컴퓨터 공학 기초 실험2 보고서

실험제목: Traffic Light Controller with/without Left Turn Signals

실험일자: 2023년 10월 06일 (금)

제출일자: 2023년 10월 20일 (금)

학 과: 컴퓨터공학과

담당교수: 이형근 교수님

실습분반: 금요일 0, 1, 2

학 번: 2020202037

성 명: 염정호

1. 제목 및 목적

A. 제목

Traffic Light Controller with/without Left Turn Signals

B. 목적

finite-state machine 중 하나인 Moore FSM을 이용하여 traffic light controller를 설계한다. 베릴로그를 이용해 struct와 behavior를 이용한 각각의 traffic light를 구현해 보고 추가로 left trun기능을 더한 traffic를 구현한다.

원리(배경지식)

finite-state machine:

유한상태 기계로 컴퓨터 프로그램과 전자 논리 회로를 설계하는 데에 쓰이는 수학적 모델이다.

유한한 개수의 상태를 가질 수 있는 오토마타 이며, 한번에 오로지 하나의 상태만을 갖는다. 현재 상태란 임의의 주어진 시간의 상태를 칭한다.

FSM은 event에 의해 다른 상태로 변화 할 수 있으며, 이를 전이라 한다.

시작상태 : 유한 오토마타가 입력값을 처리 하기 전의 상태이다.

진입동작 : 상태에 진입 할 때 수행되는 동작.

퇴장동작 : 상태로부터 벗어날 때 수행되는 동작.

Moore FSM :

이상 이벤트 시스템을 나타내기 위한 수학적 모델 중 하나로 이것은 상태, 입력, 출력, 전이 함수로 구성된다.

상태 **(States)** : 무어 머신은 여러가지 상태를 가질 수 있으며 이러한 상태들은 시스템이 놓여있는 특정한 상황을 나타낸다.

입력 (Inputs): 각 상태에서 시스템 입력이 주어질 수 있다. 입력은 해당 머신의 동작을 제어하고 상태 전이를 발생시킨다.

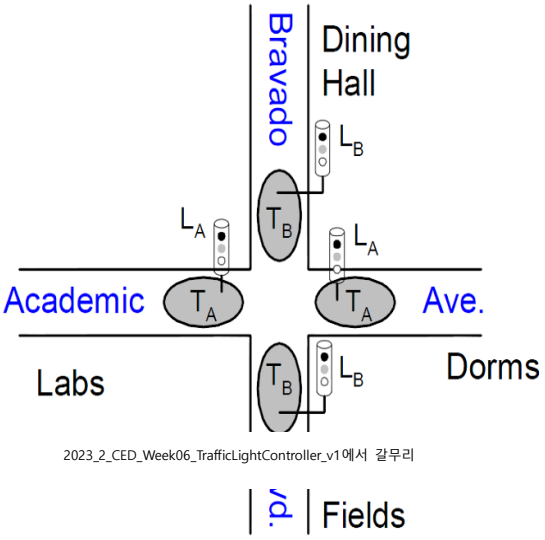
출력 (Outputs): 각 상태 및 입력 조합에 대해 시스템은 특정 출력을 생성할 수 있다.. 출력은 모어 머신의 동작에 따라 외부 환경과의 상호 작용

을 표현한.

전이 (Transitions) 함수: 이 함수는 현재 상태와 입력에 따라 다음 상태로의 전이를 정의한다. 모어 머신은 어떤 상태에서 어떤 상태로 이동할지를 결정하는 데 사용된다.

모어 머신은 이러한 구성 요소들을 사용하여 특정 상황에서 시스템이 어떻게 동작해야 하는지를 정의한다. 이러한 모델은 자동 판별기, 제어 시스템, 각종 논리 회로 등 다양한 응용 분야에서 사용된다. 시스템은 이전 상태의 정보에만 의존하여 다음 상태로 이동하는 Mealy Machine과는 달리, Moore Machine은 이전 상태의 정보만을 기반으로 다음 상태 및 출력을 결정합니다

2. 설계 세부사항
   - **Define state - Traffic Light Controller**



1. Traffic light 는 교통이 없을 때 초록색(00)에서 노란색(01)을 거쳐 빨간색으로 변한다.
2. Traffic light 는 교통이 없을 때 초록색(00)에서 노란색(01)을 거쳐 빨간색으로 변한다.
3. 만약 traffic light LA가 초록색이거나 노란색이면, traffic light LB는 빨간색(10)이다. 반대의 경우도 마찬가지이다.

- **Define inputs - Traffic Light Controller**

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

K-map

D1

TATB / Q1Q2	00	01	11	10
00				
01	o	o	o	o
11				
10	o	o	o	o

D1

TATB / Q1Q2	00	01	11	10
00	o	o		
01				
11				
10	o			o

$$D1 = Q1 \oplus Q0$$
$$D0 = \overline{Q1Q0TA} + Q1\overline{Q0TB}$$

- **Define outputs - Traffic Light Controller**

Current state		Outputs			
Q1	Q0	LA[1]	LA[0]	LB[1]	LB[0]
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	0	0
1	1	1	0	0	1

k - map

LA1

Q1 \ Q0	0	1
0		
1	o	o

LA0

Q1 \ Q0	0	1
0		o
1		

LB1

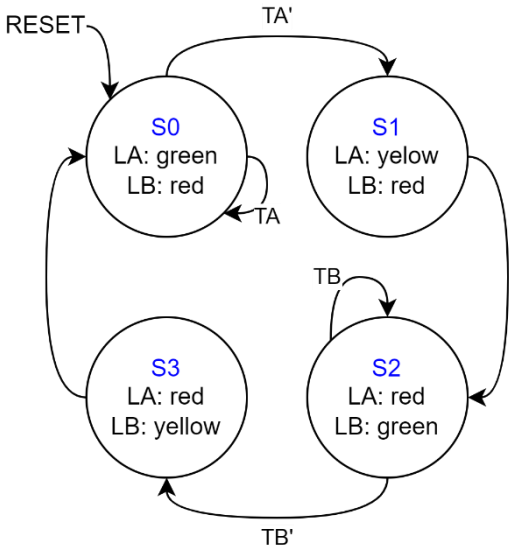
Q1 \ Q0	0	1
0	o	o
1		

LB0

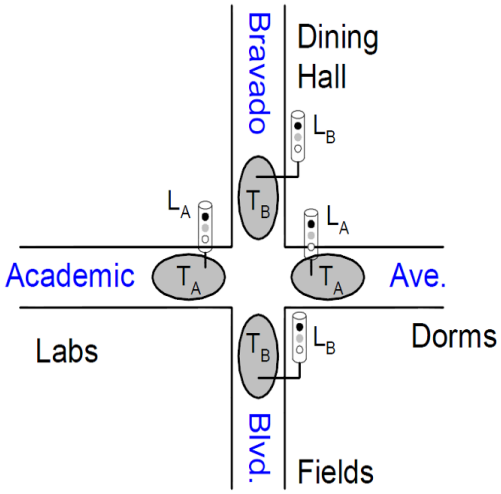
Q1 \ Q0	0	1
0		
1		o

$$\boldsymbol{LA1} = \boldsymbol{Q1}$$
$$\boldsymbol{LA0} = \overline{\boldsymbol{Q1}}\ \boldsymbol{Q0}$$
$$\boldsymbol{LB1} = \overline{\boldsymbol{Q1}}$$
$$\boldsymbol{LB0} = \boldsymbol{Q1}\ \boldsymbol{Q0}$$

- **Draw diagram - Traffic Light Controller**



- **Define state - Traffic Light Controller with Left Turn signals**



1. Traffic light는 교통이 없을 때 초록색(00)에서 노란색(01)을 거쳐 좌회전으로 변한다.
2. Traffic light는 교통이 없을 때 좌회전에서 노란색(01)을 거쳐 빨간색(10)으로 변한다.
3. Traffic light는 비록 좌회전하는 교통이 없더라도 초록색(00)에서 좌회전으로 우선 변해야 한다.

- **Define inputs - Traffic Light Controller**

Q2	Q1	Q0	TA	TAL	TB	TBL	D2	D1	D0
0	**0**	**0**	**0**	**X**	**X**	**X**	**0**	**0**	**1**
0	**0**	**0**	**0**	**X**	**X**	**X**	**0**	**0**	**0**

0	0	1	1	X	X	X	0	1	0
0	1	0	0	0	X	X	0	1	1
0	1	0	0	1	X	X	0	1	0
0	1	1	1	X	X	X	1	0	0
1	0	0	0	X	0	X	1	0	1
1	0	0	0	X	1	X	1	0	0
1	0	1	1	X	X	X	1	1	0
1	1	0	0	X	X	0	1	1	1
1	1	0	0	X	X	1	1	1	0
1	1	1	1	X	X	X	0	0	0

$$D2 = \overline{Q_2}Q_1Q_0 + Q_2\overline{Q_1Q_0T_B} + Q_2\overline{Q_1Q_0}T_B + Q_2\overline{Q_1}Q_0 + Q_2Q_1\overline{Q_0T_{BL}} + Q_2Q_1\overline{Q_0}T_{BL}$$

$$= \overline{Q_2}Q_1Q_1 + \overline{Q_1}Q_2 + Q_2Q_1\overline{Q_0}$$

$$D1 = \overline{Q_2Q_1}Q_0 + \overline{Q_2}Q_1\overline{Q_0T_{AL}} + \overline{Q_2}Q_1\overline{Q_0}T_{AL} + Q_2\overline{Q_1}Q_0 + Q_2Q_1\overline{Q_0T_{BL}} + Q_2Q_1\overline{Q_0}T_{BL}$$

$$= \overline{Q_2Q_1}Q_0 + Q_1\overline{Q_0} + Q_2\overline{Q_1}Q_0$$

$$D0 = \overline{Q_2}\ \overline{Q_1}\ \overline{Q_0TA} + \overline{Q_2}\ Q_1\ \overline{Q_0T_{AL}} + \overline{Q_2}Q_1\overline{Q_0}T_{AL} + Q_2\overline{Q_1Q_0}T_B + Q_2Q_1\overline{Q_0T_{BL}}$$

- **Define outputs - Traffic Light Controller**

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

L_{A1}

	00	01	11	10
0				1
1	1	1	1	1

L_{A0}

	00	01	11	10
0		1	1	
1	1	1	1	1

L_{B1}

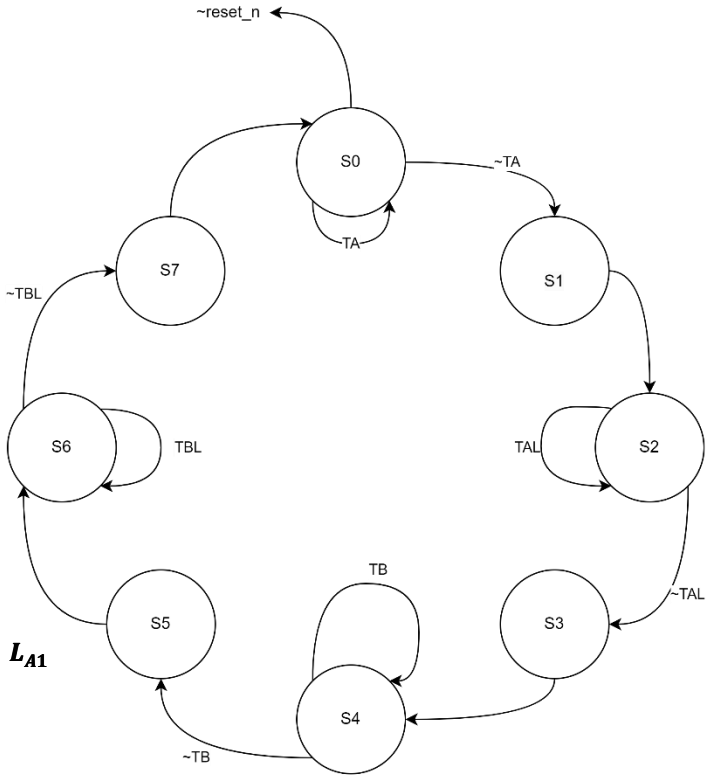
	00	01	11	10
0	1	1	1	1
1	0	0	0	1

L_{B0}

	00	01	11	10
0	1	1	1	1
1	0	1	1	0

$L_{A1} = Q_1\overline{Q_0} + Q_2$
$L_{A0} = Q_0 + Q_2$
$L_{B1} = \overline{Q_2} + Q_1\overline{Q_0}$
$L_{B0} = \overline{Q_2} + Q_0$

- **Draw diagram - Traffic Light Controller**



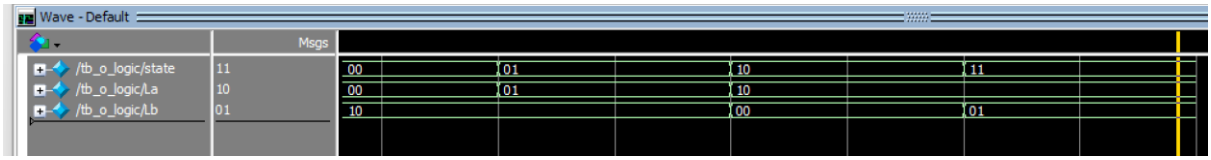
L_{A1}

3. 설계 검증 및 실험 결과

A. 시뮬레이션 결과

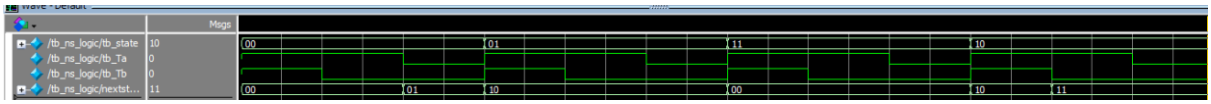
Sub module

o_logic



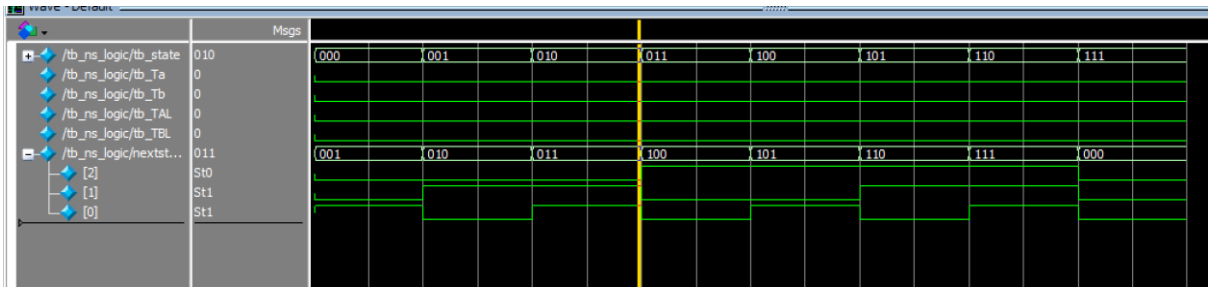
위에서 작성한 output state에 맞게 값이 변화하는 것을 확인 할 수 있다.

ns_logic



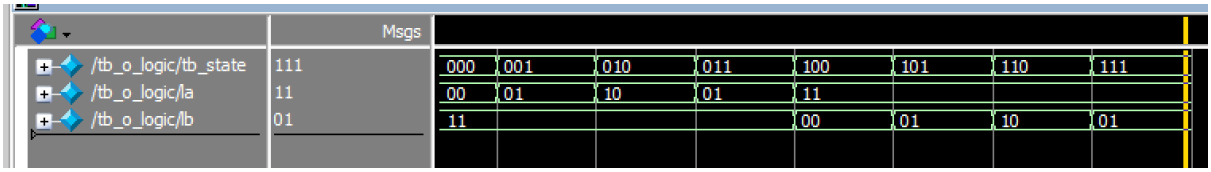
각 state 별로 ta 또는 tb값의 변화에 따라 다음 상태로 넘어가는 것을 확인 할 수 있었다.

ns_logic-turn



State가 차례대로 nextstate값으로 넘어 가는 것을 확인했다.

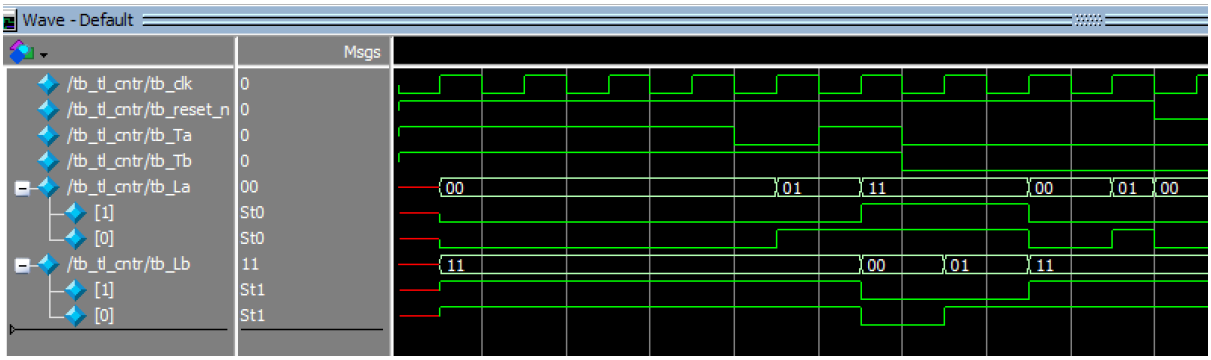
o_logic-turn



위에서 작성한 output state에 맞게 값이 변화하는 것을 확인 할 수 있다.

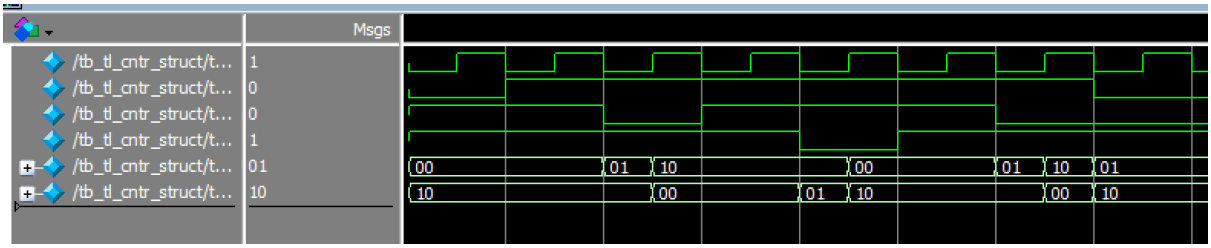
Main module

tl_cntr



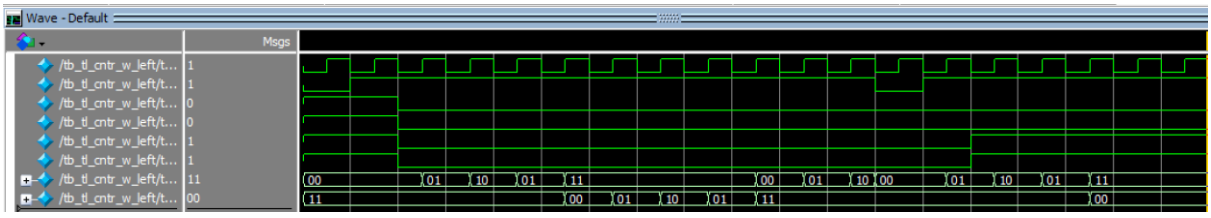
1. La ->green(00) -> yellow(01)-> red(11) 순으로 변경
2. Lb -> red(11) ->green(00) ->yellow(01)
3. Reset 상태시 state 00으로 이동 La: green, Lb: red

tl_cntr_struct



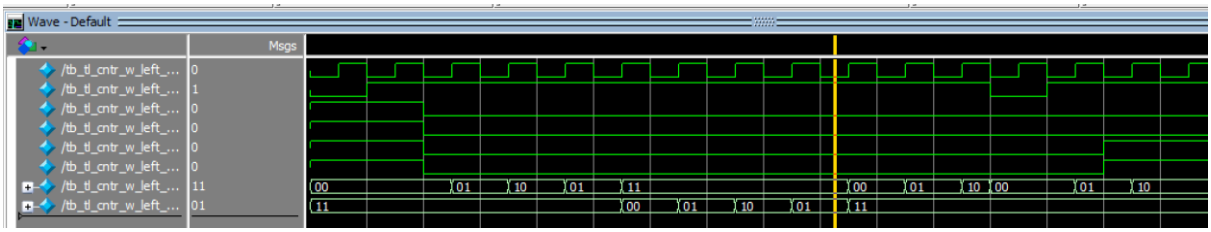
1. La ->green(00) -> yellow(01)-> red(10) 순으로 변경
2. Lb -> red(10) ->green(00) ->yellow(01)
3. Reset 상태시 state 00으로 이동 La: green, Lb: red

tl_cntr_w_left



1. La ->green(00) -> yellow(01)->left(10)-> yellow(01)->red(11) 순으로 변경
2. Lb -> red(11)-> green(00) -> yellow(01)->left(10)-> yellow(01)
3. Reset 상태시 state 00으로 이동 La: green, Lb: red

tl_cntr_w_left_struct



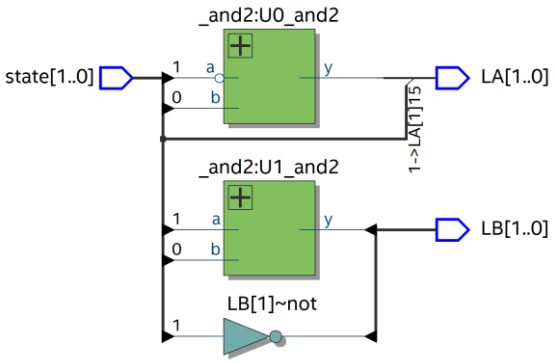
1. La ->green(00) -> yellow(01)->left(10)-> yellow(01)->red(11) 순으로 변경

2. Lb -> red(11)-> green(00) -> yellow(01)->left(10)-> yellow(01)
3. Reset 상태시 state 00으로 이동 La: green, Lb: red

B. 합성(synthesis) 결과

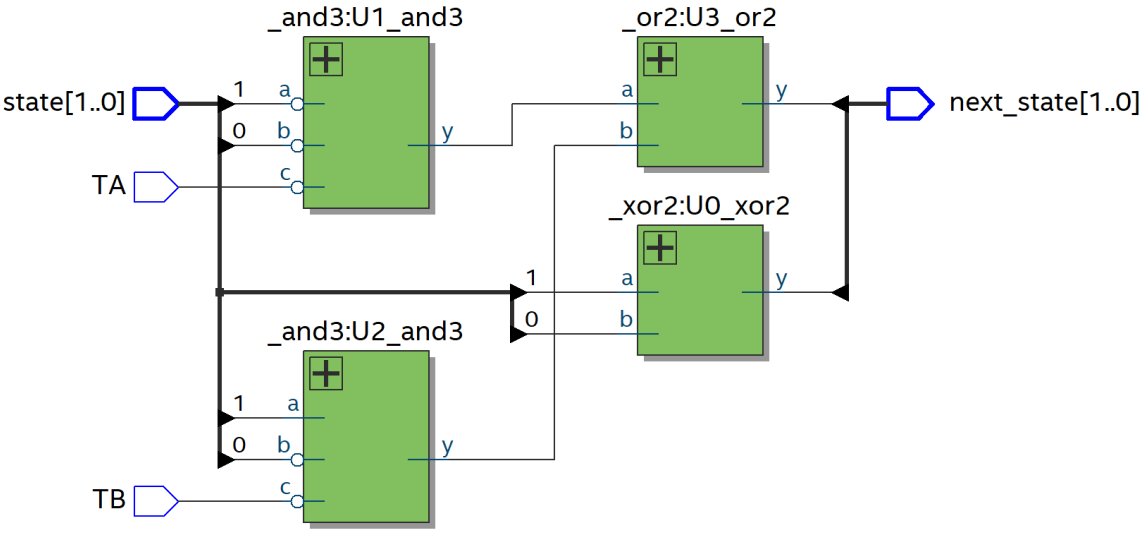
Sub module

o_logic



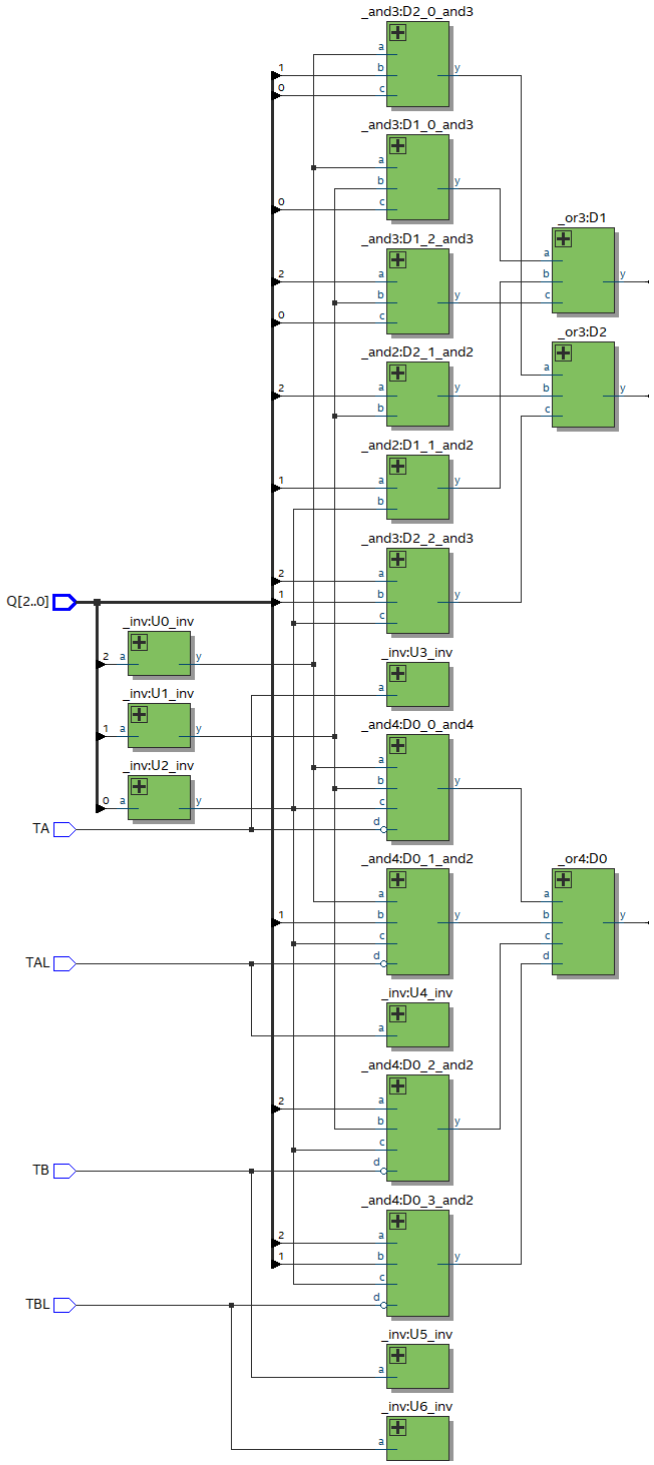
Flow Status	Successful - Fri Oct 20 17:24:20 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	tl_cntr_struct
Top-level Entity Name	o_logic
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	0
Total pins	6

ns_logic



Flow Status	Successful - Fri Oct 20 17:47:20 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	tl_cntr_struct
Top-level Entity Name	ns_logic
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	0
Total pins	6

ns_logic-trun

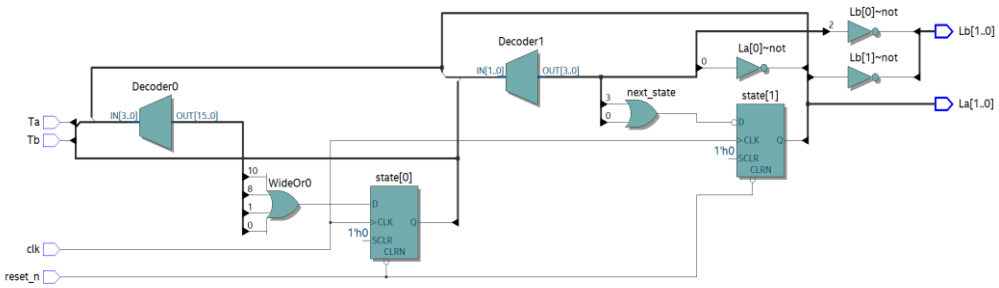

_and3:D2_0_and3
_and3:D1_0_and3
_or3:D1
_and3:D1_2_and3
_or3:D2
_and2:D2_1_and2
_and2:D1_1_and2
_and3:D2_2_and3
Q[2..0]
_inv:U0_inv
_inv:U3_inv
_inv:U1_inv
_and4:D0_0_and4
_inv:U2_inv
TA
_and4:D0_1_and2
_or4:D0
TAL
_inv:U4_inv
_and4:D0_2_and2
TB
_and4:D0_3_and2
TBL
_inv:U5_inv
_inv:U6_inv

Flow Status	Successful - Fri Oct 20 18:11:47 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	ns_logic
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	0
Total pins	10

o_logic

Main module

tl_cntr

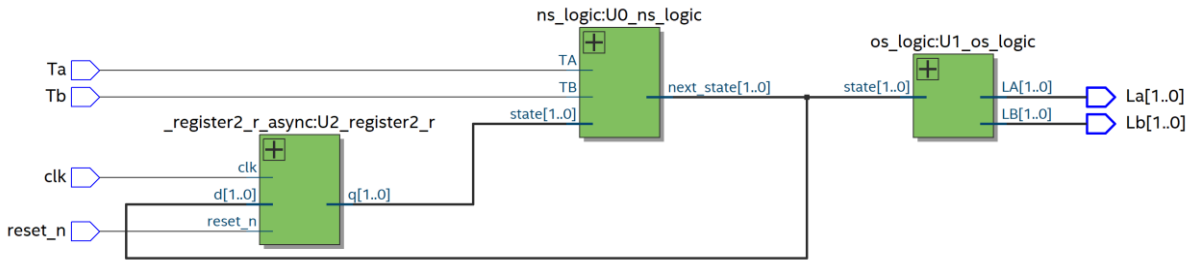


Flow Summary

<<Filter>>

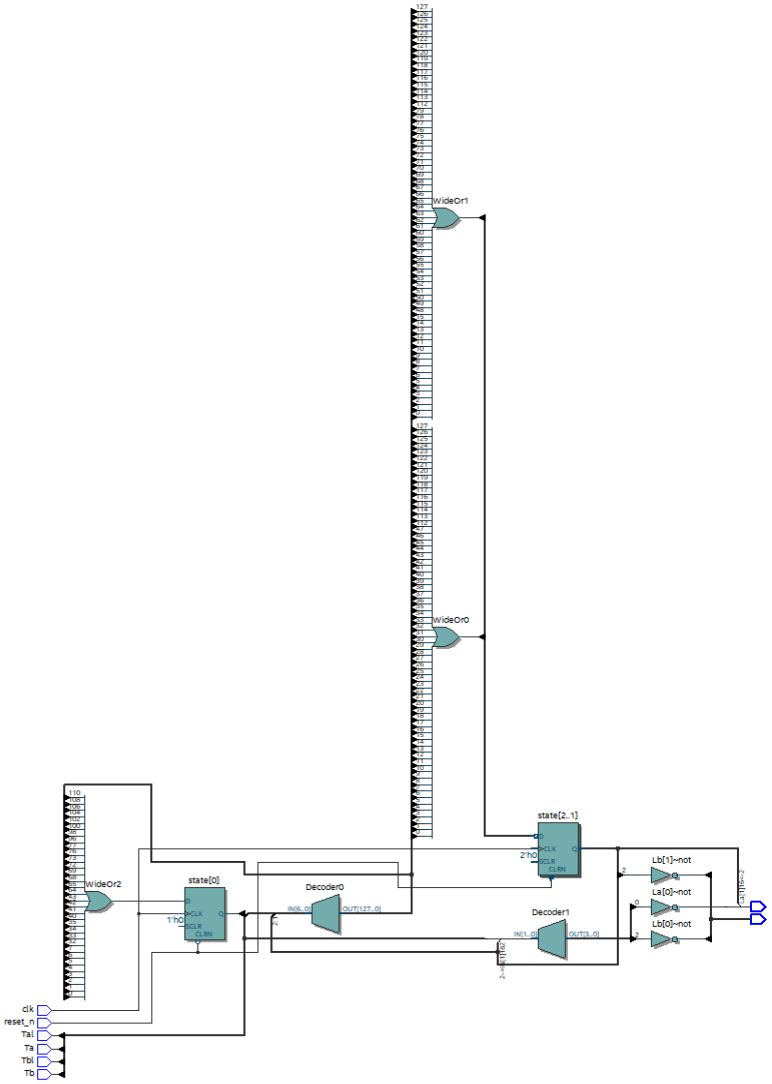
Flow Status	Successful - Fri Oct 20 13:33:01 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	tl_cntr_struct
Top-level Entity Name	tl_cntr
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	2
Total pins	8

tl_cntr_struct

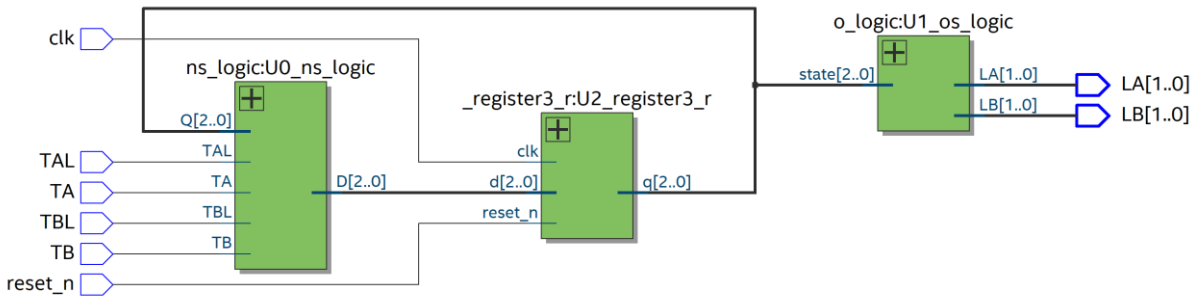


Flow Status	Successful - Fri Oct 20 14:06:49 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	tl_cntr_struct
Top-level Entity Name	tl_cntr
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	2

tl_cntr_w_left



Flow Summary	
<<Filter>>	
Flow Status	Successful - Fri Oct 20 14:23:33 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)	N/A
Total registers	3



Flow Status	Successful - Fri Oct 20 14:19:03 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)	N/A
Total registers	3
Total pins	13

4. 고찰 및 결론

A. 고찰

코드 작성 하면서 light의 변화를 눈으로 따라가는 것이 힘들었다. La, lb코드가 4bit인데 해당 코드를 따라가는데 어려움을 겪어 state로 현재 상태를 확인했다.

2bit의 반복을 관찰하니 훨씬 수월하게 문제점을 찾을 수 있었다.

코드 작성보다 tb작성이 시간이 오래 걸린 것 같다. Tb를 작성하면서 오류를 찾아야 했고 계산식을 고쳐가며 오류를 수정하는 시간이 더 오래 걸렸다. 그만큼 tb의 중요성을 알게 되었다.

B. 결론

베릴로그를 이용해 FSM을 이용한 traffic light control를 구성했다. 과제를 수행하면서 베릴로그에 익숙해 지다 보니 확실히 코드에 익숙해 졌다. 초창기에는 코드를 보고 틀린 부분을 찾았다면 이젠 tb코드를 변경해 가며 어떤 bit에 문제가 발생했는지 알아보고, 해당 비트에 영향을 주는 부분을 확인하니 코드를 수월하게 수정 가능했다.

5. 참고문헌

해당 과제를 수행하며 참고한 서적/강의자료/웹의 출처를 작성하여 줍니다.

서적의 경우에는 **저자/제목/출판사/출판년도**의 순으로 작성한다.

강의자료의 경우에는 **강사/강의제목/강의장소(학교명(과명))/강의년도**의 순으로 작성한다.
웹의 경우에는 **주제/URL**의 순으로 작성한다.
또한, 문헌이 아닌 다른 사람의 도움을 많이 받았을 경우에는 '어떠한 부분은 누구의 도움을 받아 수행하였다'는 내용을 명시한다.
이준환/디지털논리회로2/광운대학교/2023
이형근/컴퓨터공학기초실험2/광운대학교/2023
유지현/디지털논리회로1/광운대학교/2023

FSM/https://ko.wikipedia.org/wiki/%EC%9C%A0%ED%95%9C_%EC%83%81%ED%83%9C_%EA%B8%B0%EA%B3%84