

# 컴퓨터 공학 기초 실험2 보고서

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

실험일자: 2023년 10월 09일 (월)

제출일자: 2023년 10월 16일 (월)

학 과: 컴퓨터정보공학부

담당교수: 이준환 교수님

실습분반: 월요일 0, 1, 2

학 번: 2020202031

성 명: 김재현

## 1. 제목 및 목적

### A. 제목

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

### B. 목적

D latch는 clock이 활성화 상태를 유지하는 동안 입력 D값의 변화를 출력하는 구현을 한다. D flip-flop은 clock의 rising edge나 falling edge에서만 D값으로 출력이 바뀌게 되는데, 이 실험에서는 rising edge를 사용하도록 구현하는 것을 목적으로 한다.

## 2. 원리(배경지식)

### 1) D-latch

Input		Output	
CLK		Q	
0		이전 Q	
1		D	

D-latch는 clk가 0일 땐, D 값의 변화에 반응하지 않고 Q값을 동일하게 유지합니다. 반면, clk가 1일 땐, Q가 D의 값을 따라가게 됩니다.

### 2) D-flipflop

Input		Output	
CLK		Q	
↑		D	
other case		이전 Q	

D-flipflop은 CLK가 rising edge를 가질 때 D의 값을 따라가고, 그 외의 경우엔 Q값을 동일하게 유지하게 됩니다.

### 3) Resettable D-flipflop

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

Resettable D-flipflop은 D-flipflop에 reset 단자를 추가해준 것입니다. 이 때, R은 active low이므로 R이 1일 때는 기존의 D-flipflop과 동일하지만, R이 0일 때는, D, CLK 값과 상관없이 Q가 0으로 초기화됩니다.

4) Enabled D-flipflop

Enabled D flipflop은 D-flipflop에 EN 단자를 하나 추가해준 것입니다. EN신호가 0이면 D 값의 변화와 상관없이 D-flipflop으로의 입력 값으로 이전 Q값을 전달하기 때문에, 이전 Q값이 출력됩니다. 반면, EN신호가 1이면 일반 D-flipflop과 동일하게 작동합니다.

5) 32-bits register

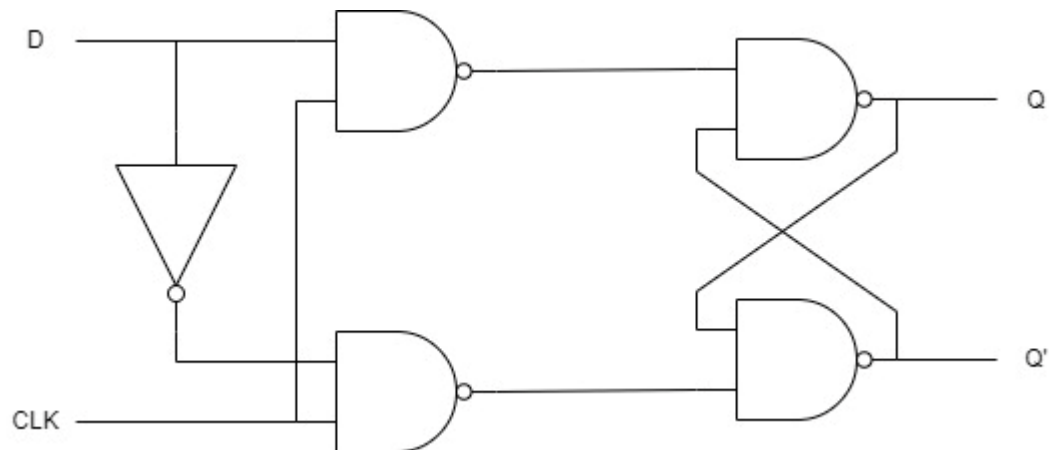
앞서 구현한 D-flipflop은 1비트짜리이다. 32-bits register를 처리하기 위해서 32개의 D-flipflop을 한 줄로 늘어놓음으로써 각 bit를 각각 D-flipflop에 입력되도록 설계했습니다.

6) Async/Sync Set/Resettable D-flipflop

Synchronous Reset 단자를 가진 Resettable D-flipflop의 경우 Reset이 CLK와 동기화되어 같은 주기를 가지기 때문에, Reset이 0이 된 후, CLK이 rising되어야 Q값이 0으로 초기화됩니다.

하지만 Asynchronous Reset 단자를 가진 Resettable D-flipflop의 경우 Reset이 CLK와 비동기화기 때문에, Reset이 0이 되면, CLK의 rising 여부와 상관 없이 Q값이 0으로 초기화됩니다.

7) nand gate 4개를 이용하여 D latch를 구현하는 방법



위의 논리회로도처럼 구현하면 D-latch가 된다.

8) Enabled D-flipflop 다른 방법으로 구현하는 방법

clk와 en 값을 and gate에 넣어주고, 그 결과 값을 D-flipflop의 clk 값으로 입력해주는 방법도 존재합니다. 하지만 이 방법은, clk가 and gate를 통과하면서 delay가 생기므로

타이밍 오류의 원인이 될 수 있습니다.

### 3. 설계 세부사항

#### 1) D latch

D latch의 경우 clk가 1일 때 q를 d로 초기화 해주는 회로입니다. 따라서 clk와 d 값에 변화가 있을 때마다 clk가 1인지를 체크하여 1이라면 q의 값에 d를 대입해주도록 설계했습니다.

#### 2) D flip-flop

D flip-flop의 경우 clk가 rising edge를 가질 때 q를 d로 초기화 해주는 회로입니다. 따라서 D latch를 두개 이어 붙여 이전 D latch의 출력 값을 다음 D latch의 입력 값으로 연결하되, clk을 서로 반대로 주어, 오로지 clk가 rising edge일 때만 q값이 변화하도록 설계했습니다.

#### 3) Enabled D flip-flop

Enabled D flip-flop의 경우 en이 1일 때 기존의 D flip-flop처럼 작동하고, en이 0일 때 이전 q값을 출력합니다. 따라서 2to1 mux를 통해 en의 값에 따라 D flip-flop의 입력 값으로 이전 q 값을 그대로 부여할지, d 값을 부여할지를 정할 수 있도록 구현했습니다.

#### 4) Resettable D flip-flop

Resettable D flip-flop의 경우 reset\_n이 0일 때, D flip-flop의 입력을 0으로 초기화해주는 회로입니다. 따라서 입력 값 d와 reset\_n을 and gate에 입력하고 그 출력 값을 dff에 입력해줌으로써 reset\_n이 0일 때 dff의 입력 값이 0이 되게 구현했습니다.

#### 5) 32 bits register

각 비트가 하나의 clk로 동기화 돼있는 D flip-flop 32개에 각각 입력되도록 구현했습니다.

#### 6) Async/sync resettable D flip-flop

sync resettable D flip-flop은 reset이 clk에 동기화되어 작동하므로, clk의 rising edge에서 reset\_n이 0인지를 판단하여 출력 값을 0으로 초기화하도록 구현했습니다.

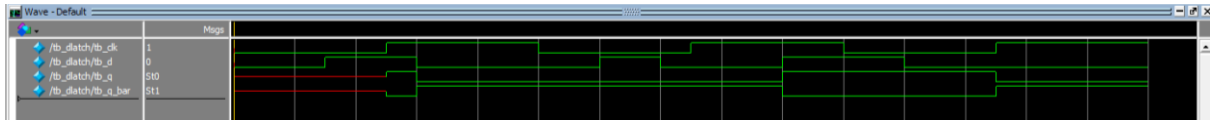
async resettable D flip-flop은 reset이 clk 값과 관계없이 작동하므로, clk의 rising edge와 더불어 reset\_n의 falling edge에서 reset\_n이 0인지를 판단하여 출력 값을 0으로

초기화하도록 구현했습니다.

#### 4. 설계 검증 및 실험 결과

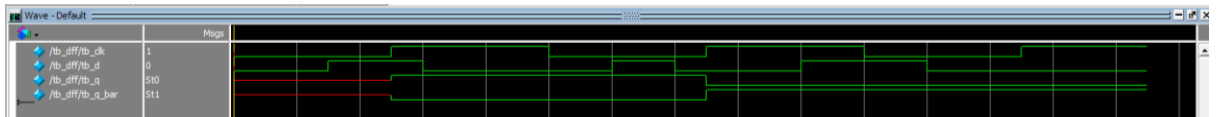
##### A. 시뮬레이션 결과

###### 1) D latch



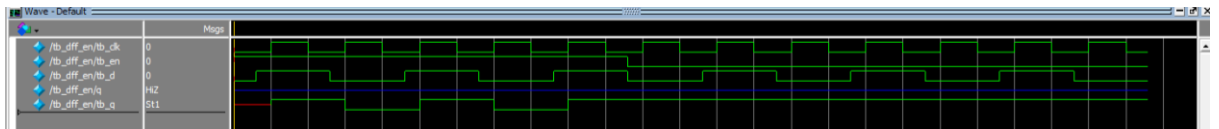
tb\_d latch의 테스트벤치 결과화면이다. clk가 1일 때, d의 값의 변화에 따라 출력 값이 변화하는 것을 확인할 수 있다.

###### 2) D flip-flop



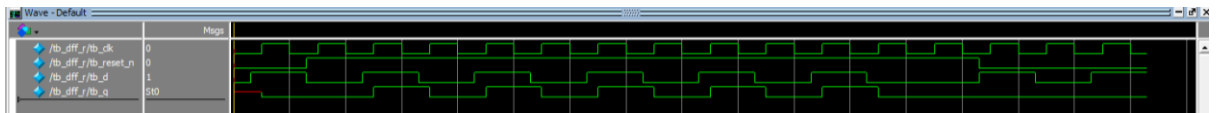
tb\_dff의 테스트벤치 결과화면이다. clk가 rising edge를 가질 때, d의 값에 따라 출력 값이 변화하는 것을 확인할 수 있다.

###### 3) Enabled D flip-flop



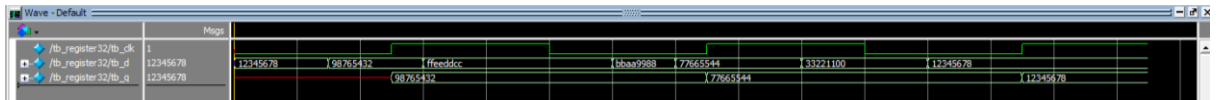
tb\_dff\_en의 테스트벤치 결과화면이다. en이 1일 때는 기존의 dff과 동일하게 작동하지만 en이 0일 때는, 이전 출력 값을 계속 유지하는 것을 확인할 수 있다.

###### 4) resettable D flip-flop



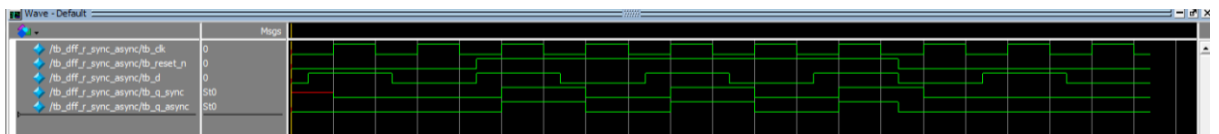
tb\_dff\_r의 테스트벤치 결과화면이다. reset\_n이 0이면 d가 0으로 초기화되고 clk가 rising edge를 가질 때, d의 값이 출력되는 것을 확인할 수 있다. reset 단자가 독립적으로 출력 값을 0으로 초기화시키지 않고 clk과 동기화 돼있는 것으로 보아 synchronous resettable D flip-flop이라고 할 수 있다.

###### 5) 32 bits register



tb\_register32의 테스트벤치 결과화면이다. clk가 rising edge를 가질 때, register로 입력되고 있는 값이 출력되는 것을 확인할 수 있다.

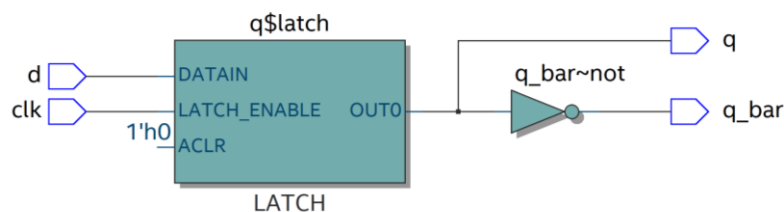
#### 6) async/sync resettable D flip-flop



tb\_dff\_r\_sync\_async의 테스트벤치 결과화면이다. 8번째 클럭 주기에서 reset\_n이 0이 될 때, synchronous dff은 clk가 rising edge일 때 출력이 0으로 초기화 되지만, asynchronous dff은 clk과 관계없이 바로 출력이 0으로 초기화 되는 것을 확인할 수 있습니다.

## B. 합성(synthesis) 결과

### 1) D latch

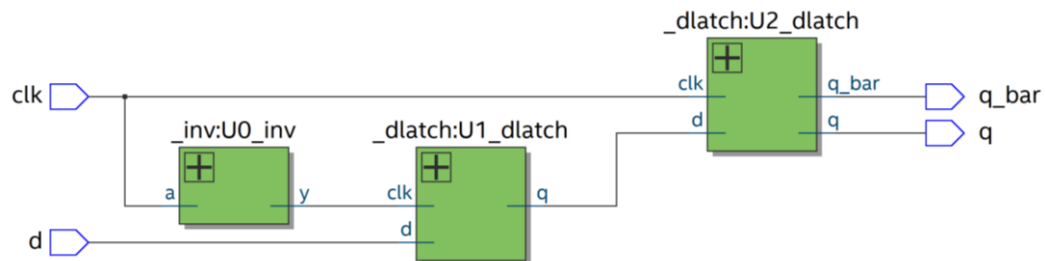


\_dlatch의 RTL Viewer 화면이다. 입력 값으로 d와 clk를 받고, clk가 1일 때 d의 값을 q로, q\_bar에는 ~q 값을 할당한다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sat Oct 14 17:03:12 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	_dlatch
Top-level Entity Name	_dlatch
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	1 / 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 %)

\_dlatch의 Flow Summary 화면이다. total pins의 값이 clk, d, q, q\_bar로 총 4임을 확인할 수 있다. Logic utilization은 1임을 확인할 수 있다.

## 2) D flip-flop

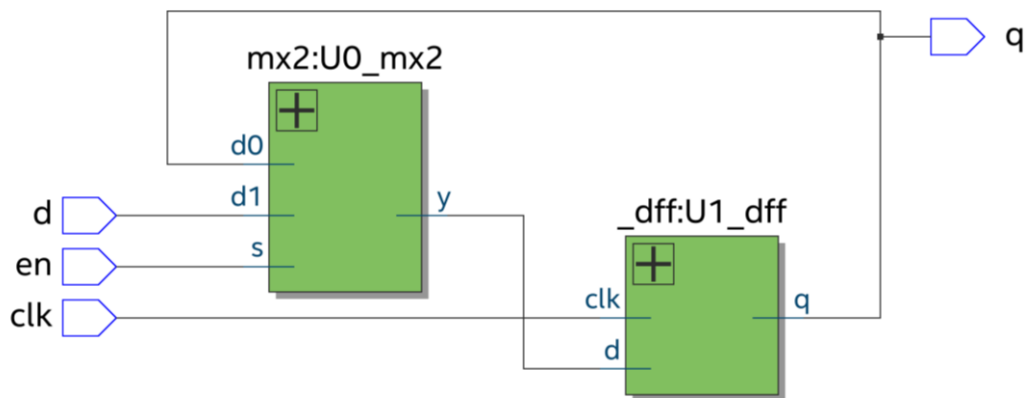


\_dff의 RTL Viewer 화면이다. 첫 번째 dlatch에 ~clk와 d가 입력되고, 그 출력 값과 clk가 두 번째 dlatch의 입력 값으로 들어간다. 두 번째 dlatch에서 나온 출력 값이 D flip-flop의 출력 값이다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sat Oct 14 17:19:22 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 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 % )

\_dff의 Flow Summary 화면이다.total pins의 값이 clk, d, q, q\_bar로 총 4임을 확인할 수 있다. Logic utilization은 2임을 확인할 수 있다.

## 3) Enabled D flip-flop

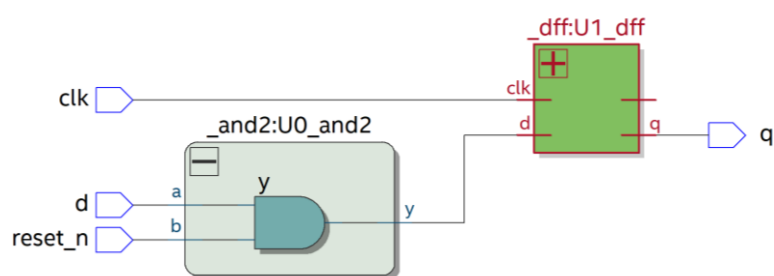


\_dff\_en의 RTL Viewer 화면이다. 이전 q값과 입력 d값이 mx2에 들어가고 en값에 따라 이전 q, d 중 하나와 clk가 dff의 입력 값으로 들어갑니다. dff의 출력 값이 곧 Enabled D flip-flop의 출력 값이 됩니다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Sat Oct 14 17:31:04 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 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 %)

\_dff\_en의 Flow Summary 화면이다. total pins는 clk, en, d, q로 총 4임을 확인할 수 있습니다. Logic utilization은 2임을 확인할 수 있습니다.

#### 4) resettable D flip-flop



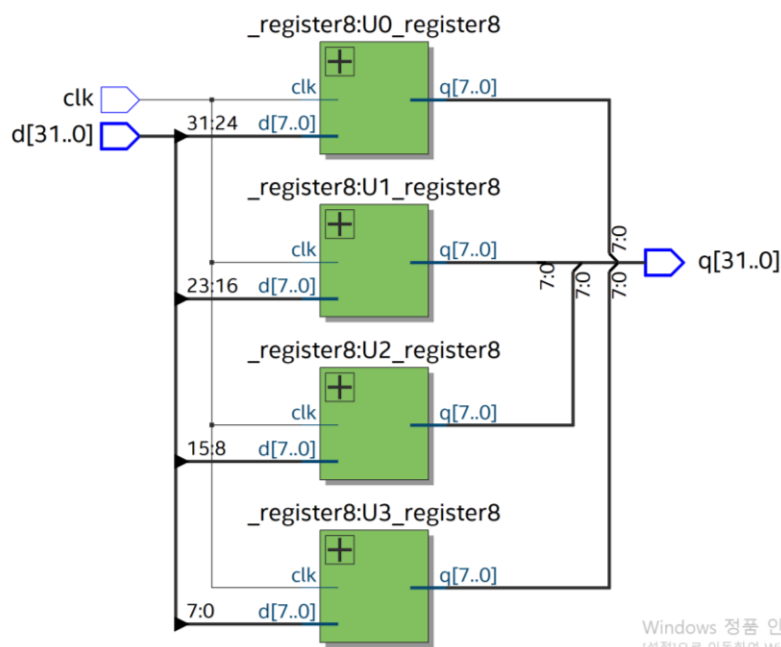


\_dff\_r의 RTL Viewer 화면이다. d와 reset\_n을 and 연산한 결과 값과 clk를 \_dff의 입력 값으로 넣어준다. reset\_n이 0이라면 d의 값에 상관 없이 0이 dff에 입력되므로 resettable하게 된다. 단, clk이 rising edge를 가질 때만 \_dff 출력 값이 0으로 초기화 되므로 synchronous함을 알 수 있다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Mon Oct 16 13:49:31 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 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 % )

\_dff\_r의 Flow Summary 화면이다. total pins가 clk, reset\_n, d, q로 총 4임을 알 수 있다. Logic utilization은 2임을 확인할 수 있다.

##### 5) 32 bits register

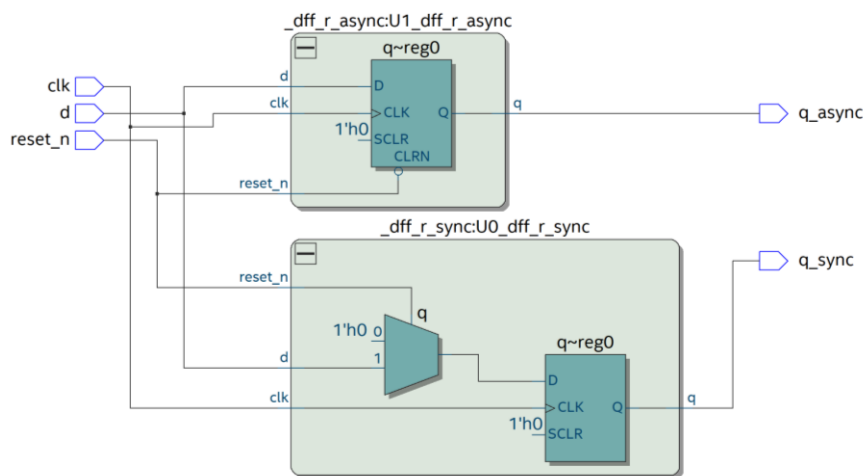


\_register32의 RTL Viewer 화면이다. register8을 4개 병렬연결로 구현했다. \_register32의 입력 값이 32비트 데이터와 clk 이므로 각 register8에 8비트씩 나누어 입력해줬다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Mon Oct 16 13:56:00 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	_register32
Top-level Entity Name	_register32
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	33 / 41,910 ( < 1 % )
Total registers	0
Total pins	65 / 499 ( 13 % )
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 % )

\_register32의 Flow Summary 화면이다. total pins가  $\text{clk } 1\text{bit} + \text{d } 32\text{bits} + \text{q } 32\text{bits} = 65\text{임}$ 을 알 수 있다. Logic utilization은 33임을 알 수 있다.

#### 6) async\_sync resettable D flip-flop



\_dff\_r\_sync\_async의 RTL Viewer 화면이다. asynchronous의 경우 모든 입력 값이 dff에서 종합되지만, synchronous의 경우 reset\_n과 d가 mx2에 입력되어 0 or 1이 dff에 입력 값으로 들어간다. 위에서 구현한 \_dff\_r과 동일한 구조라고 할 수 있다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Mon Oct 16 14:10:31 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	_dff_r_sync_async
Top-level Entity Name	_dff_r_sync_async
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	1 / 41,910 ( < 1 % )
Total registers	2
Total pins	5 / 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 % )

\_dff\_r\_sync\_async의 Flow Summary 화면이다. total pins가 clk, reset\_n, d, q\_sync, q\_async로 총 5임을 확인할 수 있다. Logic utilization은 2이다.

## 5. 고찰 및 결론

### A. 고찰

synchronous와 asynchronous를 이론으로만 접해서 헷갈리는 부분이 있었는데, 이번 실습을 통해 synchronous, asynchronous resettable D flip-flop을 직접 구현해 보고 RTL Viewer로 비교해보면서, 둘 사이에 어떠한 차이점이 있는지, 구조면에서는 어떤 차이점이 있는지를 알 수 있었습니다. 또한 1학기 때 배웠던 D latch, D flip-flop에 대해 복습할 수 있어서 가치 있는 시간이었습니다.

### B. 결론

#### 1) D-flipflop synchronous reset과 D-flipflop asynchronous reset의 차이

D-flipflop synchronous reset 같은 경우에는 reset\_n 단자가 flipflop과 별개로 분리되어 있어, reset\_n이 0일 때, 입력 값을 0으로 초기화해주는 역할을 하고, D-flipflop asynchronous reset은 reset\_n 단자가 flipflop에 포함되어 있어, reset\_n이 0일 때, 출력 값이 바로 0으로 초기화됩니다.

## 6. 참고문헌

유지현 교수님/컴퓨터공학기초실험1/광운대학교(컴퓨터정보공학부)/2023

이준환 교수님/컴퓨터공학기초실험2/광운대학교(컴퓨터정보공학부)/2023