

디지털시스템설계 Lab 5

손량(20220323)

Last compiled on: Friday 19th May, 2023, 23:35

1 개요

현대적인 컴퓨터의 기반이 되는 ALU와 flip-flop을 verilog로 구현해 보고, 시뮬레이션과 실험을 통해 작동을 확인한다.

2 이론적 배경

2.1 ALU

ALU는 입력이 주어졌을 때 산술, 논리 연산을 수행하는 회로이다. 입력에 따라 수행할 연산을 골라서 operand끼리 연산을 수행한다.

2.2 비동기 회로와 동기 회로

클럭 등의 신호를 이용하여 내부의 구성 요소들을 동기화하는 회로를 동기 회로, 그렇지 않은 회로를 비동기 회로라고 부른다. 비동기 회로의 예시로는 Combinational logic, 클럭과 상관없이 동작하는 회로가 있다. 이번 lab 5에서 다루는 회로 중 latch는 비동기 회로, flip-flop은 동기 회로에 해당한다.

2.3 JK Latch

가장 기본적인 latch인 SR latch는 forbidden state가 존재한다는 단점이 있다. JK latch는 SR latch에 게이트를 추가하여 forbidden state가 일어나지 않도록 한다. SR latch에서 forbidden state에 해당하는 입력은 JK latch에서는 저장된 비트를 반전시키는 데에 사용한다.

JK latch에 클럭 신호를 통한 동기화를 추가하면 JK flip-flop을 만들 수 있다.

2.4 Master-Slave JK Flip-Flop

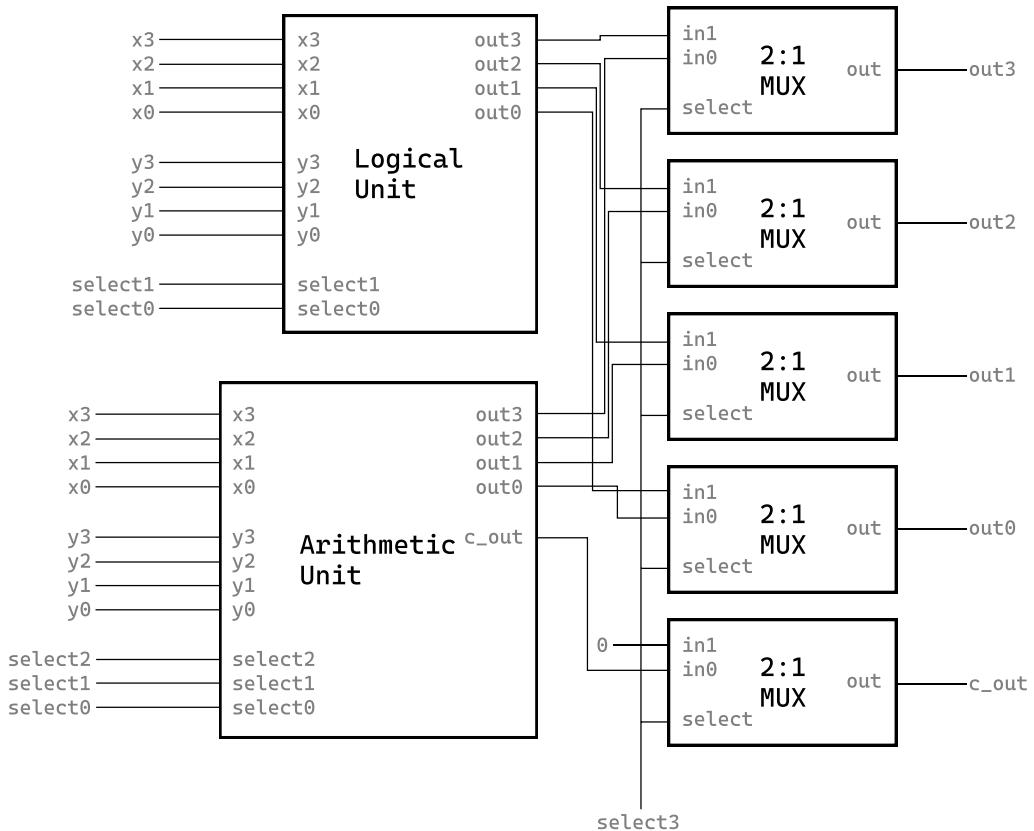
Master-slave JK flip-flop은 SR 래치 두 개를 연결해 만든 flip-flop 회로이다. 이름이 의미하듯 master latch와 slave latch가 존재한다. 클럭이 1인 동안에는 master latch가 활성화되어 입력을 저장하고, 0인 동안에는 slave latch가 활성화되어 master latch의 출력을 저장하게 된다.

클럭이 0인 동안에는 slave latch에 들어가는 입력이 일정하기 때문에 상관없지만, 클럭이 1인 동안에는 계속 입력을 받으므로 master latch에 글리치로 인해 잠깐 입력이 생기면 글리치를 정상적인 신호로 인식한다는 문제점이 존재한다. 이를 1's catch라고 부르고, edge-trigger 회로를 사용하여 해결하는 방법이 있다.

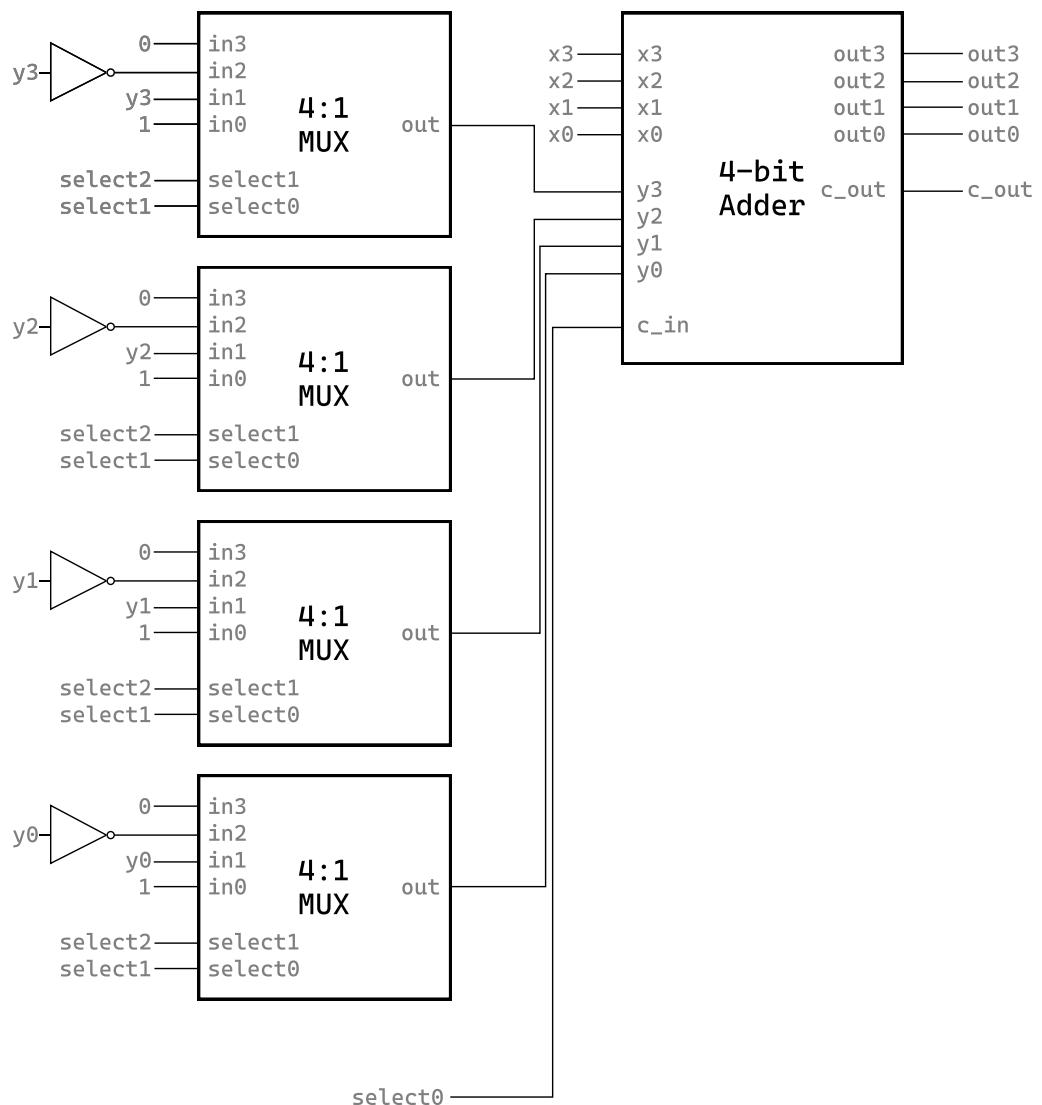
3 실험 준비

3.1 ALU

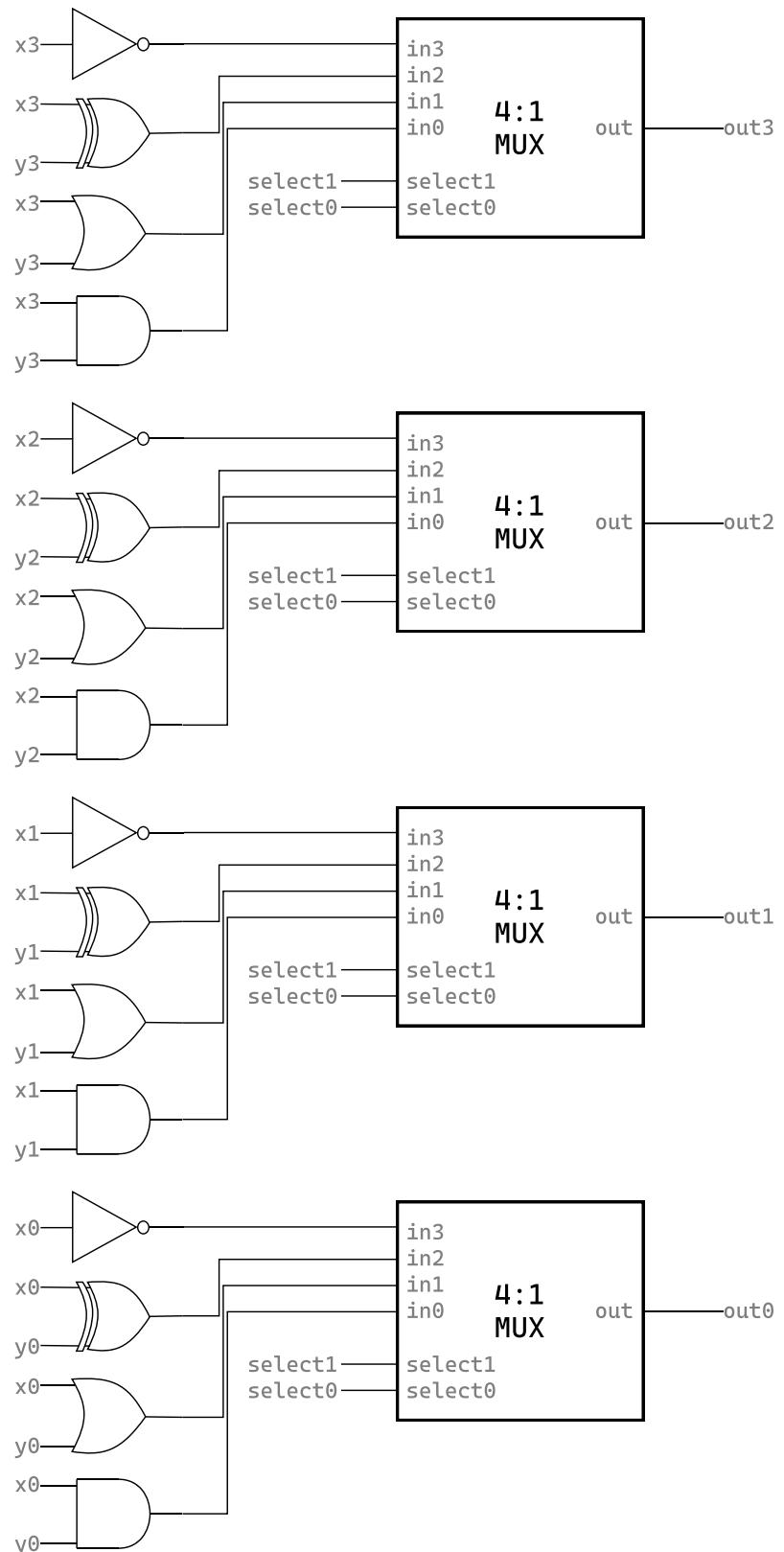
ALU의 전체적인 회로도는 다음과 같다.



Arithmetic unit의 회로도는 다음과 같다.

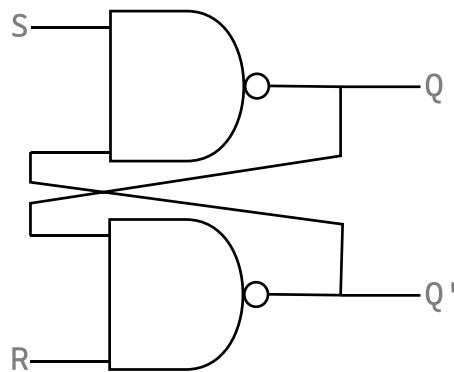


Logical unit의 회로도는 다음과 같다.

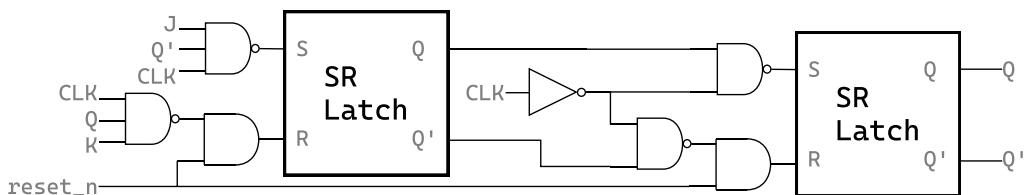


3.2 Master-Slave JK Flip-Flop

SR latch의 회로도는 다음과 같다.



Negative reset master-slave JK flip-flop의 회로도는 다음과 같다.

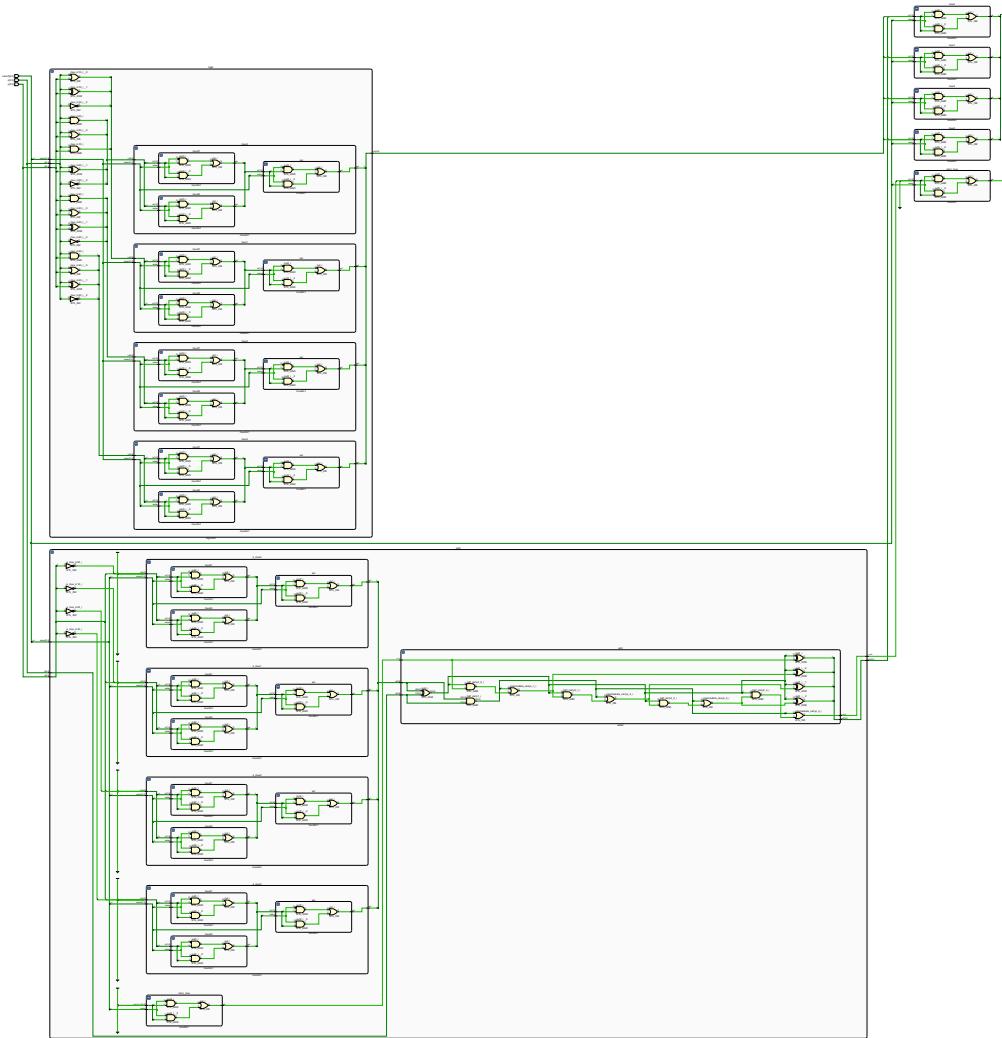


이 회로도의 SR latch는 $S = R = 0$ 입력이 금지되어 있고, $S = R = 1$ 입력 직후에 $S = R = 0$ 을 입력한다면 racing 현상과 같은 비정상적인 동작을 할 수 있다. 반면 master-slave flip-flop의 경우 SR latch에 게이트와 클럭을 추가함으로써 $S = R = 0$ 과 같은 금지된 입력이 들어가지 못하도록 한다. 하지만 master-slave flip-flop의 설계에서 클럭이 1인 동안에는 master latch가 입력을 계속 받는 상태이기 때문에 글리치로 생긴 입력 값이 회로 동작에 영향을 끼치는, 1's catch가 일어날 수 있다는 단점은 SR latch와 마찬가지로 해결하지 못한다.

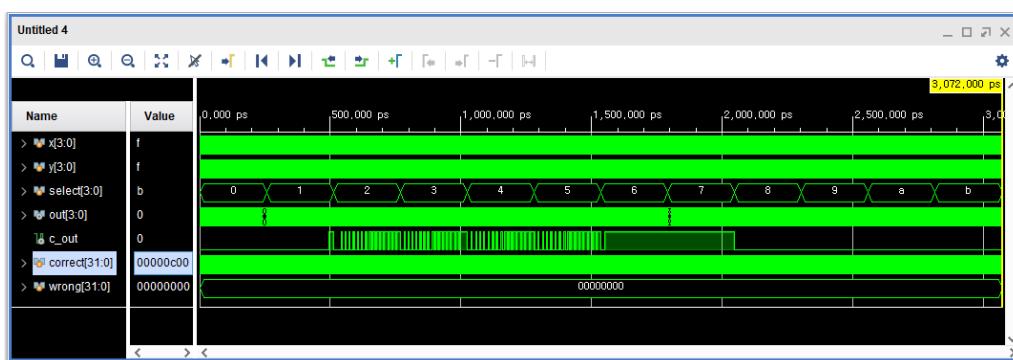
4 실험 결과

4.1 lab5_1.v – ALU 구현

화로도는 다음과 같다.



테스트벤치 실행 결과 다음 파형을 얻을 수 있었다.



4.1.1 FPGA 실험 결과

다음은 FPGA에서 코드를 실행한 결과들이다. $x = 13, y = 6$ 입력을 넣었고, sel 입력을 0부터 11까지 바꿔가며 실행한 결과는 다음과 같다.

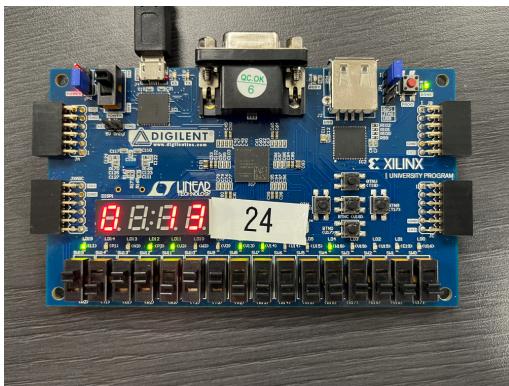


Figure 1: $\text{sel} = 0$

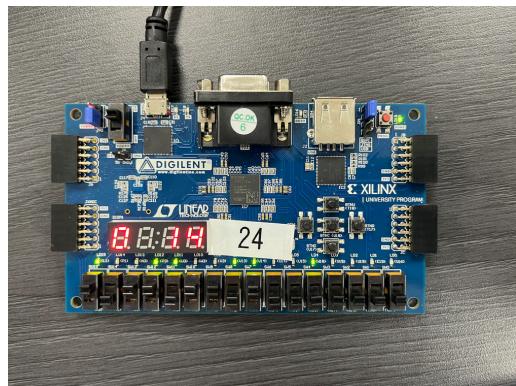


Figure 2: $\text{sel} = 1$

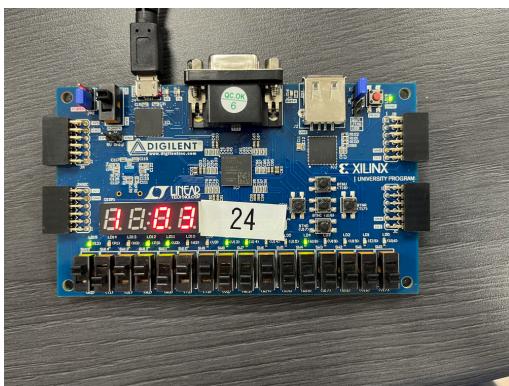


Figure 3: $\text{sel} = 2$

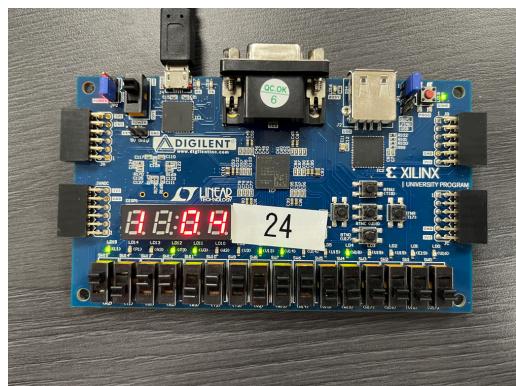


Figure 4: $\text{sel} = 3$

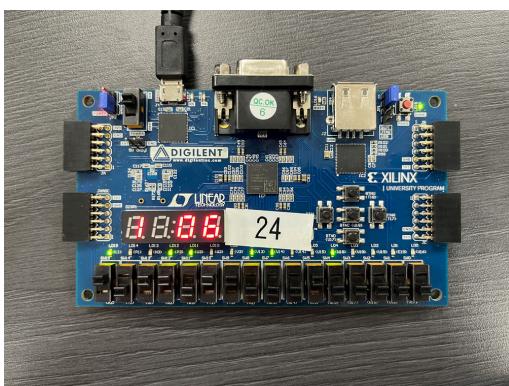


Figure 5: $\text{sel} = 4$

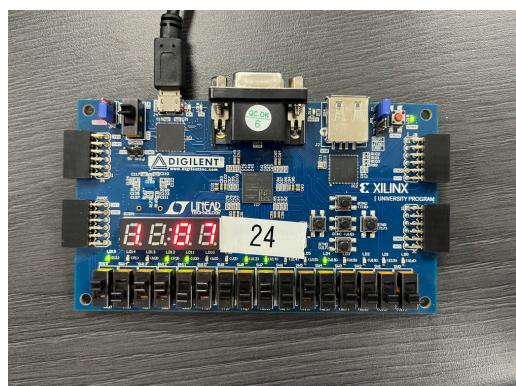


Figure 6: $\text{sel} = 5$

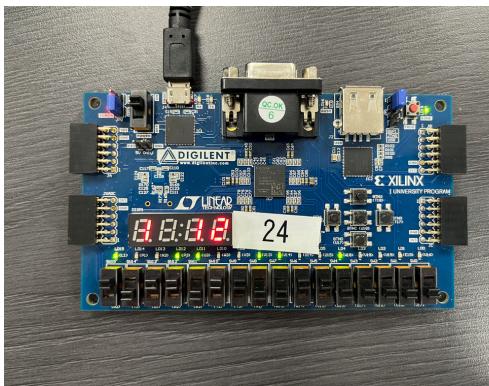


Figure 7: $\text{sel} = 6$

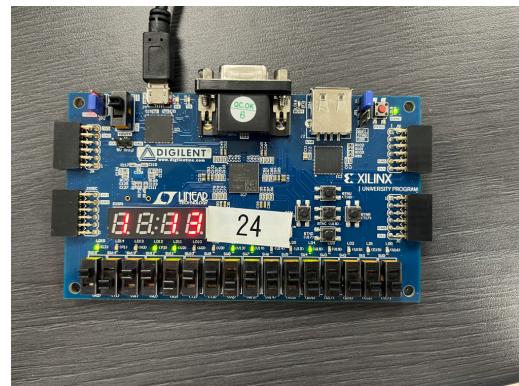


Figure 8: $\text{sel} = 7$

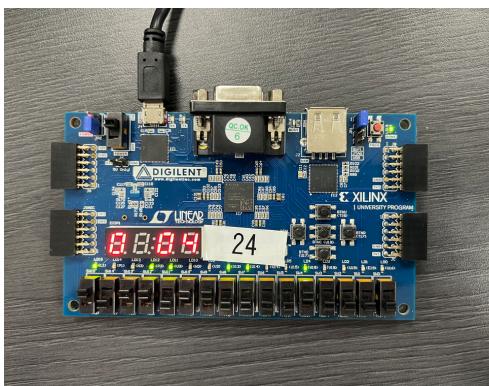


Figure 9: $\text{sel} = 8$

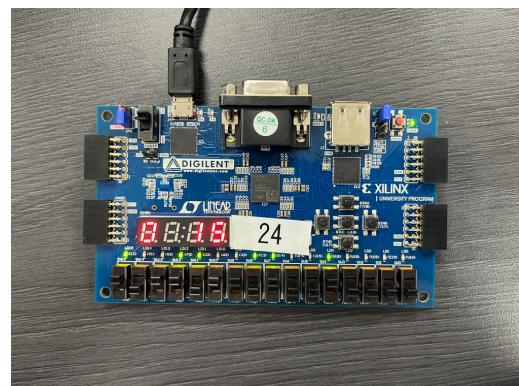


Figure 10: $\text{sel} = 9$

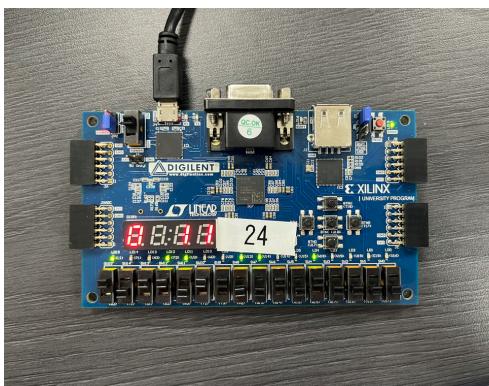


Figure 11: $\text{sel} = 10$

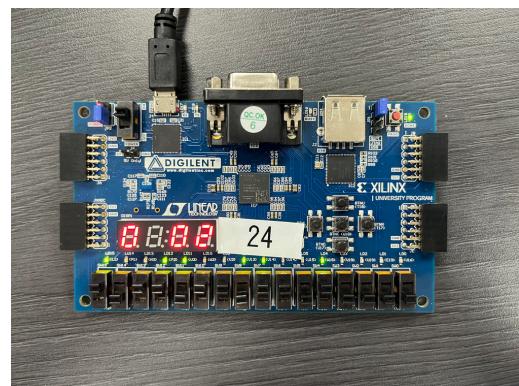
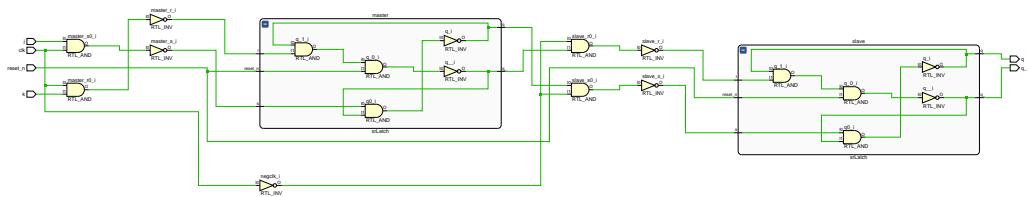


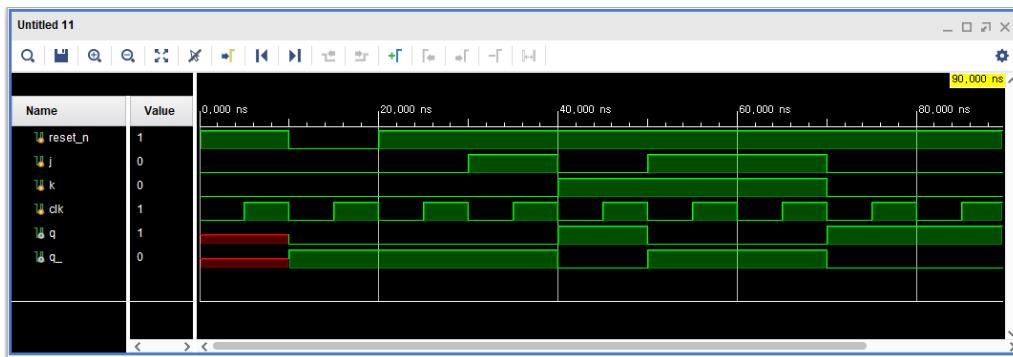
Figure 12: $\text{sel} = 11$

4.2 lab5_2.v – Negative Reset Master-Slave JK Flip-Flop

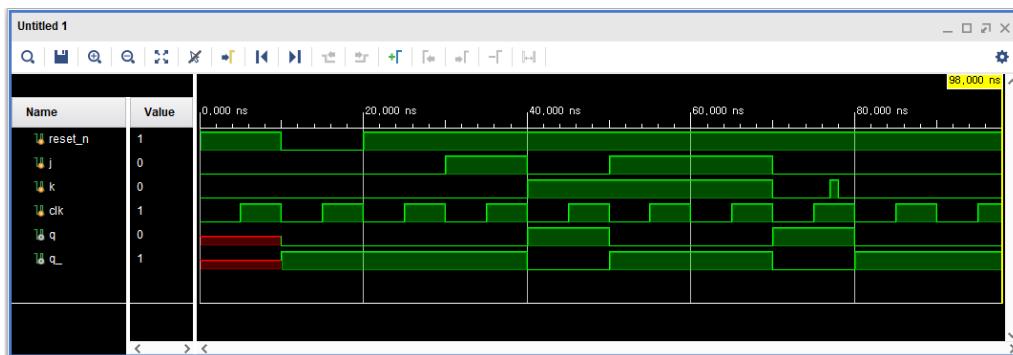
SR latch 두 개를 연결한 회로도는 다음과 같다.



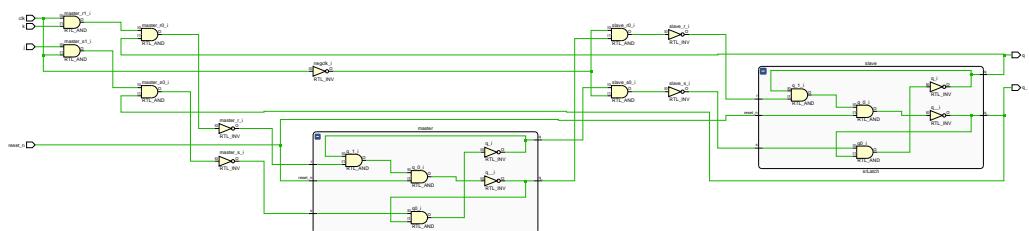
테스트벤치 실행 결과 다음 파형을 얻을 수 있었다.



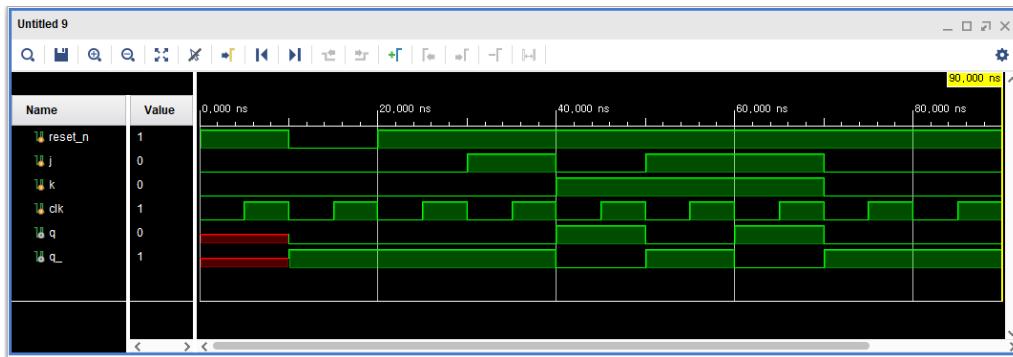
50ns 지점부터 20ns 동안 $J = K = 1$ 입력이 주어졌지만 toggle이 작동하지 않는 모습을 볼 수 있다. 또한, 이 회로에서는 짧은 시간동안 1이 입력되었을 때 출력에 영향을 미치는 1's catch 현상이 일어난다.



1ns 동안 잠시 $K = 1$ 입력이 주어졌을 때, 출력이 바뀌는 것을 볼 수 있다.
Negative reset master-slave JK flip-flop을 구현한 회로도는 다음과 같다.



테스트벤치 실행 결과 파형은 다음과 같다.



단순히 SR latch 두 개를 연결했을 때와 달리, toggle이 작동하는 것을 볼 수 있다. 한편, 이 회로에서도 1's catch 현상은 일어난다. 1ns 동안 잠시 $J = 1$ 입력이 주어졌을 때 출력이 바뀌었다.



5 논의

ALU와 flip-flop을 구현하면서 컴퓨터의 구조에 대해 알 수 있는 시간이었다. 구현과 실험 등에서 몇 가지 시행착오를 거쳤지만 지금까지 배운 내용으로 실제 동작하는 회로를 만들었다는 사실이 뜻깊었다.