

컴퓨터 공학 기초 실험2 보고서

실험제목: Latch & flip-flop design with/without
reset/set

실험일자: 2023년 10월 06일 (금)

제출일자: 2023년 10월 19일 (목)

학 과: 컴퓨터공학과

담당교수: 이형근 교수님

실습분반: 금요일 0, 1, 2

학 번: 2020202037

성 명: 엄정호

1. 제목 및 목적

A. 제목

Latch & flip-flop design with/without reset/set

B. 목적

Clock의 입력에 따라 값을 불러와 저장하는 D latch, D flip-flop을 구성해 보고, 추가로 enable, resettable 플립플롭을 구성한다. 주어진 플립플롭 회로 이외로 4개의 nand게이트를 이용한 플립플롭 구성 방법과 동기/비동기 신호의 차이점에 대해 이해한다.

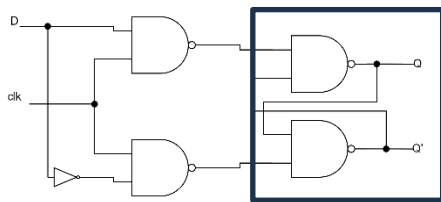
2. 원리(배경지식)

D latch : 2개의 입력(clk, D)을 갖는 레지, 데이터

입력(D) : latch에 저장될 데이터를 나타낸다.

클럭입력(clk) : latch가 어떤 경우에 동작할지를 결정하는 입력이다.

clk가 1일 경우 현재 D값을 불러오고, clk가 0일 경우 이전 Q값을 그대로 유지한다.



아래 설계에서 SR레지를 이용해 구성도 가능하지만 위처럼 nand게이트 4개를 이용한 구성도 가능하다. (검정박스가 SR latch이다.)

D flip-flop : 논리회로에서 사용되는 가장 기본적인 저장 장치이다. 데이터를 저장-전달하는 역할을 수행한다.

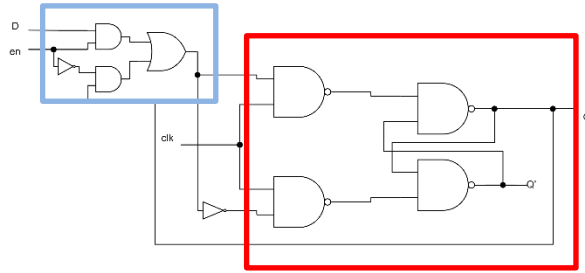
D : 데이터 입력 플립플롭에서 저장할 데이터 비트를 나타낸다.

Clk : 클럭 입력은 데이터를 저장할 타이밍을 정해준다.

Q : 플립플롭에 저장된 데이터를 나타낸다. D플립플롭에서 저장된 값이 D에 나타난다.

데이터(D)를 입력받고 클럭(clk)이 상승하거나 하강할 때 입력 데이터를 저장한다. 이러한 D flip-flop여러개를 조합하면 register를 구성할 수 있다.

enabled D flip-flop : 데이터를 저장할 때 clk신호 이외에도 활성화 신호(Enable)를 이용하여 데이터를 저장하거나 전달하는 플립플롭이다. Clk가 상승, 또는 하강할 때 작동하는 일반적인 플립플롭과 달리 enable신호가 추가로 들어가 enable신호가 활성화되지 않으면 데이터를 업데이트 하지 않는다.



Filp-flop(붉은색)에 멀티플렉서(푸른색)를 연결하여 구성할 수 있다.

Resettable-Filp-Flop : resettable-flip-flop은 특정신호(reset)가 활성화 될 때 저장된 데이터를 리셋하고 초기상태로 되돌리는 회로이다. D flip-flop과 유사하게 작동하지만, 리셋 신호가 활성화 될 때 저장된 데이터를 초기 상태로 되돌린다는 점에서 다르다.

Register : 레지스터는 플립플롭으로 구성된 메모리 단위이다. 특정 비트수로 구성된 메모리 셀의 배열로, 각 메모리 셀은 플립플롭으로 구현 될 수 있다. 8bit으로 구성된 레지스터의 경우 8개의 플립플롭이 이용되며, 32bit레지스터의 경우 4개의 8bit레지스터 즉 32개의 filp-flop으로 구성된다.

Asynchronous : 동시에 모든 일이 수행되는 것이 아닌, 각자가 각자의 타이밍에 맞게 일을 수행하는 것이다. 즉 작업의 요청과 응답의 타이밍이 같지 않을 수 있다.

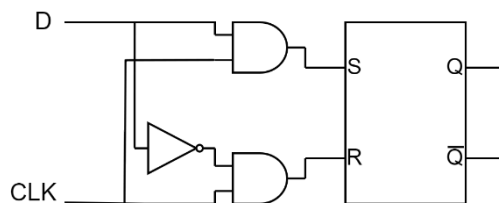
- 설계가 복잡하지만 결과가 나오지 않아도 다른 일들을 병렬적으로 수행 할 수 있다는 장점이 있다.

Synchronous : 동기식 - 모든 일의 진행이 동시에 진행되도록 설계하는 것이다.

즉 blocking처럼 호출된 함수가 자신이 할 일을 모두 마칠 때 까지 제어권을 계속 가지고 호출한 함수에게 돌려주지 않는 것이다.

- 설계가 간단하지만 이전 일의 수행이 끝나지 않을 경우 다음 일을 수행할 수 없다.

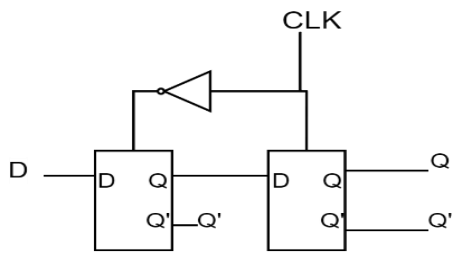
3. 설계 세부사항



D-latch Schematic

CLK	Q
0	이전Q
1	D

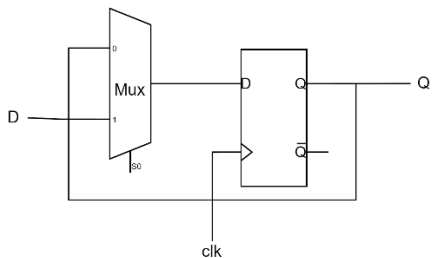
D-latch Truth table



D-Flip-Flop Schematic

CLK	D(t)
↑	D
In other case	이전 D

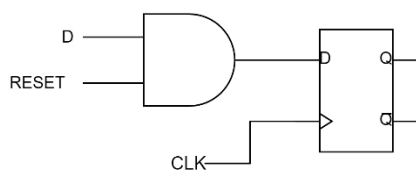
D-Filp-Flop truth table



enabled D flip-flop Schematic

EN	clk	D(t)
1	↑	D
0	X	이전 D

enabled D flip-flop truth table



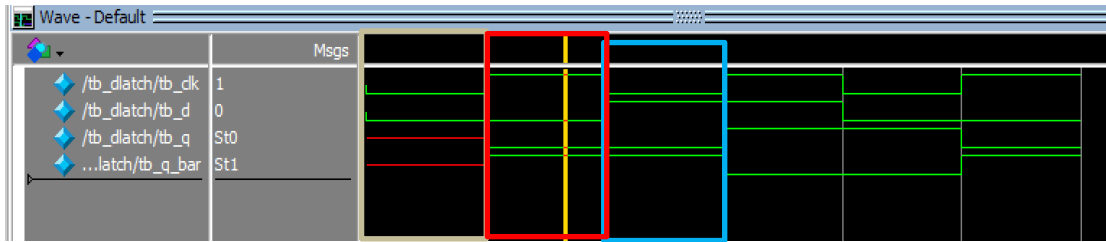
Resttable-Filp-Flop Schematic

Input			Output
R	D	CLK	Q
0	X	X	0
1	0	↑	0
1	1	↑	1
1	X	↓ or 0 or 1	이전 Q

Resttable D-Filp-Flop truth table

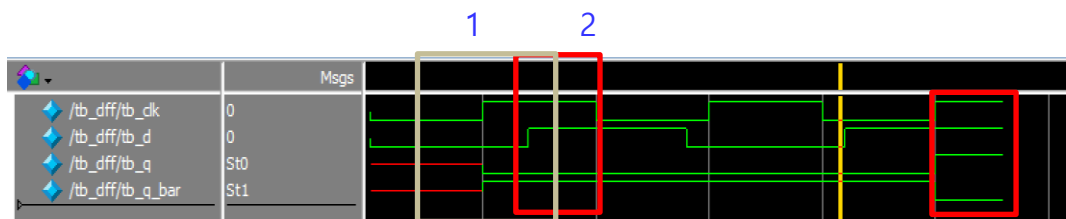
4. 설계 검증 및 실험 결과

A. 시뮬레이션 결과



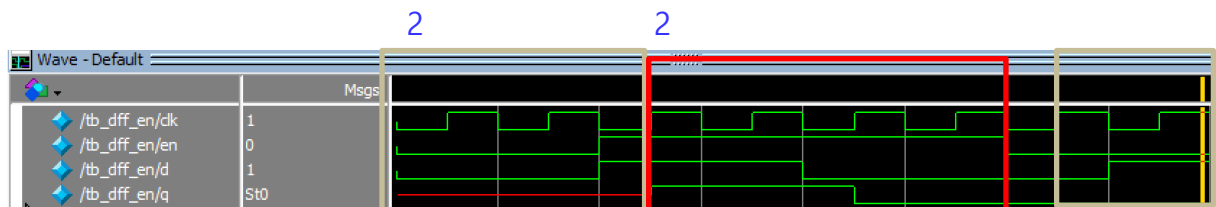
D-latch

1. CLK의 초깃값이 0이기 때문에 Q값이 Known이다.
2. CLK가 1이 됐을 때 Q값을 이어받는다.
3. CLK가 0이면 이전 값을 유지한다.



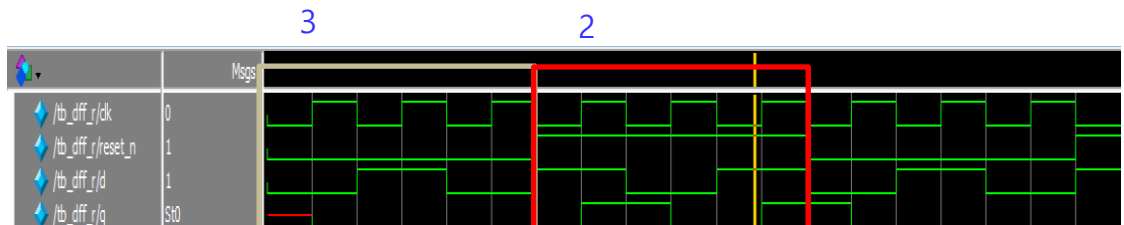
D-Flip-Flop

1. CLK의 초깃값이 0이기 때문에 Q값이 Known이다.
2. Clock이 rising됐을 때 D값을 받아오고 이 외의 경우에는 이전 값을 유지한다.



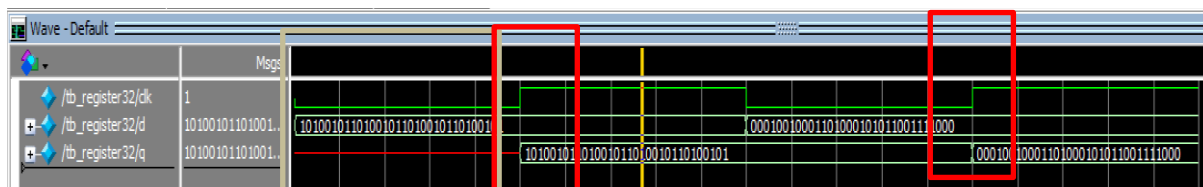
enabled D flip-

1. Enable이 0인 상태에서는 CLK가 rise되어도 값을 입력 받지 않는다.
2. Enable이 1인 상태일 때에는 D-flip-flop처럼 작동하는 것을 확인할 수 있다.



resettable D flip-flop

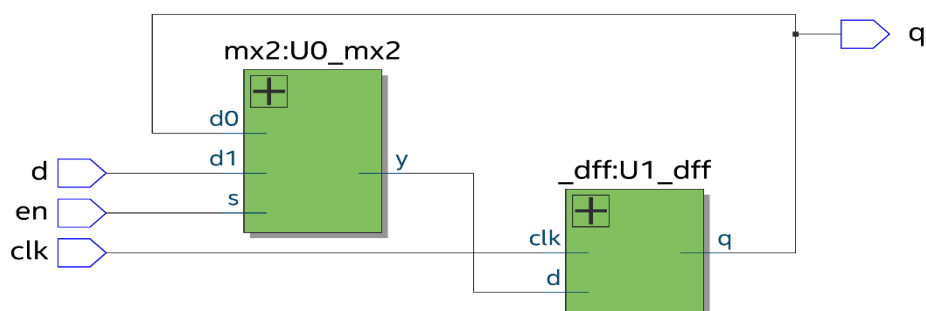
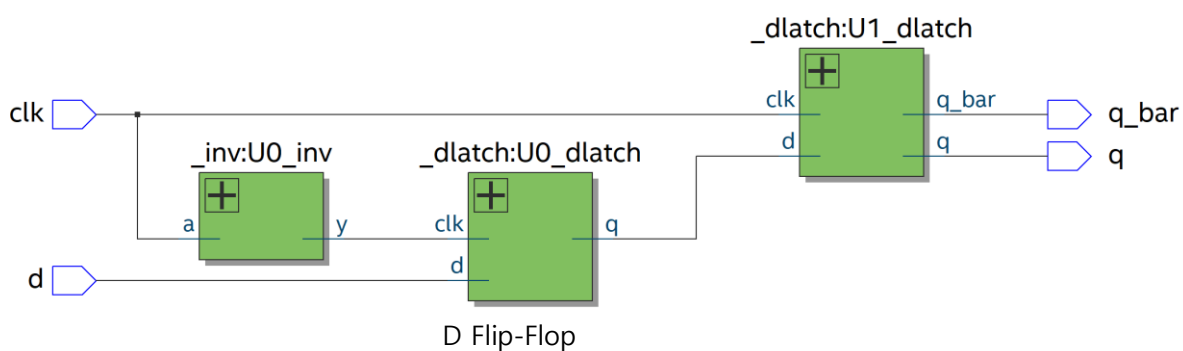
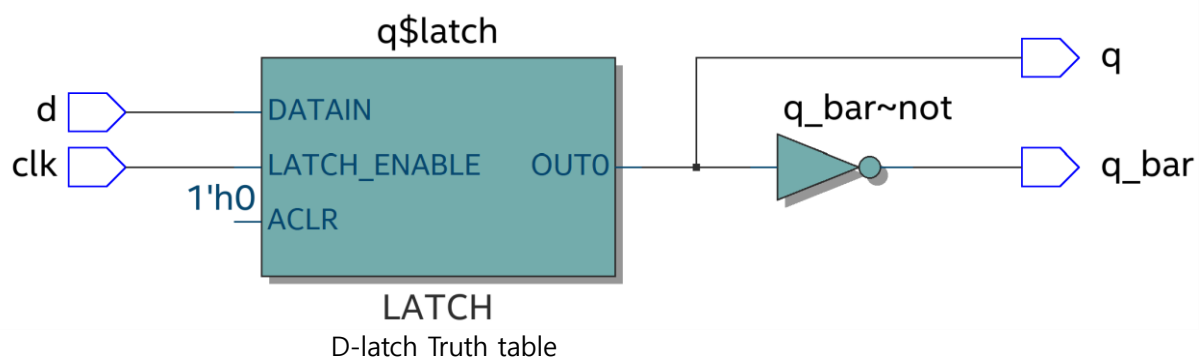
1. RESET이 0일땐 d값에 관계 없이 0으로 유지된다.
2. RESET이 1일땐 D-flip-flop으로 작동
3. Active high에 동작하는 리셋의 경우는 reset값이 1일 때 reset



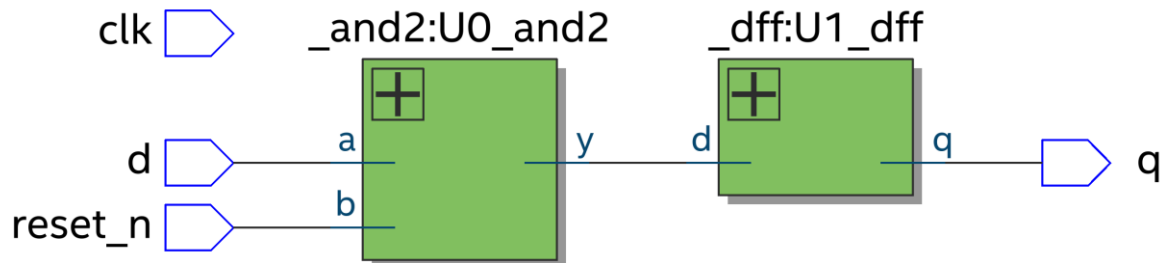
32 bits register

1. 초기 상태
2. CLK이 RISE됐을 때 d값이 Q에 저장되는 것을 확인 할 수 있다.

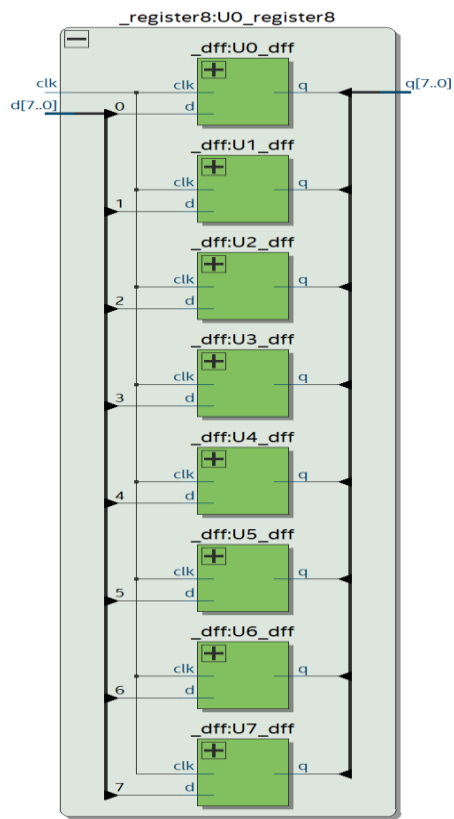
B. 합성(synthesis) 결과



Enabled D Flip-Flop

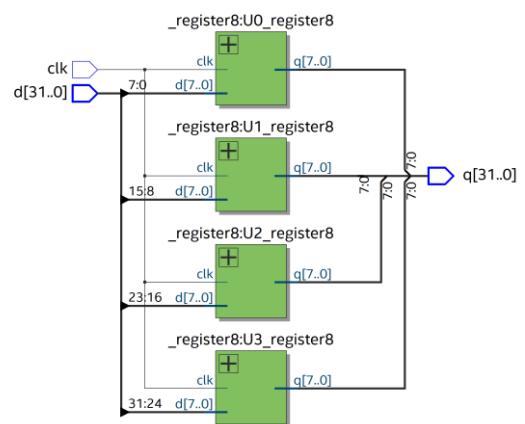


Resettable D Flip-Flop



8bit_reg

8개의 flip-flop를 한 줄로 연결



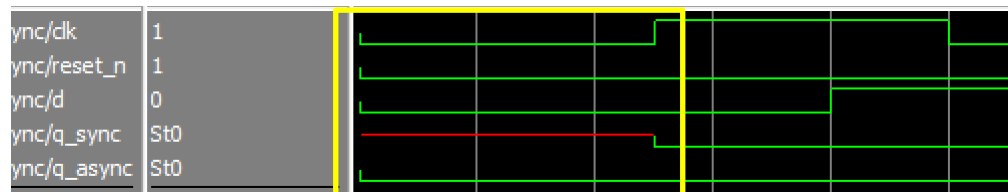
32bit_reg

4개의 8bit-flip-flop를 한 줄로 연결

Async/Sync(비동기, 동기 신호)

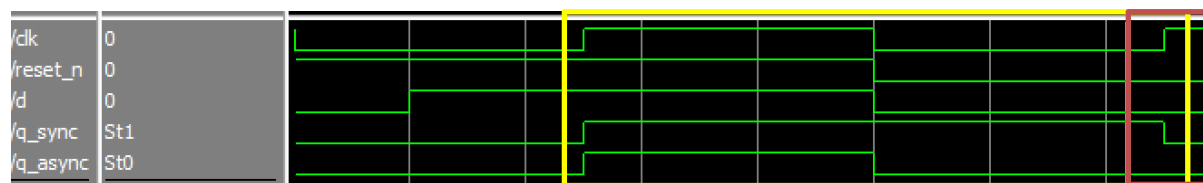
Async/Sync 두 신호 모두 active low에서 작동하도록 설계되어 있지만, Sync의 경우 첫 clk rise되기 전까지 값이 unknown이 되는 것을 확인 할 수 있고 Async신호는 바로 0신호가 들어오는 것을 확인 할 수 있다.

이유는 동기 신호의 경우 clk가 rise 됐을 때 reset이 0이 되는지 확인 후 q값을 초기화 하기 때문이고, 비동기식 신호의 경우 clk가 rise 됐을 때에도 값을 전달하고 reset이 0이 되었을 때에서 q를 초기화 시키기 때문에 초깃값이 들어간 것을 확인 할 수 있다.



아래 사진에서도

비동기 신호의 경우 reset이 0일 경우 q에 반영되는 반면 동기 신호의 경우 clk가 한번 rise 된 후에 값이 반영되는 것을 확인 할 수 있다.



5. 고찰 및 결론

A. 고찰

4번째 실험을 진행 하면서 이상한 점을 확인했었다. 배웠던 대로라면 reset이 0이 되었을 때 Q값이 0이 되었어야 했는데, 이전 값을 유지하다가 clk가 rise 되었을 때 값을 반영한다는 것을 확인했다. 실험 5번을 진행하면서 내가 구성했던 회로가 동기식 회로 이런 일이 발생했다는 것을 확인했다.

실험 5번에서 30ns정도의 실험을 진행 했는데 wave form의 차이를 확인하지 못했다.

Clk와 resetable를 다른 간격으로 두고 d값을 바꿔가며 300ns시간의 실험 결과를 확인하니 차이점을 찾을 수 있었다.

B. 결론

실험을 통해 latch, flip-flop와 같은 회로에 값이 어떻게 저장되는지 알 수 있었다. 또한 어셈블리 수업 초반 당시 교수님이 레지스터는 플립플롭의 뭉치라는 말씀해 주셨는데 플립플롭을 이용한 레지스터를 구상해 보니 명확한 이해가 가능했다.

동기식, 신호와 비동기식 신호의 차이를 알 수 있었다. 이를 통해 다음 traffic light

control에서 동기/비동기 신호 모두를 사용 하면 편리할 것 같았다.

6. 참고문헌

D latch/

https://ktword.co.kr/test/view/view.php?m_temp1=5300

Flip flop/

<http://www.ktword.co.kr/test/view/view.php?nav=2&no=4712&sh=enabled+d+flip+flop>

이준환/디지털논리회로2/광운대학교/2023

이형근/컴퓨터공학기초실험2/광운대학교/2023