD 가산기를 구현할 수 있다.
이러한 점을 토대로 구현한 회로는 아래와 같다.

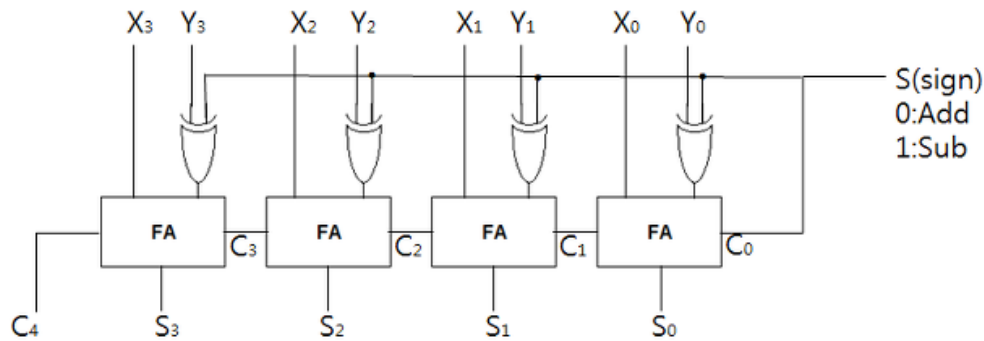


병렬 가감산기에 대해 조사하시오.

전자산기 여러개를 병렬로 연결하면 2bit 이상의 가산기를 만들 수 있는데, 이를 병렬가산기라고 한다. 병렬가산기는 다음과 같은 구조를 이룬다.



이러한 병렬가산기에 부호를 나타내는 Sign과 입력값 Y를 XOR gate를 통해 뺄셈을 구현할 수 있다. 즉, 부호인 S가 1이면 Y값과 XOR 하여 1의 보수가 되고, Carry값 역시 1로 되어 결과적으로 Y의 2의 보수가 되는 방식이다. 이를 구현한 회로는 아래와 같다.



이를 병렬가감산기 라고 한다.

Carry Look-Ahead Adder을 Ripple Carry Adder와 비교하여 설명하시오.

Ripple Carry Adder은 가산기의 각 bit는 아래 bit로부터 Carry가 출력되는 걸 기다려야 하는데, Carry Look-Ahead Adder는 모든 Carry 출력을 한번에 계산하기 때문에, 보다 적은 지연시간으로 계산을 완료할 수 있다. 또한, Ripple Carry Adder보다 적은 수의 논리게이트를 활용하기 때문에 더 빠르다. Carry Look-Ahead Adder는 가산기의 비트 수가 n이라면 $O(n)$ 개의 논리게이트가 필요한 반면, Ripple Carry Adder는 $O(n^2)$ 이상의 논리 게이트가 필요하다. 따라서 큰 가산기에서는 Ripple Carry Adder와 Carry Look-Ahead Adder를 혼합해서 사용하는 경우가 잦다.

Carry-Look Ahead Adder의 작동원리는 아래와 같다.

두개의 비트 A,B의 입력이 들어오면, 두 비트를 AND 및 XOR 한 결과를 각각 G, P라고 한다. 이들을 이용해 다음과 같은 수식을 활용한다.

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + (P_i \cdot C_i)$$

높은 자리의 Carry를 계산하는데 기존의 연산을 기다리지 않고 빠르게 처리할 수 있으므로 Ripple Carry Adder보다 빠른 속도의 계산이 가능하다.

기타이론

맨체스터 자리올림수 회로는 트랜지스터의 수를 줄이기 위한 Carry Look-Ahead Adder의 변종으로, 최상위 Carry 값을 계산하는 게이트에서 노드를 꺼내 중간 Carry를 생성한다. 이러한 회로의 주요한 단점은 용량 부하와 트랜지스터 저항 때문에 일반적인 Carry Look-Ahead Adder보다 전달 지연이 더 빠르게 증가한다는 점이다. 따라서, 맨체스터 자리올림수 회로는 일반적으로 4bit를 초과하지 않는다. 이러한 회로를 사용하는 주요 예시는 CMOS (Complementary metal-oxide-semiconductor) 등이 있다.