

컴퓨터 공학 기초 실험2 보고서

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

실험일자: 2023년 10월 11일 (수)

제출일자: 2023년 10월 12일 (목)

학 과: 컴퓨터정보공학부

담당교수: 이혁준 교수님

실습분반: 수요일 0, 1, 2

학 번: 2022202075

성 명: 우나륜

1. 제목 및 목적

A. 제목

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

B. 목적

Flip-Flop과 Latch를 설계하고 Verilog로 Flip-Flop과 register는 structural implementation 구현을 하고 behavioral implementation을 통해 asynchronous/synchronous의 차이점을 확인하는 데 목적을 둔다.

2. 원리(배경지식)

A. D Latch

D Latch는 비트를 임시 저장할 수 있는 소자로, 입력되는 clock의 상태에 따라 입력값 D에 대해 변화하는 결과값 Q를 출력한다. Clock의 값이 enable의 상태를 유지하는 동안 Q는 D의 값을 가지며, clock이 enable의 상태가 아니면 이전의 Q값을 유지한다. 아래의 표1은 clock에 따른 Q의 상태를 표로 나타낸 것이고, 그림1은 D Latch의 symbol을 나타낸 것이다. D Latch는 4개의 NAND gate와 1개의 inverter를 사용하여 설계할 수 있다. 그림2에서 확인할 수 있듯이, Q와 \bar{Q} 의 값을 서로의 NAND 게이트에 입력값으로 넣어 구현할 수 있다.

CLK	Q
0	이전 Q
1	D

표1. D Latch truth table

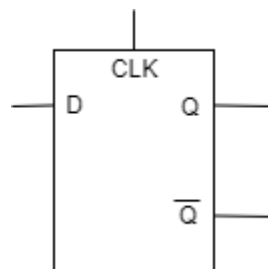


그림1. D Latch Symbol

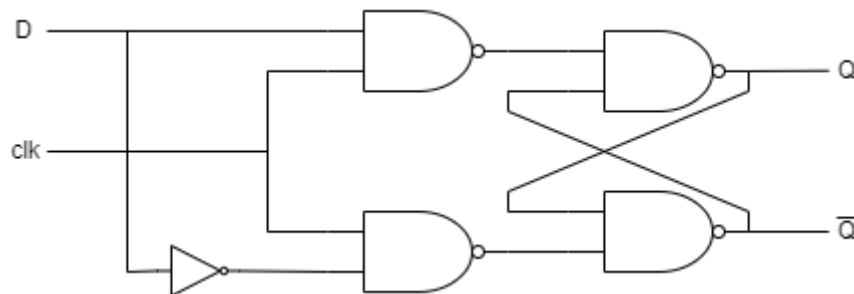


그림2. D Latch with 4 NAND gates

B. D Flip-Flop

D Flip-Flop은 D Latch와 달리, clock의 rising edge 또는 falling edge일 때의 D의 값을 가지며 clock이 enable인 상태일 동안 변화하는 D의 값을 따라가지 않는다. 아래의 표2는 D Flip-Flop의 상태를 표로 나타낸 것이고 그림3은 D Flip-Flop의 symbol이다.

CLK	Q
↑	D
Other case	이전 Q

표2. D Flip-Flop truth table

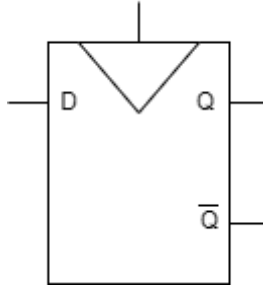


그림3. D Flip-Flop symbol

C. Enable D Flip-Flop

Enable D Flip-Flop은 EN의 값이 0이면 Flip-Flop의 입력이 모두 0이 되고, Q의 값이 유지되는 회로이다. 만약 EN의 값이 1이 되면 입력값 S와 D가 같고 R의 값은 $\sim D$ 가 된다. 그렇게 되면 D의 값은 Q가 되고 이때 EN의 값이 다시 0이 되면 이전의 D를 유지한다. 본 과제에서 구현한 Enable D Flip-Flop의 설계 외에 그림4와 같이 5개의 NAND gate으로 구현할 수 있다.

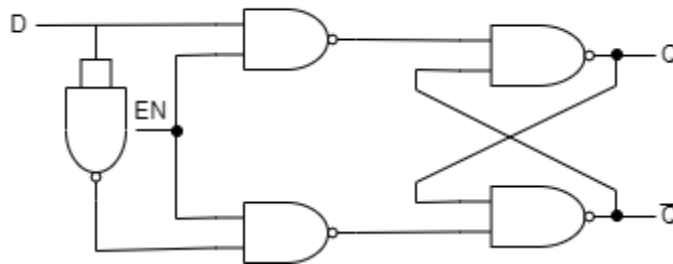


그림4. Enable D Flip-Flop

D. Resettable D Flip-Flop

Resettable D Flip-Flop은 기본 D Flip-Flop에 reset 기능이 추가된 D Flip-Flop이다. 본 과제에서 구현한 resettable D Flip-Flop은 reset의 값이 0일 때 reset 기능을 수행한다.

Input			Output
R	D	CLK	Q
0	X	X	0
1	0	↑	0

1	1	↑	1
1	X	↓ or 0 or ↑	이전 Q

표3. Resettable D Flip-Flop truth table

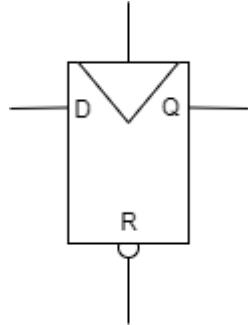


그림5. Resettable D Flip-Flop Symbol

3. 설계 세부사항

A. D Latch

`_dlatch` module은 입력값으로 clock과 입력값 d를 가지고 q와 q_bar를 출력한다. Clock 또는 d가 변화할 때마다 clock의 값이 1이면 q는 d의 값을 가지게 된다.

B. D Flip-Flop

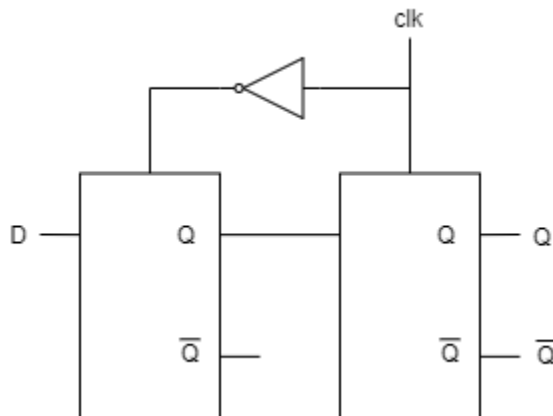


그림6. D Flip-Flop schematic

`_dff` module은 clock과 입력 d를 입력받고 q와 q_bar를 출력한다. D Flip-Flop은 D Latch를 2개를 사용하여 구현할 수 있으며 첫번째 D Latch는 \sim Clock과 D를 입력하고 출력값 Q를 다음의 D Latch의 입력 D로 입력한다. 그리고 그에 대한 결과값으로 Q와 \sim Q가 출력된다.

C. Enable D Flip-Flop

`_dff_en` module은 데이터 D와 clock, reset_n을 입력받는다. 이때 reset_n은 active-low에 동작한다. 즉, reset_n의 값이 0일 때 reset 기능을 수행하고 1일 때는 수행하

지 않는다.

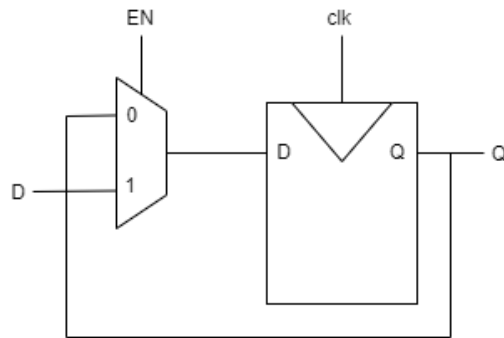


그림7. Enabled D Flip-Flop schematic

D. Resettable D Flip-Flop

Resettable D Flip-Flop은 기본적인 D Flip-Flop 앞에 AND gate를 추가하여 구현하였다. `_dff_r` module은 `clock`, `reset_n`, `d`를 입력받고 `q`를 출력한다.

E. 32-bit register

32-bit register는 32개의 D Flip-Flop을 한 줄로 연결하여 구현한다. 본 과제에서는 8-bit register module을 생성하여 완성하고, 8-bit register module을 instance화하여 총 4개를 사용하였다.

F. Async/Sync Resettable D Flip-Flop

Async/Sync Resettable D Flip-Flop은 Asynchronous Resettable D Flip-Flop과 Synchronous Resettable D Flip-Flop 모듈을 생성한 뒤, 인스턴스화하여 생성하였다. 이때 `set`과 `reset` 또한 active-low에서 동작하도록 하였다.

4. 설계 검증 및 실험 결과

A. 시뮬레이션 결과

1. D Latch

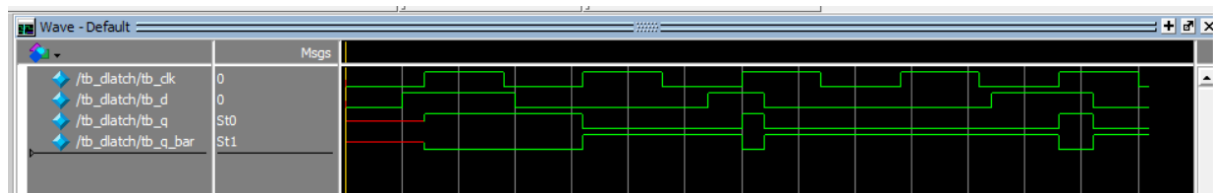


그림8. D Latch waveform

위 그림에서 `clk`의 값이 1이 되었을 때, `q`는 변화하는 `d`의 값을 갖는다. 만약 `clk`의 값이 0이 되면 `d`가 변화하여도 이전의 `q`값을 유지한다.

2. D Flip-Flop

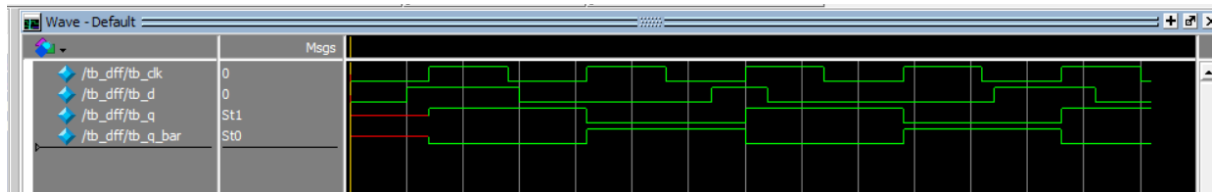


그림9. D Flip-Flop waveform

이전의 Latch와는 다르게, clk의 rising할 때 d의 값을 q가 가지게 된다. Clk이 1로 유지되는 동안에 d의 값이 변화하여도 q는 변화하지 않는다. 그리고 clk의 값이 0이 되면 이전의 q값을 유지한다.

3. Enable D Flip-Flop

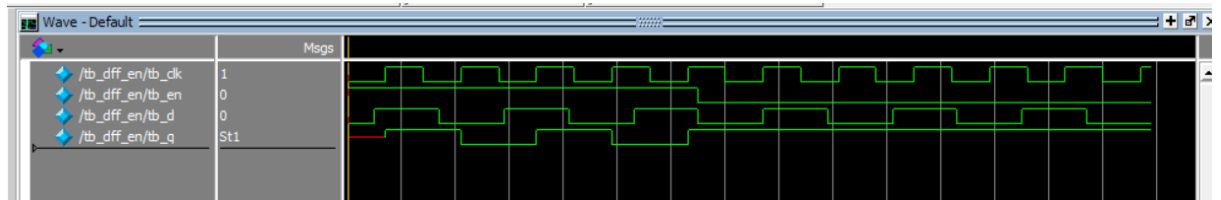


그림10. Enabled D Flip-Flop waveform

EN의 값이 1일 때는 이전의 D Flip-Flop과 같이 clock의 rising edge에서 현재 d의 값을 저장하고 그렇지 않을 때는 이전 q의 값을 유지한다. 하지만 EN의 값이 0이 되면, q는 이전의 q값을 계속 유지한다.

4. Resettable D Flip-Flop

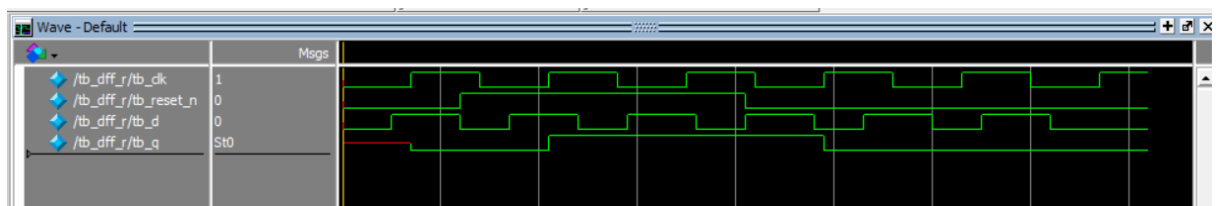


그림11. Resettable D Flip-Flop waveform

Resettable D Flip-Flop에서 reset은 active-low일 때 동작하며, 위의 waveform에서 확인할 수 있듯이, tb_reset_n의 값이 0이 되면 다음 clock의 rising edge에서 q의 값이 0이 된다. 그 이후에는 D의 값이 clock의 rising edge에서 0 또는 1이 되어도 q의 값은 0으로 D의 값을 따라가지 않는다.

5. 32-bit Register

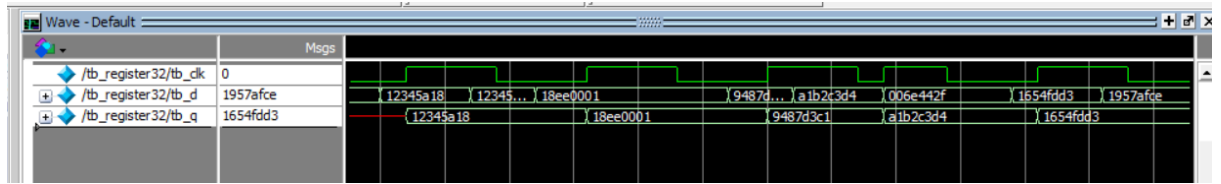


그림12. 32-bit Register waveform

Clock의 rising edge에서 Q는 32-bit D의 값을 가지는 것을 확인할 수 있다.

6. Asyncn/Sync Resettable D Flip-Flop

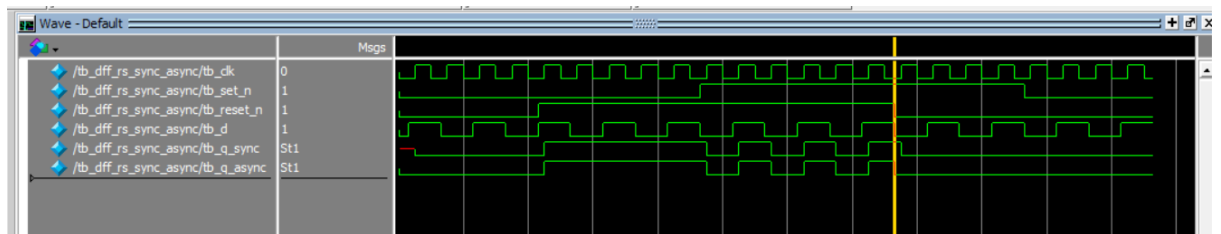


그림13. Asyncn/Sync Resettable D Flip-Flop

위의 waveform은 synchronous Flip-Flop과 asynchronous Flip-Flop을 보여주는 waveform이다. Synchronous Flip-Flop은 clock의 rising edge일 때 Q가 값을 가지므로 처음 rising edge에서 값을 가지지 않는다. 하지만 Asynchronous Flip-Flop은 clock과는 상관없이 set과 reset의 값에 따라 Q의 값이 변화하기 때문에 0 bit라는 값을 처음에 가지게 된다.

B. 합성(synthesis) 결과

1. D Latch

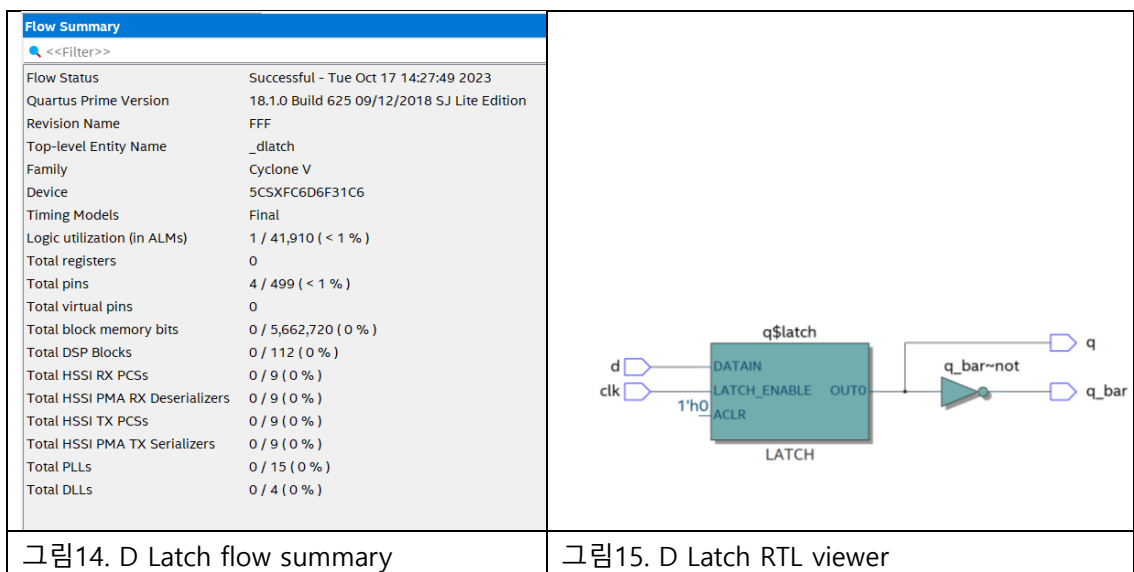


그림14. D Latch flow summary

그림15. D Latch RTL viewer

위 flow summary를 통해 _dlatch module의 Logic utilization은 1로 1% 미만이고, pin은

4개로 1% 미만 사용했다는 것을 알 수 있다. RTL viewer에서 출력값 q_bar는 ~q의 값을 가지므로 값을 가지기 전에 q가 not 게이트를 통과한 후의 결과값이 저장된다.

2. D Flip-Flop

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Oct 17 14:29:13 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	FFF
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 %)

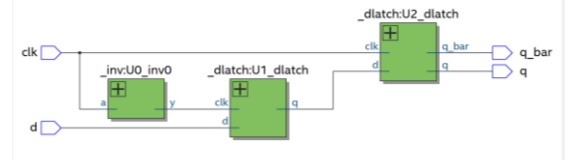


그림16. D Flip-Flop flow summary

그림17. D Flip-Flop RTW viewer

_dff module의 logic utilization은 2로 1% 미만이고 pins는 4로 1% 미만 사용되었다. RTL viewer를 통해 D Flip-Flop은 1개의 inverter와 2개의 D Latch로 구성된 것을 확인할 수 있다.

3. Enable D Flip-Flop

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Oct 17 14:30:26 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	FFF
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 %)

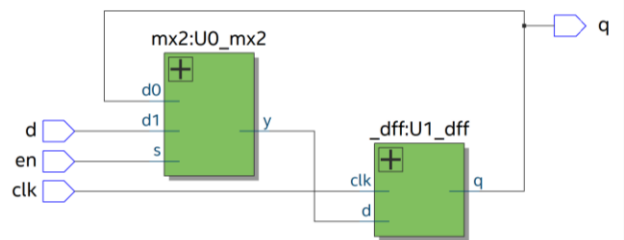


그림18. Enable D Flip-Flop flow summary

그림19. Enable D Flip-Flop RTW viewer

_dff_en module의 logic utilization은 2로 1%미만이고 pins는 4로 1% 미만 사용하였다.

RTL viewer를 통해 Enable D Flip-Flop은 1개의 D Flip-Flop과 1개의 2-to-1 MUX로 구성된 것을 확인할 수 있다.

4. Resettable D Flip-Flop

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Oct 17 14:31:46 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	FFF
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 %)

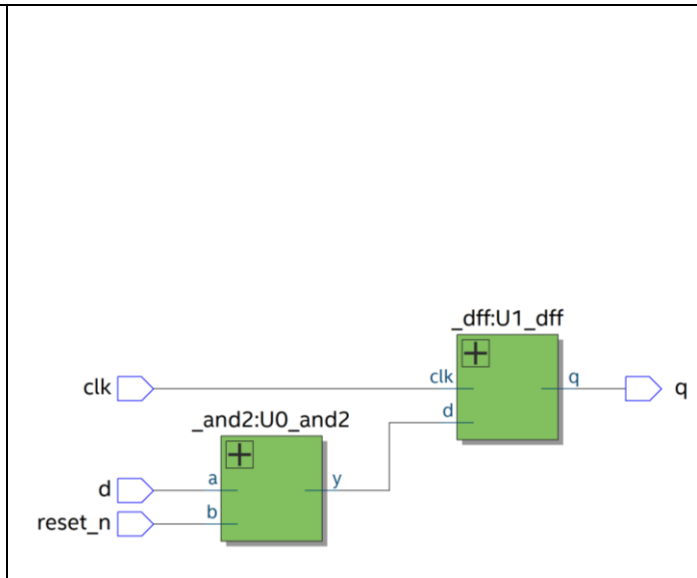


그림20. Resettable D Flip-Flop flow summary

그림21. Resettable D Flip-Flop RTL viewer

_dff_r module의 logic utilization은 2로 1% 미만이고 pins는 4로 1% 미만 사용되었다. RTL viewer를 통해 Resettable D Flip-Flop은 1개의 AND 게이트와 1개의 D Flip-Flop으로 구성되어 있는 것을 확인할 수 있다.

5. 32-bit Register

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Oct 17 14:33:07 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	FFF
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 %)

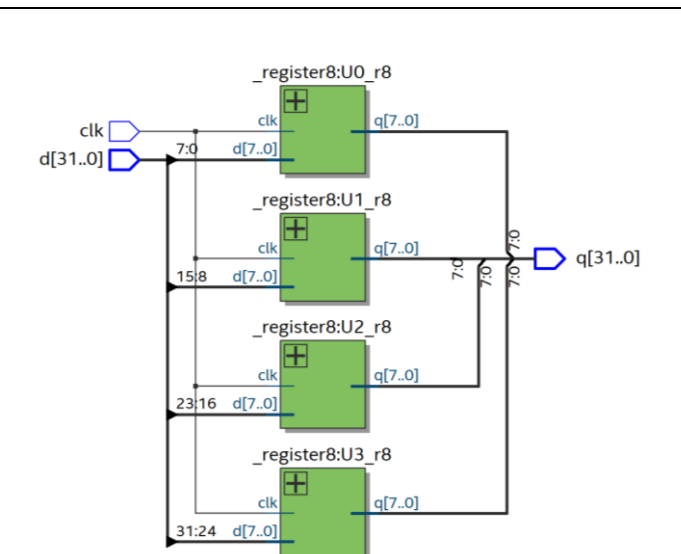
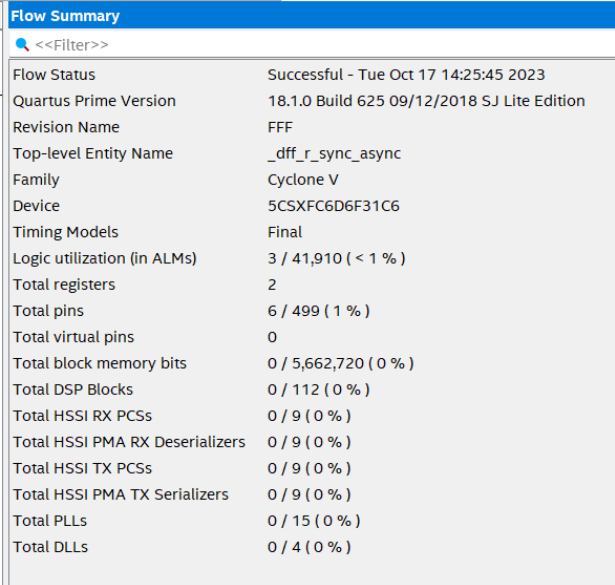
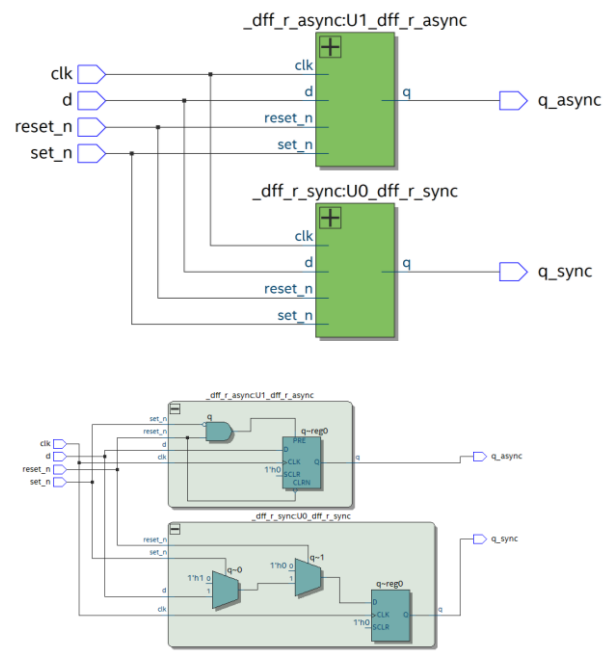


그림22. 32-bit Register flow summary

그림23. 32-bit RTL viewer

_register32 module의 logic utilization은 33으로 1%이고 pins는 65로 13% 사용되었다. RTL viewer를 통해 8-bit register instance가 4개 사용된 것을 확인할 수 있다.

6. Async/Sync Set/Resettable D Flip-Flop

 <p>Flow Summary</p> <p><<Filter>></p> <table border="1"> <tr> <td>Flow Status</td> <td>Successful - Tue Oct 17 14:25:45 2023</td> </tr> <tr> <td>Quartus Prime Version</td> <td>18.1.0 Build 625 09/12/2018 SJ Lite Edition</td> </tr> <tr> <td>Revision Name</td> <td>FFF</td> </tr> <tr> <td>Top-level Entity Name</td> <td>_dff_r_sync_async</td> </tr> <tr> <td>Family</td> <td>Cyclone V</td> </tr> <tr> <td>Device</td> <td>5CSXFC6D6F31C6</td> </tr> <tr> <td>Timing Models</td> <td>Final</td> </tr> <tr> <td>Logic utilization (in ALMs)</td> <td>3 / 41,910 (< 1 %)</td> </tr> <tr> <td>Total registers</td> <td>2</td> </tr> <tr> <td>Total pins</td> <td>6 / 499 (1 %)</td> </tr> <tr> <td>Total virtual pins</td> <td>0</td> </tr> <tr> <td>Total block memory bits</td> <td>0 / 5,662,720 (0 %)</td> </tr> <tr> <td>Total DSP Blocks</td> <td>0 / 112 (0 %)</td> </tr> <tr> <td>Total HSSI RX PCSs</td> <td>0 / 9 (0 %)</td> </tr> <tr> <td>Total HSSI PMA RX Deserializers</td> <td>0 / 9 (0 %)</td> </tr> <tr> <td>Total HSSI TX PCSs</td> <td>0 / 9 (0 %)</td> </tr> <tr> <td>Total HSSI PMA TX Serializers</td> <td>0 / 9 (0 %)</td> </tr> <tr> <td>Total PLLs</td> <td>0 / 15 (0 %)</td> </tr> <tr> <td>Total DLLs</td> <td>0 / 4 (0 %)</td> </tr> </table>	Flow Status	Successful - Tue Oct 17 14:25:45 2023	Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition	Revision Name	FFF	Top-level Entity Name	_dff_r_sync_async	Family	Cyclone V	Device	5CSXFC6D6F31C6	Timing Models	Final	Logic utilization (in ALMs)	3 / 41,910 (< 1 %)	Total registers	2	Total pins	6 / 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 %)	
Flow Status	Successful - Tue Oct 17 14:25:45 2023																																						
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition																																						
Revision Name	FFF																																						
Top-level Entity Name	_dff_r_sync_async																																						
Family	Cyclone V																																						
Device	5CSXFC6D6F31C6																																						
Timing Models	Final																																						
Logic utilization (in ALMs)	3 / 41,910 (< 1 %)																																						
Total registers	2																																						
Total pins	6 / 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 %)																																						
<p>그림24. Async/Sync Set/Resettable D Flip-Flop flow summary</p>	<p>그림25. Async/Sync Set/Resettable D Flip-Flop RTL viewer</p>																																						

_dff_r_sync_async module의 logic utilization은 3으로 1% 미만이고, 사용한 register는 2개이며 pins는 6으로 1%미만 사용하였다. RTL viewer를 통해 위에서 Asynchronous D Flip-Flop 1개의 Synchronous D Flip-Flop 1개가 사용된 것을 알 수 있다.

5. 고찰 및 결론

A. 고찰

1. Testbench 실행시에 Error loading design 오류가 발생하였다. 이는 testbench 파일에서 문법적 오류가 발생하여 생긴 오류였다. 오류가 발생하는 line의 코드를 알맞게 수정하니 해결할 수 있었다.
2. DFF with active-low synchronous reset and set과 DFF with active-low asynchronous reset and set의 차이점은 clock과 관련되어 있다. Synchronous Flip-Flop은 출력 Q의 값은 clock이 rising할 때의 reset, set, D의 값에 따라 결정된다. 반대로 Asynchronous Flip-Flop은 clock의 값과 관계없이 clock의 rising edge, reset, set 중 하나라도 값이 세팅되면 Q의 값이 Q 값에 따라 적용된다.

B. 결론

회로에서 상태를 설정할 때 active-low와 active-high 두 가지 상태로 구현할 수 있다는

것을 배웠다. Active-low 상태는 신호의 상태가 0일 때 동작하고, active-high 상태는 신호의 상태가 1일 때 동작하는 것이다. 대표적으로 본 과제에서 구현한 reset과 set이 그 예시이다.

6. 참고문헌

[1] Enabled Flip Flop / <https://wh00300.tistory.com/21>

[2] 이준환 / Lab#6 Latch & Flip-Flop design with/without reset/set / 광운대학교 / 2023