실험 4. 이진수 연산

20210661 오승준

1. 개요

이번 실험은 이진수 연산에 필요로 하는 여러 회로들을 직접 구현하는 것을 목표로 한다. 세부목표는 아래와 같다.

-반가산기, 전가산기 구현

-5-bit 리플 가산기/감산기 구현

-5x3 이진 곱셈기 구현 및 구현 회로 회로 검증

2. 이론적 배경

1)반가산기

반가산기는 1-bit 이진수 두 개를 입력받아 그 합과 Carry를 출력하는 가장 기초적인 형태의 가산기 회로이다.

2)전가산기

반가산기 2개와 or gate 1개를 통하여, 1-bit 이진수 두 개와 이전 가산기의 Carry in을 입력받아 합과 Carry Out을 출력한다.

3)N- bit 리플가산기/감산기

N-bit 리플 가산기는 N개의 전가산기를 순차적으로 이어 구현한 가산기이다. 각 전가산기는 각 bit의 한 자릿수 연산을 맡고, Carry out을 다음 가산기의 Carry in으로 보낸다.

가장 낮은 자리부터 순차적으로 연산이 수행되기 때문에, bit의 수가 커질수록 연산시간이 더욱 많이 걸린다.

감산기의 경우, 빼려는 수를 2의 보수를 취함으로써 가산기를 그대로 이용할 수 있다.

A-B = A+(-B)이기 때문이다.

4)MxN 이진 곱셈기

MxN 이진 곱셈은 M-bit Multiplicand와 N-bit Multiplier의 각 자릿수의 부분 곱을 통해 얻은 이진수 N개를 합하여 계산한다. 각 부분곱을 구한 후 가산기로 부분곱을 더하여 답을 구한다.

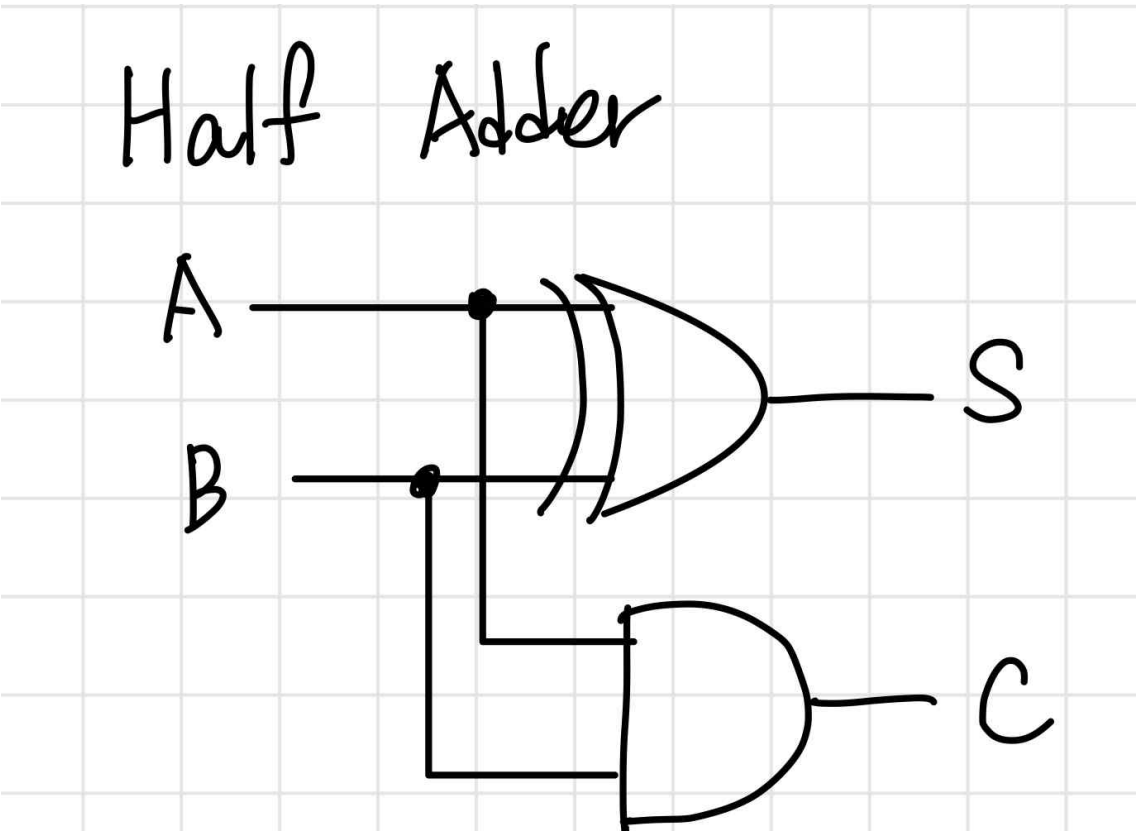
3. 실험준비

1) 반가산기의 진리표와 식을 구하고 회로를 그린다.

반가산기의 진리표는 아래와 같다.

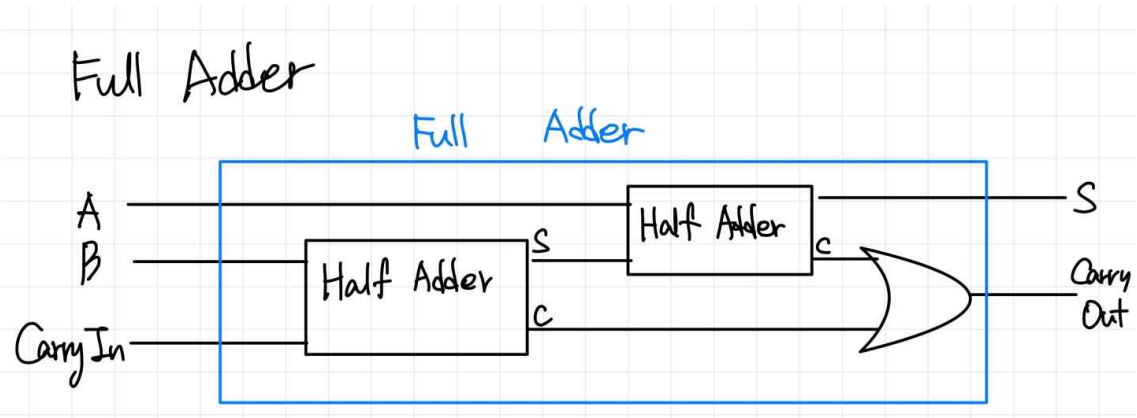
Input		Output	
a	b	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1

1	1	1	0

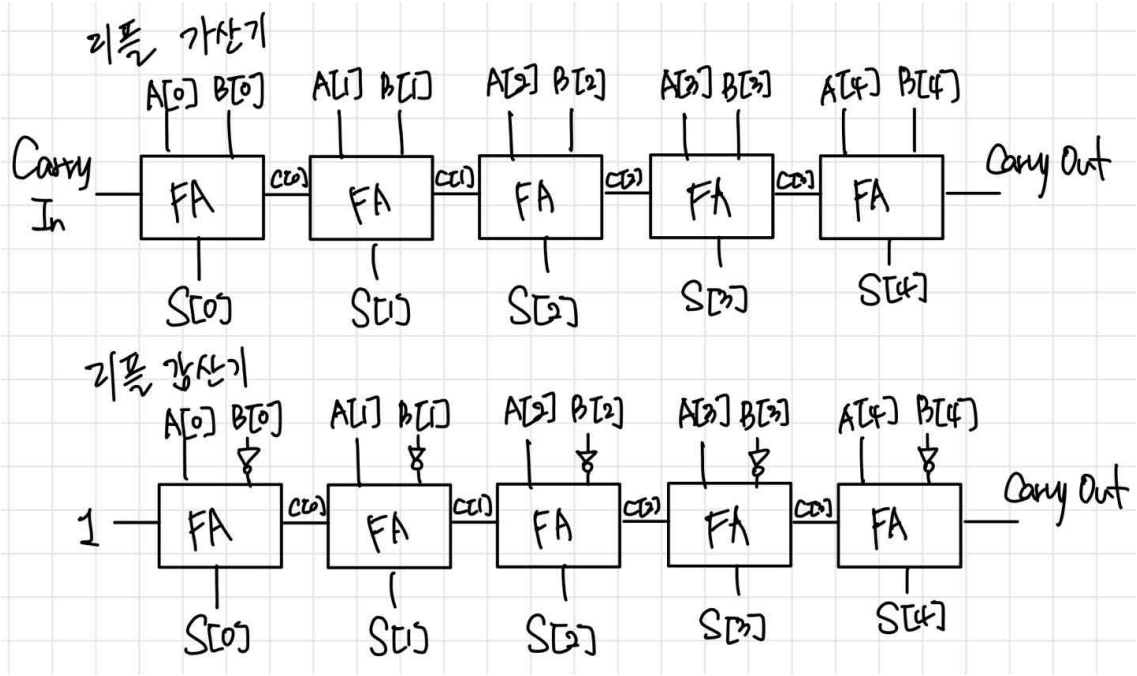


2) 전가산기의 진리표와 식을 구하고 반가산기를 사용해 전가산기 회로를 그린다.

Input			Output	
Carry in	a	b	Carry out	Sum
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

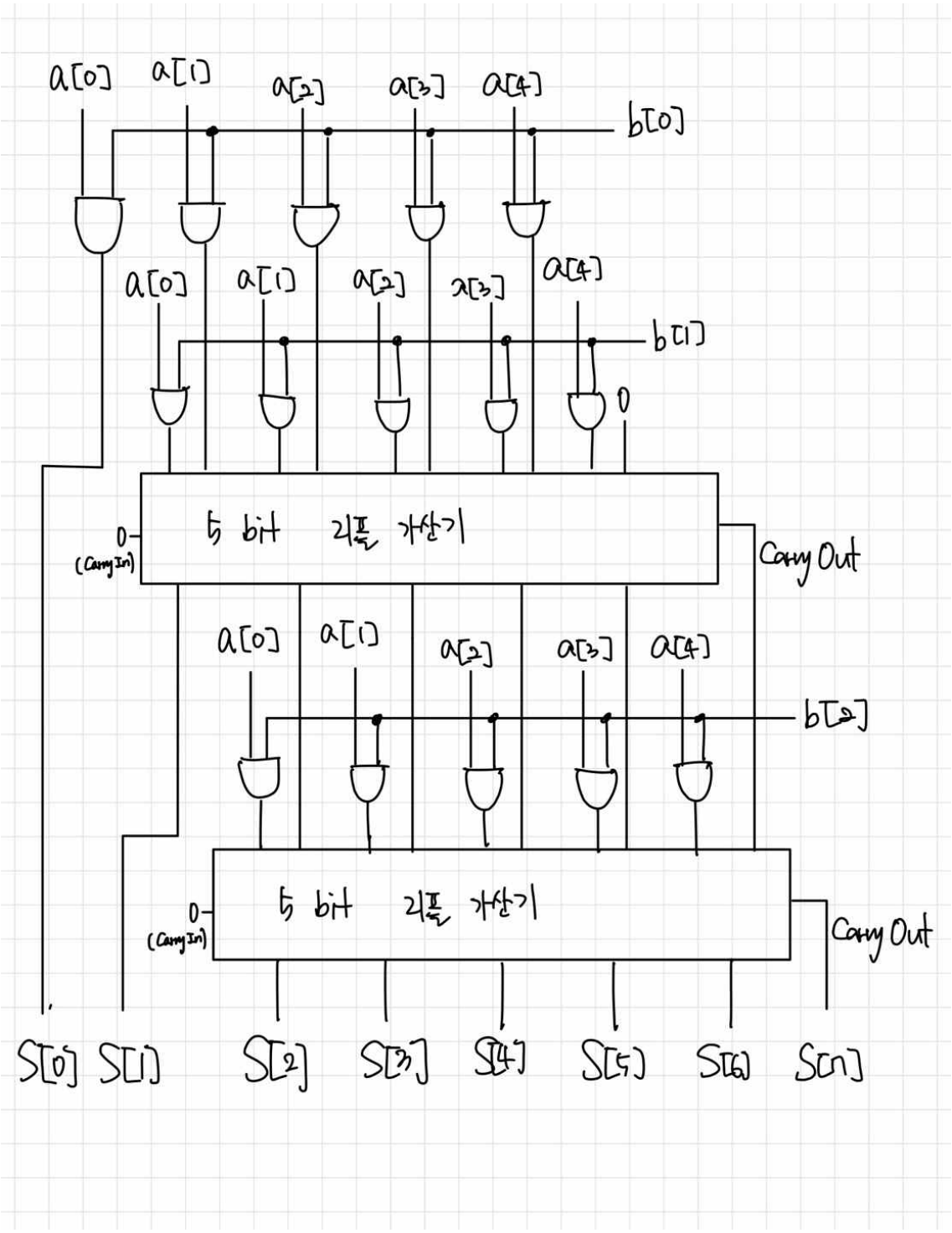


3) 전가산기를 사용해 5비트 리플 가산기와 감산기 회로를 그린다.



리플 감산기에서 B의 값을 전부 부정(NOT) 한 후, Carry In에 1을 넣어주면 2의 보수를 입력하는 것과 동일한 역할을 한다.

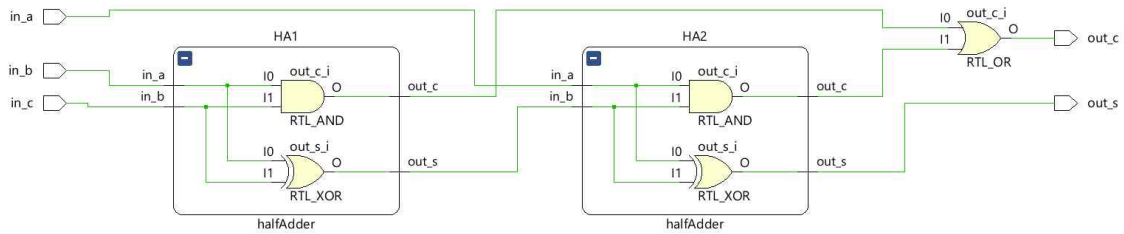
4) 5비트 리플가산기를 이용해 5x3 이진 곱셈기 회로를 그린다.



4. 실험

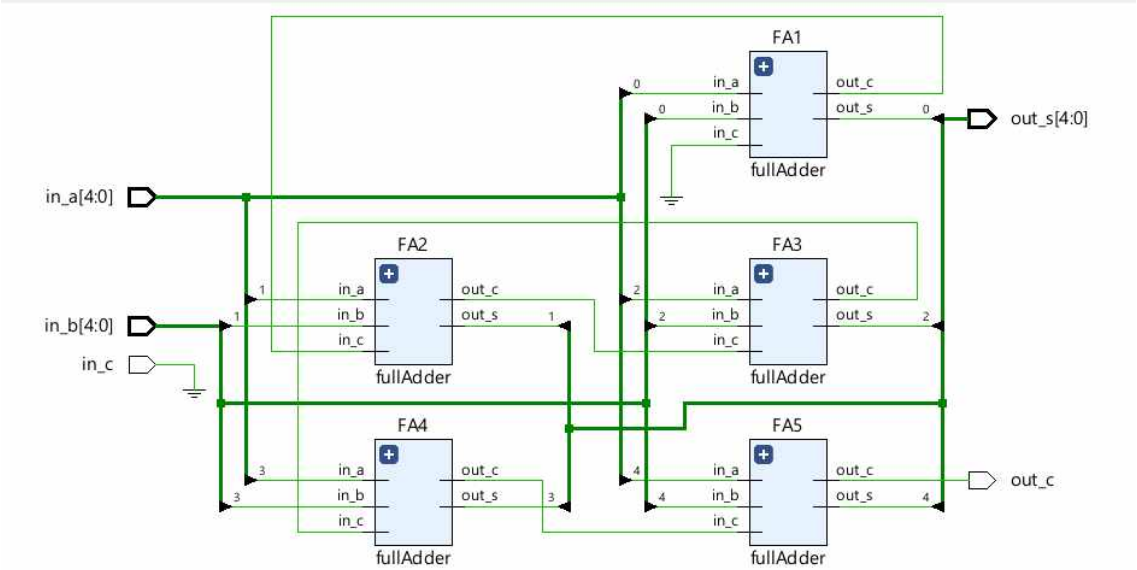
1)반가산기, 전가산기

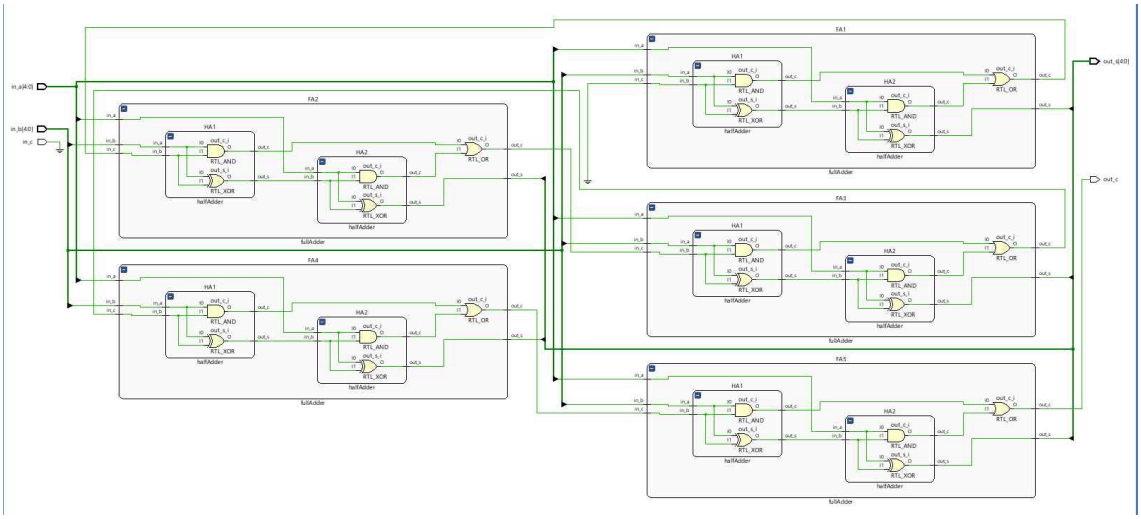
- 실험 결과 회로는 아래와 같다.



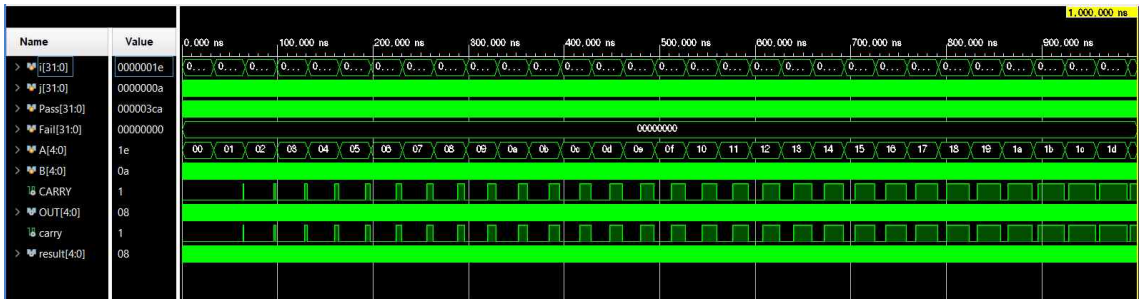
2)5비트 리플 가산기

- 실험 결과 회로는 아래와 같다.





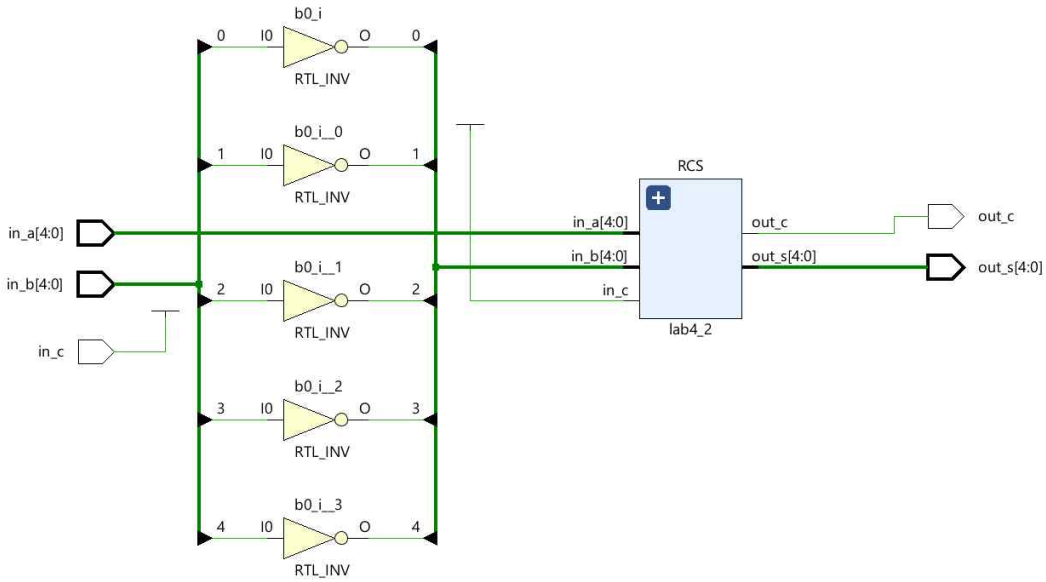
- 실험 결과 시뮬레이션 결과는 아래와 같다.

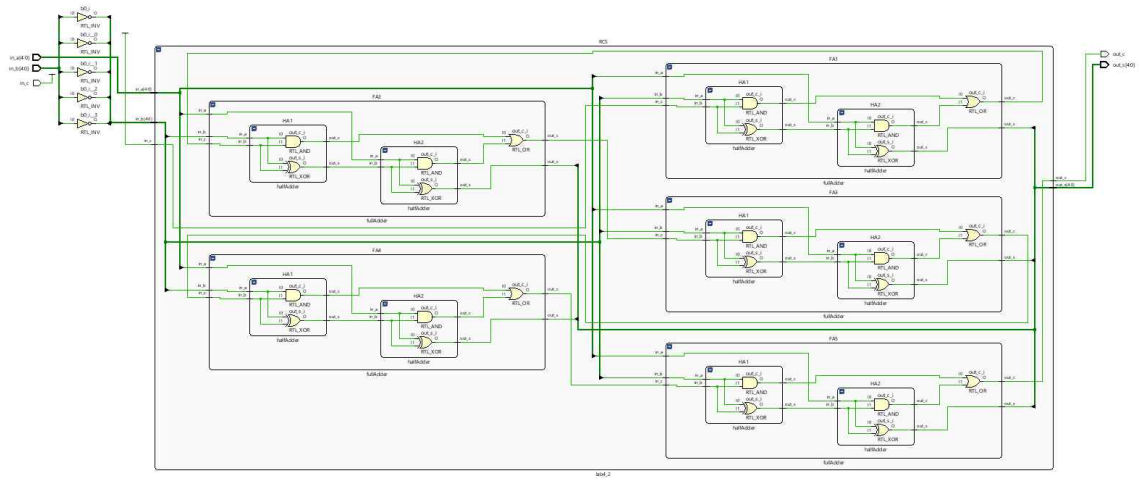


결과에서 오류가 없고, 실험 값이 이론값과 동일하므로 올바르게 회로가 짜임을 확인할 수 있다.

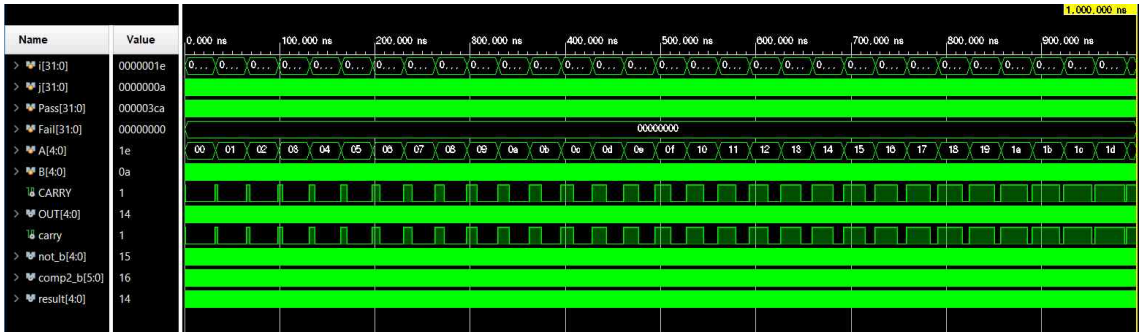
3)5비트 리플 감산기

- 실험 결과 회로는 아래와 같다.





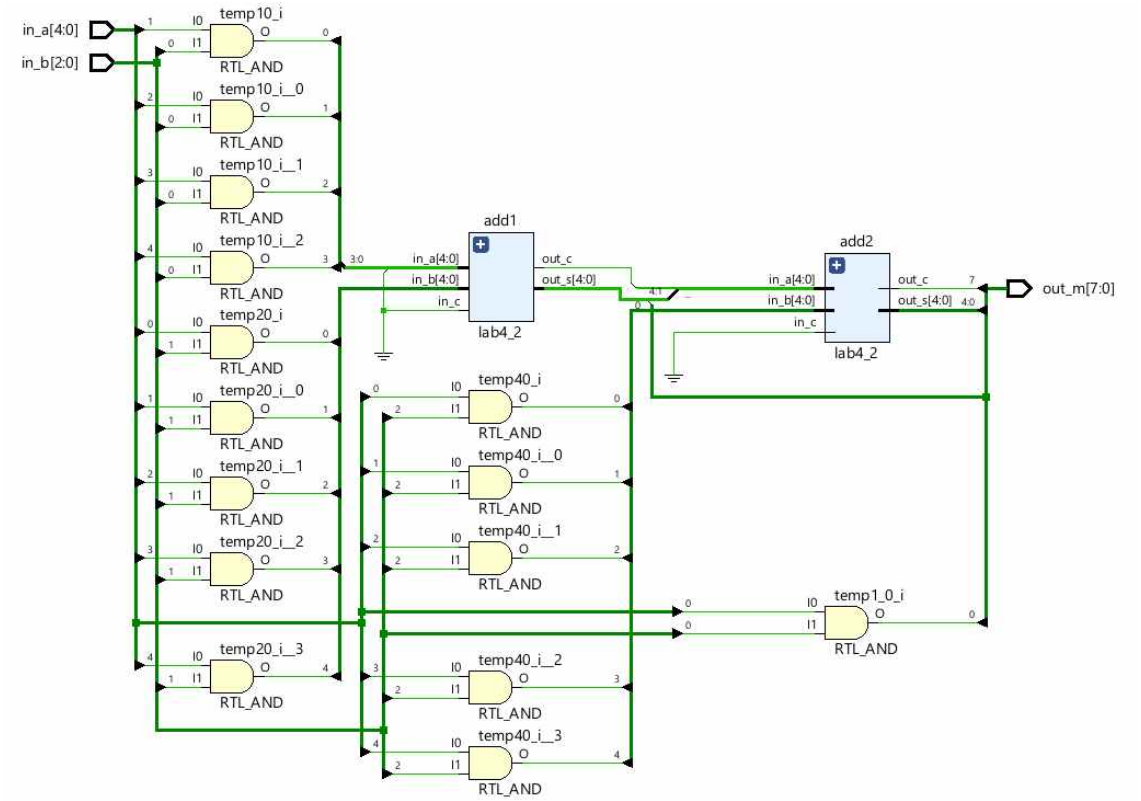
- 시뮬레이션 결과는 아래와 같다.

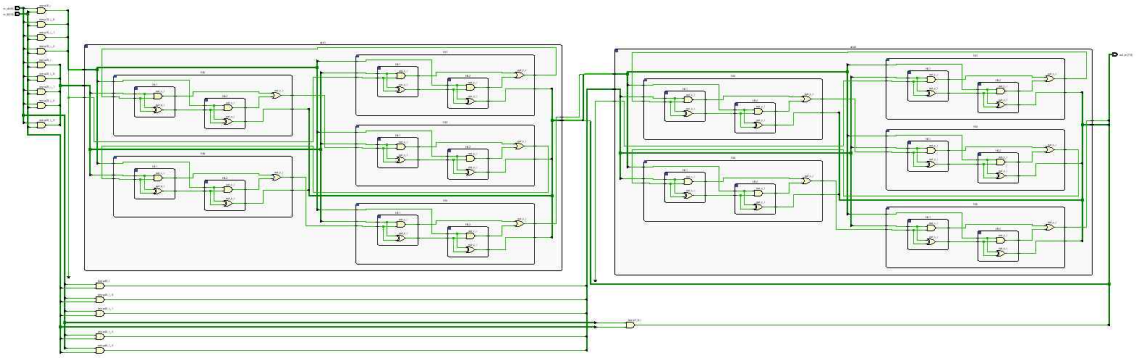


결과에서 오류가 없고, 실험 값이 이론값과 동일하므로 올바르게 회로가 짜임을 확인할 수 있다.

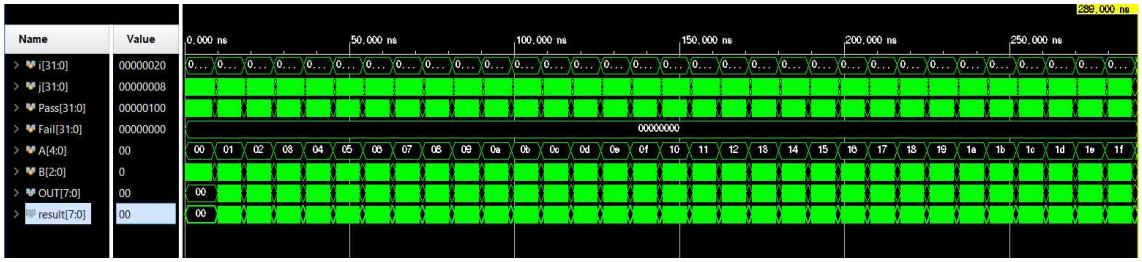
4) 5x3 이진 곱셈기

- 실험 결과 회로는 아래와 같다.





-시뮬레이션 결과, 아래와 같다.



결과에서 오류가 없고, 실험 값이 이론값과 동일하므로 올바르게 회로가 짜임을 확인할 수 있다.

5. 논의

- 이번 실험을 진행하며 이진 연산에 관한 여러 회로들을 직접 구성하고 시뮬레이션을 진행할 수 있었다. 이진 연산(덧셈, 뺄셈, 곱셈)을 구현하는데 있어 모두 덧셈을 위한 회로(반가산기, 전가산기)를 통해 구현한다는 것이 우리가 일상에서 써왔던 연산과 비슷하게 느껴졌다.

실은 일전에 반가산기와 전가산기에 대해 배워본 적이 있었으나, 이를 통해 연산을 어떻게 한다는 것인지 이해를 하지 못했는데, 리플 가산기의 회로를 직접 짜보며 전가산기를 직렬로 이음으로써 여러 비트의 연산을 할 수 있는 회로를 구성할 수 있음을 깨달았다.

이번 코드를 작성하면서는 모듈 작성의 중요성을 느꼈다. 반가산기 모듈, 전가산기 모듈, 리플 가산기 모듈을 각각 여러 곳에서 활용하는데, 이를 모듈을 만들어 두지 않고 연산을 했다면 회로 구성에 어려움을 겪었을 것이다. 모듈을 선언하는 것이 프로그래밍에서 함수 선언과 유사한 느낌을 받았다.