

실험 5. ALU와 JK 플립플롭

20210661 오승준

1. 개요

이번 실험은 컴퓨터의 기초가 되는 산술 논리 장치(Arithmetic Logic Unit: ALU)와, 플립플롭에 대해 이해하고, 플립플롭의 종류 중 하나인 정보를 저장할 수 있는 JK 플립플롭(Flip-flop)을 구현한다.

2. 이론적 배경

1) ALU (Arithmetic Logic Unit)

ALU는 입력에 대해 여러 산술(Arithmetic) 및 논리(Logic) 연산을 수행한다. 연산의 종류에 따라 산술 장치와 논리 장치 두 부분으로 나눌 수 있는데, 산술 장치는 사칙 연산 등을, 그리고 논리 장치는 Bitwise 논리 연산 등을 맡는다. 산술장치는 구현과정에서 Adder가 필요하고, 논리장치는 MUX가 필요하다.

2) 비동기 회로 / 동기 회로

비동기 회로란, 모든 조합 회로와 클록을 따르지 않는 순차 회로를 말한다.

동기 회로는 다른 회로와 같은 순간에 맞춰 작동하기 위해 클록 신호를 따르는 것을 말한다.

3) JK 래치 / JK 플립플롭

JK 래치는 SR 래치에 추가적인 회로를 더해 S와 R이 동시에 1인 상황에서도 정상적으로 작동하도록 수정한 래치이다. JK 래치에서 J와 K가 동시에 1일 경우 현재 상태에 상관없이 값을 반전시킨다.

래치가 입력이 바뀔 때 출력도 바로 바뀌는 비동기 회로이나, 플립플롭은 입력이 바뀌더라도 출력이 클록에 맞추어 반영되는 동기 회로이다. JK 플립플롭은 클록 신호를 추가로 받아 이에 맞추어 작동한다는 특징을 지닌다.

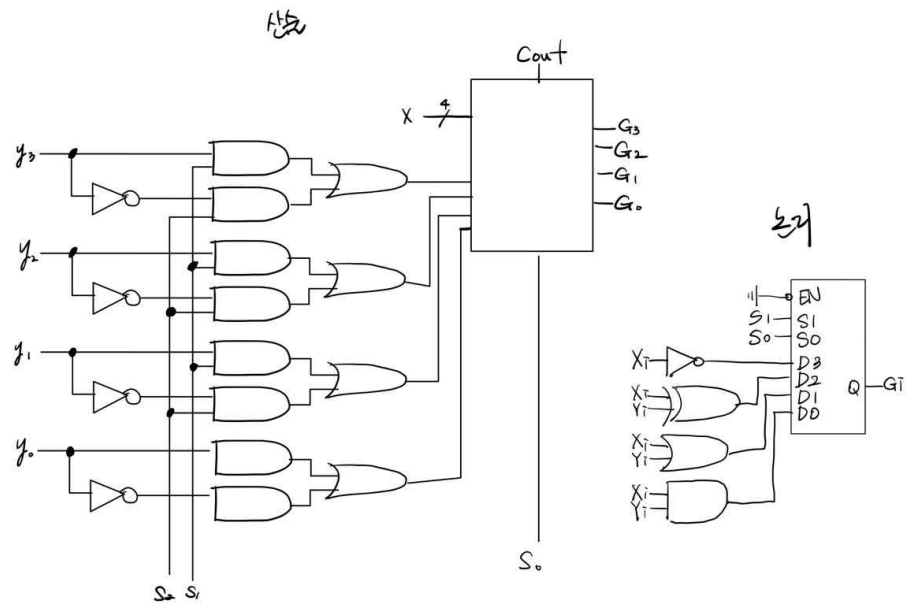
4) Master-slave JK 플립플롭

Master-slave JK 플립플롭은 SR 래치 두 개를 연결하여 만든 JK 플립플롭의 종류 중 하나이다. 이 플립플롭은 클록이 1인 동안 Master 래치를 활성화해 입력을 임시로 저장한 뒤 클록이 0이 되는 순간 Slave 래치로 전달한다. Master 래치가 활성화되어있는 동안 글리치로 잠깐 입력값이 생기면 이 값이 Master 래치에 저장되어있다가 다음 클록이 0이 되는 순간에 Slave 래치로 전파되는 문제가 있다. 이는 클록이 1인 동안 계속 입력을 받기 때문에 생기는 문제로 클록이 0에서 1로, 혹은 1에서 0으로 바뀌는 순간에만 입력을 받는 Edge-trigger 회로를 사용하여 해결할 수 있다.

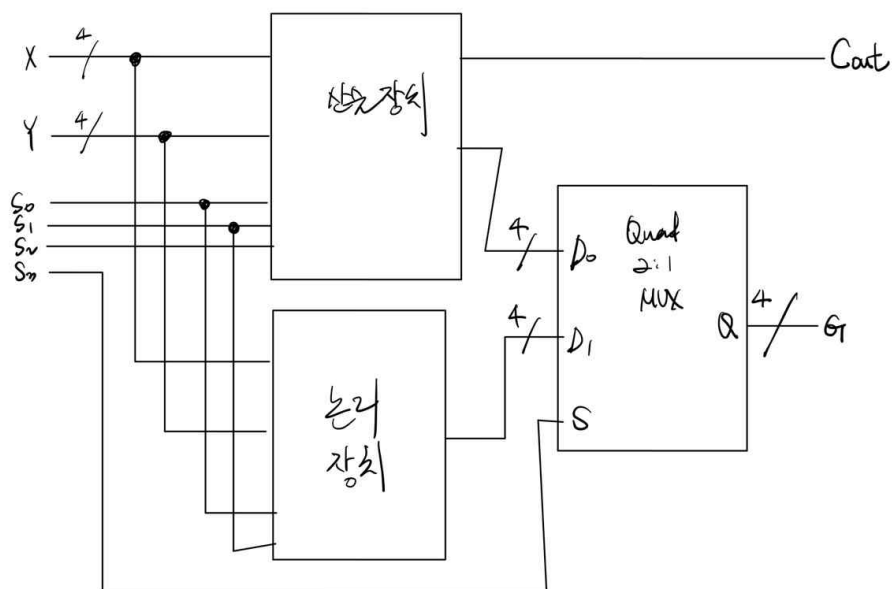
3. 실험준비

1) ALU

1. 산술장치와 논리장치의 회로도를 그린다.



2. 1에서 그린 모듈을 묶어 ALU의 회로도를 그린다.

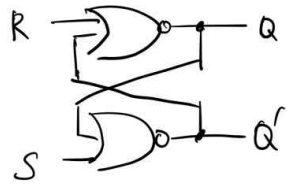


2. Master-Slave JK 플립플롭

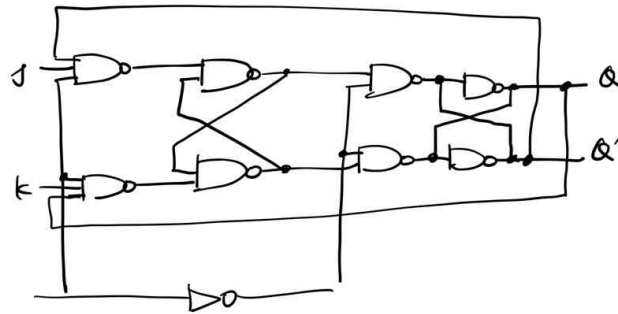
1. SR 래치의 회로도 그리기

2. SR 래치를 사용해 Negative reset Master-slave JK 플립플롭의 회로도 그리기

7. SR Latch



↳



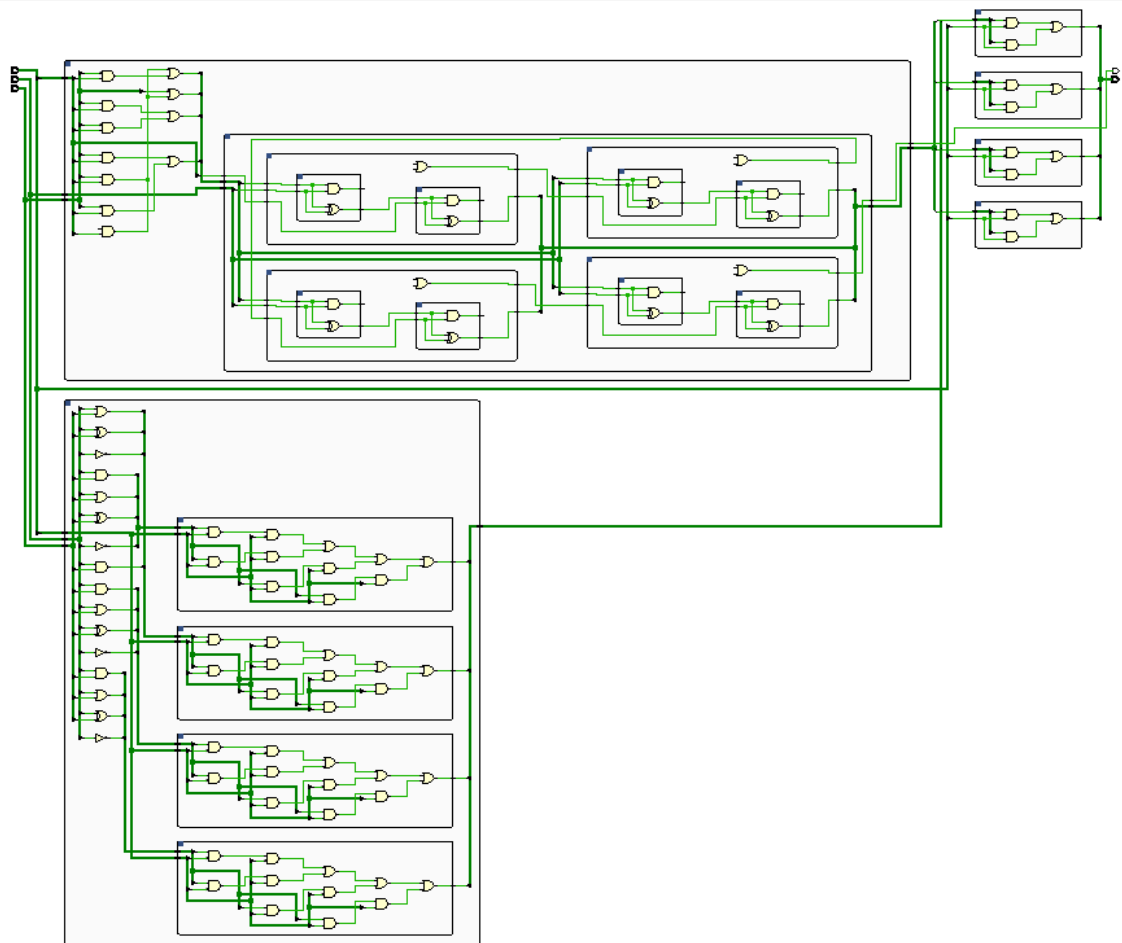
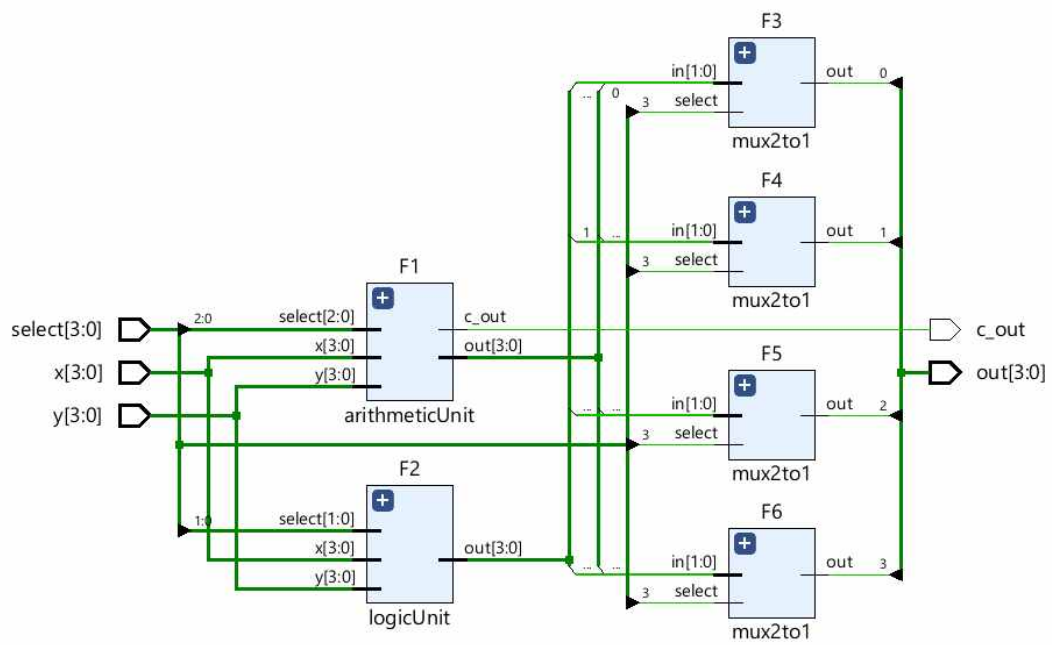
3. SR 래치와 비교해 Master-slave JK 플립플롭이 해결가능한 글리치와 해결불가능한 글리치를 예상하고 분석하기

Master-slave JK 플립플롭은 SR 플립 플롭의 한계를 극복한다. SR 플립 플롭에서 $S = R = 1$ 조건이되면 출력이 불확실해지나, JK 마스터 슬레이브에서 $J = K = 1$ 이면 출력이 토글되며, 출력이 Clock와 함께 계속 변경된다.

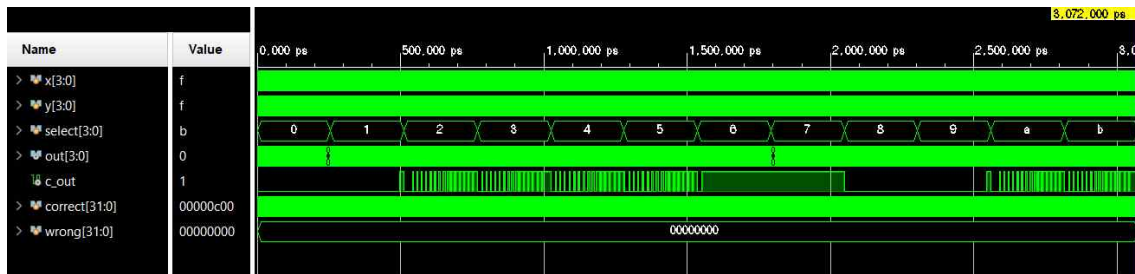
4. 실험

1. ALU

- 회로도 구성은 아래와 같다.



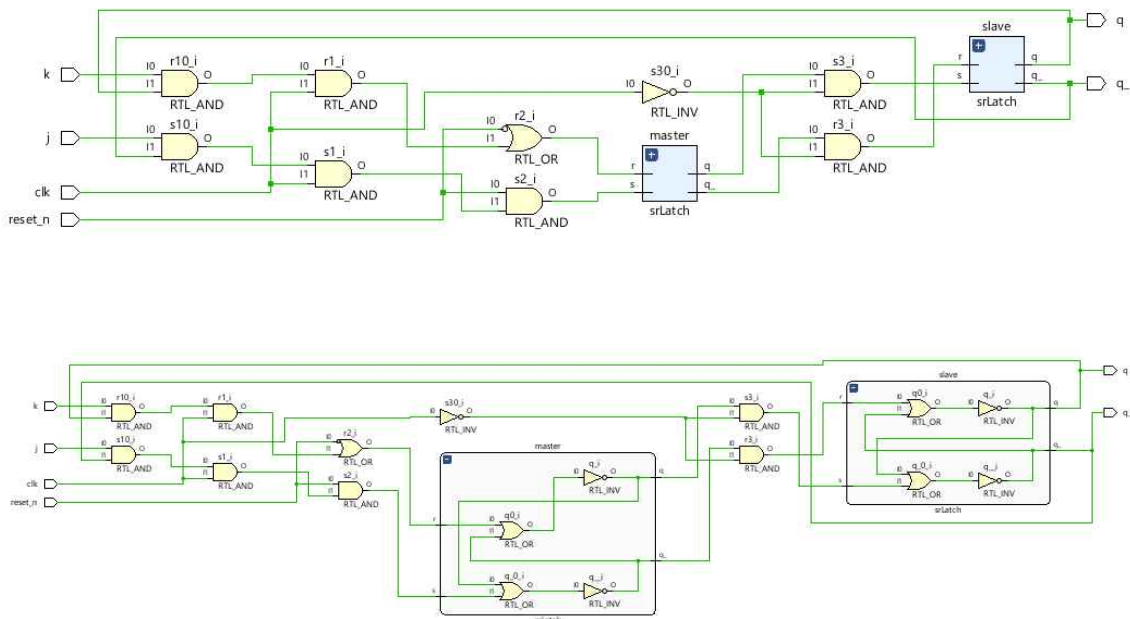
이에 따른 테스트벤치의 결과는 아래와 같다.



오류표현이 나타나지 않는 것으로 보아 실행이 옳게 된 것으로 확인된다.

2. Master-Slave JK 플립플롭

- 회로도



-테스트벤치 결과



대부분의 결과에서 오류가 나타나지 않으며, q, q_에서의 초기 오류는 q, q_의 선언시간에 의한 문제로 프로그램 작동에 문제가 없을 것으로 생각된다.

테스트벤치를 수정함으로써, 기존 SR 랜치에서 J=K=1인 case에서 오류가 나는 것을 해결한 것을 확인할 수 있다.

5. 논의

- 이번 실험을 통하여 ALU와 플립플롭에 대해 학습할 수 있었다. 실은 이번 실험은 Verilog 파일을 완성하는 것 보다 테스트벤치 작성에 너무 많은 시간을 사용하였다. 베릴로그 언어와 문법에 대해 좀 더 학습할 필요가 있고, 테스트벤치 작성법에 관한 공부를 해야할 필요성을 느꼈다.