컴퓨터 공학 기초 실험2 보고서

실험제목: Traffic Light Controller with/without Left Turn Signals

실험일자: 2023년 10월 9일 (월)

제출일자: 2023년 10월 21일 (일)

학 과: 컴퓨터정보공학부

담당교수: 이준환 교수님

실습 분반: 월요일 0, 1, 2

학 번: 2020202031

성 명: 김재현

1. 제목 및 목적

A. 제목

Traffic Light Controller with/without Left Turn Signal

B. 목적

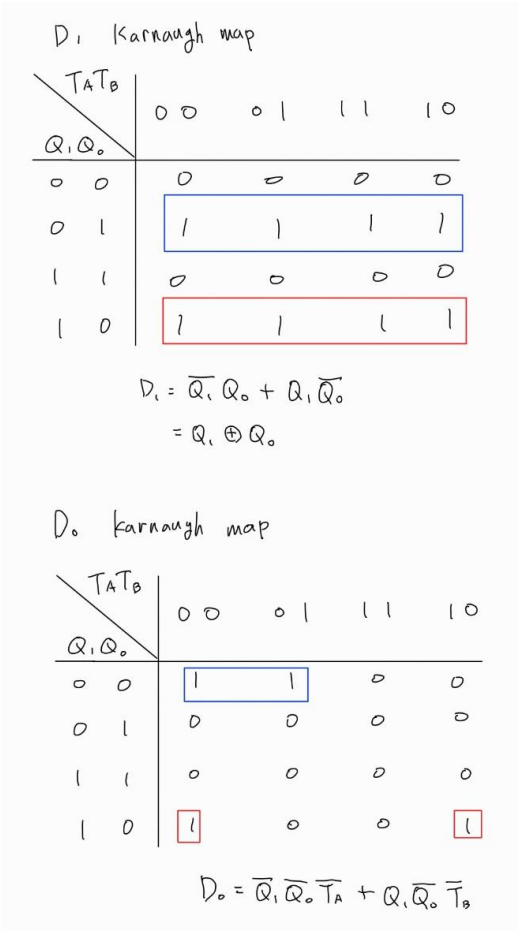
이번 실습에서는 FSM 의 기법 중 하나인 Moore FSM 을 적용하여 traffic light controller 를 설계한다. Behavioral Implementation로도 traffic light controller를 설계해보고 이를 Structural Implementation과 비교해본다.

2. 원리(배경지식)

A. Traffic Light Controller

1) ns_logic

ns_logic은 current state 값과 Ta, Tb에 따라 next state 값을 정해주는 회로입니다. 다음은 Q1, Q0, Ta, Tb에 대한 D1, D0의 카르노 맵과 그에 따른 논리식입니다.

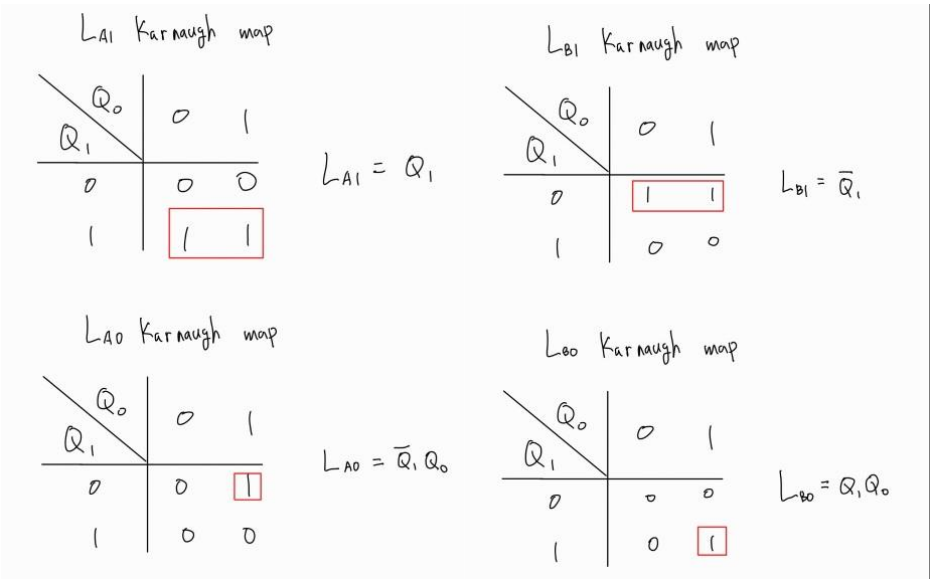


2) _register2_r_async

clk이 rising 될 때, 입력 D 값을 출력 Q로 내보내는 D flipflop register입니다. 이전에 구현한 _dff_r_async은 1비트짜리이므로 _dff_r_async를 2개 사용하여 2비트짜리 값을 다루는 register를 구현해줍니다.

3) o_logic

o_logic은 current state 값에 따라 La, Lb 값을 정해주는 회로입니다. 다음은 Q1, Q0에 대한 La, Lb의 카르노 맵과 그에 따른 논리식입니다.



B. Traffic Light Controller With Left Turn Signal

1) ns_logic

D2, D1, D0를 카르노맵으로 구하기엔 input이 Q2, Q1, Q0, Ta, Tal, Tb, Tbl 총 7개로, 너무 많기 때문에 Quine-McCluskey algorithm을 사용하여 구합니다. 1학기에 디지털논리회로1 과제로 구현했던 QM프로그램을 사용하여 D에 대한 논리식을 구해줬습니다.

- D2

```
1    011-----
2    10------
3    1-0-----
4
5    Cost (# of transistor): 34
```

$$D_2 = \overline{Q_2}Q_1Q_0 + Q_2\overline{Q_1} + Q_2\overline{Q_0}$$

- D1

```
-01----
-10----

Cost (# of transistor): 22
```

$$D_1 = Q_1 \oplus Q_0$$

- D0

```
0000---
010-0--
100--0-
110---0

Cost (# of transistor): 64
```

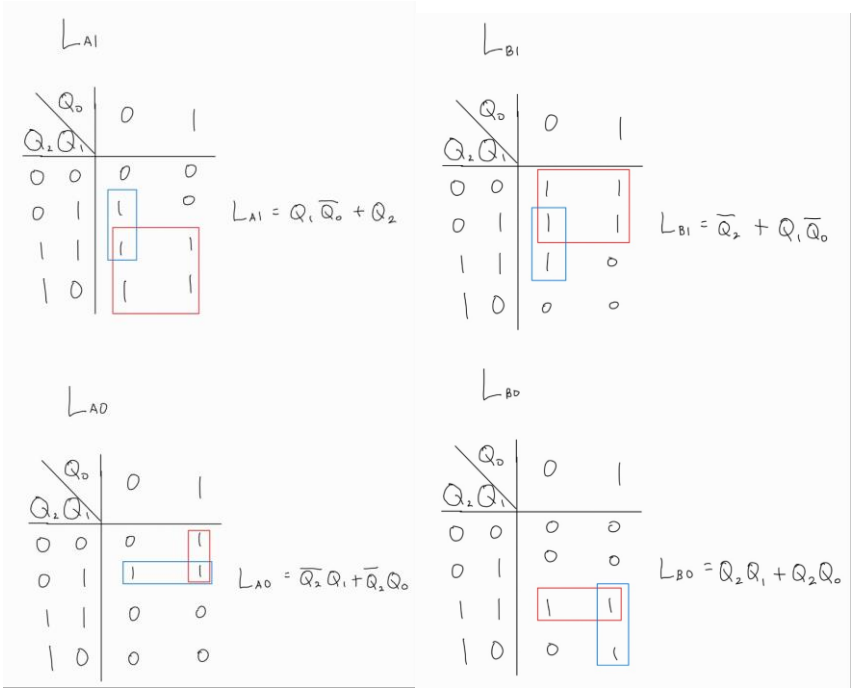
$$D_2 = \overline{Q_2 Q_1 Q_0 T_A} + \overline{Q_2} Q_1 \overline{Q_0 T_{AL}} + Q_2 \overline{Q_1 Q_0 T_B} + Q_2 Q_1 \overline{Q_0 T_{BL}}$$

2) _register3_r

clk이 rising 될 때, 입력 D 값을 출력 Q로 내보내는 D flipflop register입니다. 이전에 구현한 _dff_r_async은 1비트짜리이므로 _dff_r_async를 3개 사용하여 3비트짜리 값을 다루는 register를 구현해줍니다.

3) o_logic

o_logic은 current state 값에 따라 La, Lb 값을 정해주는 회로입니다. 다음은 Q1, Q0에 대한 La, Lb의 카르노 맵과 그에 따른 논리식입니다.

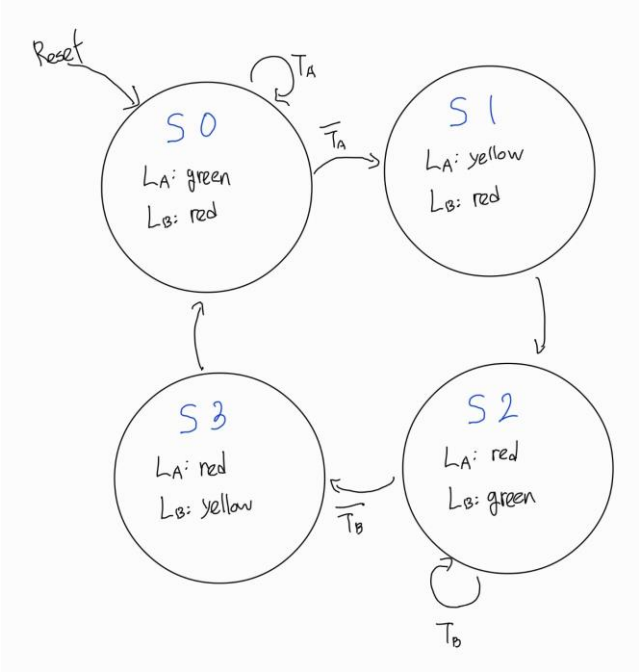


3. 설계 세부사항

A. Traffic Light Controller

i. Structural Implementation

- State Transition Diagram



- Encoded State Transition Table

State	Encoding
S0	00
S1	01
S2	10
S3	11

Color	Encoding
Green	00
Yellow	01
Red	10

Current state		Inputs		Next state	
Q1	Q0	TA	TB	D1	D0
0	0	0	X	0	1
0	0	1	X	0	0
0	1	X	X	1	0
1	0	X	0	1	1
1	0	X	1	1	0

1	1	X	X	0	0

- Output Table

Current state		Outputs			
Q1	Q0	LA1	LA0	LB1	LB0
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	0	0
1	1	1	0	0	1

1) ns_logic

ns_logic에서는 ~Ta, ~Tb, ~Q1, ~Q0을 사용합니다. 따라서 inverter를 통해 각 값들을 Ta_bar, Tb_bar, Q1_bar, Q0_bar wires에 저장합니다. 그리고 원리(배경지식)에서 도출한 D1, D0의 논리식을 바탕으로 3 input and gate, 2 input or gate, 2 input xor gate 등을 활용하여 D1, D0에 적절한 값을 저장합니다.

2) o_logic

원리(배경지식)에서 도출한 La1, La0, Lb1, Lb0의 논리식을 바탕으로 inverter, 2 input and gate 등을 활용하여 La, Lb에 적절한 값을 저장합니다.

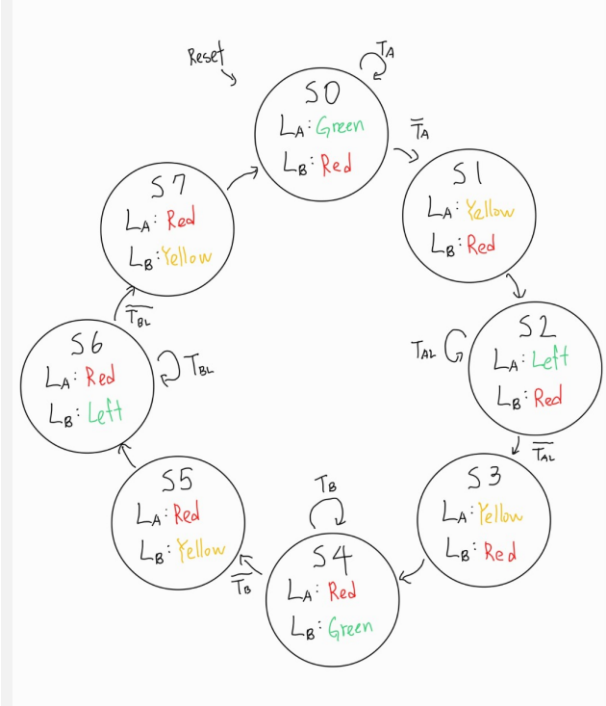
ii. Behavioral Implementation

clk의 posedge와 reset_n의 negedge가 나올 때 현재 state 값을 변화시킵니다. 또한 state와 Ta, Tb의 변화가 감지되면 next_state 값을 변화시킵니다.
마지막으로 state의 변화가 감지되면 현재 state에 따른 신호등 색을 변경시켜줍니다.

B. Traffic Light Controller with Left Turn Signals

i. Structural Implementation

- State Transition Diagram



- Encoded State Transition Table

State	Encoding
S0	000
S1	001
S2	010
S3	011
S4	100
S5	101
S6	110
S7	111

Color	Encoding
Green	00
Yellow	01
Red	10
Left	11

Current State	Inputs				Next State
Q	TA	TAL	TB	TBL	D
S0	1	X	X	X	S0

S0	0	X	X	X	S1
S1	X	X	X	X	S2
S2	X	1	X	X	S2
S2	X	0	X	X	S3
S3	X	X	X	X	S4
S4	X	X	1	X	S4
S4	X	X	0	X	S5
S5	X	X	X	X	S6
S6	X	X	X	1	S6
S6	X	X	X	0	S7
S7	X	X	X	X	S0

- Output Table

Q2	Q1	Q0	LA1	LA0	LB1	LB0
0	0	0	0	0	1	0
0	0	1	0	1	1	0
0	1	0	1	1	1	0
0	1	1	0	1	1	0
1	0	0	1	0	0	0
1	0	1	1	0	0	1
1	1	0	1	0	1	1
1	1	1	1	0	0	1

1) ns_logic

ns_logic에서는 ~Ta, ~Tb, ~Q1, ~Q0을 사용합니다. 따라서 inverter를 통해 각 값들을 Ta_bar, Tb_bar, Q1_bar, Q0_bar wires에 저장합니다. 그리고 원리(배경지식)에서 도출한 D1, D0의 논리식을 바탕으로 3 input and gate, 2 input or gate, 2 input xor gate 등을 활용하여 D1, D0에 적절한 값을 저장합니다.

2) o_logic

원리(배경지식)에서 도출한 La1, La0, Lb1, Lb0의 논리식을 바탕으로 inverter, 2 input and gate 등을 활용하여 La, Lb에 적절한 값을 저장합니다.

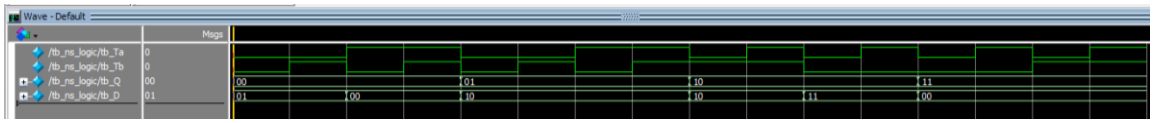
4. 설계 검증 및 실험 결과

A. 시뮬레이션 결과

i. Traffic Light Controller

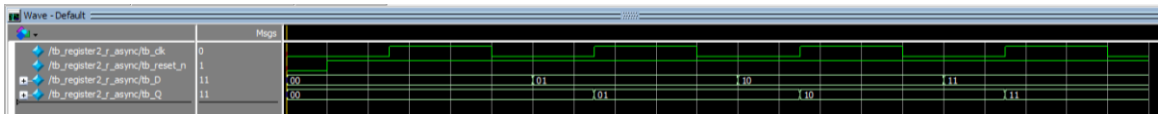
- Structural Implementation

1) tb_ns_logic



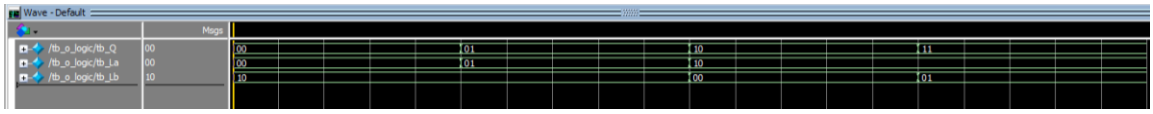
tb_ns_logic의 waveform입니다. Q가 00이면 Ta가 1일 때 D가 00, Ta가 0일 때 D가 01이 됩니다. Q가 01이면 D가 10이 됩니다. Q가 10이면 Tb가 1일 때 D가 10, Tb가 0일 때 D가 11이 됩니다. Q가 11이면 D가 00이 됩니다.

2) tb_register2_r_async



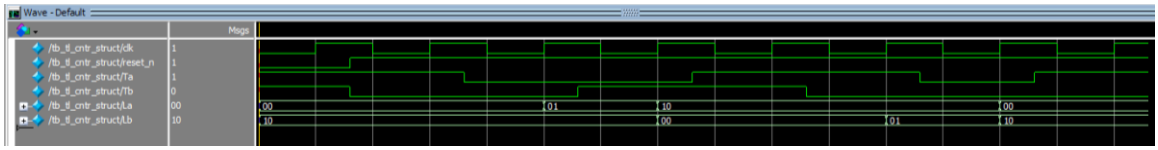
tb_register2_r_async의 waveform입니다. tb_clk이 rising 될 때마다 tb_D가 tb_Q를 따라가는 것을 확인할 수 있습니다.

3) tb_o_logic



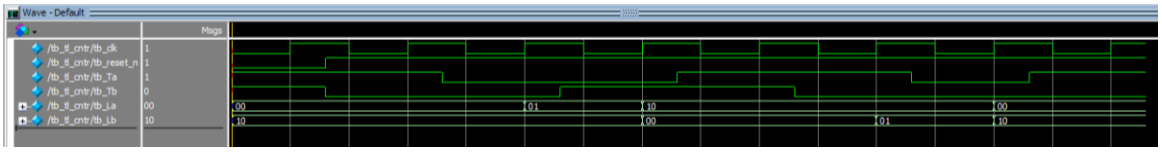
tb_o_logic의 waveform입니다. tb_Q가 S0일 때, La: green, Lb: red입니다. tb_Q가 S1일 때, La: yellow, Lb: red입니다. tb_Q가 S2일 때, La: red, Lb: green입니다. tb_Q가 S3일 때, La: red, Lb: yellow입니다.

4) tb_tl_cntr_struct



tb_tl_cntr_struct의 waveform입니다. 테스트벤치 시작과 동시에 reset_n이 0이므로, current state가 00 즉, S0입니다. 따라서 La가 초록불 Lb가 빨간불이며, Ta가 1일 땐, Tb의 값과 상관없이 현상태를 유지합니다. 하지만 Ta가 0이 되면, clk이 rising 함과 동시에 current state가 S1로 넘어가고, La는 노란불, Lb는 빨간불이 됩니다. 다음 clk가 rising 할 때는 Ta, Tb 값에 상관 없이 current state가 S2로 넘어갑니다. S2에서는 La가 빨간불, Lb가 초록불이며, Tb가 1이면 Ta 값에 상관 없이 현상태를 유지하며, Tb가 0이면 clk가 rising함에 따라 current state가 S3로 넘어갑니다. 여기서는 La가 빨간불, Lb가 노란불이며, 다음 clk rising에 S0로 넘어갑니다.

- Behavioral Implementation

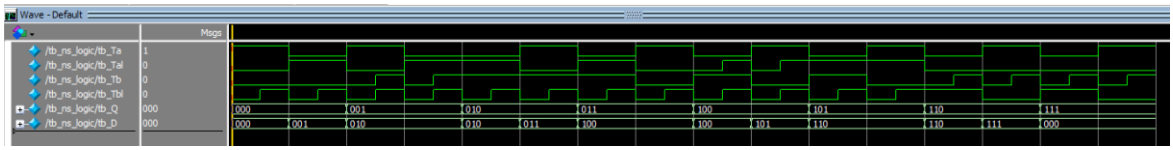


tb_tl_cntr의 waveform입니다. 테스트벤치 경우의 수는 tb_tl_cntr_struct와 동일하게 부여하여 정확한 비교가 가능하게 만들었습니다. waveform 확인 결과, tb_tl_cntr과 tb_tl_cntr_struct의 waveform이 완벽히 일치하는 것을 확인함으로써 tl_cntr이 잘 작동함을 알 수 있습니다.

ii. Traffic Light Controller With Left Turn Signal

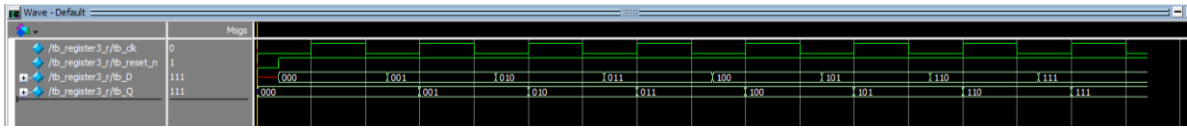
- Structural Implementation

1) tb_ns_logic



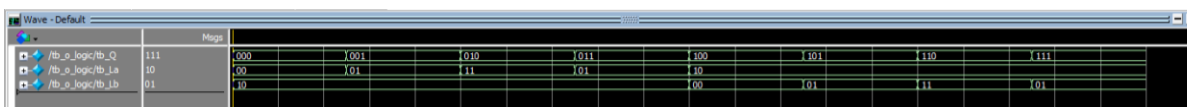
tb_ns_logic의 waveform입니다. Q가 000이면 Ta가 1일 때 D가 000, Ta가 0일 때 D가 001이 됩니다. Q가 001이면 D가 010이 됩니다. Q가 010이면 Tal이 1일 때 D가 010, Tal이 0일 때 D가 011이 됩니다. Q가 011이면 D가 100이 됩니다. Q가 100일 때 Tb가 1일 때 D가 100이 됩니다. Tb가 0일 때 D가 101가 됩니다. Q가 101일 때, D가 110이 됩니다. Q가 110일 때, Tbl이 1일 때 D가 110이 됩니다. Tbl이 0일 때 D가 111이 됩니다. Q가 111이면 D가 000이 됩니다.

2) tb_register3_r



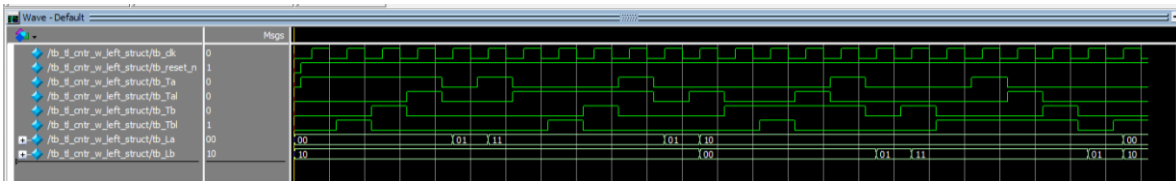
tb_register3_r 의 waveform입니다. reset_n이 0이면 D 값에 상관없이 Q가 0이 되고, tb_clk이 rising 될 때마다 tb_D가 tb_Q를 따라가는 것을 확인할 수 있습니다.

3) tb_o_logic



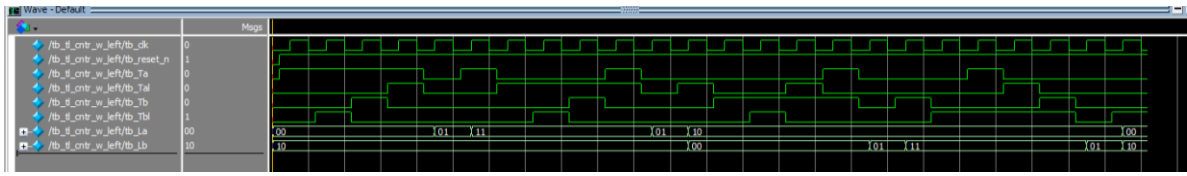
tb_o_logic의 waveform입니다. tb_Q가 S0일 때, La: green, Lb: red입니다. tb_Q가 S1일 때, La: yellow, Lb: red입니다. tb_Q가 S2일 때, La: left, Lb: red입니다. tb_Q가 S3일 때, La: yellow, Lb: red입니다. tb_Q가 S4일 때, La: red, Lb: green입니다. tb_Q가 S5일 때, La: red, Lb: yellow입니다. tb_Q가 S6일 때, La: red, Lb: left입니다. tb_Q가 S7일 때, La: red, Lb: yellow입니다.

4) tb_tl_cntr_w_left_struct



tb_tl_cntr_w_left_struct의 waveform입니다. 설계 세부사항에서 diagram으로 표현한 것과 동일한 waveform이 형성되는 것을 확인할 수 있습니다. Q의 current state와, Ta, Tal, Tb, Tbl 신호에 따라 next state인 D가 결정되고, Q의 current state에 따른 La, Lb가 결정되고 곧, 신호등의 색이 결정되게 됩니다.

- Behavioral Implementation

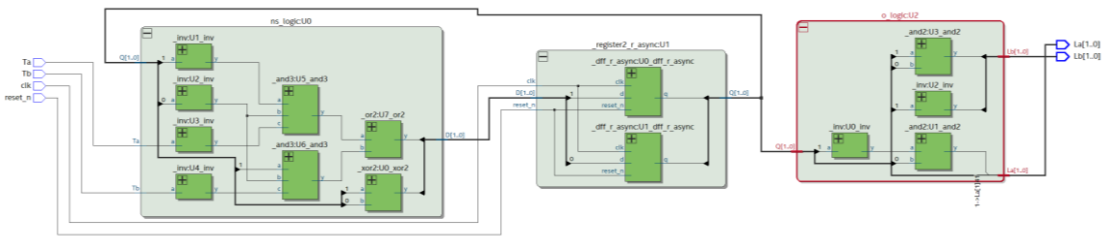


tb_tl_cntr_w_left의 waveform입니다. 테스트벤치 경우의 수는 tb_tl_cntr_w_left와 동일하게 부여하여 정확한 비교가 가능하게 만들었습니다. waveform 확인 결과, tb_tl_cntr과 tb_tl_cntr_struct의 waveform이 완벽히 일치하는 것을 확인함으로써 tl_cntr이 잘 작동함을 알 수 있습니다.

B. 합성(synthesis) 결과

1. Traffic Light Controller

A. Structural Implementation



tl_cntr_struct의 RTL Viewer입니다. ns_logic에서 Ta, Tb, Q을 입력으로 D값을

출력하여 register로 D를 입력합니다. register에서는 clk이 rising 할 때, 입력으로 받은 D 값을 Q로 출력합니다. 이 Q는 ns_logic의 입력으로 들어감과 동시에 o_logic의 입력으로 들어가서 La, Lb 값을 출력하게 됩니다.

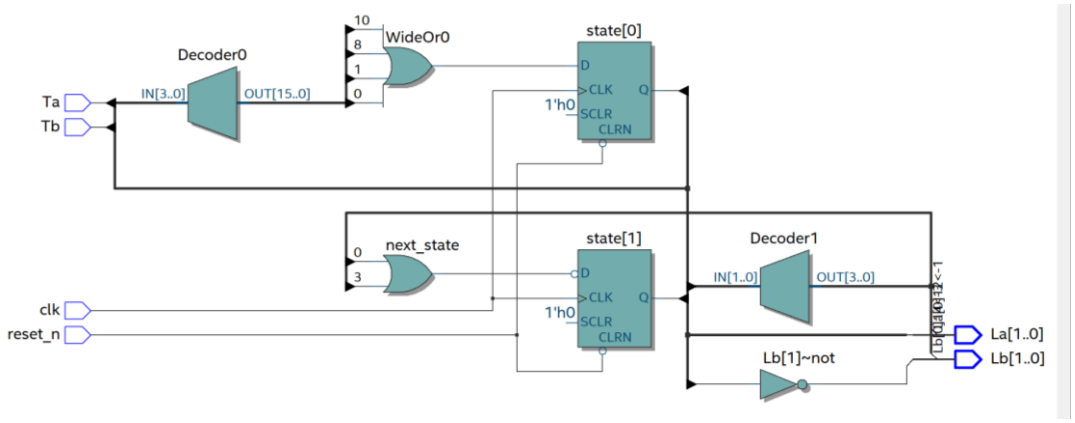
Flow Summary

<<Filter>>

Flow Status	Successful - Thu Oct 19 21:57:34 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	tl_cntr_struct
Top-level Entity Name	tl_cntr_struct
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	3 / 41,910 (< 1 %)
Total registers	3
Total pins	8 / 499 (2 %)
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 %)

tl_cntr_struct의 Flow Summary입니다. Total pins는 clk 1bit + reset_n 1bit + Ta 1bit + Tb 1bit + La 2bits + Lb 2bits = 8bits 입니다.
Logic utilization은 3임을 확인할 수 있습니다.

B. Behavioral Implementation



tl_cntr의 RTL Viewer입니다. Decorder0은 reset_n이 1이고, clk이 rising할 때 Ta, Tb, state 총 4비트를 input으로 16bit를 출력합니다. 그리고 적절한 값을 dff(state[0])에 입력해줍니다. dff(state[1])은 next_state 값을 입력으로 state를 출력합니다. 현재 state 값을 Decorder1에 입력으로 하여 적절한 La, Lb 값을 결정합니다.

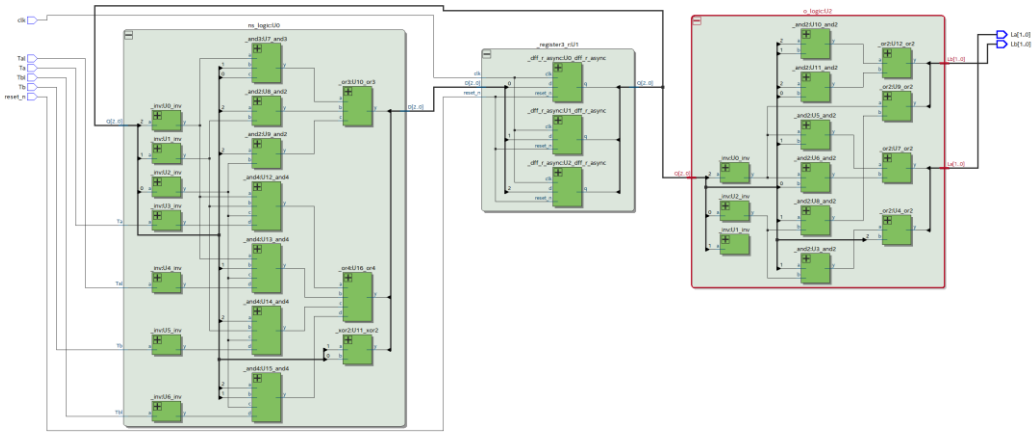
Flow Summary	
<<Filter>>	
Flow Status	Successful - Fri Oct 20 13:43:18 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	tl_cntr
Top-level Entity Name	tl_cntr
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	3 / 41,910 (< 1 %)
Total registers	3
Total pins	8 / 499 (2 %)
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 %)

tl_cntr의 Flow Summary입니다. Total pins는 clk 1bit + reset_n 1bit + Ta 1bit + Tb 1bit + La 2bits + Lb 2bits = 8bits 입니다.

Logic utilization은 3임을 확인할 수 있습니다.

2. Traffic Light Controller With Left Turn Signal

   A. Structural Implementation



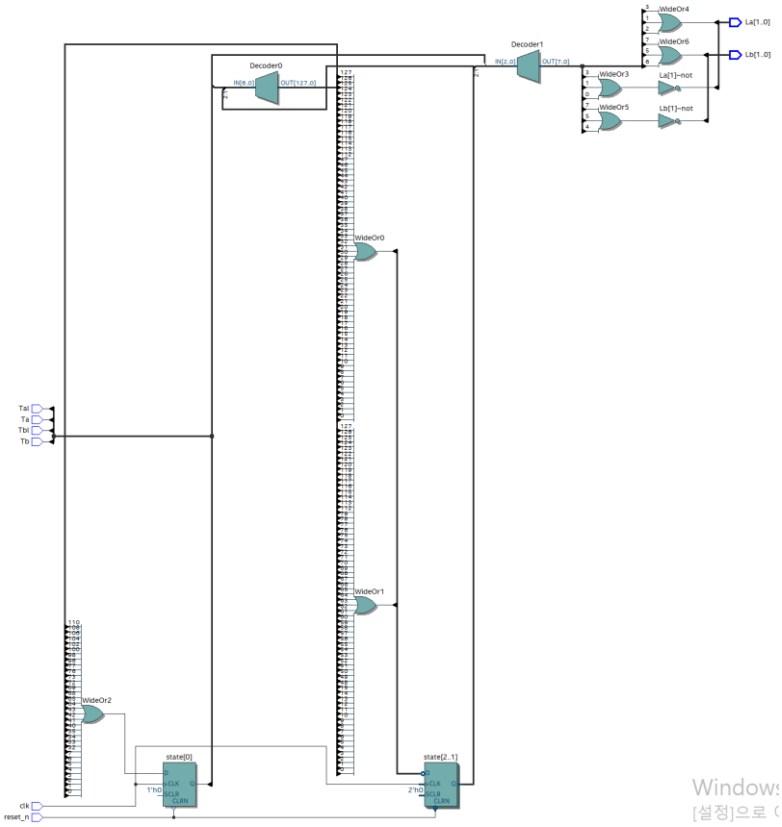
tl_cntr_w_left_struct의 RTL Viewer입니다. ns_logic에서 Ta, Tal, Tb, Tbl, Q을 입력으로 D값을 출력하여 register로 D를 입력합니다. register에서는 clk이 rising 할 때, 입력으로 받은 D 값을 Q로 출력합니다. 이 Q는 ns_logic의 입력으로 들어감과 동시에 o_logic의 입력으로 들어가서 La, Lb 값을 출력하게 됩니다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Fri Oct 20 20:46:56 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	tl_cntr_w_left_struct
Top-level Entity Name	tl_cntr_w_left_struct
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	5 / 41,910 (< 1 %)
Total registers	4
Total pins	10 / 499 (2 %)
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 %)

tl_cntr_w_left_struct의 Flow Summary입니다. Total pins는 clk 1bit + reset_n 1bit + Ta 1bit + Tal 1bit + Tb 1bit + Tbl 1bit + La 2bits + Lb 2bits = 10bits 입니다.

Logic utilization은 5임을 확인할 수 있습니다.

B. Behavior Implementation



tl_cntr_w_left의 RTL Viewer입니다. 감지센서가 Tal, Tbl이 추가됐고, State가

S0~S7로 늘어난 것을 제외하고는 tl_cntr_의 RTL Viewer와 동일한 구조임을 확인할 수 있습니다.

Flow Summary

<<Filter>>

Flow Status	Successful - Sat Oct 21 15:21:37 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	tl_cntr_w_left
Top-level Entity Name	tl_cntr_w_left
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	5 / 41,910 (< 1 %)
Total registers	4
Total pins	10 / 499 (2 %)
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 %)

tl_cntr_w_left의 Flow Summary입니다. Total pins는 clk 1bit + reset_n 1bit + Ta 1bit + Tal 1bit + Tb 1bit + Tbl 1bit + La 2bits + Lb 2bits = 10bits 입니다. Logic utilization은 5임을 확인할 수 있습니다.

5. 고찰 및 결론

A. 고찰

ns_logic에서는 current state인 Q가 입력, next_state인 D가 출력입니다. _register2_r_async에서는 next_state인 D가 입력, current state인 Q가 출력입니다. o_logic에서는 current state인 Q가 입력이고, 신호등의 색을 결정하는 La, Lb가 출력입니다. 이처럼 각 회로에서 D, Q 값이 입력으로 쓰이기도 하고, 출력으로 쓰이기도 해서 헷갈리는 부분이 있었지만 각 모듈에서 입출력 값을 잘 설정해주었더니 문제 없이 잘 작동했습니다.

B. 결론

Moore Finite State Machine을 적용하여 traffic light controller w/wo left turn signal을 설

계하고, 베릴로그를 통해 설계한 회로를 직접 구현해 봄으로써 combinational circuit, sequential circuit을 코딩하는 방법을 체득할 수 있었습니다. Behavioral Implementation로 TLC를 구현해 봄으로써 case문을 사용하는 법 또한 터득할 수 있었습니다.

6. 참고문헌

이준환 교수님/컴퓨터공학기초실험2/광운대학교(컴퓨터정보공학부)/2023
이준환 교수님/디지털논리회로2/광운대학교(컴퓨터정보공학부)/2023