컴퓨터 공학 기초 실험2 보고서

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

실험일자: 2022년 10월 11일 (화)

제출일자: 2022년 10월 13일 (목)

학 과: 컴퓨터정보공학부

담당교수: 공영호 교수님

실습분반: 화요일 0,1,2

학 번: 2019202050

성 명: 이강현

1. 제목 및 목적

A. 제목

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

B. 목적

이번 실험을 통해 latch와 flip-flop의 차이를 이해하고 set과 reset이 어떤 역할을 수행하는지 또한 있는 회로와 없는 회로의 차이를 안다. Synchronous와 Asynchronous의 차이점을 통해 동작 원리를 이해하고 verilog를 통해 구현해본다.

2. 원리(배경지식)

<Latch>

래치(Latch)란 클럭이 존재하지 않아 클럭 신호에 상관없이 input의 변화에 따라 언제든지 output 값을 변화시킬 수 있는 synchronous Sequential logic이다.

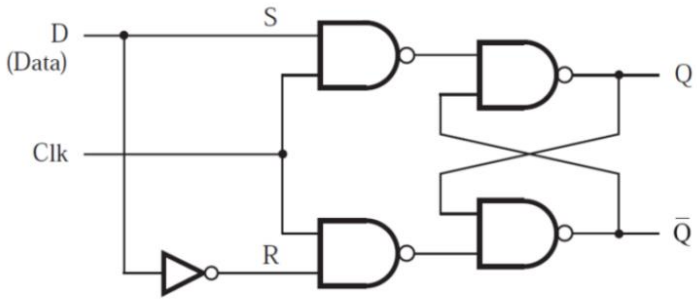
<Flip-Flop>

플립플롭(flip-flop)이란 latch와 달리 클럭이 존재하여 input 값의 변화에 따라 output 값이 바로 변하지 않고 클럭이 rising edge 혹은 falling edge일 때 output 값이 변하는 Asynchronous Sequential logic이다.

<D-latch>

D-latch는 latch이지만 clock이 존재한다. 따라서 clock의 영향을 받지만 변하는 타이밍인 edge에서 값의 변화가 생기지 않고 단순히 clock값이 1이면 D의 값이 반영되고 0이면 이전 값을 유지하도록 작동한다. 이는 SR latch에서 S와 R이 모두 1일 때 생기는 문제를 해결할 수 있게 한다.

<4개의 NAND gate로 D-latch 구현해보기>



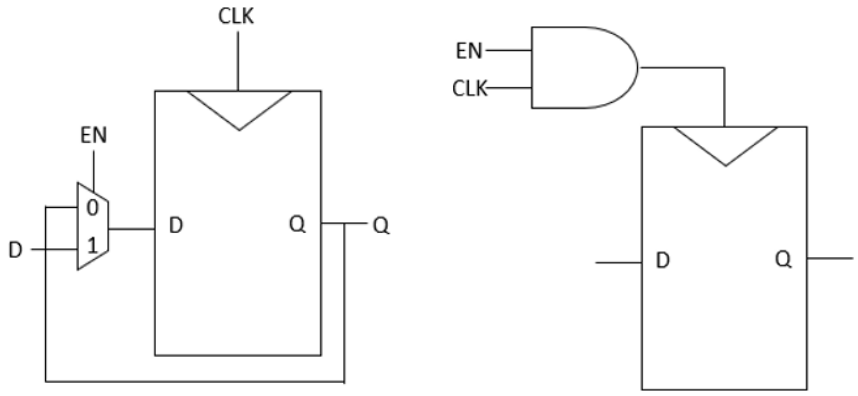
위는 4개의 NAND gate로 D-latch를 구현한 것이다.

Input		Output	
Clk	**D**	**Q**	**~Q**
0	X	Prev Q	~Prev Q
1	0	0	1
1	1	1	0

위와 같은 논리식을 가진다.

and게이트 2개와 or게이트 2개로 Dlatch를 구현할 수 있으나 bubble push를 통해서 동일한 nand 게이트 4개로 회로를 위와 같이 구현할 수 있다.

<다른 방법으로 enabled d-flip flop 구현하기>



실습에서 구현한 enabled d flip flop은 왼쪽의 회로이다. 2 to 1 mux를 사용하여 0 또는 1을 enable의 상태에 맞게 출력했으나 이를 오른쪽과 같은 회로로 구현한다면 and게이트는 둘다 1이어야 D값을 출력하므로 en이 1이어야 1이 나갈 수 있다는 동일한 결과를 얻게 된다. en이 0인 경우에는 clk이 작동되지 않아 이전값을 유지한다는 것도 동일하게 구현할 수 있다.

아래는 enabled flip flop의 truth table이다.

Input			Output
CLK	**EN**	**D**	**Q**

↑	1	1	1
↑	1	0	0
↑	0	X	Prev Q

3. 설계 세부사항

D-Latch

D-Latch는 input으로 clk와 d가 존재하여 clk가 1일 때만 d 값에 의해 output Q의 값이 변하고 clk가 0인 경우 이전 값을 유지한다

Input		Output	
CLK	**D**	**Q**	**~Q**
0	x	Prev Q	~Prev Q
1	0	0	1
1	1	1	0

D Flip-Flop

D flip-flop은 D Latch와 원리는 같지만 clock이 rising edge일때만 output 값이 D값에 따라 변한다.

Input		Output	
CLK	**D**	**Q**	**~Q**
↑	x	Prev Q	~Prev Q
other	0	0	1

Enabled D Flip-Flop

Enabled D Flip-Flop은 D flip-flop에 mux가 추가되어 EN=0일 때는 CLK과 D값에 상관없이

Q는 이전 값을 유지하고 EN=1일때만 D flip-flop과 똑같이 동작한다.

Input			Output
CLK	**EN**	**D**	**Q**
↑	1	1	1
↑	1	0	0

↑	0	X	Prev Q

Resettable D Flip-Flop

Resettable D Flip-Flop은 reset이 추가된 D Flip-flop이므로 reset=0일 때는 d 값에 상관없 이 clk가 rising edge를 가지면 output이 무조건 0으로 나오고 reset=1일 때 D Flip-Flop과 똑같이 동작한다.

Input		Output	
CLK	**reset**	**Q**	**~Q**
↑	1	Prev Q	~Prev Q
other	1	0	1
X	0	Prev Q	~Prev Q

Set/Resettable D Flip-Flop

Set/Resettable D Flip-Flop은 set과 reset이 존재하는 D flip-flop으로 input d값이 D와 set'이 OR gate의 input으로 들어가 나온 output 값과 reset이 AND gate의 input으로 들어가 나온 값이 D Flip-Flop의 output 값이 된다. 따라서 set과 d 값에 상관없이 reset이 0이면 q값은 무조건 0이 되고 reset이 1인 경우 set이 1이면 d 값에 상관없이 Q값이 1이 된다.

Input				Output
R'	**S'**	**D**	**CLK**	**Q**
0	X	X	X	0
1	0	X	X	1
1	1	0	↑	0
1	1	1	↑	1
1	1	X	else	Prev Q

32-bits Register

32-bit register는 D Flip-Flop의 input과 output을 8-bit로 만든 8-bit register 4개로 만든 것으로 D Flip-Flop과 동작원리는 같고 input, output이 32-bit이다.

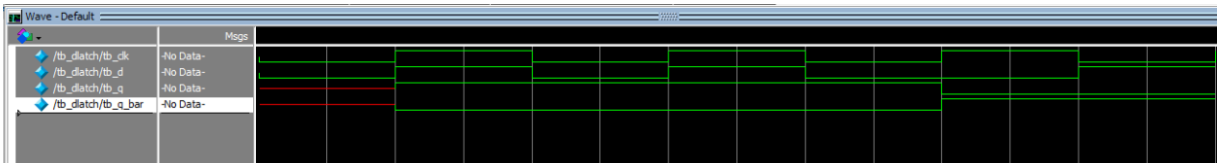
Async/Sync Set/Resettable D Flip-Flop

Sync set/resettable D flip-flop은 clock에 의해서만 값이 변하는데 Async set/resettable D flip-flop은 clk, set, reset들의 값에 의해서도 Q 값이 변할 수 있다. 동작 원리는 set/resettable D flip-flop과 같으나 Sync인지 Async인지의 차이점이 있다.

4. 설계 검증 및 실험 결과

A. 시뮬레이션 결과

D-Latch

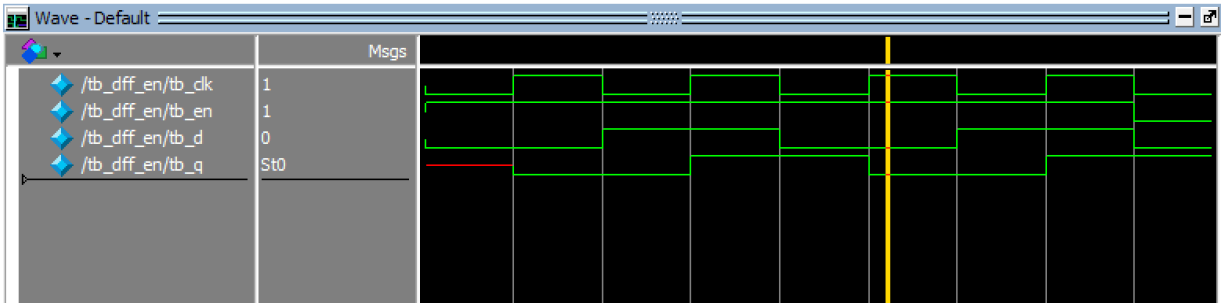


CLK이 1일 때, q가 d의 값을 따라가는 것을 waveform을 통해 확인할 수 있다.

D Flip-Flop

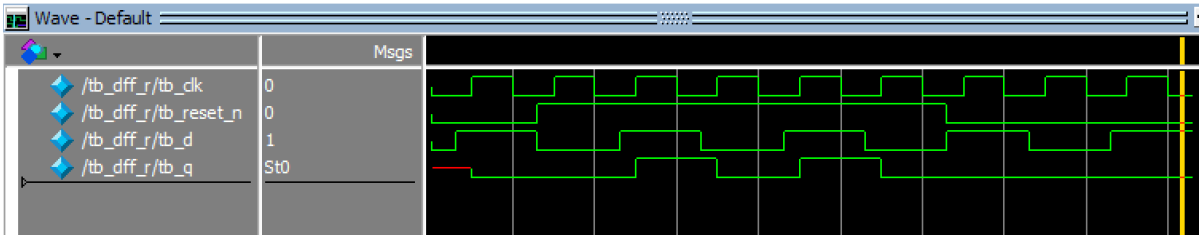
CLK이 Rising edge일 때, q값이 변하는 것을 waveform을 통해 확인할 수 있다.

Enabled D Flip-Flop



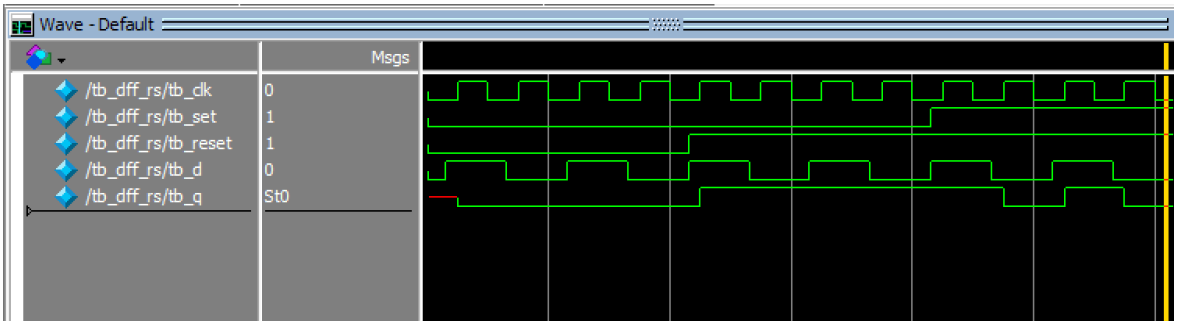
EN=0일 때 q 값이 이전 값을 유지하고 en= 1이고 clk가 rising edge일때만 q 값이 변하는 것을 확인할 수 있다.

Resettable D Flip-Flop



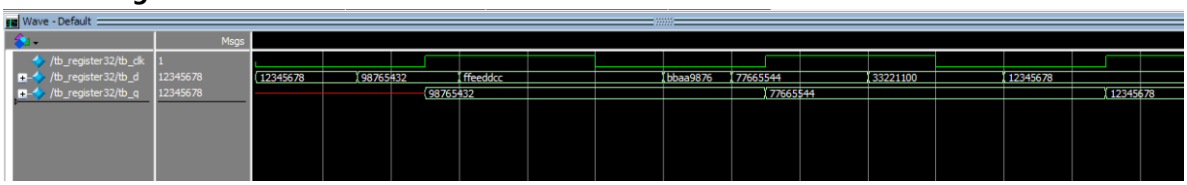
reset=0 일 때 clk가 rising edge를 가지면 D 값에 상관없이 Q 값이 무조건 0이 된다. reset=1인 경우 clk가 rising edge일 때 d 값에 따라 q 값이 변하는 것을 볼 수 있다.

Synchronous Set/Resettable D Flip-Flop



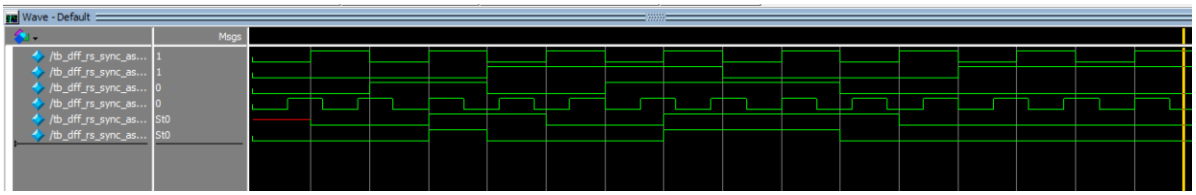
reset=0일 때, q는 0이 되고, reset=1, set=0일 때는 q는 1이 된다. Reset=1 set=1일 때, clk의 rising edge에서 q가 d 값이 따라 변경되는 것을 확인할 수 있다.

32-bits Register



위의 waveform과 같이 clk가 rising edge가 아닐 때는 q가 이전 값을 유지하고 있다가 rising edge일 때 d 값에 따라 q 값이 변한다.

Async/Sync Set/Resettable D Flip-Flop

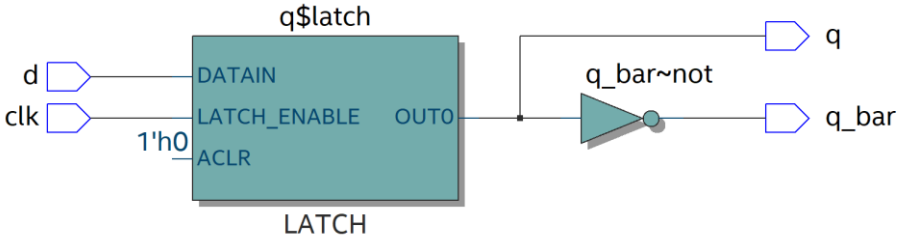


Synchronous은 clk가 rising edge일 때만 reset, set, d에 의해 q의 값이 변하고, Asynchronous은 clk가 rising edge이거나 reset이나 set이 falling edge 일 때 reset, set, d의 값에 의해 q값이 변하는 것을 waveform으로 검증할 수 있다.

B. 합성(synthesis) 결과

D-Latch

<RTL Viewer>



Dlatch 구현모습이다. 게이트를 사용하지 않아 위처럼 나왔다.

<Flow Summary>

Flow Summary

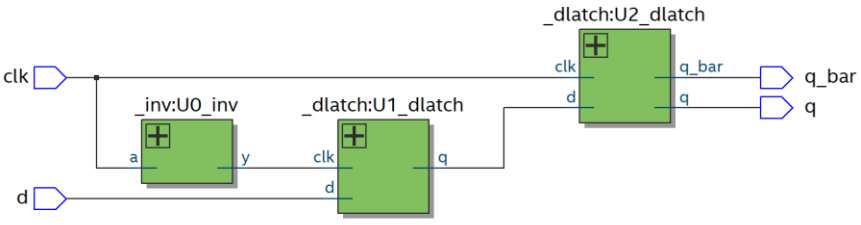
<<Filter>>

Flow Status	Successful - Wed Oct 12 20:15:27 2022
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 %)

성공적인 flow summary 결과를 얻었다.

D Flip-Flop

<RTL Viewer>



D-latch 2개와 1개의 inverter를 사용하여 구현하였다.

<Flow Summary>

Compilation Report - _dff

Flow Summary

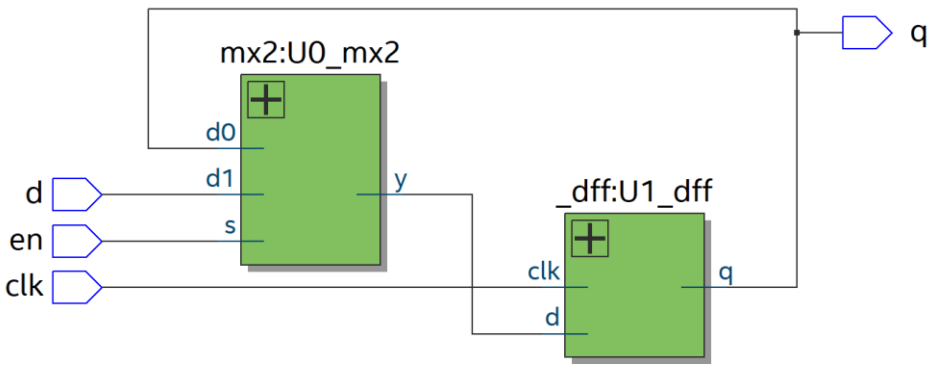
<<Filter>>

Flow Status	Successful - Wed Oct 12 20:30:29 2022
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)	N/A
Total registers	0
Total pins	4
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

성공적인 flow summary 결과를 얻었다.

Enabled D Flip-Flop

<RTL Viewer>



이전에 구현한 dff에 2 to 1 mux를 연결하여 en의 입력에 따른 값을 판단하게 하였다.

<Flow Summary>

Flow Summary

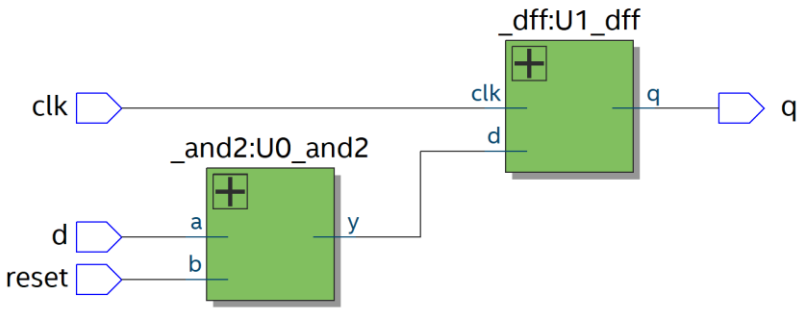
<<Filter>>

Flow Status	Successful - Wed Oct 12 21:15:29 2022
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)	N/A
Total registers	0
Total pins	4
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

성공적인 flow summary 결과를 얻었다.

Resettable D Flip-Flop

<RTL Viewer>



reset과 d를 2 input and gate에 연결하여 D flip flop에 연결하여 구현하였다.

<Flow Summary>

Flow Summary

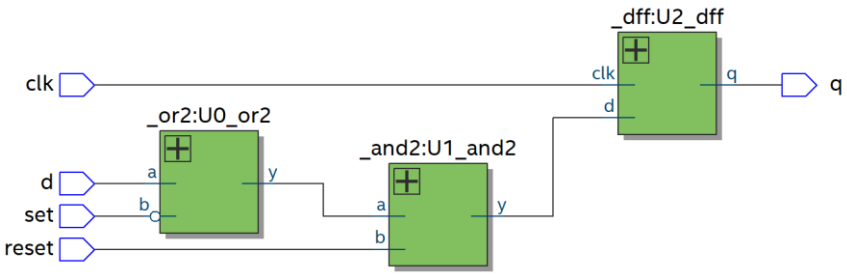
<<Filter>>

Flow Status	Successful - Wed Oct 12 22:28:52 2022
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)	N/A
Total registers	0
Total pins	4
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

성공적인 flow summary 결과를 얻었다.

Synchronous Set/Resettable D Flip-Flop

<RTL Viewer>



Active low에서 작동하게끔 set의 입력값에 버블을 붙여주었다.

<Flow Summary>

Flow Summary

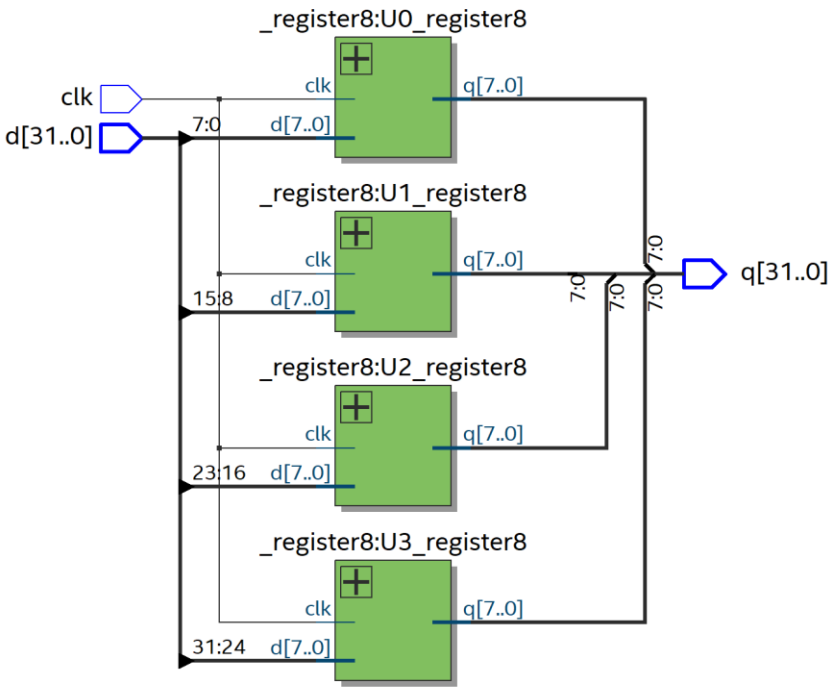
<<Filter>>

Flow Status	Successful - Wed Oct 12 23:53:07 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	_dff_rs
Top-level Entity Name	_dff_rs
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	0
Total pins	5
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

성공적인 flow summary 결과를 얻었다.

32-bits Register

<RTL Viewer>



8 bit register를 4개 연결해서 구현하였다. 각 8 bit register은 총 8 개의 flip-flop으로 연결되어 있다.

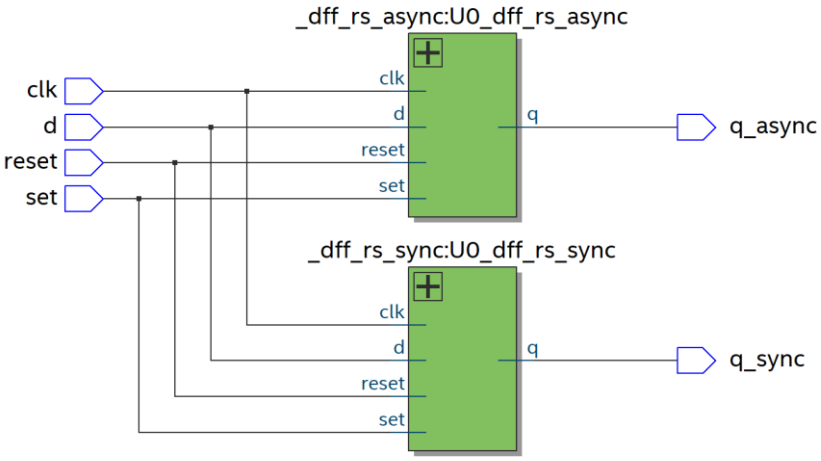
<Flow Summary>

Flow Summary	
<<Filter>>	
Flow Status	Successful - Thu Oct 13 00:10:03 2022
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)	N/A
Total registers	0
Total pins	65
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

성공적인 flow summary 결과를 얻었다.

Async/Sync Set/Resettable D Flip-Flop

<RTL Viewer>



Sync flip-flop은 clk에 의해서만 output이 변하고 async flip-flop은 clk, reset, set에 의해서 output이 변하도록 구현해 주었다.

<Flow Summary>

Flow Summary

<<Filter>>

Flow Status	Successful - Thu Oct 13 00:16:03 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	_dff_rs_sync_async
Top-level Entity Name	_dff_rs_sync_async
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	2
Total pins	6
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

성공적인 flow summary 결과를 얻었다.

5. 고찰 및 결론

A. 고찰

이번 실험에서는 latch & flip flop에 대한 실험이었다. 회로에 대한 기본적인 개념은 모두 알고 있었으나 막상 verilog로 구현하는데는 낯설어서 어려움이 있었다. Testbench를 통해 결과값이 잘 나오는지 확인하기 위해 고의적으로 값을 맞추는 것이 시간이 걸렸고 가장 기본이 되는 d latch를 구현할 때 sr latch를 이용해서 구현하는 방법을 알고 있었는데 always 구문을 이용하여 구현하는 것은 새롭고 간단하게 코드를 구현할 수 있었어서 놀라웠다. 동기식과 비동기식의 차이가 잘 이해하기가 어려웠고 코드를 보면 always @(posedge clk or negedge set or negedge reset)이런식으로 조건이 걸려있는데 서로 반대되는 입력을 넣었을 때 누가 우선시되는지 같은 것이 헷갈렸다.

B. 결론

D FF with active-low synchronous reset and set과 D FF with active-low asynchronous resetand set의 차이는 입력에 따른 결과가 나타나는 것이 동시에 일어나면 sync, 동시에 일어나지 않으면 async이다. sync에서는 reset,set,d의 값이 clk의 변화에 따라 q의 값에 반영이 되는데 async에서는 clk이 d의 값만 처리하고 set,reset은 clk와 독립적으로 q값을 변화 시킨다는 점에서 차이가 생긴다.

6. 참고문헌

공영호 교수님/컴퓨터공학기초실험2/2022