

실험 4. 이진수 연산

2022. 04. 28

20210054 정하우

1. 개요

이진수 덧셈에 사용하는 반가산기와 전가산기의 기능을 이해하고 회로를 구성한 후, 이를 사용해 덧셈, 뺄셈, 곱셈 회로를 구성한다. 자세한 학습 목표는 다음과 같다.

- 반가산기와 전가산기 구현
- 5-bit 리플 가산기/감산기 구현
- 5x3 이진 곱셈기 구현
- 테스트 벤치를 이용한 회로 검증

2. 이론적 배경

1) 반가산기 (Half adder)

1-bit 이진수 2개를 입력받아 합과 Carry out을 출력한다. 보통 1-bit 이상의 다중 비트 연산을 할 때는 사용되기 어렵고, 전가산기(Full adder)을 보편적으로 사용한다.

2) 전가산기 (Full adder)

1-bit 이진수 2개와 이전 가산기의 Carry in 총 3개를 입력받아 합과 Carry out을 출력한다. 두 개의 반가산기를 사용해 구현할 수 있다. 보편적으로 많이 사용된다.

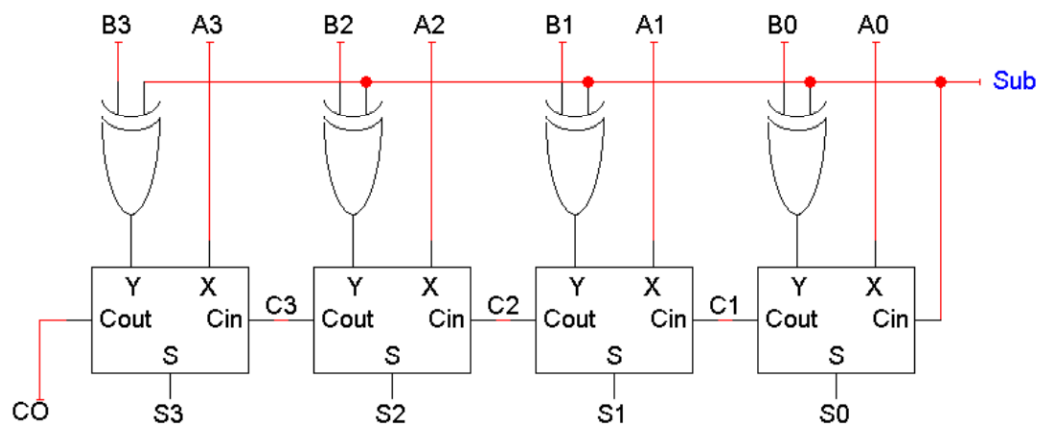
3) N-bit 리플 가산기 / 감산기 (N-bit Ripple Adder / Subtractor)

N-bit 가산기는 N-bit 이진수 두 개를 더하여 N-bit 덧셈 결과와 Carry out을 출력한다. N-bit 가산기는 다양한 방법으로 구현할 수 있는데, 그중 하나로 N-bit 리플 가산기가 있다.

N-bit 리플 가산기는 N-bit 의 각 자릿수를 전가산기를 통해 계산한다. 이때 k 번째 자릿수를 계산하는 전가산기의 Carry out이 k+1번째 자릿수를 담당하는 전가산기의 Carry in에 연결되어 가장 낮은 자릿수부터 가장 높은 자릿수까지 Carry가 순차적으로 계산된다.

한편, A-B 곱의 뺄셈은 $A+(-B)$ 곱의 덧셈으로 변환하여 계산한다. 보통의 경우 1의 자리수에서 계산될 때 Carry in은 0이다. 하지만 뺄셈을 하는 경우에는 빼는 수를 1's complement로 전환하기 위해 NOT를 적용시킨다. 최종적으로 계산할 때는 2's complement 형식으로 전환시켜줘야 하므로 1의 자리수에서 carry in 값을 1로 받아야 한다. 이것을 그림을 나타내면 다음과 같다.

4) MxN 이진 곱셈기 (MxN Binary Multiplier)



MxN 이진 곱셈은 M-bit Multiplicand와 N-bit Multiplier의 각 자릿수의 부분 곱을 통해 얻은 이진수 N개를 합하여 계산할 수 있다. 가산기로 부분 곱을 모두 더하여 답을 구한다.

이때, 다음과 같이 각 부분 곱을 더하는 과정을 M-bit 가산기 N-1개로 구할 수 있다.

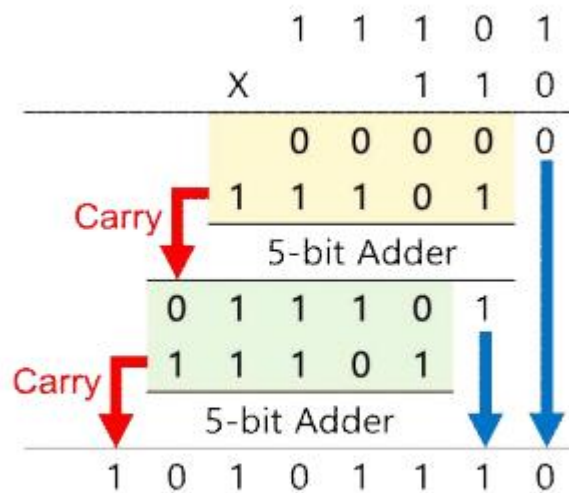
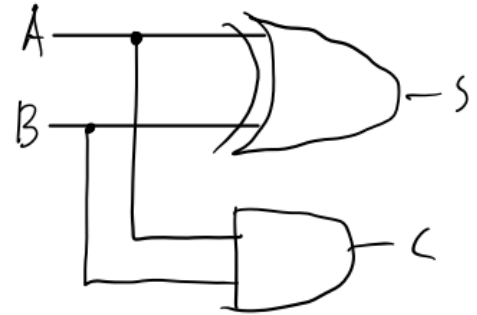


그림 2. 5x3 이진 곱셈 예시

3. 실험 준비

- 1) 반가산기의 진리표와 식을 구하고 회로를 그린다.

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

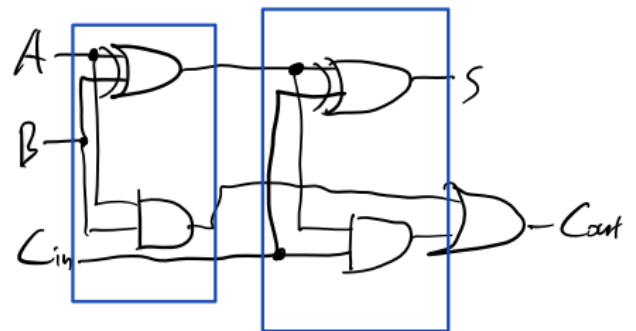


$$S = A \oplus B$$

$$C = AB$$

- 2) 전가산기의 진리표와 식을 구하고 반가산기를 사용해 전가산기 회로를 그린다.

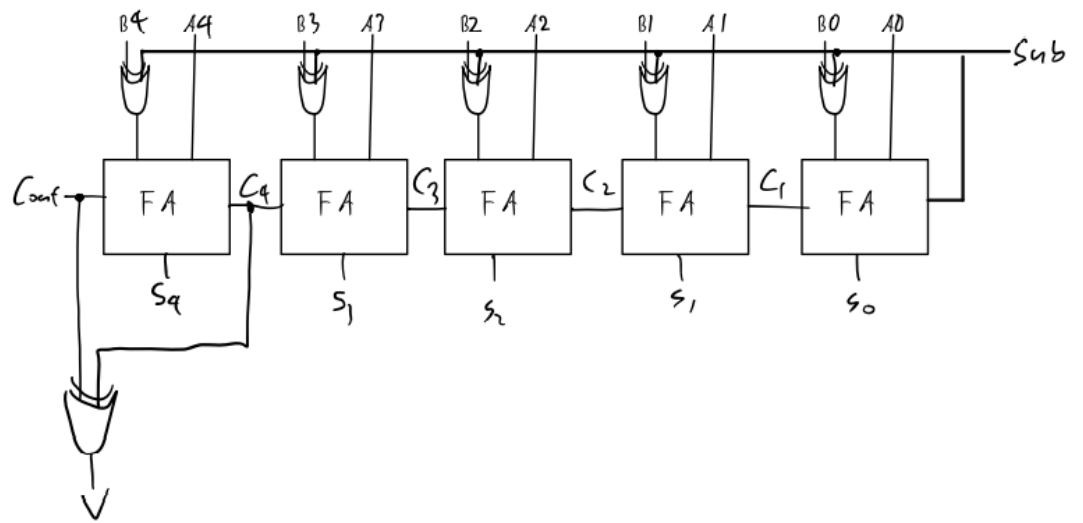
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$$S = A \oplus B \oplus C_{in}$$

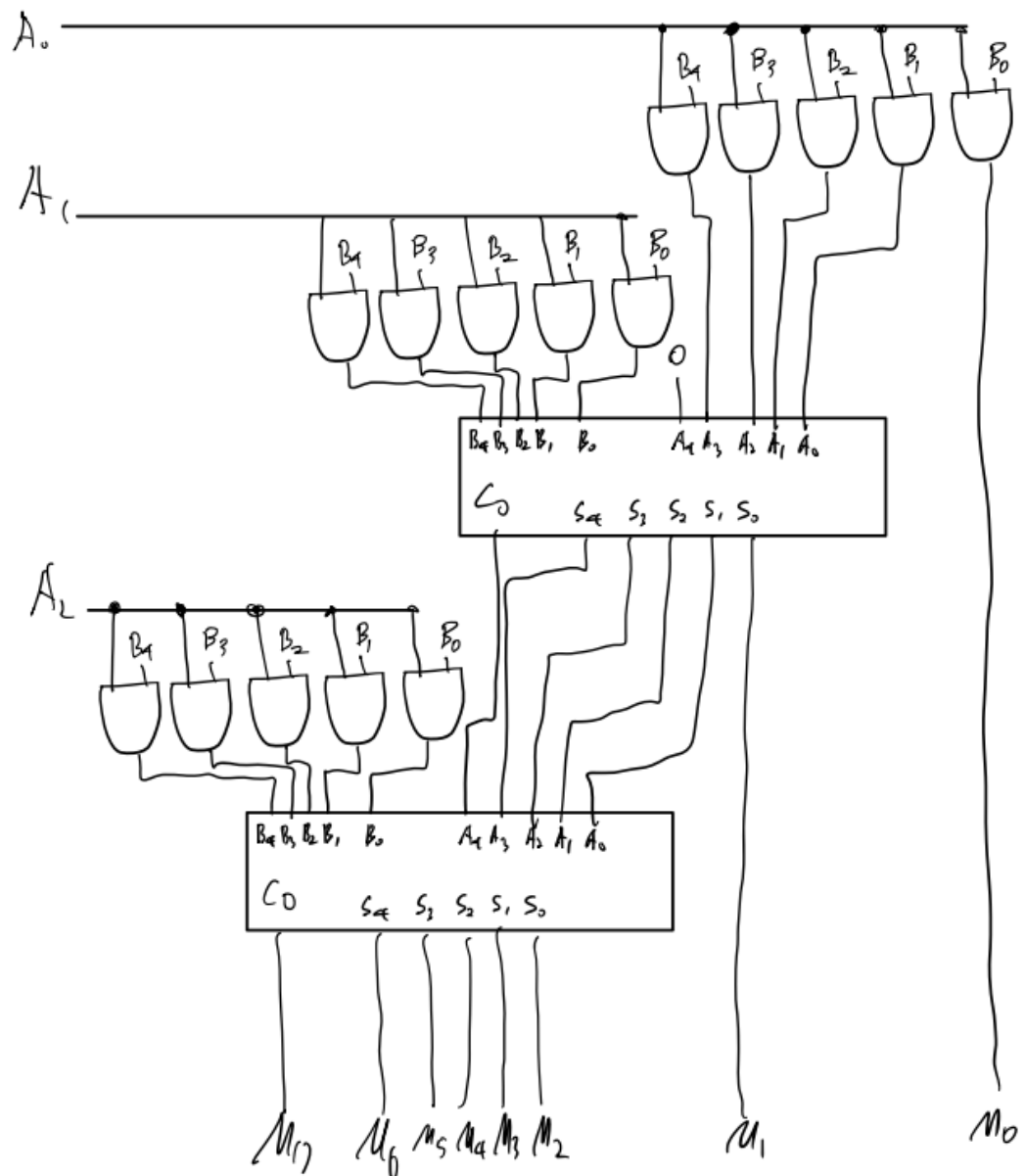
$$C_{out} = AB + (A \oplus B)C_{in}$$

- 3) 전가산기를 사용해 5비트 리플 가산기와 감산기 회로를 그린다. - 음수는 2의 보수로 표현한다.



Sub이 0이면 가산기, 1이면 감산기이다.

- 4) 5비트 리플 가산기를 사용해 5x3 이진 곱셈기 회로를 그린다.

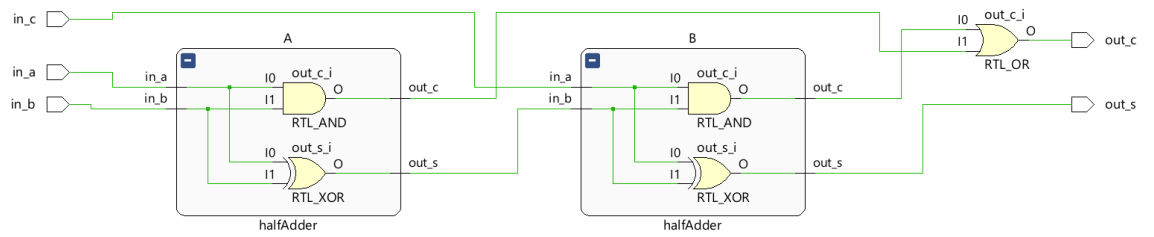


4. 결과

1) 반가산기, 전가산기 - lab4_1.v

ㄱ. 반가산기를 구현하고, 이를 사용해 전가산기를 구현한다.

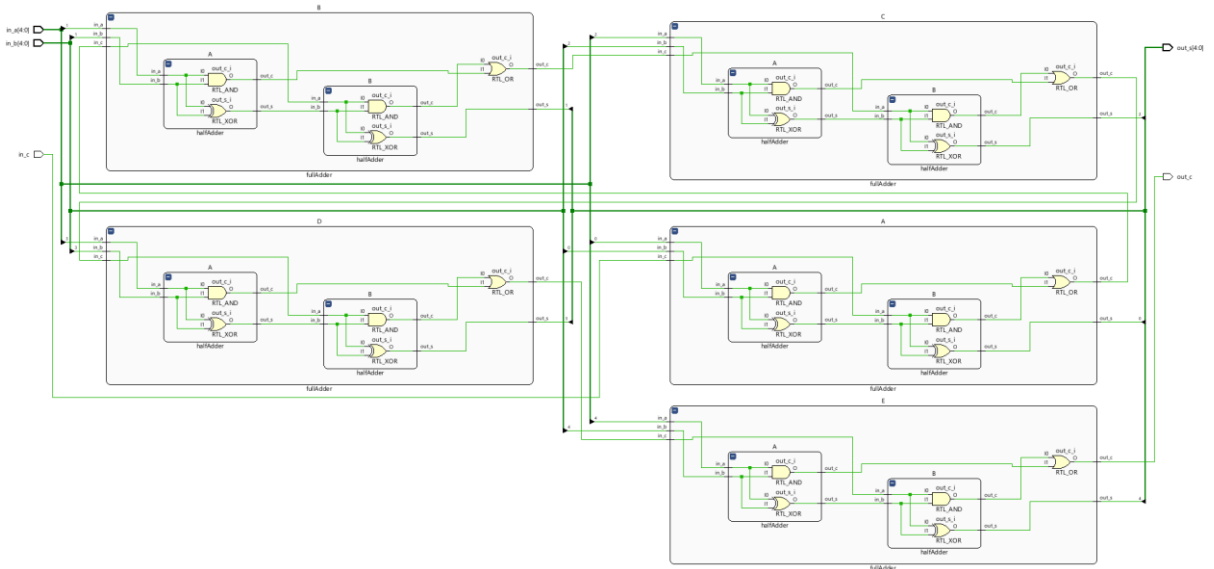
ㄴ. Schematic 기능으로 회로를 확인한다.



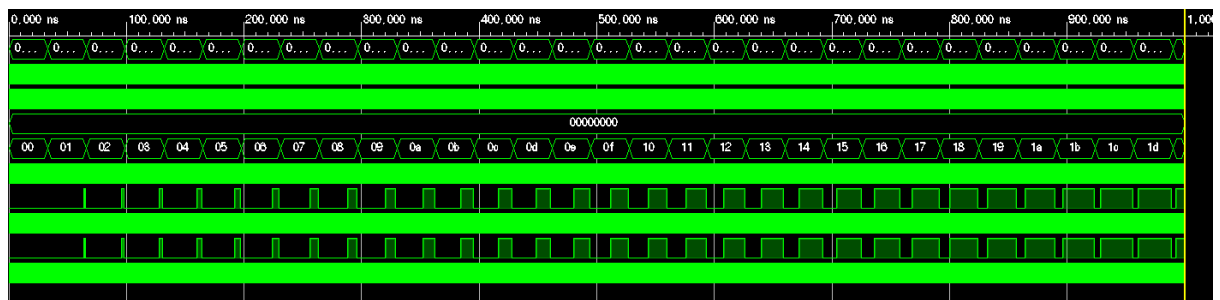
2) 5비트 리플 가산기 – lab4_2.v

ㄱ. 실험 1의 전가산기 모듈을 사용해 5비트 리플 가산기를 구현한다.

ㄴ. Schematic 기능으로 회로를 확인한다.



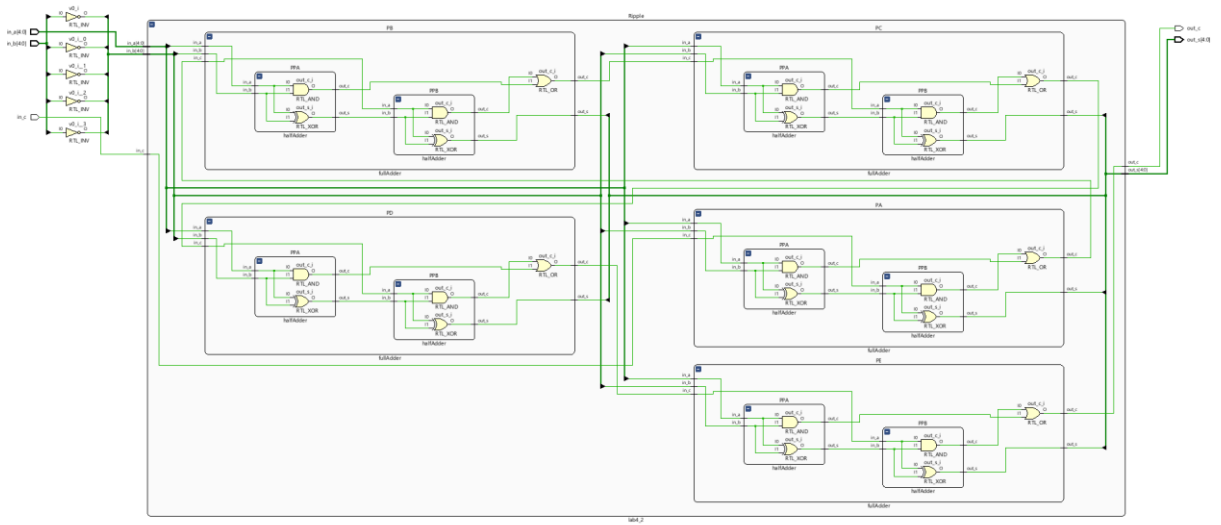
ㄷ. 주어진 테스트벤치로 시뮬레이션을 실행해 정상 작동을 확인한다.



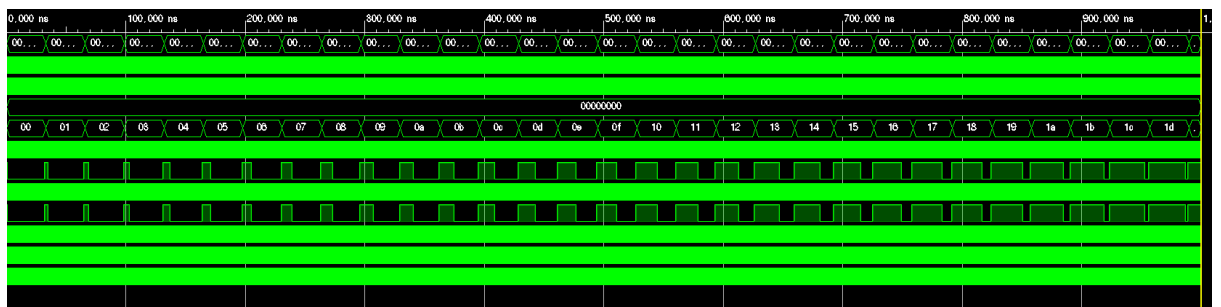
3) 5비트 리플 감산기 – lab4_3.v

ㄱ. 실험 2의 5비트 리플 가산기 모듈을 사용해 5비트 리플 감산기를 구현한다. - 음수는 2의 보수로 표현한다.

↳ Schematic 기능으로 회로를 확인한다.



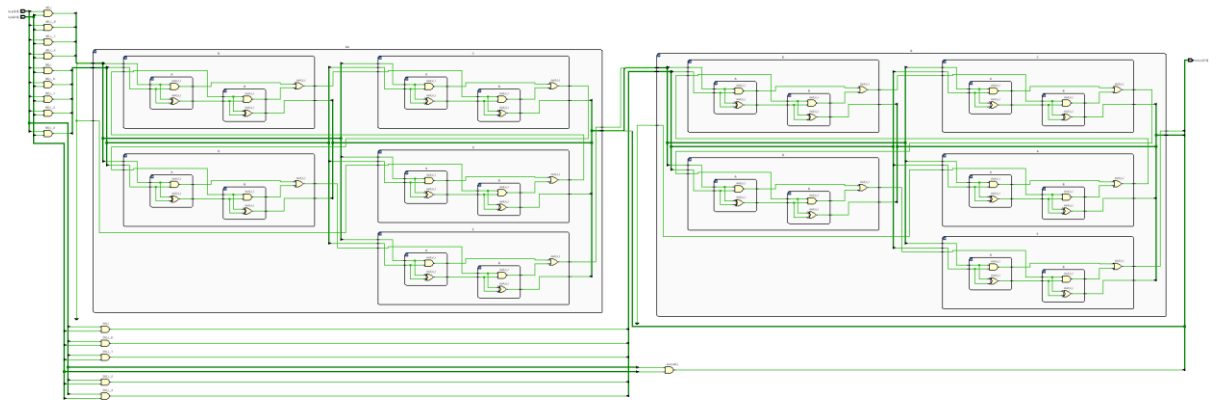
㉔. 주어진 테스트벤치로 시뮬레이션을 실행해 정상 작동을 확인한다.



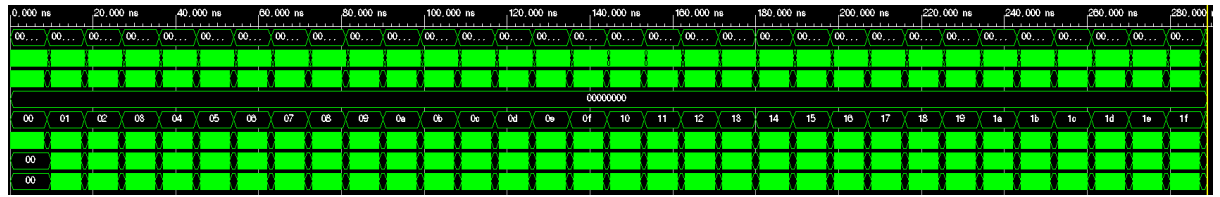
4) 5x3 이진 곱셈기 - lab4_4.v

7. 실험 2의 5비트 리플 가산기 모듈을 사용해 5x3 이진 곱셈기를 구현한다.

↳ Schematic 기능으로 회로를 확인한다.



ㄷ. 주어진 테스트벤치로 시뮬레이션을 실행해 정상 작동을 확인한다.



5. 논의 및 결론

이번 실험을 통해 리플 가산기와 감산기의 원리를 정확하게 이해할 수 있었다. 수업시간에 배운 adder 개념은 계산 알고리즘을 대략적으로 파악하는데 그쳤지만, 본 실험을 통해 곱셈기가 가산기와 감산기를 어떻게 이용하여 알고리즘을 설계하는지 완벽하게 파악할 수 있었다.