

# 컴퓨터 공학 기초 실험2 보고서

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

실험일자: 2019년 09월 17일 (화)

제출일자: 2019년 09월 30일 (월)

학 과: 컴퓨터정보공학과

담당교수: 이준환 교수님

실습분반: 화요일 0, 1, 2

학 번: 2015722031

성 명: 박 태 성

## 1. 제목 및 목적

### A. 제목

Latch and Flip Flop with/without Reset/Set

### B. 목적

Latch와 Flip Flop의 동작 원리를 이해하고 직접 구현해 보면서 reset, set이 있을 때와 없을 때의 동작 원리를 이해함을 목적으로 한다. 그리고 동기식과 비동기식의 reset, set의 동작 원리를 waveform을 통하여 비교하여 그 차이를 이해하는데 목적을 둔다.

## 2. 원리(배경지식)

### A. Sequential Circuit

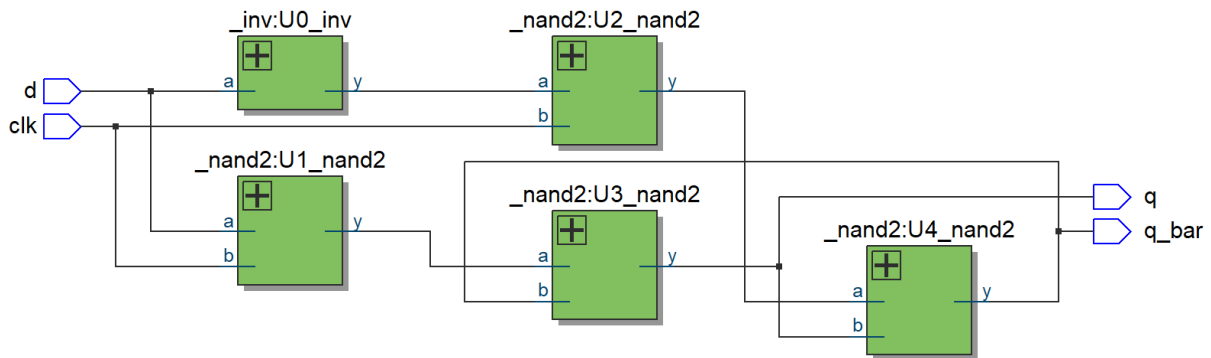
Sequential circuit은 입력과 이전 입력에 의해 출력이 결정되는 회로를 의미한다. 따라서 저장소자인 latch와 flip-flop이 사용된다. 동작 방식에 따라 synchronous와 asynchronous로 구분된다. 동기식은 clock과 동기화되어 clock의 영향으로 (거의)동시에 결과가 출력되는 반면, 비동기식은 clock과 동기화되어 있지 않아 clock의 rising 또는 falling edge에 영향을 받지 않고 결과를 출력하기도 한다.

### B. Latch

Latch에는 대표적으로 SR latch와 D latch가 있다. SR latch(Set-Reset)는 set과 reset의 값에 따라 출력된다. 단, set과 reset이 모두 1이면 결과가 출력되지 않는다. 이 점을 개선한 latch가 JK latch이다. JK latch는  $J = K = 1$ 이면 현재 값을 역으로 변화시킨다. D latch는 clock이나 enable에 따라서 d를 q로 출력하거나 하지 않는다. Clock이나 enable이 없다면 이전 q값을 계속해서 출력한다.

#### i. Nand gate 4개를 이용하여 D latch를 구현하는 방법

D latch는 and gate 2개, inverter 1개, SR latch 1개를 instantiate해서 만든다. SR latch는 nor gate 2개로 구성되어 있다. Bubble pushing(드모르간 법칙)을 응용하여 nand gate 4개로 만들 수 있다. Nor gate를 and gate으로 바꾸어 주면서 bubble(inverter)의 위치를 옮기면 nand gate 4개로 구성한 D latch를 만들 수 있다. Verilog HDL로 코드를 작성한다면 inverter 1개와 nand gate 4개를 instantiate하여 만들 수 있다. 다음은 Verilog HDL 코드로 작성한 D latch의 RTL Viewer이다.



### C. Flip-Flop

Flip-Flop은 클럭 신호의 변화에 동작하여 한 클럭 주기 동안 1bit의 정보를 저장하는 회로이다. 클럭이 rising하거나 falling하는(변화하는) 시점에 입력 값을 출력값에 반영한다는 점에서 latch와 차이가 있다.

#### i. Enabled D Flip-Flop(F/F)

Enabled D F/F은 2-to-1 mux와 D latch 2개로 구성되어 있다. 다른 structural 구조를 사용하여 설계하는 방법은 D latch를 nand gate 4개로 구성된 D latch로 바꾸는 것이다. 코드를 수정하고자 한다면, behavior implementation을 통하여 구현할 수 있다. 다음은 슈도 코드이다.

```
always @(posedge clk)
```

```
if ( en == 1)
```

```
    q <= d;
```

```
else
```

```
    q <= q;
```

마찬가지로 2-to-1 mux도 삼항 연산자를 사용하면 다른 방식으로 구현할 수 있다.

### D. Synchronous/Asynchronous

동기식은 clock과 동기화 되어 clock 신호의 변화 따라서 결과 값이 변한다. 반면, 비동기식은 clock외에도 set이나 reset의 상태의 변화에도 결과 값이 변한다. 일반적으로 synchronous를 선호한다. 왜냐하면 asynchronous circuit은 비교적 복잡하기 때문이다.

## 3. 설계 세부사항

### A. SR Latch

#### i. Truth Table

Input		Output	
S	R	Q	~Q
0	0	Qprev	~Qprev
0	1	0	1

1	0	1	0
1	1	X	X

## B. D Latch

### i. Truth Table

Input		Output	
CLK	D	Q	~Q
0	X	Qprev	~Qprev
1	0	0	1
1	1	1	0

## C. D Flip-Flop

### i. Truth Table

Input		Output	
CLK		Q	
나머지		Qprev	
rising		D	

## D. Enabled D Flip-Flop

### i. Truth Table

Input		Output	
Enable	D	Q	~Q
0	X	Qprev	~Qprev
1	0	0	1
1	1	1	0

## E. Resstable D Flip-Flop

### i. Truth Table

Input		Output	
R	CLK	D	Q
0	X	X	0
1	rising	0	0
1	rising	1	1
1	나머지	X	Qprev

## F. Set/Resstable Flip-Flop

i. Truth Table

Input				Output
R	S	CLK	D	Q
0	X	X	X	0
1	0	X	X	1
1	1	rising	0	0
1	1	rising	1	1
1	1	나머지	X	Qprev

Reset의 우선 순위가 set보다 높기 때문에 reset이 0이면 set의 값에 관계 없이 Q는 0이다.

G. Async/Sync Set/Resetable D Flip-Flop

i. Truth Table

Input				Output	
R	S	CLK	D	Q_sync	Q_async
0	X	X	X	0	0
1	0	X	X	1	1
1	1	rising	0	0	0
1	1	rising	1	1	1
falling	X	X	X	Qprev	0
1	falling	X	X	Qprev	1

Asynchronous의 경우 reset이나 set이 falling 일 때 0 또는 1을 출력한다.

## 4. 설계 검증 및 실험 결과

### A. 시뮬레이션 결과

i. SR Latch



SR Latch는 r과 s의 값에 따라 결과 q를 출력한다. SR Latch가 r과 s의 입력으로 각각 1을 받으면 제대로 동작하지 않는다.

ii. D Latch



D Latch가 clk이 rising 하기 전에 d의 변화에 영향이 없음을 확인할 수 있다. Clk가 0일 때는 이전 q값을 유지한다.

### iii. D Flip-Flop



Clk이 상승 전에 d의 변화에 영향을 받지 않음을 확인하였다. Clk이 rising 할 때의 d값이 q로 출력된다. Clk이 rising 하지 않는 경우에는 이전 d값을 유지한다.

### iv. Enabled D Flip-Flop



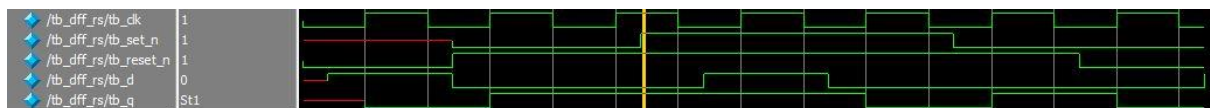
Enable이 0일 때는 q가 이전 값을 유지하지만, 이전 값이 존재하지 않기 때문에 알 수 없는 값이 출력된다. Enable이 1이면, clk의 rising edge에 따라 값이 변한다.

### v. Resettable D Flip-Flop



Reset일 때 0을 결과로 출력한다. Reset이 1이라면 D Flip-Flop과 동작이 같다. Clk 신호가 발생하기 전까지는 알 수 없는 값이 출력되다가 상승 엣지가 발생하자마자 출력을 0으로 내보낸다.

### vi. Set/Resettable D Flip-Flop



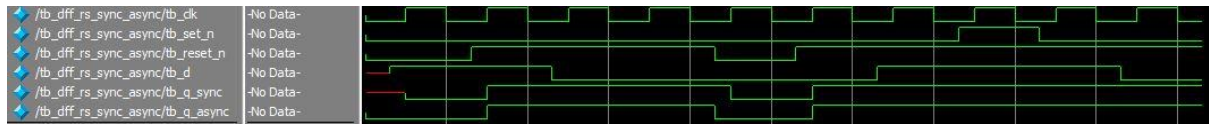
D flip-flop의 결과에 영향을 미치는 변수가 2개이다. Reset이 0이면 set, d, clk에 관계 없이 q는 0이 된다. Clk 신호가 발생 전에는 알 수 없는 값이 출력되다가 상승 엣지가 발생하면 출력을 0으로 내보낸다. Reset이 1, set이 0이면 q는 clk에 관계 없이 1이 된다. set이 1이 된다면 d flip-flop과 같이 동작한다.

### vii. 32-bits Register



Clk이 rising edge일 때 d를 q로 출력함을 확인할 수 있다. Clk의 rising edge일 때 동시에 d가 바뀌면 이전 값이 출력된다. 일반적으로는 clk의 rising과 d의 변화가 동시에 일어나면 바뀐 d값이 반영된다고 한다.

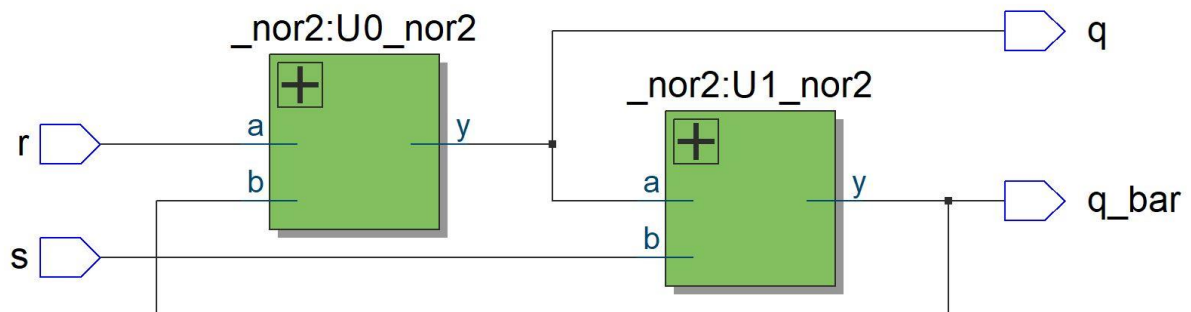
### viii. Asynchronous/Synchronous set/resettable D Flip-Flop



동기식은 clock 상승전까지 값을 알 수 없지만, 비동기식은 Reset의 영향으로 출력이 0이다. 동기식은 clock과 동기화되어 있는 반면, 비동기식은 clock 상승외에 reset과 set에 영향을 받는다.

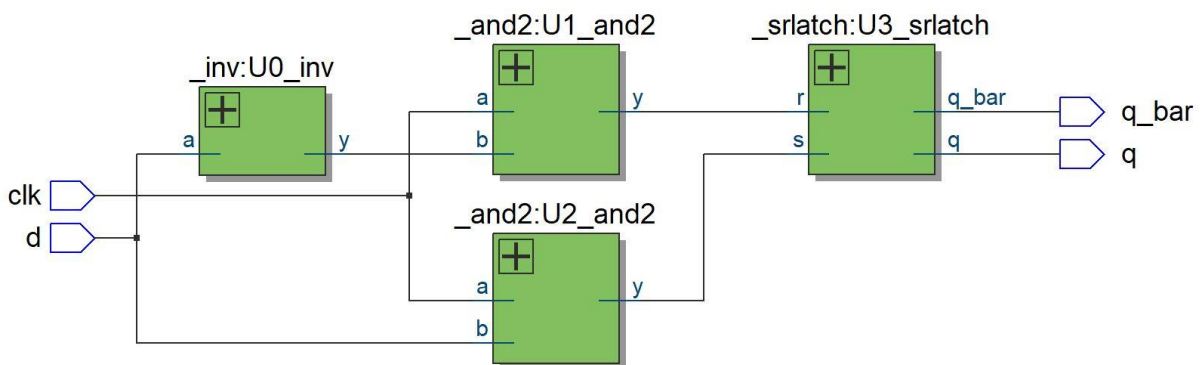
## B. 합성(synthesis) 결과

### i. SR Latch



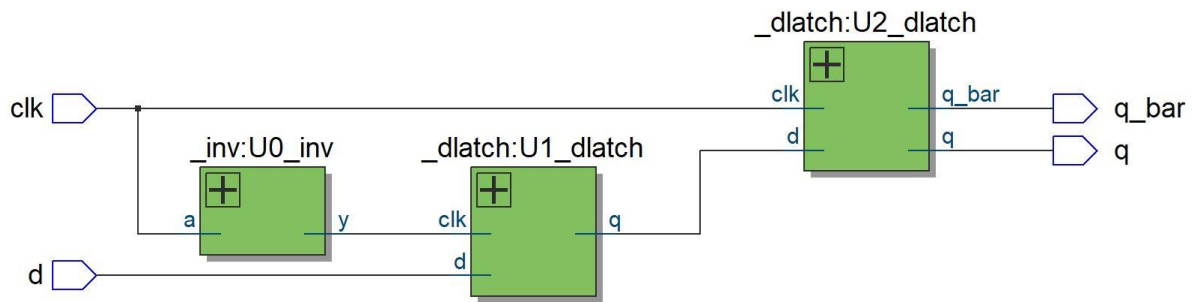
출력을 다시 입력으로 사용하는 형태이다.

### ii. D Latch



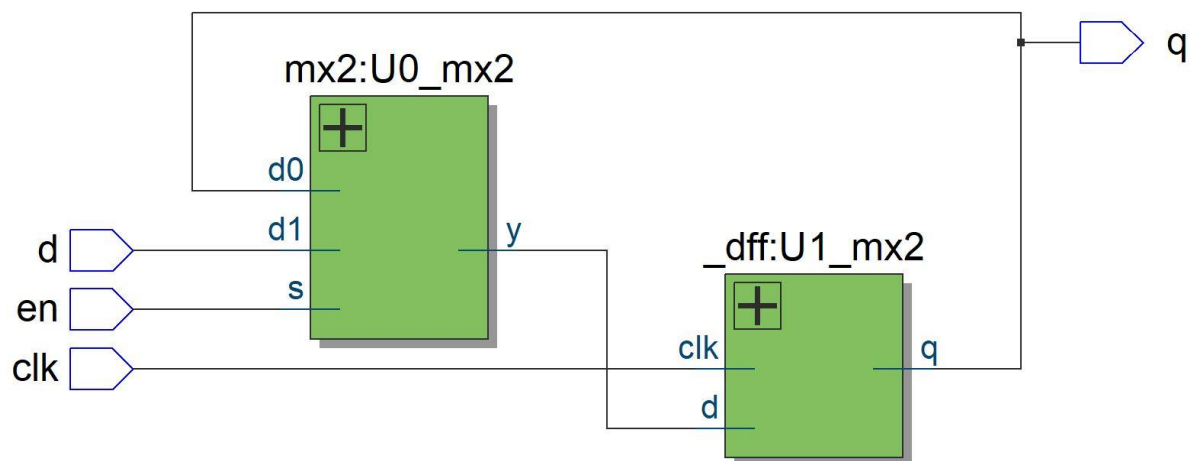
SR Latch에 gate를 추가하여 구현한다.

### iii. D Flip-Flop



D Latch 2개와 inverter를 사용하여 구현한다. Clk가 0이면 d를 다음 D Latch에 보내고 1로 변하는 순간 D가 출력된다.

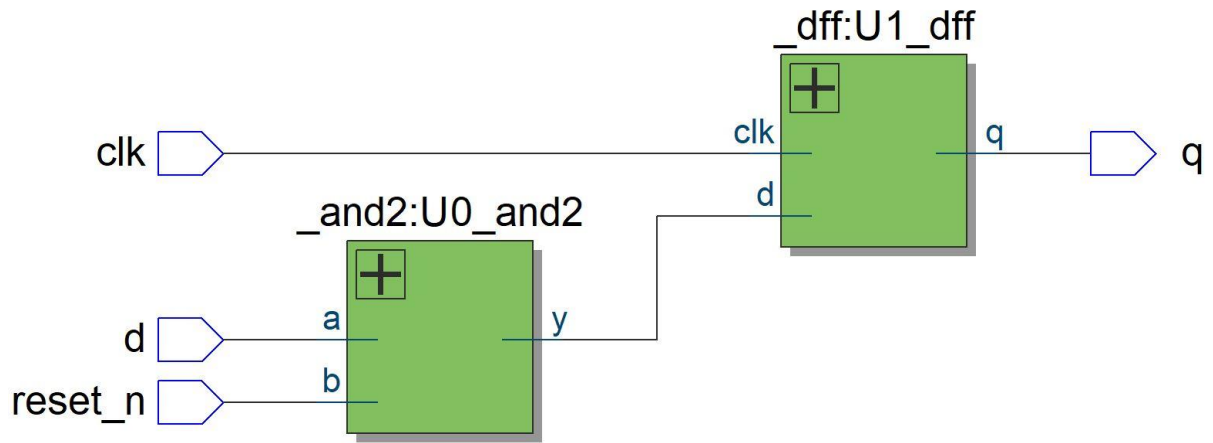
#### iv. Enabled D Flip-Flop



D Flip-Flop에 mux를 추가하여 enable 기능을 구현하였다. Enable이 0이면 이전 q를 입력으로 하고 enable이 1이면 d를 flip-flop의 입력으로 넣는다.

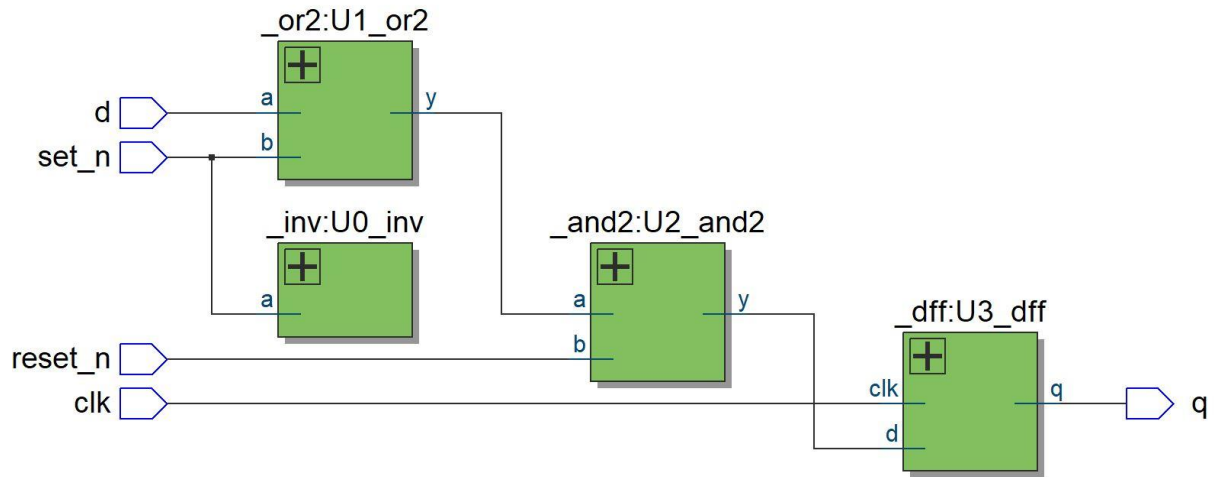
#### v. Resettable D Flip-Flop



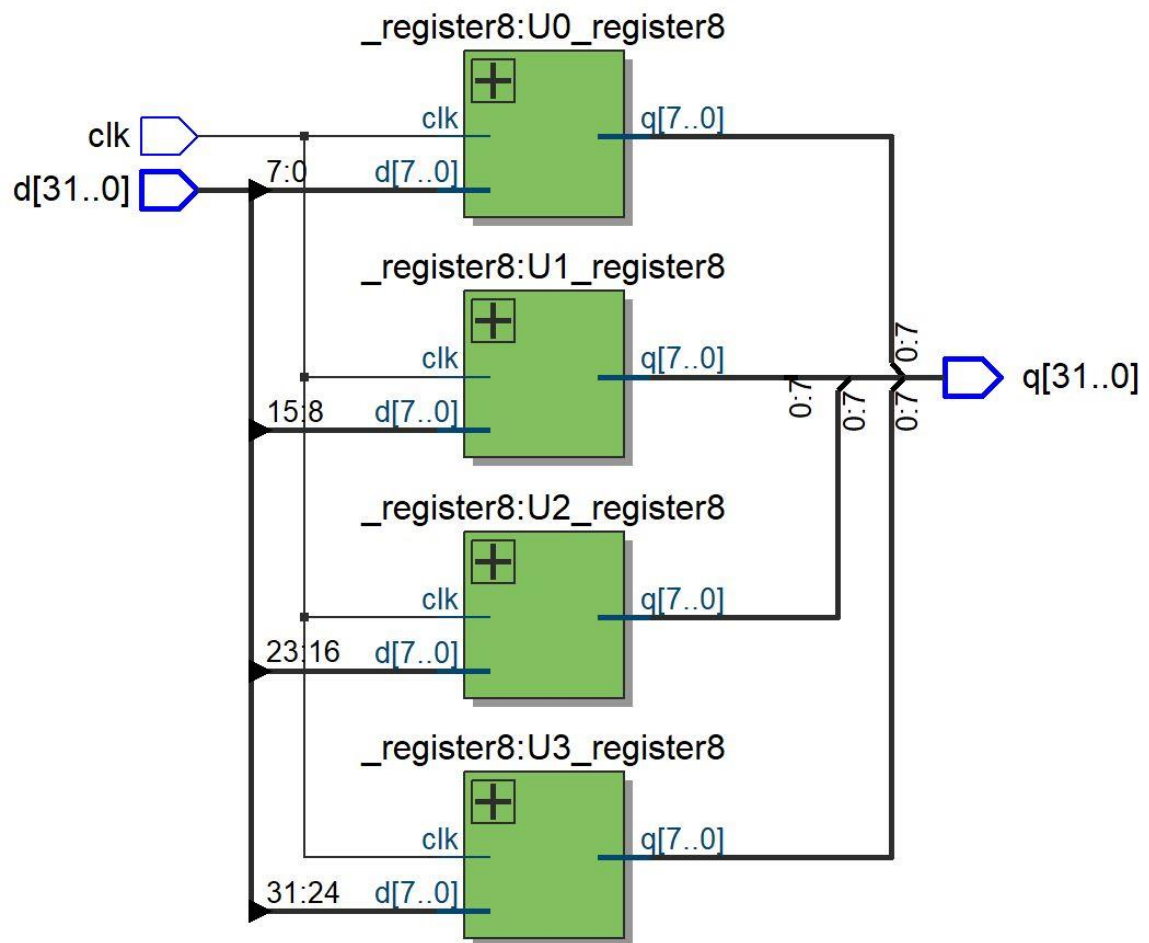


Reset은 active low이므로 1일 때 d 값에 의해 출력이 결정된다.

#### vi. Set/resettable D Flip-Flop

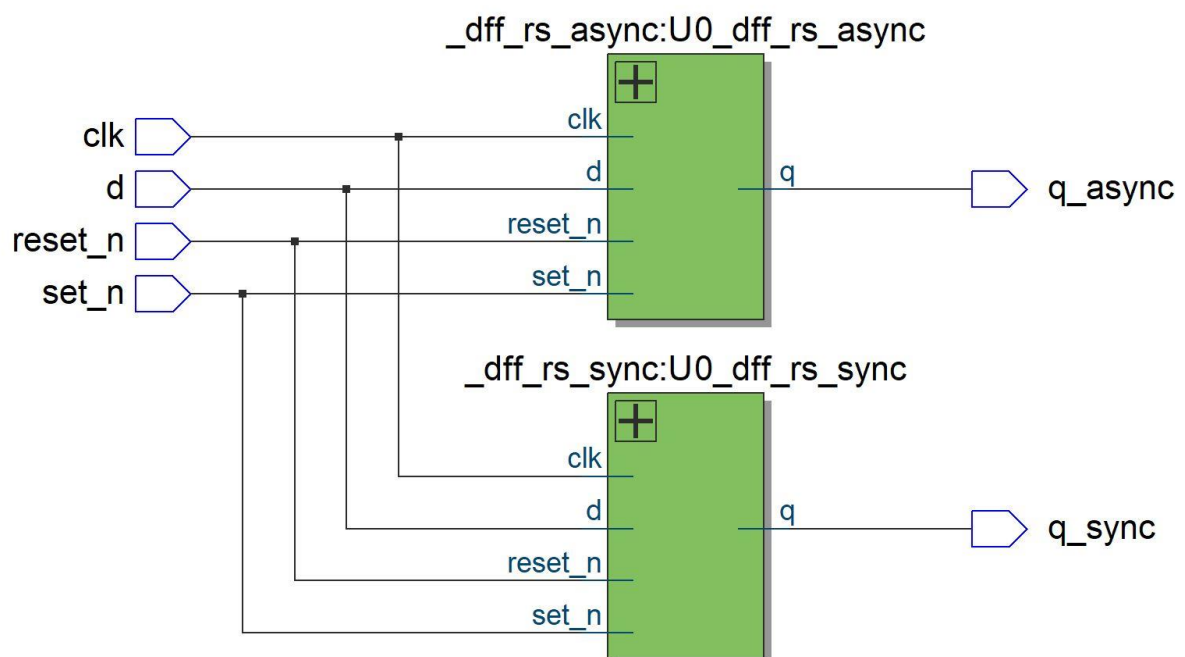


#### vii. 32-bits register



8-bits register 4개를 연결하여 구현하였다. Flip-flop 1개가 1bit을 저장한다.

#### viii. Asynchronous/Synchronous set/resettable D Flip-Flop



위의 블록이 asynchronous를 보여주고 아래 블록이 synchronous를 보여준다.

## 5. 고찰 및 결론

### A. 고찰

D Flip-Flop을 simulation할 때 clock의 rising과 d값이 동시에 변하면 일반적으로 변한 d 값을 출력으로 내보낸다. 그러나, D F/F과 8-bits register을 simulation했을 때, 이전 d 값이 출력되었다. Tool의 기본 설정 또는 다른 컴퓨터 환경 때문에 이러한 차이가 발생하는 것으로 생각된다.

### B. 결론

동기식은 clock의 rising edge 이전에 d, reset, set 값에 영향을 받지 않는다. 그러나, 비동기식은 clock을 포함하여 reset, set이 output q값에 영향을 준다. 예로, reset이 1이고 set이 0이면 clock에 관계없이 1을 출력한다. 둘의 공통점은 clock의 rising edge에서의 d값에 따라 결과가 출력된다.

## 6. 참고문헌

David Money Harris & Sarah L. Harris / Digital Design and Computer Architecture / Elsevier Korea L.L.C / November 2013