

컴퓨터 공학 기초 실험2 보고서

실험제목: Shifter & Counter

실험일자: 2022년 10월 18일 (화)

제출일자: 2022년 10월 24일 (월)

학 과: 컴퓨터공학과

담당교수: 공영호 교수님

실습분반: 화요일 0, 1, 2

학 번: 2021202058

성 명: 송채영

1. 제목 및 목적

A. 제목

Shifter & Counter

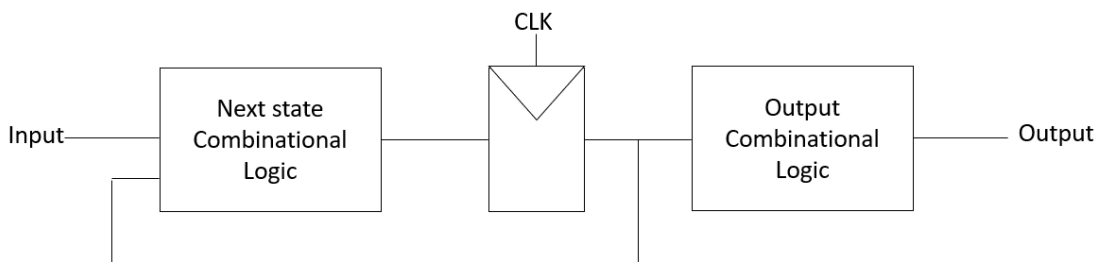
B. 목적

Flip-flop과 combinational logic을 이용하여 sequential logic인 shifter와 counter를 설계해 본다. 지난 실습에 이어 FSM 설계 과정을 익히고 FSM 설계 과정을 통해 5-way counter와 8-bits shifter와 8-bit loadable counter를 설계한다. 또한 이를 testbench를 이용해 검증해 본다.

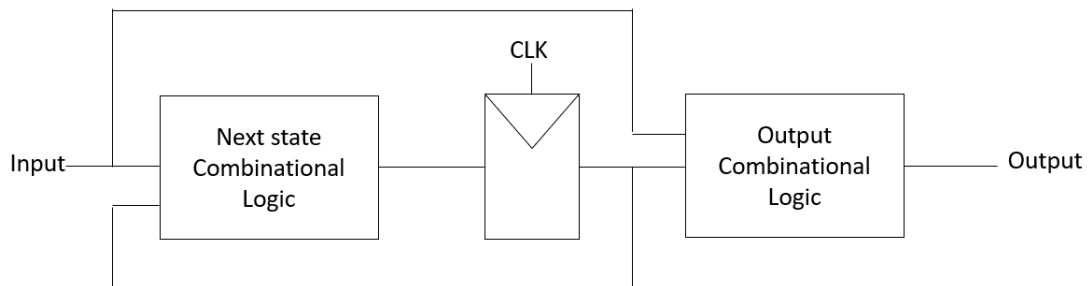
2. 원리(배경지식)

- Moore FSM 과 Mealy FSM 의 장단점

Moore Fsm과 Mealy Fsm은 output의 결정 방식이 다르다.



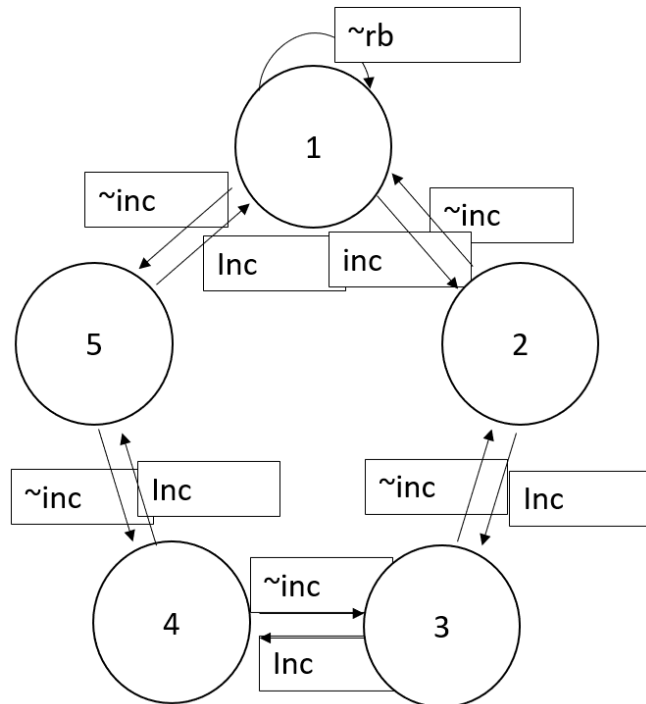
먼저 위의 사진은 Moore FSM으로, 현재 상태에 따라 다음 상태가 결정된다.



위의 사진은 Mealy FSM으로, 현재 상태와 input값에 따라 다음 상태가 결정된다.

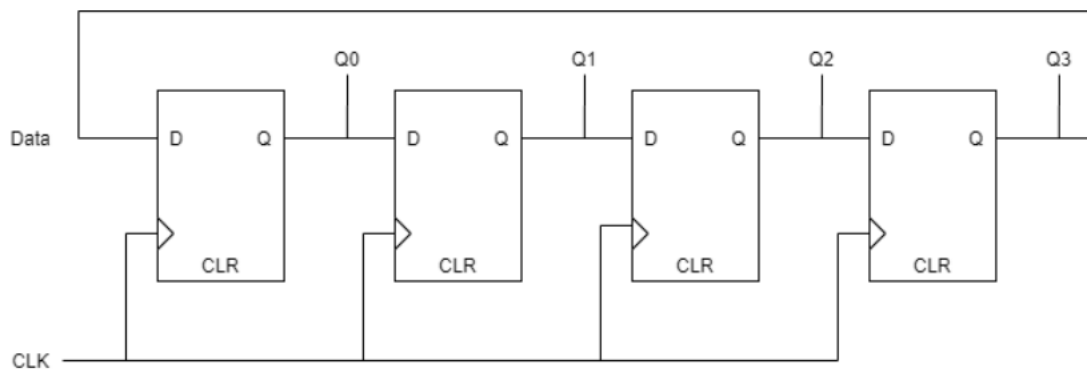
두 FSM을 비교했을 때 어느 것이 좋다고는 말 할 수 없지만 각각의 장단점에 맞게 사용하면 좋다. Mealy FSM은 Moore FSM보다 적은 수의 state를 가지므로, 작은 사이즈로 구현 가능하다. 또한 Mealy FSM은 input과 현재 상태에 의해 출력이 결정되므로, Moore FSM보다 빠르게 응답할 수 있다. 반면 Moore FSM은 출력이 input의 영향을 받지 않으므로 Mealy FSM보다 Delay를 고려할 필요가 적어 복잡한 회로일 경우 사용하면 좋다.

- ring counter



Ring counter는 shift register로 연결된 flip-flop들로 이루어진 counter이다. Ring counter는 위의 사진과 같이 마지막 flip-flop의 output이 첫 번째 flip-flop의 input으로 들어가는 구조이다. Input 값으로 inc를 가지며, inc의 값이 들어오면 값이 증가하고 그 반대일 경우 값이 감소한다.

Ring counter의 회로도는 다음과 같다.



- shifter

shifter는 레지스터에 저장되어 있는 정보를 이동시킬 수 있는 하드웨어이다.

NOP : no operation-현재 register의 값을 그대로 출력한다.

Load : 입력된 데이터를 출력

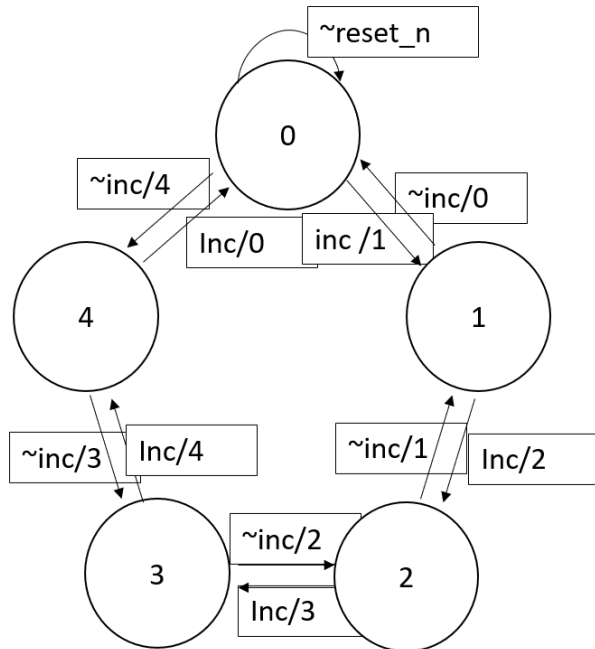
LSL : Logical Shift Left를 수행

LSR : Logical Shift Right를 수행

ASR : Arithmetic Shift Right를 수행

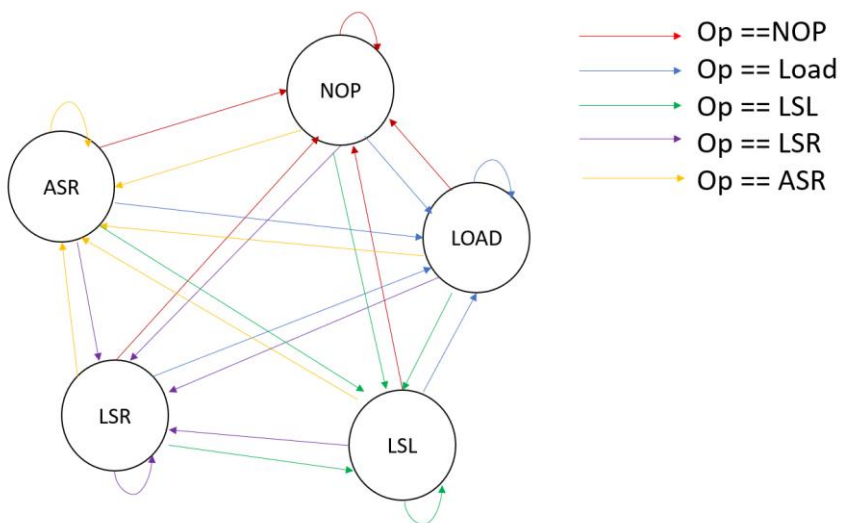
3. 설계 세부사항

- 5 way counter



위와 같은 방법으로 설계하였다. 0~4까지 총 5개의 state가 있으며, input으로 inc, output으로 현재상태가 있다. Reset을 통해 state를 0state로 reset하며, clk는 positive edge에서, reset_n은 negative edge에서 동작한다. 따라서 cnt는 clk가 positive edge일 때 동작한다.

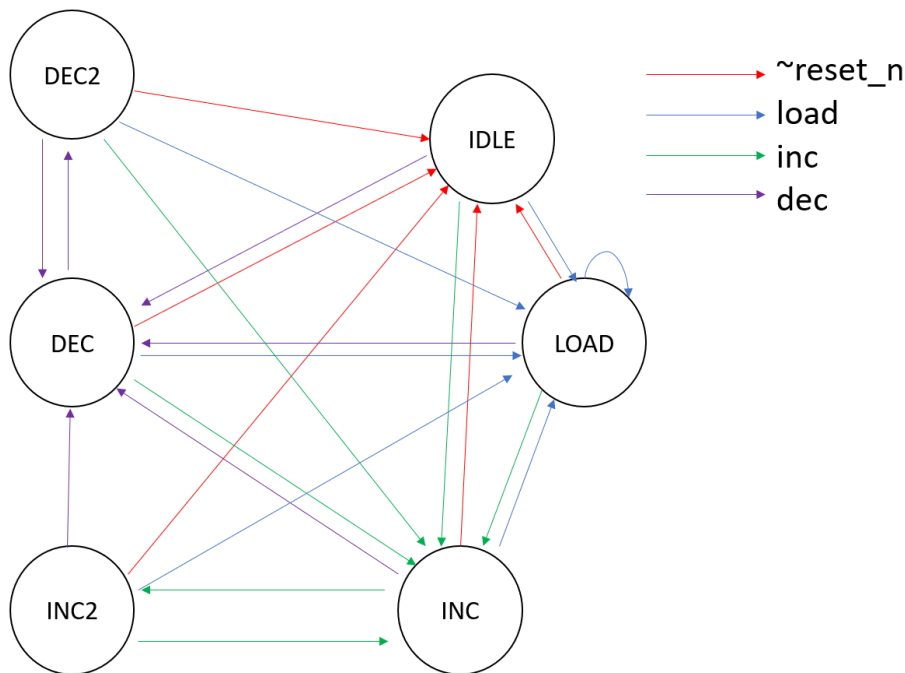
- 8-bit Shifter



위와 같은 방법으로 설계하였다. NOP, LOAD, LSL, LSR, ASR, 총 5개의 state가 있으며 어떤

state든지 op 값에 따라 이동하는 next state의 값은 같다. NOP는 현재 register의 값을 그대로 출력, LOAD는 register에 입력된 값을 출력, LSL, LSR, ASR은 각각의 shift 역할을 한다.

- 8-bit loadable up/down Counter

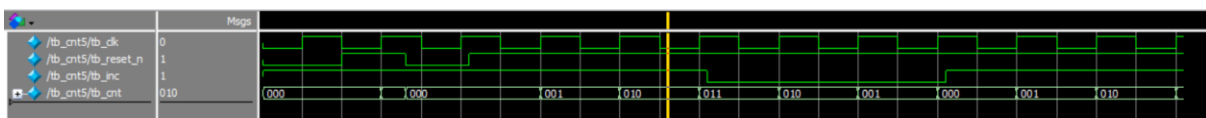


- 위와 같은 방법으로 설계하였다. IDLE은 reset되었을 때의 상태로 출력이 0이 되게 한다. Load는 8-bit Shifter와 같다. 현재 상태가 INC이고 입력, inc가 1이 들어오면 INC2의 state로 가고, INC2에서 inc가 1이 들어오면 INC의 state로 간다. 현재 상태가 DCE이고 입력 inc가 0이 들어오면 DCE2의 state로 가고, DCE2에서 inc가 0이 들어오면 DCE의 state로 간다. 즉 INC와 INC2는 inc가 1일 때의 상태로 1을 더한다. 반면 DCE와 DCE2는 inc가 0일 때의 상태로 1을 빼준다.

4. 설계 검증 및 실험 결과

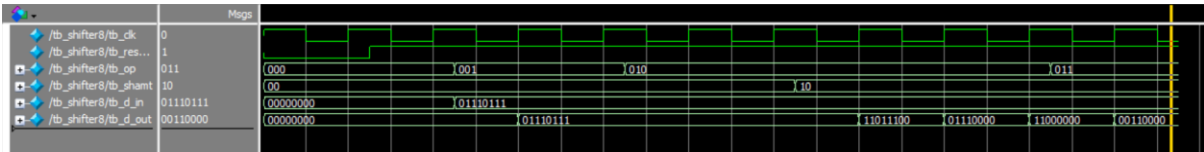
A. 시뮬레이션 결과

- cnt5



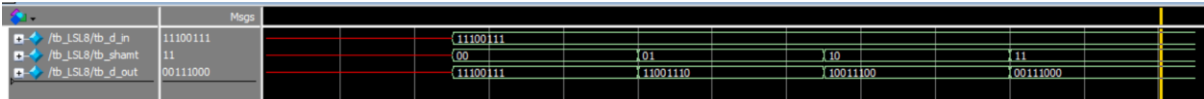
Reset을 통해 inc가 바뀌며 0 -> 1로 갈 때 값이 증가하고, 1 -> 0으로 갈 때 값이 감소하는 것을 cnt를 통해 확인할 수 있다.

- shifter 8



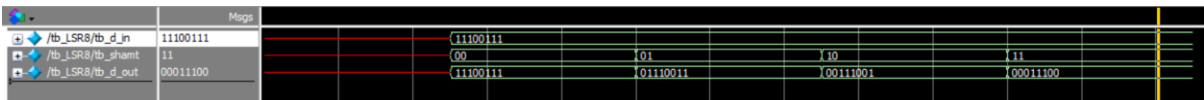
Binary encoding을 이용한 Encoding states는 다음과 같다. Nop 000, Load 001, LSL 010, LSR 011, ASR 100로, 각 명령어에 맞게 결과가 잘 출력이 된 것을 확인할 수 있다.

- LSL8



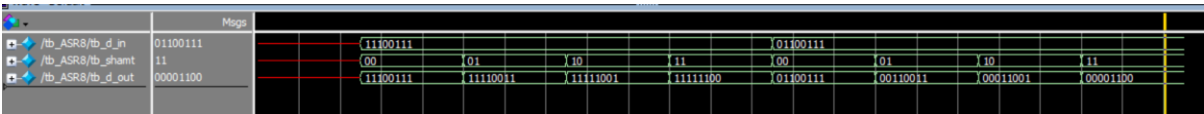
Sub module의 testbench이다. 빈공간이 0으로 채워지는 것을 확인할 수 있다.

- LSR8



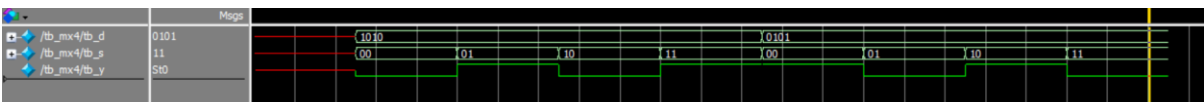
Sub module의 testbench이다. 빈공간이 0으로 채워지는 것을 확인할 수 있다.

- ASR8



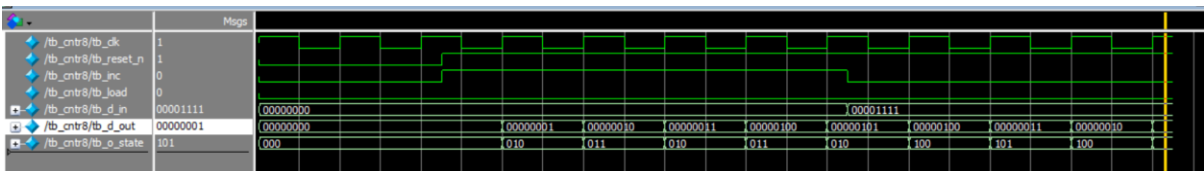
Sub module의 testbench이다. MSB를 알맞게 넣는 것을 확인할 수 있다.

- mx4

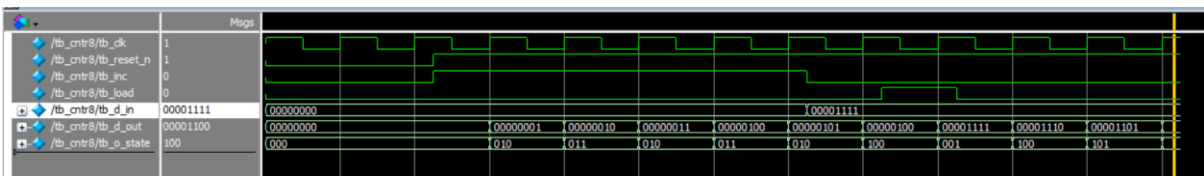


Sub module의 testbench이다. Mx2 module 3개로 만들었다.

- Cnrt8



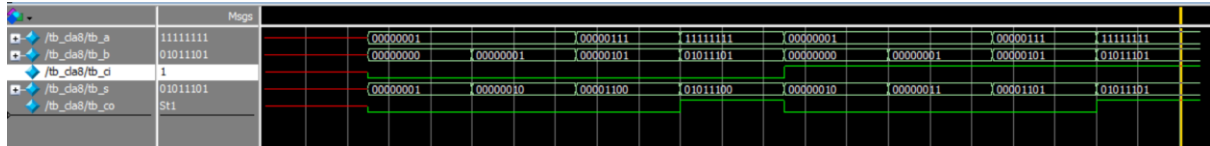
Load가 0일 때를 나타낸 사진이다.



Load가 1일 때를 나타낸 사진이다. 코드 내에서는 load가 0일 때를 나타냈으며 load가 1

일 때의 부분은 주석처리를 했다. inc가 0인지 1인지에 따라 결과값이 잘 출력됨을 확인할 수 있다.

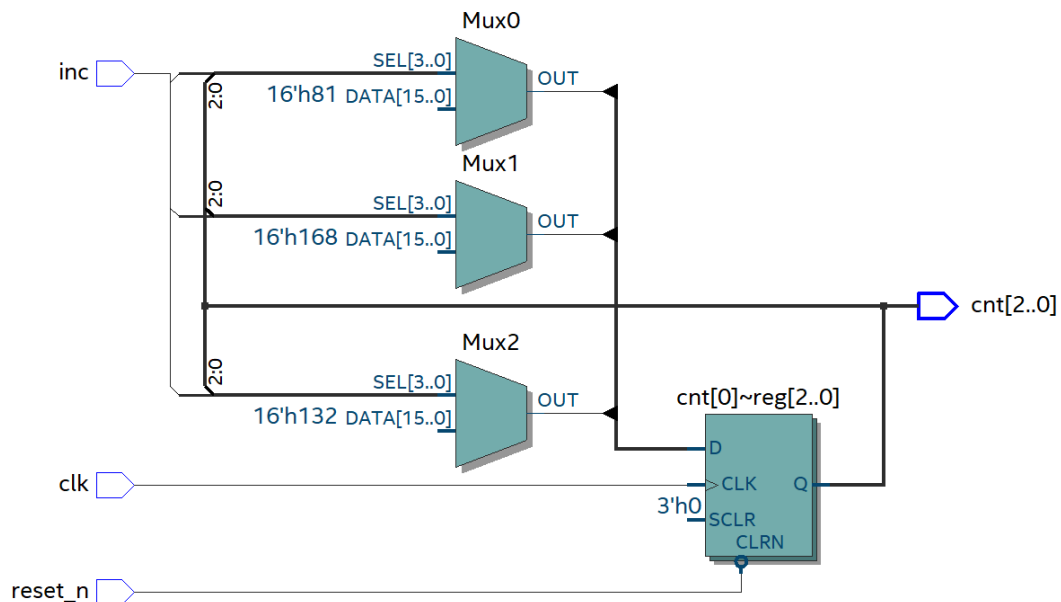
- cla8



Sub module의 testbench이다. Cla4 module 2개로 만들었으며, Ci가 0, 1일 때로 나누어 코드를 구현한 testbench이다.

B. 합성(synthesis) 결과

- cnt5

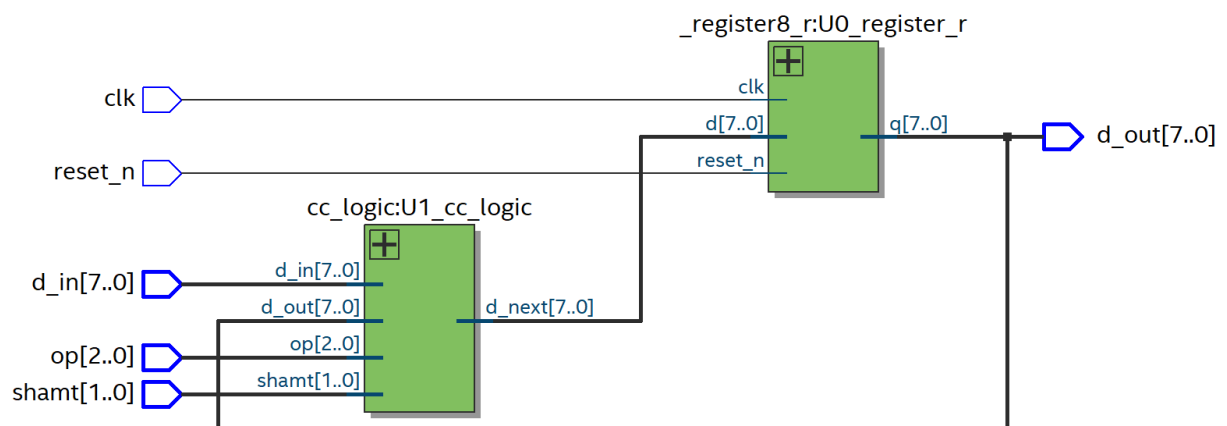


cnt5의 rtl viewer로, mux 3개와 d flip flop이 연결되어 설계된 것을 확인할 수 있다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Oct 18 03:28:13 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	cnt5
Top-level Entity Name	cnt5
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	3
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

Cnt5의 flow summary이다. 총 3개의 register와 6개의 pin을 사용한 것을 알 수 있으며 성공적으로 컴파일이 완료된 것을 알 수 있다.

- shifter 8

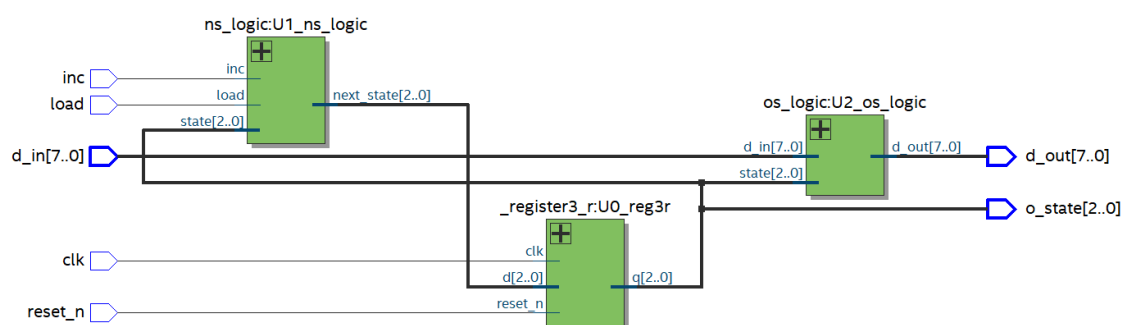


Shifter8의 rtl viewer로, `cc_logic`과 `_register8_r`이 연결되어 설계된 것을 확인할 수 있다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Oct 18 04:11:08 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	shifter8
Top-level Entity Name	shifter8
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	8
Total pins	23
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

Shifter8의 flow summary이다. 총 8개의 register와 23개의 pin을 사용한 것을 알 수 있으며 성공적으로 컴파일이 완료된 것을 알 수 있다.

- Cnrt8



Cnrt8의 rtl viewer로, ns_logic과 _register3_r, os_logic이 연결되어 설계된 것을 확인할 수 있다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Tue Oct 18 04:44:44 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	cntr8
Top-level Entity Name	cntr8
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	3
Total pins	23
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

Cntr8의 flow summary이다. 총 3개의 register와 23개의 pin을 사용한 것을 알 수 있으며 성공적으로 컴파일이 완료된 것을 알 수 있다.

5. 고찰 및 결론

A. 고찰

Ppt에 제공된 5-way counter, 8-bit shifter, 8-bit loadable up/down Counter의 RTL viewer를 참고하여 코드를 짜니, 비교적 쉽게 구현할 수 있었다. Shift 하기 위해 사용하는 op 코드, 즉 명령어를 어셈블리 실습 시간에도 사용해 친숙하게 느껴졌다. 또한 1학기, 디지털 논리회로 1에서 배웠던 것들을 2학기 실험을 통해 복습하고 이론으로만 이해했던 것들을 직접 코드로 구현하는 과정이 재밌고 그때 미처 이해하지 못했던 부분들을 이해할 수 있는 것 같아 좋았다. 강의를 통해 이론을 이해하고 실습을 진행하는데, 모르는 부분은 여러 번 돌려보며 이해할 때 까지 들을 수 있어 좋은 것 같다. 또한 실습이 진행될수록 어려워지는 것이 느껴져 겁이 났지만, 앞서 말한 점이 장점으로 적용되는 것 같다. 이번 실습을 통해 Moore FSM과 Mealy FSM 그리고 counter에 대해 보다 자세히 알 수 있었다.

B. 결론

loadable counter 와 ring counter의 장단점 및 응용분야에 대해 말해보면, 우선 loadable

counter는 load가 가능한 카운터이며 ring counter는 전체적으로 데이터가 회전하는 카운터이다. 먼저 loadable counter는 외부에서 시작 값을 받아들일 수 있으며 그 값으로부터 하나씩 카운트 해 나가는 카운터이다. Loadable counter의 장점은 먼저 multiplexer 없이 사용할 수 있다는 것이다. 두 번째로, 조합 회로의 최적화로 데이터의 증감이 가능해 회로의 면적을 줄일 수 있다. Loadable counter의 단점은 decoding을 해야 한다는 것이다. 따라서 Loadable counter는 각 변수의 따라 다른 state를 가진다는 또다른 장점이 있는데 그런 분야에서 사용될 것 같다. 다음으로 ring counter는 일정 시간 동안 1개의 flip-flop 만 1이 되고 나머지는 0이 되는 카운터이다. Ring counter의 장점은 먼저 디코딩이 필요하지 않다는 것이다. 두 번째로 ring counter의 delay는 일정하다. 하지만 디코딩이 필요 없는 대신, loadable counter가 없을 시 load값을 입력해주어야 한다는 단점이 있다. 따라서 ring counter는 순차적으로 작동하는 장치에서 사용될 것 같다.

다음으로 barrel shifter에 대해 말해보면, barrel shifter는 순차논리를 사용하지 않고 한 번의 연산으로 bit 수를 이동하거나 회전, shift 할 수 있는 카운터이다. Barrel shifter는 조합 논리를 사용해 n-bit의 길이를 가지는 register를 n-bit 만큼 shift 시키려면 multiplexer $n \log_2 n$ 개가 필요하며 bandwidth는 n개가 필요하다.

6. 참고문헌

공영호 교수님/디지털논리회로2 강의자료/광운대학교 컴퓨터 공학과 2022 강의자료
공영호 교수님/컴퓨터공학기초실험2/광운대학교/2022 강의자료