

## Lab5. ALU와 JK 플립플롭

학번 : 20210273 이름 : 하태혁

### 1. 개요

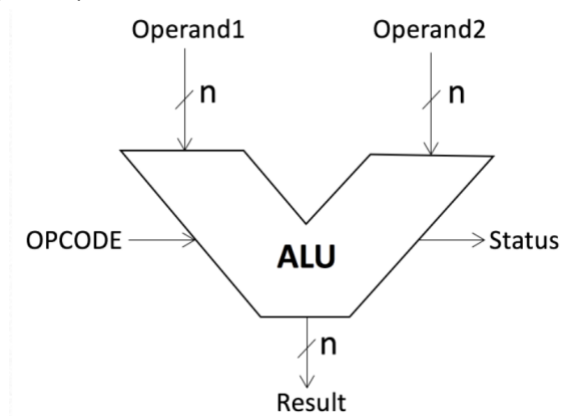
(컴퓨터의 기초가 되는 산술 논리 장치(Arithmetic Logic Unit; ALU)와 정보를 저장할 수 있는 JK 플립플롭(Flip-flop)을 구현한다.)

CPU의 연산에 사용되는 ALU의 기능을 이해하고 adder의 input을 수정하여 원하는 연산을 구현해본다. Memory element가 가져야 하는 특징을 이해하고 값을 저장하기 위한 JK FF의 작동을 이해하고 구현해 본다.

### 2. 이론적 배경

#### 1) ALU (Arithmetic Logic Unit)

산술 논리 장치 ALU는 입력에 따라 산술연산과 논리연산을 계산하는 디지털 회로이다. 본 실험에서 만드는 ALU는 select 인풋으로 기능을 조절하여 동작을 수행한다. ALU에서 산술연산 부분을 구현할 때는 Ripple adder를 사용하며 논리연산을 구현할 때는 4:1 MUX를 사용한다. 최종적으로 두 부분 중 어떤 동작을 사용할지를 구현할 때는 2:1 MUX를 사용한다.



#### 2) 비동기 회로 / 동기 회로

클럭 신호에 따라 동기화가 진행되는 디지털회로를 동기 회로 그렇지 않은 회로를 비동기 회로라 부른다. 회로에서는 CK라는 값으로 나타냈다.

#### 3) JK 래치 / JK 플립플롭

SR 래치에 추가적인 기능을 더하여 값의 toggle이 가능하게 한 래치를 JK래치라고 부른다. 그리고 JK 래치에 클럭 신호를 더해 만든 동기 회로는 JK 플립플롭이라 부른다.

#### 4) Master-slave JK 플립플롭

SR 래치 두개를 연결하고 클럭 신호를 더해서 만든 플립플롭이다. CK가 1일 때

master 부분에 값이 저장된 뒤 CK가 0이 될 때 slave 부분에 전달된다. Glitch가 발생했을 때도 CK = 1이면 값이 바뀌어 master에 저장되기 때문에 값이 바뀌는 문제가 발생한다.

### 3. 실험 준비

#### 1) Lab5\_1

산술 장치	Select				동작	Adder 입력		
	$S_3$	$S_2$	$S_1$	$S_0$	$out = A + B + C_{in}$	$A$	$B$	$C_{in}$
	0	0	0	0	$x$	0000	$x$	0
	0	0	0	1	$x + 1$	0000	$x$	1
	0	0	1	0	$x + y$	$y$	$x$	0
	0	0	1	1	$x + y + 1$	$y$	$x$	1
	0	1	0	0	$x + \bar{y}$	$\bar{y}$	$x$	0
	0	1	0	1	$x + \bar{y} + 1$	$\bar{y}$	$x$	1
	0	1	1	0	$x - 1$	1111	$x$	0
	0	1	1	1	$x$	1111	$x$	1
논리 장치	Select				동작			
	$S_3$	$S_2$	$S_1$	$S_0$	$out_i$			
	1	0	0	0	$x_i \text{ AND } y_i$			
	1	0	0	1	$x_i \text{ OR } y_i$			
	1	0	1	0	$x_i \text{ XOR } y_i$			
	1	0	1	1	$\bar{x}_i$			

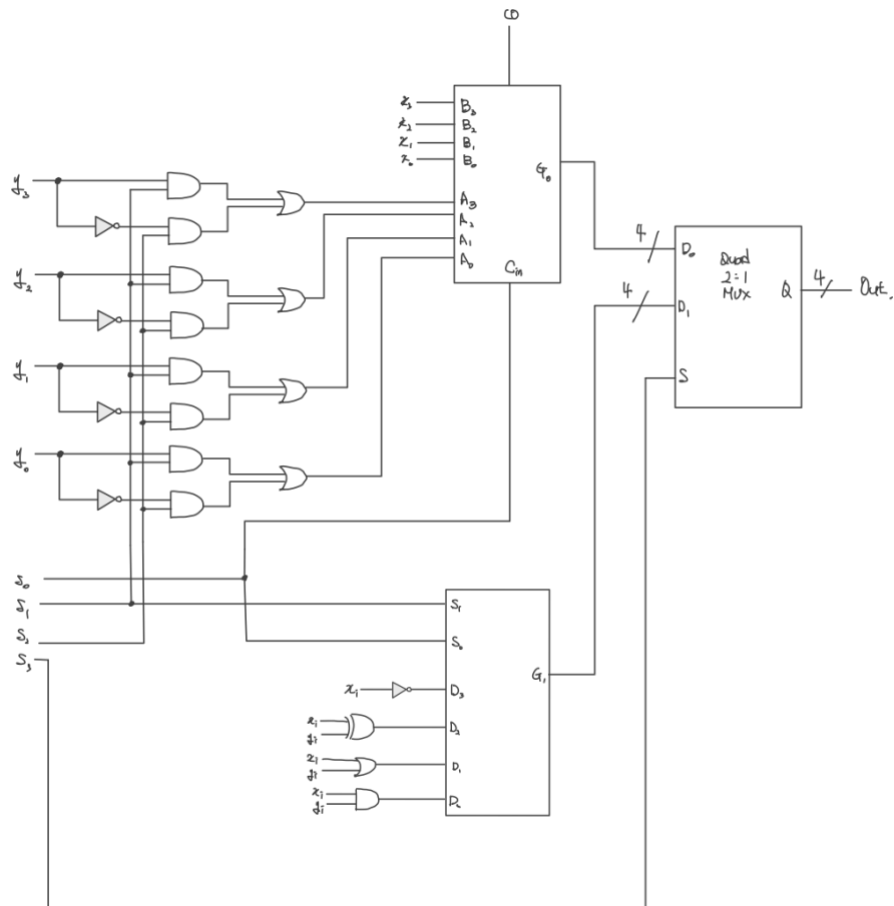
A에 대한 Arithmetic equation을 만들기 위해 function table을 그리면 다음과 같고 K-map 으로 나타낸 뒤 식을 구하면,

$S_2$	$S_1$	$Y_i$	$A_i$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$$A = S_2 y_i' + S_1 y_i$$

$$B = x$$

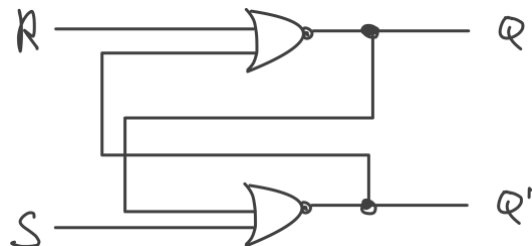
$$C_{in} = S_0$$



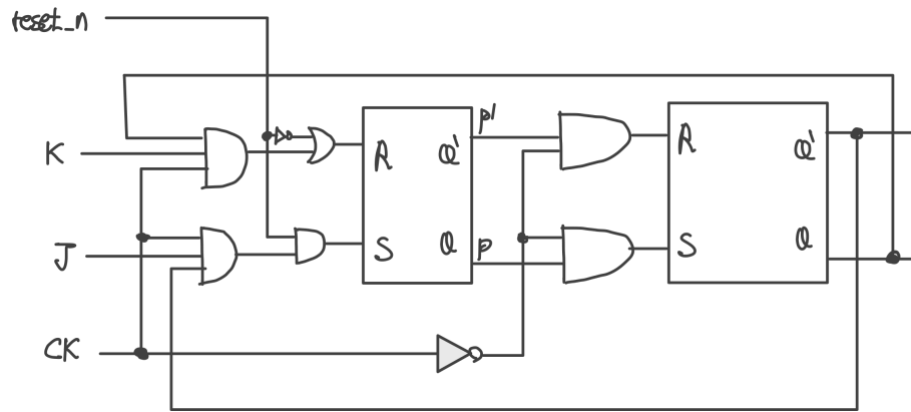
산술 장치와 논리 장치를 구현한 뒤 2:1 MUX로 묶어서 회로도를 그리면 위 사진과 같다.

## 2) Lab5\_2

ㄱ) SR 래치의 회로도를 그리면 다음과 같다.



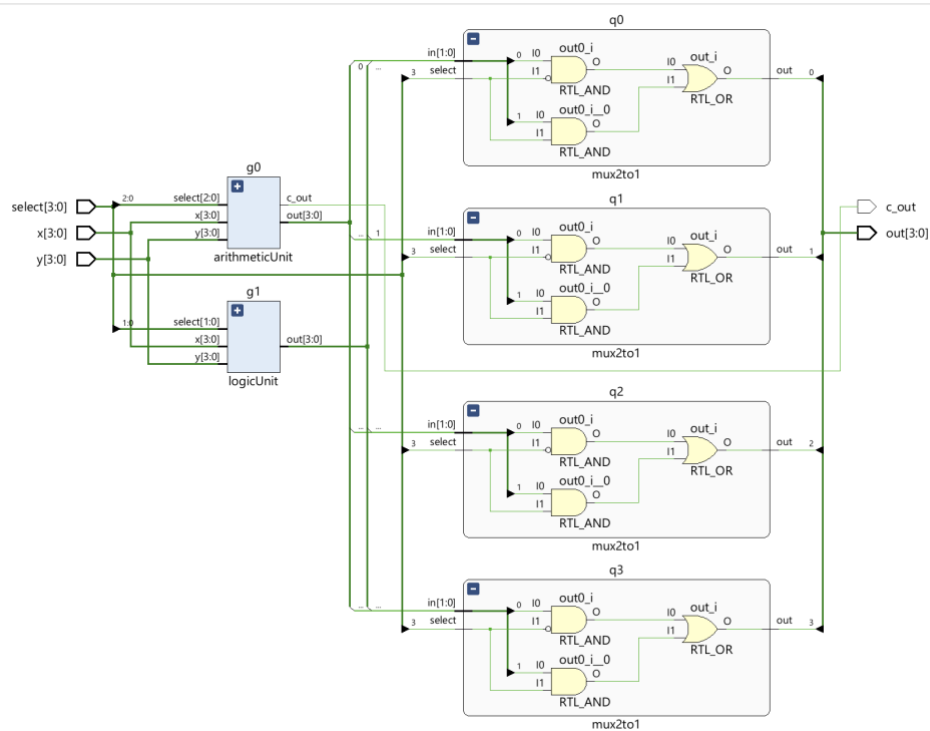
ㄴ) SR 래치를 사용해 Negative reset Master-slave JK 플립플롭의 회로도를 그리면 다음과 같다.



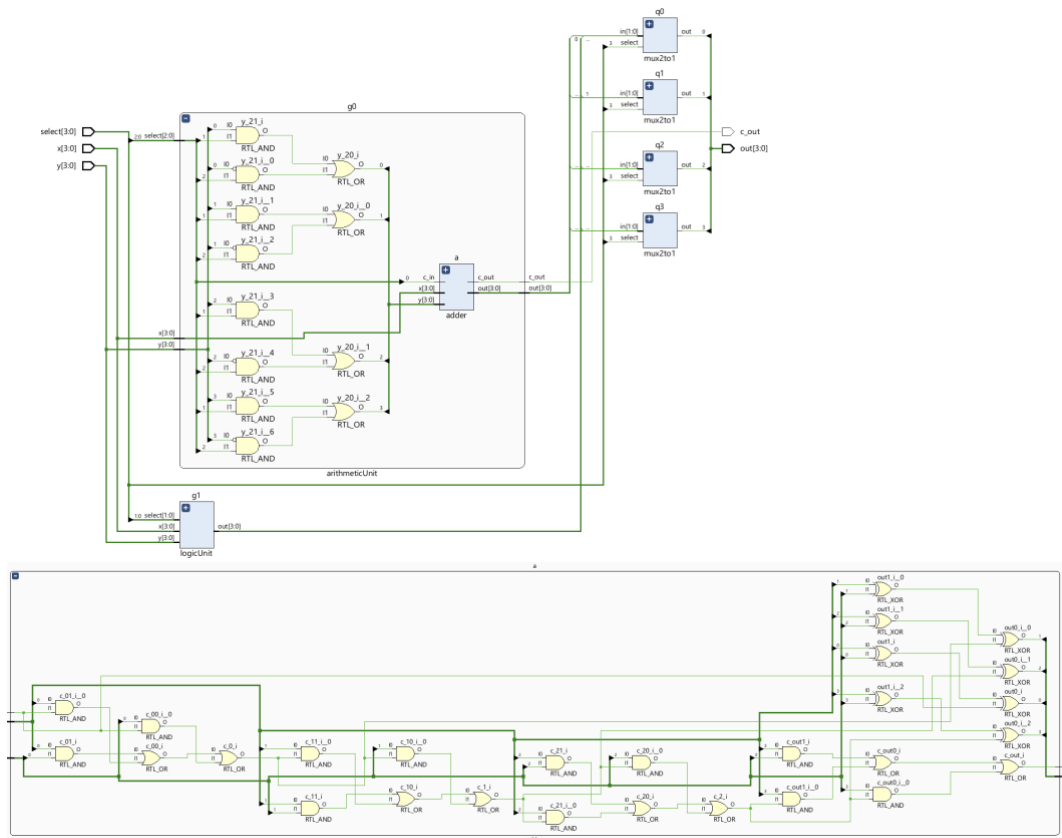
ㄷ) SR 래치와 비교해 Master-slave JK 플립플롭이 해결 가능한 클리치와 해결할 수 없는 클리치를 분석해보면,

## 4. 결과

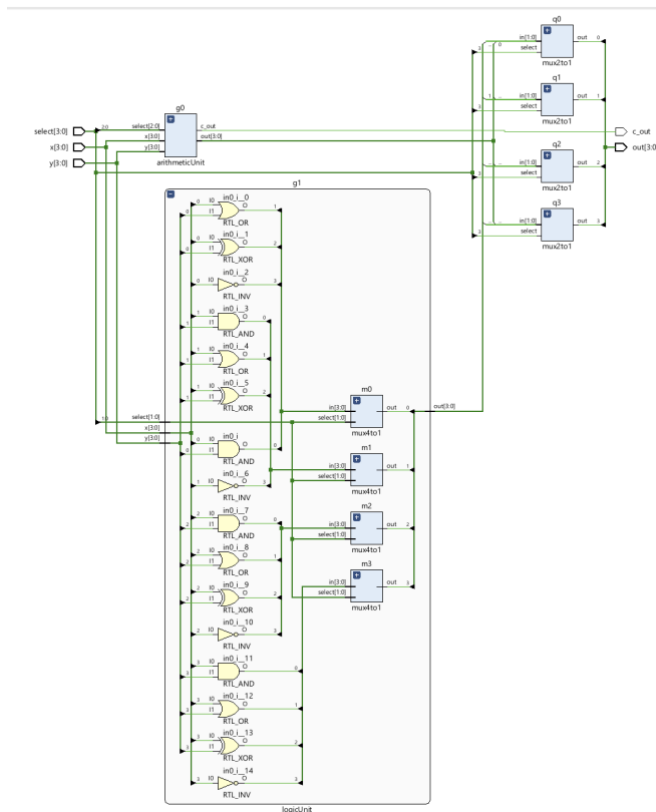
1) Lab5\_1

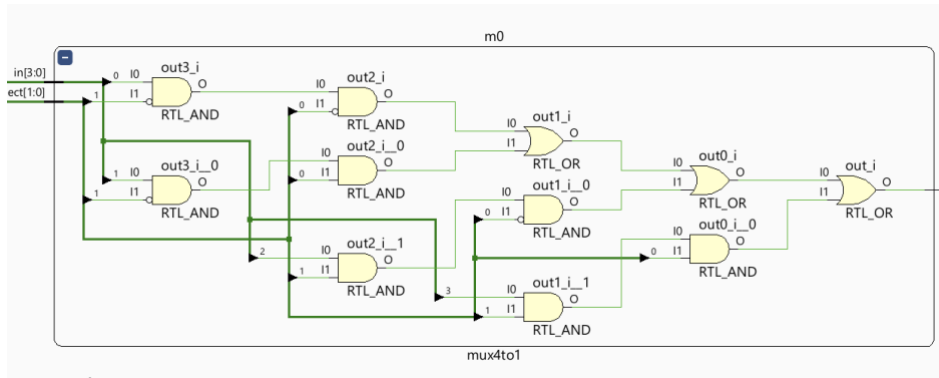


전체 회로도에 대한 그림은 위 그림과 같다.

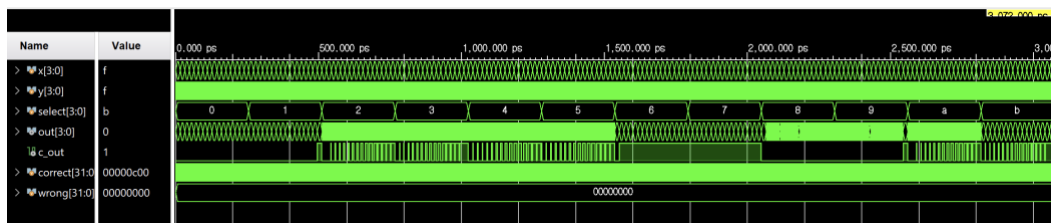


산술 장치와 adder에 대한 회로도 는 다음과 같다.



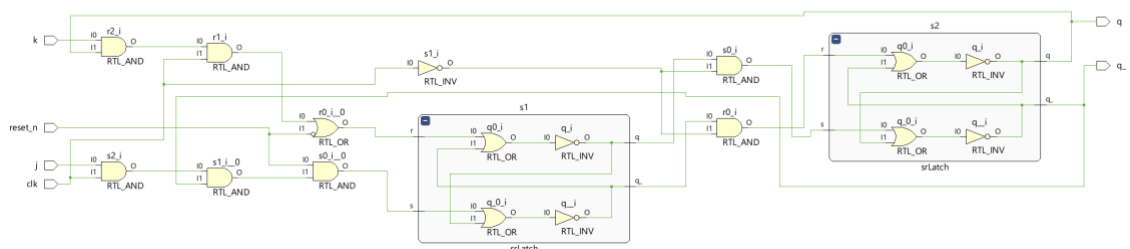


논리 장치 모듈과 4:1 MUX에 대한 회로도도 다음과 같다.

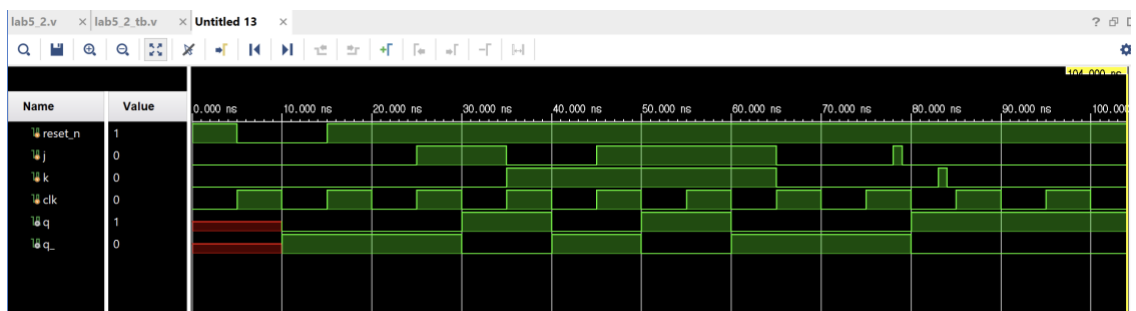


시뮬레이션 파형을 캡처하면 다음과 같다. CK=1일 때는 글리치 값이 반영되어 q값이 바뀌었지만, CK=0일 때는 글리치 값이 반영되지 않아서 q값이 바뀌지 않았음을 알 수 있다.

## 2) Lab5\_2



구현한 회로도 위 사진과 같다.



시뮬레이션 파형은 다음과 같다.

## 5. 논의

테스트벤치 코드도 작성해 보면서 베릴로그 문법에 대한 이해도를 높일 수 있었다.