

# 컴퓨터 공학 기초 실험2 보고서

실험제목: Latch & flip-flop design

실험일자: 2020년 10월 09일 (금)

제출일자: 2020년 10월 14일 (수)

학 과: 컴퓨터공학과

담당교수: 이준환 교수님

실습분반: 금요일 5,6,7

학 번: 2019202009

성 명: 서여지

## 1. 제목 및 목적

### A. 제목

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

### B. 목적

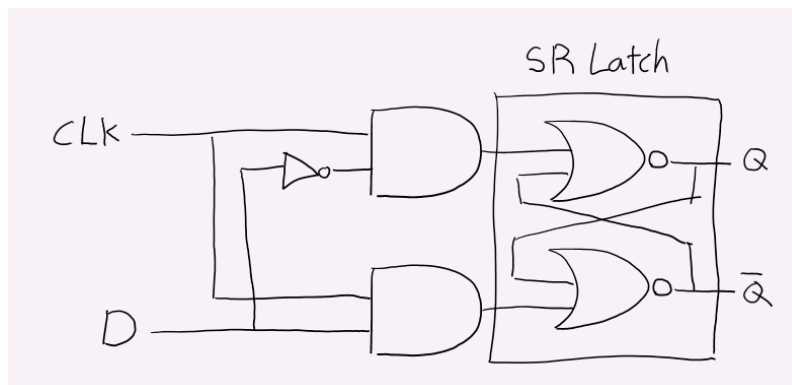
값을 저장할 수 있는 소자인 D latch와 D flip-flop을 structural implementation하고, enable, set/reset 기능을 추가한다. Synchronous 형태로 제작한 flip-flop을 이용하여 32bit register를 구현해본다. Behavioral implementation을 이용하여 Set/Resettable Flip flop을 설계하고, 이를 통해 Synchronous와 Asynchronous의 차이를 확인한다.

## 2. 원리(배경지식)

### (1) Latch & Flip-flop

Latch와 Flip-Flop은 데이터를 저장할 수 있는 소자이다. 가장 단순한 형태의 Latch는 SR latch로, 두 개의 NOR게이트를 이용하여 구현할 수 있다. R과 S가 다른 값을 가질 때, SR Latch는 Q로 S값을 출력하고, Q bar는 R값이 출력된다. 반면 R과 S의 값이 같은 값일 때는 특이한 결과를 확인할 수 있다. 먼저 두 값이 모두 1인 경우, 두 개의 NOR gate의 결과값이 0으로 같아지므로, Q와 Q의 부정이 같은 값을 가지게 되어 모순이 발생한다. 다음으로, 두 값이 모두 0인 경우, 두 개의 NOR gate는 이전 Q와 Q bar의 값을 출력하게 된다.

SR Latch에서 1과 1이 입력되어 모순이 발생하는 단점을 보완하고, CLK를 추가한 것이 D Latch이다. D Latch는 CLK와 D bar를 AND 연산한 결과를 R입력으로, CLK와 D를 and연산한 결과를 s입력으로 하는 SR Latch이다. Clk가 1일 때 R과 S는 서로 부정관계이므로 같은 값이 입력되는 일이 없고, clk가 0인 경우 두 입력이 모두 0이 되어 이전의 Q, Q bar값을 출력하게 된다.

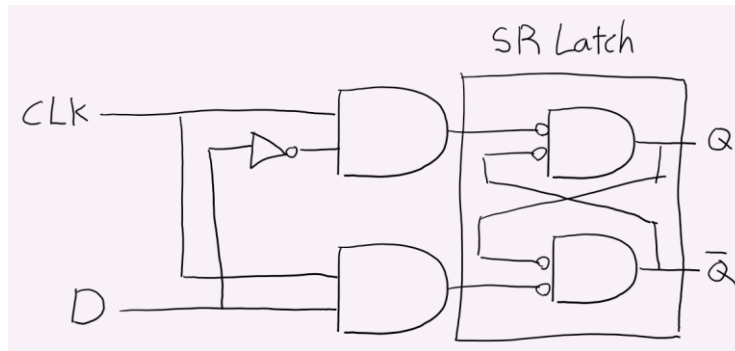


D Latch는 CLK 뿐만 아니라 D의 값이 변화했을 때에도 출력 값이 바뀌게 된다. 이 때문에 회로의 출력 값이 짧은 시간동안 자주 변하는 glitch 현상이 일어날 수 있다. 이것을 방지할 수 있는 회로가 D Flip-flop이다. D flip flop은 2개의 D latch를 연결하여

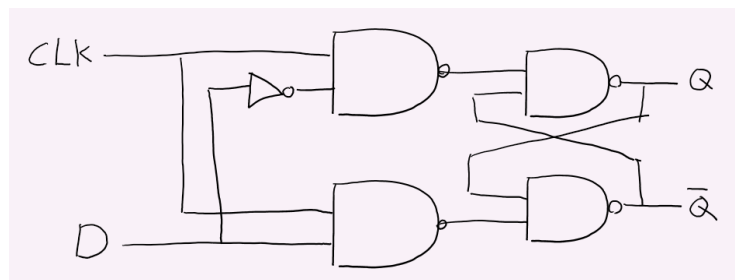
얻는다. 이때 두 Latch의 clk는 서로 반대로 연결한다. 따라서 D Flip Flop은 CLK의 값이 1이 되는 순간 출력Q와 Q bar를 update할 수 있다.

(2) NAND gate 4개를 이용하여 D latch를 구현하는 방법

위에서 살펴본 D latch는 2개의 NOR gate와 2개의 AND gate를 이용하였다. 이것을 4개의 NAND gate를 이용하는 것으로 바꾸면 다음과 같다.

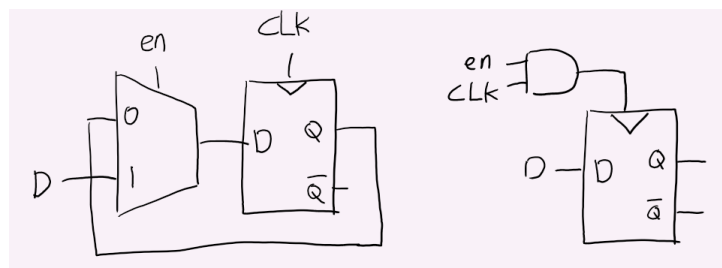


NOR gate를 드모르간 정리를 이용하여 변형한다.



(3) 다른 방법으로 enabled D flip – flop을 구현하는 방법

실습을 진행하며 작성한 enabled D flip flop의 구조는 2 to 1 MUX를 이용하여 enable신호를 이용하여 Data와 Q prev 값 중 하나를 선택하는 구조이다. 다른 방법으로는 enable신호를 CLK와 and 연산한 값을 D Flip-flop에 clock신호로 전달하는 것이다. 이 경우 enable이 0인 경우 Q prev가, 1인 경우 data가 동일하게 출력된다.



(4) Reset & Set

Reset signal은 Flip-flop의 출력을 0으로 초기화하는 신호이고, set은 출력을 1로 지정하는 신호이다. 이번 실습에서 reset과 set은 active low이므로 각각 신호가 0일 때 해당 동작이 실행된다. Reset과 set중 하나의 신호만 0일 때, 해당 동작을 실행하고, 두 신호가 모두 0인 경우 Q prev, 모두 1인 경우 d의 값을 출력한다.

### (5) Synchronous & Asynchronous

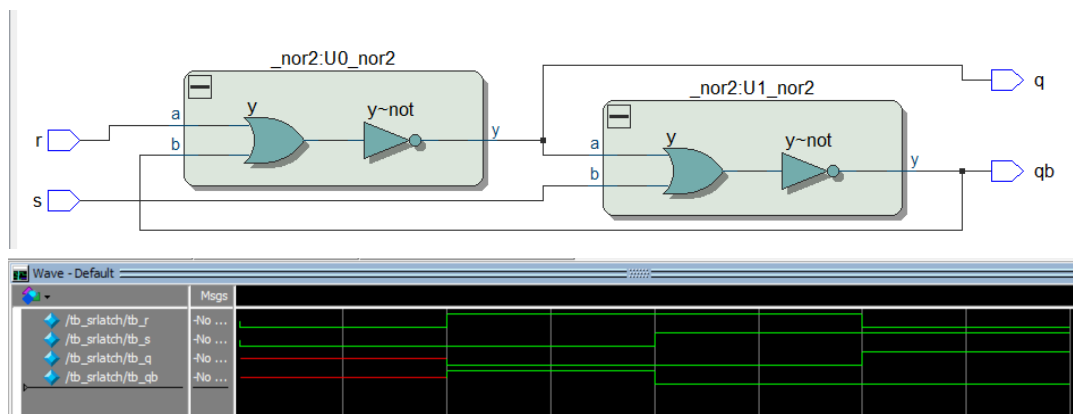
Synchronous는 회로가 clk에 동기화 되어있음을 의미한다. Clk의 rising edge에서 출력 값을 update하고, 그 외의 경우에는 값이 변경되지 않는다. D Flip-flop은 해당 조건을 만족하기 때문에 동기회로이다. Asynchronous는 반대로 Latch의 경우이다. Latch는 data값이 변화하면 그 즉시 출력결과가 변한다. 이것은 clk에 동기화 되어있지 않다.

## 3. 설계 세부사항

### 1) SR Latch

2개의 NOR gate를 이용하여 SR Latch를 설계하였다. Reset과 set의 값이 다르고, Reset의 값이 1일 때 Q가 0으로, set이 1일 때 Q가 1로 출력되는 것을 확인할 수 있다.

S	R	Q	Q bar
0	0	Q prev	Q bar prev
0	1	0	1
1	0	1	0
1	1	0	0



### 2) D Latch

CLK와 D를 AND연산한 결과를 SR Latch의 S입력에, CLK와 D bar를 AND연산한 결과를 R에 연결하여 D Latch를 structural implementation 한다. D Latch는 clk가 1인 동안 data가 변화하면 그에 따른 출력의 변화가 바로 나타난다.

### 3) D Flip-flop

D Latch 2개를 반대되는 CLK에 연결하여 D Flip-flop을 설계한다. 앞의 D Latch에서 출력된 Q의 값을 다음 D Latch의 D 입력으로 연결한다. D Flip-flop은 clk가 0에서 1로 변화하는 rising edge에서 출력이 변화한다.

### 4) Enabled D Flip flop

Enable의 값에 따라 d와 q의 값 중 하나를 선택하게 되는 2 to 1 MUX를 이용하여 enabled d flip-flop을 설계한다. En이 1일 때 D의 값을 D Flip-flop의 d input으로 연결

하고, en이 0일 때 d flip-flop의 이전 출력인 q를 d의 입력으로 연결한다. Enable이 1일 때, 기존의 D Flip-flop과 동일하게 작동하고, 0일 때는 이전의 출력을 그대로 유지한다.

5) Resettable D Flip flop

reset기능이 추가된 d flip-flop을 설계한다. 이 실험에서 reset은 active low이므로, reset이 0일 때 동작한다. D와 reset을 AND 연산한 결과를 D Flip-flop의 d 입력으로 연결한다. Reset이 0일 때 d의 값에 상관없이 AND의 결과가 0이므로, D Flip-flop의 결과가 0이 되고, reset이 1일 때 d의 값이 d input으로 들어가 기존의 d flip flop과 동일하게 작동한다.

6) Synchronous Set/Resettable D Flip Flop (active-low)

reset기능과 함께 set기능을 추가한 d flip flop을 설계한다. Reset과 마찬가지로 set signal은 active low이다 따라서 0일 때 D Flip-flop의 결과를 1로 만들고, 1일 때 기존의 D Flip-flop과 동일하게 동작하도록 한다. 5)에서 설계한 resettable d flip flop의 d input에 set bar와 d의 OR연산 결과를 연결한다. Set이 0일 때 D의 값에 상관없이 OR 연산의 결과가 1이 되어 D Flip-flop의 결과를 1로 지정하고, set이 1이면 d의 값에 따라 기존 d flip-flop과 동일하게 작동한다. Set과 reset이 모두 1인 경우 두 동작이 모두 실행되지 않기 때문에 일반적인 d flip flop과 동일하게 동작한다. 한편 set과 reset이 모두 0인 경우에는 reset을 고려하는 회로가 더 나중에 계산되기 때문에 reset의 실행결과로 0이 출력된다.

7) 32bit register

D Flip-flop을 동일한 clk를 이용하여 연결하여 32bit register를 얻는다. 이 실습에서는 8개의 D Flip-flop을 연결한 8bit register를 만든 후, 다시 4번 instance하여 32bit register를 설계하였다. 각 Flip-flop의 동작은 동일하고, 총 32개의 D Flip-flop이 32bit의 2진수 값을 저장한다.

8) Synchronous/Asynchronous Set/Resettable D Flip Flop (active-low)

Behavioral implementation을 통해 생성한 Synchronous Set/Resettable D Flip Flop과 Asynchronous Set/Resettable D Flip Flop은 전체 구조가 유사하지만 always문에서 차이를 보인다. \_dff\_rs\_async모듈은 \_dff\_rs\_sync모듈과 달리 always문의 sensitivity list에 negedge set\_n과 negedge reset\_n이 추가된 것을 확인할 수 있다. Set과 reset signal이 변화할 때 asynchronous의 경우 출력 값을 변화시킨다는 뜻이고, Active-low기 때문에 negedge 조건으로 추가되었다. 또한 if문에서 reset signal의 조건을 먼저 확인하여 set과 reset이 모두 0인 경우에도 기존에 작성한 structural implementation의 경우와 동일한 결과를 얻을 수 있도록 작성된 것을 볼 수 있다.

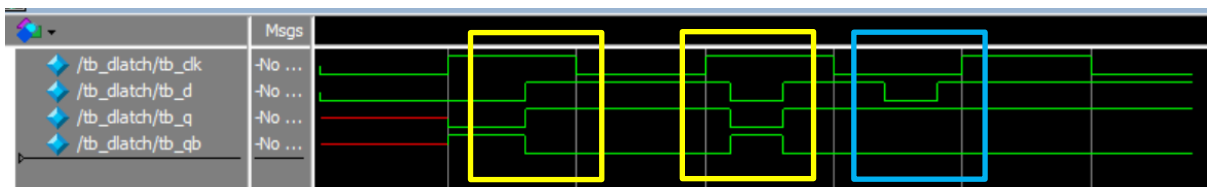
#### 4. 설계 검증 및 실험 결과

##### A. 시뮬레이션 결과

###### 1) D Latch

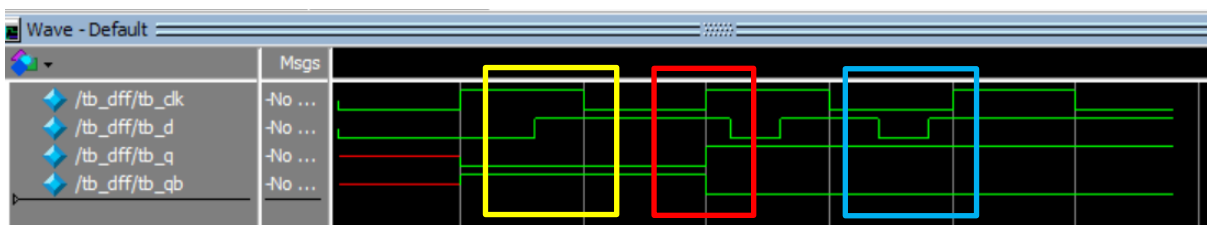
Clk는 주기적으로 전환되고, d의 값을 변경하여 testbench를 작성하였다. 해당 testbench는 강의자료에 주어진 timing diagram을 참고하여 작성하였다.

노란색으로 체크한 부분에서 clk가 1일 때 d의 값이 변경되면 곧바로 결과값이 변하는 것을 확인할 수 있고, 파란색으로 강조한 부분에서는 clk가 0일 때 d의 값이 변하더라도 결과가 바뀌지 않는 것을 볼 수 있다.



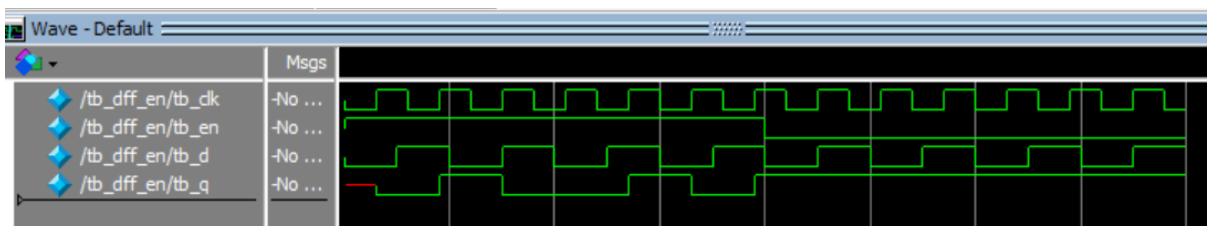
###### 2) D Flip flop

위의 1) D Latch와 동일한 testbench를 이용하여 실험을 진행하였다. 노란색으로 강조한 부분에서 D Latch와 다르게 d의 값이 변경되어도 결과에 바로 반영되지 않은 것이 확인된다. 이후 clk의 rising edge에서 출력이 update됨을 빨간색 사각형에서 확인할 수 있다. Clk가 0인 경우 d가 변해도 출력이 변하지 않는다.



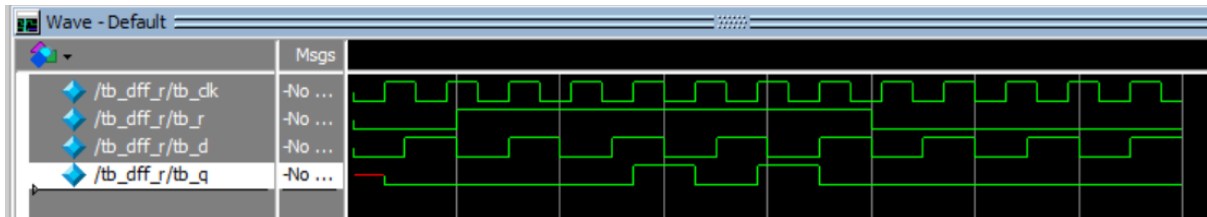
###### 3) Enabled D Flip flop

Clk와 d가 주기적으로 변화하고, en의 값을 1에서 0으로 한 번 변경하였다. En이 1인 동안 clk의 rising edge에서 d값이 q로 출력되는 것을 확인할 수 있고, en이 0일 때 출력 결과는 직전 출력인 1에서 변화하지 않는 것을 볼 수 있다.



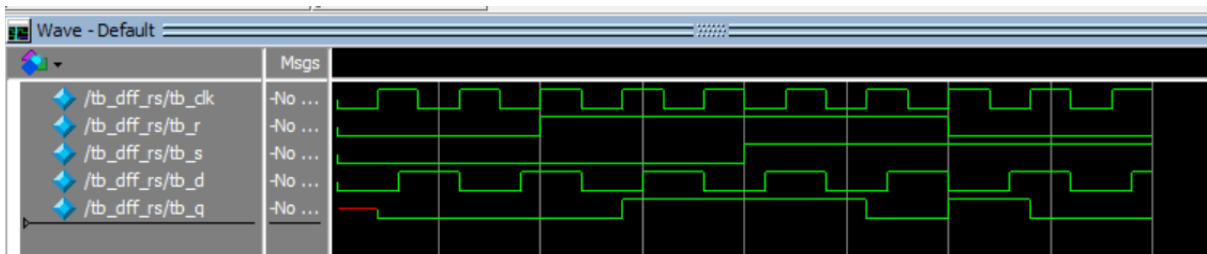
#### 4) Resettable D Flip flop(active-low)

Clk와 d가 주기적으로 변화하고, r의 값을 변경하였다. R이 0인 구간에서 동작하여 출력이 모두 0이 된 것을 확인할 수 있다. R이 1인 구간에서는 clk의 rising edge에서의 d값이 q로 출력된다. R의 값이 1이 된 후의 첫 번째 clk주기에서는 clk와 d가 동시에 1로 변하여 delay에 의해 d flip-flop의 출력이 변화하지 않은 것을 볼 수 있다.



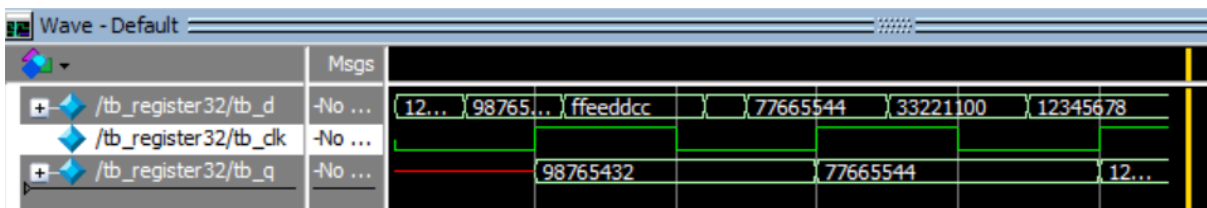
#### 5) Synchronous Set/Resettable D Flip Flop (active-low)

Clk와 d가 주기적으로 변하고, r과 s의 모든 경우의 수에 대해 testbench를 작성하였다. R과 s가 모두 0인 경우 Set/Resettable D Flip Flop의 구조상 더 늦게 계산되는 reset이 동작하여 출력결과 q가 0이 되었다. S의 값만 0인 경우 동기 회로이기 때문에 다음 clk의 rising edge에서 q가 1로 지정되었고, r과 s가 모두 1인 경우 일반적인 d flip-flop과 동일하게 동작한다. 마지막으로 r의 값만 0인 경우 q가 0으로 변화하였다. 이 경우 r이 0이 되는 동시에 clk가 1로 변화했기 때문에 delay에 의하여 첫 번째 주기에서 값이 바로 0으로 지정되지 않았다.



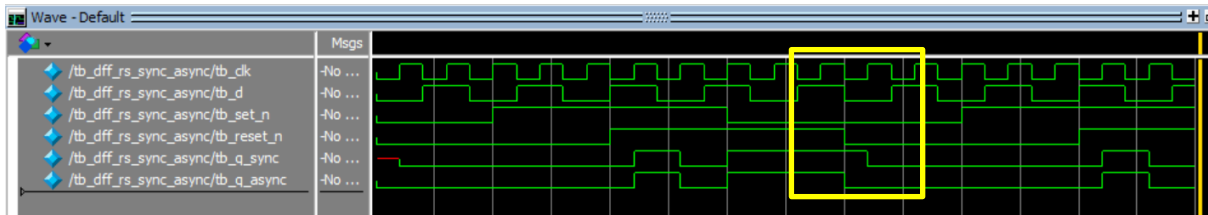
#### 6) 32bit register

Testbench는 강의자료 내용을 이용하여 작성하였다. Clk의 rising edge에서 d의 값이 q로 출력되는 것을 확인할 수 있다.



## 7) Synchronous/Asynchronous Set/Resettable D Flip Flop (active-low)

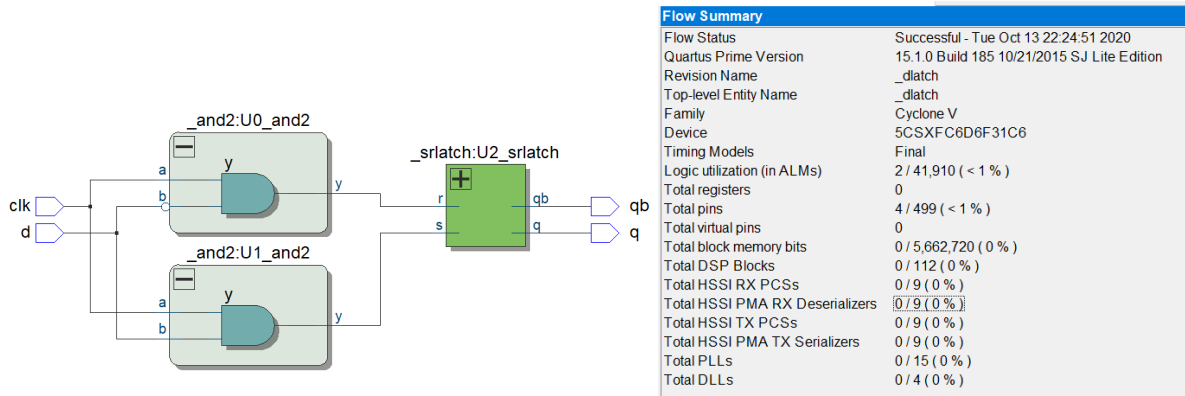
Clk와 d가 주기적으로 변하고, set\_n과 reset\_n의 모든 경우에 대해 testbench를 작성하였다. 노란색으로 강조한 부분에서 Synchronous와 asynchronous의 차이가 나타나는 것을 볼 수 있다. Reset\_n의 값이 1에서 0으로 바뀌어 reset기능이 실행되는 경우, synchronous회로는 그 다음 clk의 rising edge에서 출력값이 변화한 반면, asynchronous회로는 reset\_n의 값이 바뀌는 즉시 바뀐 결과값을 출력하였다.



## B. 합성(synthesis) 결과

### 1) D Latch

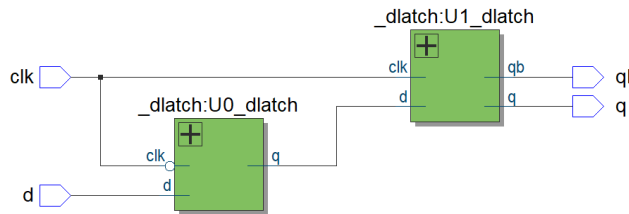
SR latch에 clk, d를 이용한 두 개의 and gate를 이용하여 d Latch를 설계하였다. input으로 clk, d 2개의 pin을 사용하였고, output으로 qb, q 2개의 pin을 사용하여 총 4개의 pin을 사용한 것을 flow summary를 통해 확인할 수 있다.



### 2) D Flip flop

2개의 D latch에 clk와 clk bar를 연결하여 D Flip-flop을 설계하였다. input으로 clk, d 2개의 pin을 사용하였고, output으로 qb, q 2개의 pin을 사용하여 총 4개의 pin을 사용한 것을 flow summary를 통해 확인할 수 있다.

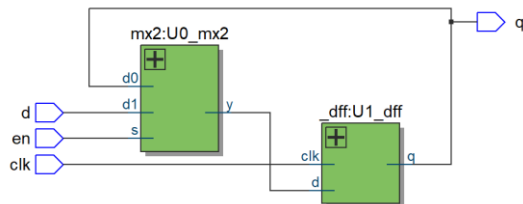




Flow Summary	
Flow Status	Successful - Tue Oct 13 22:21:55 2020
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	_dff
Top-level Entity Name	_dff
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	2 / 41,910 (< 1 %)
Total registers	0
Total pins	4 / 499 (< 1 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

### 3) Enabled D Flip flop

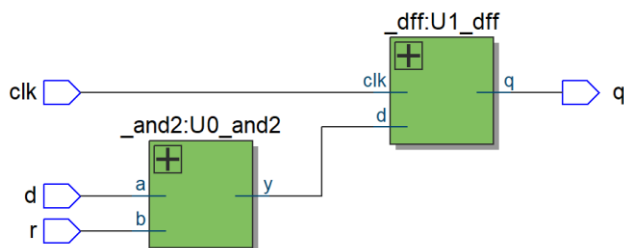
D flip flop에 en을 이용한 MUX를 추가하여 Enabled d flip-flop을 구현하였다. Input으로 clk, d, en의 3개의 pin을 사용하였고, output으로 q를 사용하여 총 4개의 pin을 사용한 것을 확인할 수 있다.



Flow Summary	
Flow Status	Successful - Tue Oct 13 22:27:54 2020
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	dff_en
Top-level Entity Name	_dff_en
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	2 / 41,910 (< 1 %)
Total registers	0
Total pins	4 / 499 (< 1 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

### 4) Resettable D Flip flop

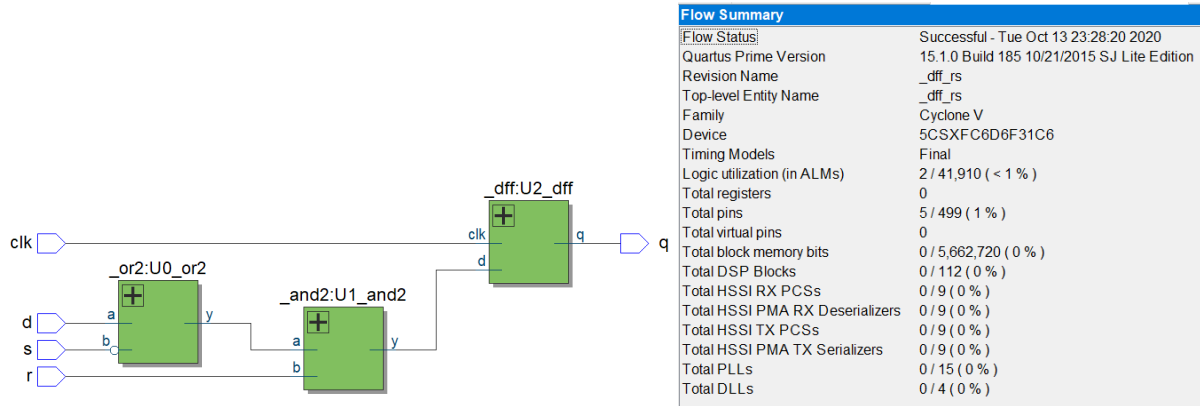
D flip flop에 r을 이용한 and gate를 추가하여 resettable d flip-flop을 설계하였다. Input으로 clk, d, r을 사용하였고, output으로 q를 사용하여 총 4개의 pin을 사용한 것을 확인할 수 있다.



Flow Summary	
Flow Status	Successful - Tue Oct 13 23:34:29 2020
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	_dff_r
Top-level Entity Name	_dff_r
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	2 / 41,910 (< 1 %)
Total registers	0
Total pins	4 / 499 (< 1 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

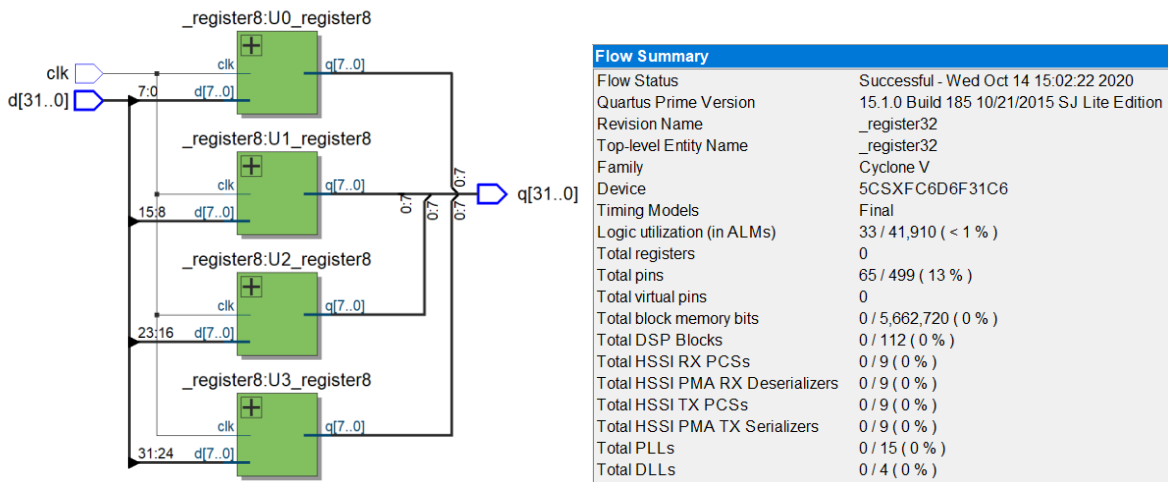
##### 5) Synchronous Set/Resettable D Flip Flop (active-low)

Resettable d Flip flop에 set을 추가하여 Synchronous Set/Resettable D Flip Flop을 구현하였다. Input인 clk, d, s, r가 4개의 pin을 차지하고, output으로 q를 사용하여 총 5개의 pin을 사용한 것을 확인할 수 있다.



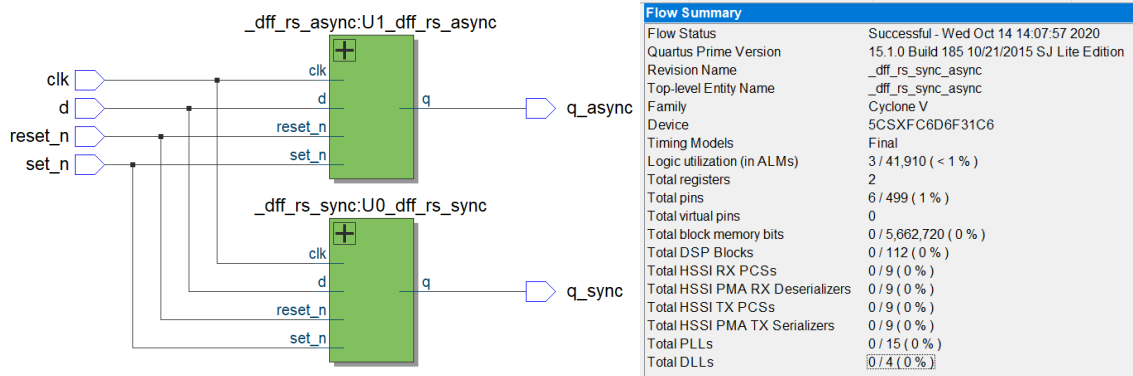
##### 6) 32bit register

하나의 clk로 8개의 D flip-flop을 연결한 \_register8 module을 4개 instance 하여 32bit register를 구현하였다. Input과 output으로 각각 32개의 pin을 사용하고, clk가 하나의 pin을 차지하여 총 65개의 pin이 사용되었다.



##### 7) Synchronous/Asynchronous Set/Resettable D Flip Flop (active-low)

Synchronous와 Asynchronous한 Set/resettable d flip-flop을 behavior 설계 하여 하나의 module에 instance하였다. Input인 Clk, d, reset\_n, set\_n이 4개, output인 q\_async, q\_sync가 2개의 pin을 사용하여 총 6개의 pin이 사용되었다.



## 5. 고찰 및 결론

### A. 고찰

맨 처음 SR Latch를 제작하는 과정에서 두 개의 NOR gate가 각각의 output값인 `q`와 `q_bar`를 input으로 받는데, 이것을 작성하는 것이 혼란스러웠다. 그러나 구조에서 나타나는 그대로 output을 input에 연결하는 것으로 코드를 작성할 수 있었다. Resettable d flip-flop에서 reset의 값이 active-low임을 간과하여 강의자료의 결과를 해석하는 것에서 어려움을 겪었다. Active-low는 reset의 값이 0일 때 reset동작을 수행함을 확인하고, 정상적으로 과제를 진행할 수 있었다.

### B. 결론

Synchronous circuit에 사용되는 기본적인 소자인 D Flip-flop을 직접 설계하며 다양한 종류의 d flip-flop을 접하게 되었다. Enable, reset, set/reset 이 각 기능을 추가하며 flip flop의 구조에 변경된 부분을 확인하고, 그 원리를 이해할 수 있었다. D latch가 asynchronous하고, d flip-flop이 synchronous함을 알고 있었지만, d flip-flop의 set/reset 신호에 따라 다시 asynchronous와 synchronous 한 구조로 나뉠 수 있음을 새로 알게되었다.

### C. D Flip Flop with active-low synchronous reset and set 과 D Flip-Flop with active-low asynchronous reset and set의 차이 설명하기

Testbench를 이용한 실험 결과에서 드러나듯이, reset 과 set의 값이 변화하였을 때 변화에 따른 D Flip-flop의 출력이 즉시 반응하면 asynchronous한 회로이고, 그 다음 CLK의 rising edge에서 set, reset상태에 대하여 출력이 반응하면 synchronous한 회로이다. D Flip-Flop with active-low asynchronous reset and set의 경우 Clk와 data값의 변화에 의한 `q`의 값은 clk의 rising edge에서 update되지만, reset, set신호에 의한 값 변화는 즉시 변경되는 것이다.

## 6. 참고문헌

이준환교수님, 디지털논리회로2 강의자료, 광운대학교 컴퓨터정보공학과,2020

이준환교수님, 컴퓨터공학기초실험2 강의자료, 광운대학교 컴퓨터정보공학과,2020

David Money Harris 외 1인, Digital Design and Computer Architecture, Elsevier

SIMPLIS, D-Type Flip-Flop with Set/Reset

[https://www.simplistechnologies.com/documentation/simplis/dp\\_flip\\_flops/topics/dtypeflip\\_flopwsetrst.htm](https://www.simplistechnologies.com/documentation/simplis/dp_flip_flops/topics/dtypeflip_flopwsetrst.htm)