

컴퓨터 공학 기초 실험2 보고서

실험제목: Traffic Light Controller with/without
Left Turn Signals

실험일자: 2019년 09월 17일 (화)

제출일자: 2019년 09월 30일 (월)

학 과: 컴퓨터정보공학과

담당교수: 이준환 교수님

실습분반: 화요일 0, 1, 2

학 번: 2015722031

성 명: 박 태 성

1. 제목 및 목적

A. 제목

Traffic Light Controller with/without Left Turn Signals

B. 목적

Finite State Machine(FSM)의 기법 중 하나인 Moore FSM을 적용하여 Traffic Light Controller(TLC) with/without Left Turn Signals을 설계해보면서 karnaugh map과 sequential logic에 대한 이해를 높이고 응용력을 높이는데 목적을 둔다.

2. 원리(배경지식)

A. Karnaugh Map

Karnaugh Map(K-map)은 boolean equation을 최소화하는데 유용하다. 식을 최소화하여 transistor의 개수를 줄일 수 있다. 다시 말하면 회로의 크기를 줄일 수 있다.

B. FSM

FSM은 유한한 개수의 state을 가지고 있는 기계의 state 간의 천이에 의해 출력을 내보내는 digital circuit을 의미한다. FSM은 입력과 current state에 따라 next state을 결정하는 combinational circuit인 next state logic과 current state을 저장하는 state register 그리고 input 또는 current state에 따라 output을 내보내는 combinational circuit인 output logic으로 구성된다.

가장 많이 사용되는 FSM은 Moore model과 Mearly model이 있다. Moore model은 current state에 따라서 output이 결정된다. Mearly model은 current state과 input에 따라서 output이 결정되는 회로다.

FSM encoding은 binary encoding과 one-hot encoding이 있다. Binary encoding은 binary number system을 사용하여 state encoding한다. State 변화가 많으면 power 소비가 많다. One-hot encoding은 하나의 state에 하나의 flip-flop을 사용한다. 따라서 회로의 크기가 비교적 클 수 있다. 그러나 상태의 변화가 많은 회로에서는 power 소비가 비교적 적을 수 있다.

C. TLC

시간에 따라서 바뀌는 신호등이 아닌 대기 차량 존재 유무에 따라 신호가 바뀌는 신호등이다. 좌회전을 추가한 경우 초록색 신호에서 빨간색 신호로 변화할 때 신호를 표시한다.

3. 설계 세부사항

A. TLC without Left Turn Signals

i. Drawing the FSM

1. Define States

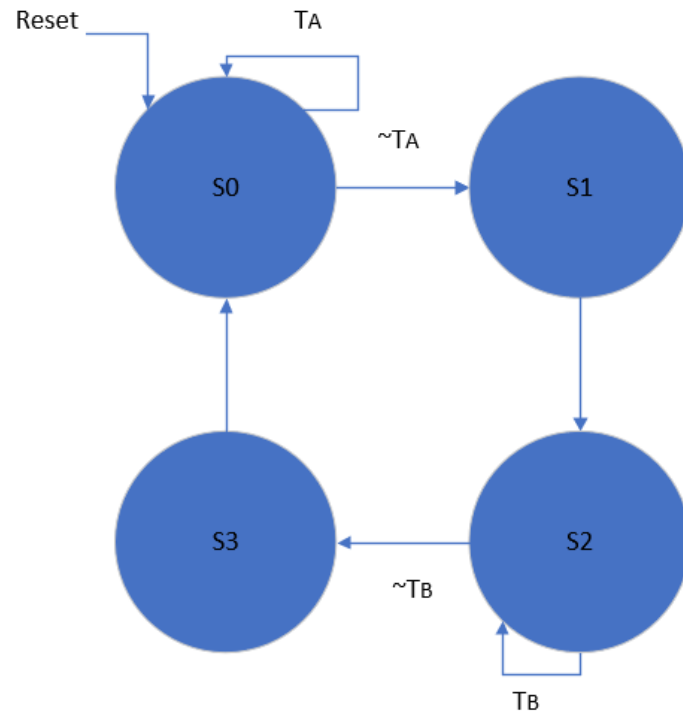
State	Encoding	L _A	L _B
S0	00	green	red
S1	01	yellow	red
S2	10	red	green
S3	11	red	yellow

Output	Encoding
green	00
yellow	01
red	10

2. Define inputs/outputs

Module 이름	구분	이름	비트 수	설명
tl_cntr	input	clk	1-bit	Clock
		reset_n	1-bit	Active low에 동작하는 reset 신호로 state를 초기화
		T _a	1-bit	Traffic sensor A('Academic Ave.'에 위치)
		T _b	1-bit	Traffic sensor B('Bravado Blvd.'에 위치)
	output	L _a	2-bits	신호등 값 출력 A('Academic Ave.'에 위치)
		L _b	2-bits	신호등 값 출력 B('Bravado Blvd.'에 위치)

3. Draw the Diagram



4. Draw State Transition Table

Current State		Inputs		Next State	
Q_1	Q_0	T_A	T_B	D_1	D_0
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
0	1	X	X	0	0

$$D_1 = Q_1 \oplus Q_0$$

$$D_0 = \sim Q_1 * \sim Q_0 * \sim T_A + Q_1 * \sim Q_0 * \sim T_B$$

5. Draw Output Table

Current State		Outputs			
Q_1	Q_0	L_{A1}	L_{A0}	L_{B1}	L_{B0}
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	0	0

1	1	1	0	0	1
---	---	---	---	---	---

$$L_{A1} = Q_1$$

$$L_{A0} = \sim Q_1 * Q_0$$

$$L_{B1} = \sim Q_1$$

$$L_{B0} = Q_1 * Q_0$$

ii. Module Configuration

구분	이름	설명
Top module	tl_cntr	TLC의 top module
Sub module	ns_logic	TLC의 next state을 결정하는 combinational logic
	_reister2_r	2-bits resettable register with active low asynchronous reset module(d_ff_r_async instantiation) – Current state 값 저장
	_dff_r	Resettable D F/F with active low asynchronous reset
	o_logic	Current state에 기반하여 output을 결정하는 combinational circuit

B. TLC with Left Turn signals

i. Drawing the FSM

1. Define states

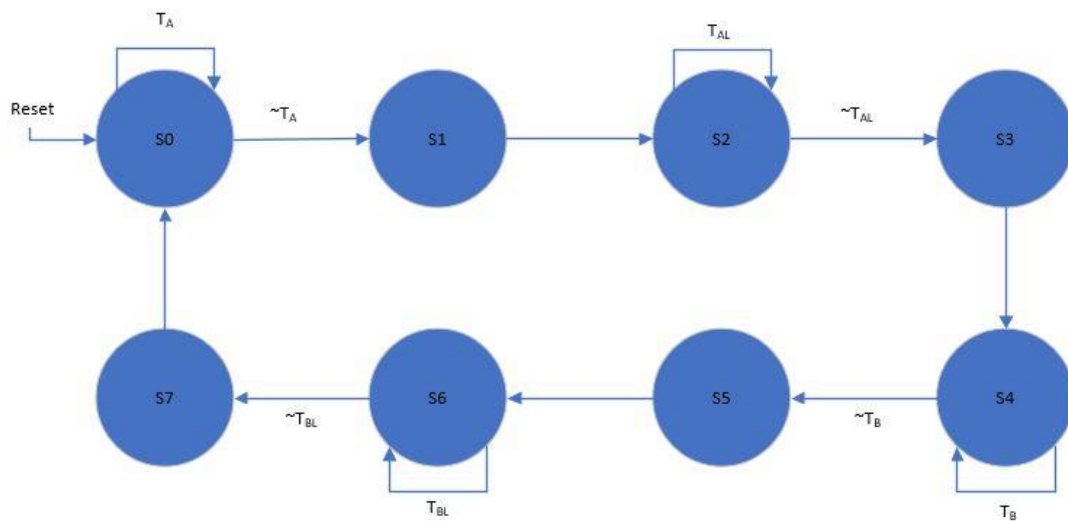
State	Code
S0	000
S1	001
S2	010
S3	011
S4	100
S5	101
S6	110
S7	111

Color	Code
Green	00
Yellow	01
Left	10
Red	11

2. Define inputs/outputs

구분	이름	비트 수	설명
input	clk	1-bit	Clock
	reset_n	1-bit	Active low에 동작
	Ta	1-bit	Traffic Sensor A
	Tal	1-bit	Traffic Sensor AL
	Tb	1-bit	Traffic Sensor B
	Tbl	1-bit	Traffic Sensor BL
output	La	2-bit	신호등 값 출력 A
	Lb	2-bit	신호등 값 출력 B

3. Draw the Diagram



4. Draw state transition table

Q ₂	Q ₁	Q ₀	T _A	T _{AL}	T _B	T _{BL}	D ₂	D ₁	D ₀
0	0	0	0	X	X	X	0	0	1
0	0	0	1	X	X	X	0	0	0

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

$$\begin{aligned}
D_2 &= \sim Q_2 * Q_1 * Q_0 + Q_2 * \sim Q_1 * \sim Q_0 * \sim T_B + Q_2 * \sim Q_1 * \sim Q_0 * T_B \\
&\quad + Q_2 * \sim Q_1 * Q_0 + Q_2 * Q_1 * \sim Q_0 * \sim T_{BL} + Q_2 * Q_1 * \sim Q_0 * T_{BL} \\
&= \sim Q_2 * Q_1 * Q_0 + Q_2 * \sim Q_1 * \sim Q_0 * (\sim T_B + T_B) \\
&\quad + Q_2 * \sim Q_1 * Q_0 + Q_2 * Q_1 * \sim Q_0 * (\sim T_{BL} + T_{BL}) \\
&= \sim Q_2 * Q_1 * Q_0 + Q_2 * \sim Q_1 * \sim Q_0 + Q_2 * \sim Q_1 * Q_0 + Q_2 * Q_1 * \sim Q_0 \\
&= (\sim Q_2 * Q_1 * Q_0) + (Q_2 * \sim Q_1) + (Q_2 * Q_1 * \sim Q_0) \\
D_1 &= \sim Q_2 * \sim Q_1 * Q_0 + \sim Q_2 * Q_1 * \sim Q_0 * \sim T_{AL} + \sim Q_2 * Q_1 * \sim Q_0 * T_{AL} \\
&\quad + Q_2 * \sim Q_1 * Q_0 + Q_2 * Q_1 * \sim Q_0 * \sim T_{BL} + Q_2 * Q_1 * \sim Q_0 * T_{BL} \\
&= \sim Q_2 * \sim Q_1 * Q_0 + Q_1 * \sim Q_0 + Q_2 * \sim Q_1 * Q_0 \\
&= (\sim Q_2 + Q_2) * \sim Q_1 * Q_0 + Q_1 * \sim Q_0 \\
&= Q_1 \oplus Q_0 \\
D_0 &= \sim Q_2 * \sim Q_1 * \sim Q_0 * \sim T_a + \sim Q_2 * Q_1 * \sim Q_0 * \sim T_a \\
&\quad + Q_2 * \sim Q_1 * \sim Q_0 * \sim T_b + Q_2 * Q_1 * \sim Q_0 * \sim T_b
\end{aligned}$$

5. Draw output table

Current State			Outputs			
Q_2	Q_1	Q_0	L_{A1}	L_{A0}	L_{B1}	L_{B0}
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	0	1

$$L_{A1} = Q_1 * \sim Q_0 + Q_2$$

$$L_{A0} = Q_2 + Q_0$$

$$L_{B1} = \sim Q_2 + Q_1 * \sim Q_0$$

$$L_{B0} = \sim Q_2 + Q_0$$

ii. Module Configuration

구분	이름	설명
Top module	tl_cntr_w_left	TLC의 top module
Sub module	ns_logic	TLC의 next state을 결정하는 combinational logic
	_reister3_r	3-bits resettable register with active low asynchronous reset module(d_ff_r_async instantiation) – Current state 값 저장
	_dff_r	Resettable D F/F with active low asynchronous reset
	o_logic	Current state에 기반하여 output을 결정하는 combinational circuit

- Combinational logic인 next state logic과 output logic을 case문을 사용한 behavior design으로 구현하였을 경우와 gate들을 사용한 structural design으로 구현한 경우의 크기를 비교하라

i. TCL without Left Turn Signals

A. Nest State Logic

gate

case

Flow Summary		Flow Summary	
Flow Status	In progress - Sun Sep 22 15:38:28 2019	Flow Status	Successful - Sun Sep 22 15:35:26 2019
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition	Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	tl_cntr	Revision Name	tl_cntr
Top-level Entity Name	ns_logic	Top-level Entity Name	ns_logic
Family	Cyclone V	Family	Cyclone V
Device	5CSXFC6D6F31C6	Device	5CSXFC6D6F31C6
Timing Models	Final	Timing Models	Final
Logic utilization (in ALMs)	2 / 41,910 (< 1 %)	Logic utilization (in ALMs)	2 / 41,910 (< 1 %)
Total registers	0	Total registers	0
Total pins	6 / 499 (1 %)	Total pins	6 / 499 (1 %)
Total virtual pins	0	Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)	Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)	Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)	Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)	Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)	Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)	Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)	Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)	Total DLLs	0 / 4 (0 %)

Logic utilization은 2, Total pins는 6으로 같다.

B. Output Logic

Flow Summary		Flow Summary	
Flow Status	Successful - Mon Sep 23 08:57:13 2019	Flow Status	Successful - Mon Sep 23 08:53:02 2019
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition	Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	tl_cntr	Revision Name	tl_cntr
Top-level Entity Name	o_logic	Top-level Entity Name	o_logic
Family	Cyclone V	Family	Cyclone V
Device	5CSXFC6D6F31C6	Device	5CSXFC6D6F31C6
Timing Models	Final	Timing Models	Final
Logic utilization (in ALMs)	2 / 41,910 (< 1 %)	Logic utilization (in ALMs)	2 / 41,910 (< 1 %)
Total registers	0	Total registers	0
Total pins	6 / 499 (1 %)	Total pins	6 / 499 (1 %)
Total virtual pins	0	Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)	Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)	Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)	Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)	Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)	Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)	Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)	Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)	Total DLLs	0 / 4 (0 %)

behavior design

structural design

Logic utilization은 2, Total pins는 6으로 같다.

ii. TCL with Left Turn Signals

A. Next State Logic

Flow Summary		Flow Summary	
Flow Status	Successful - Sun Sep 22 16:42:55 2019	Flow Status	Successful - Sun Sep 22 16:45:35 2019
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition	Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	tl_cntr_w_left	Revision Name	tl_cntr_w_left
Top-level Entity Name	ns_logic	Top-level Entity Name	ns_logic
Family	Cyclone V	Family	Cyclone V
Device	5CSXFC6D6F31C6	Device	5CSXFC6D6F31C6
Timing Models	Final	Timing Models	Final
Logic utilization (in ALMs)	3 / 41,910 (< 1 %)	Logic utilization (in ALMs)	3 / 41,910 (< 1 %)
Total registers	0	Total registers	0
Total pins	10 / 499 (2 %)	Total pins	10 / 499 (2 %)
Total virtual pins	0	Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)	Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)	Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)	Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)	Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)	Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)	Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)	Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)	Total DLLs	0 / 4 (0 %)

behavior design

structural design

Logic utilization은 3, Total pins는 10으로 같다.

B. Output Logic

Flow Summary		Flow Summary	
Flow Status	Successful - Sun Sep 22 16:50:41 2019	Flow Status	Successful - Sun Sep 22 16:53:54 2019
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition	Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	tl_cntr_w_left	Revision Name	tl_cntr_w_left
Top-level Entity Name	o_logic	Top-level Entity Name	o_logic
Family	Cyclone V	Family	Cyclone V
Device	5CSXFC6D6F31C6	Device	5CSXFC6D6F31C6
Timing Models	Final	Timing Models	Final
Logic utilization (in ALMs)	3 / 41,910 (< 1 %)	Logic utilization (in ALMs)	3 / 41,910 (< 1 %)
Total registers	0	Total registers	0
Total pins	7 / 499 (1 %)	Total pins	7 / 499 (1 %)
Total virtual pins	0	Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)	Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)	Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)	Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)	Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)	Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)	Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)	Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)	Total DLLs	0 / 4 (0 %)

behavior design

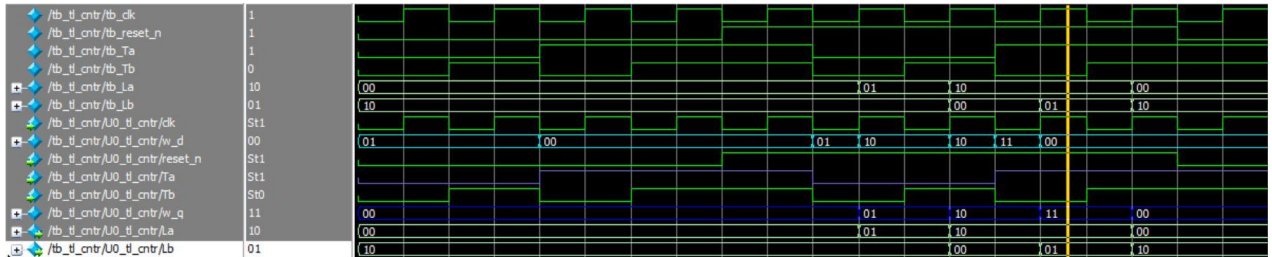
structural design

Logic utilization은 3, Total pins는 10으로 같다.

4. 설계 검증 및 실험 결과

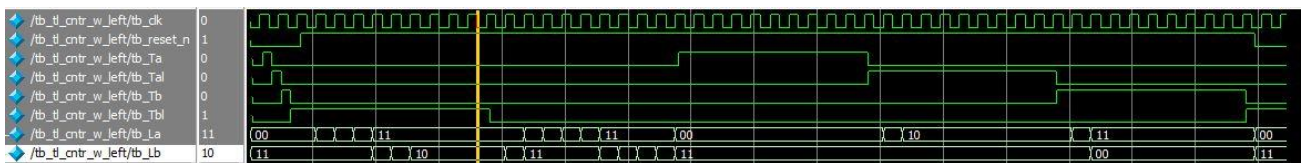
A. 시뮬레이션 결과

i. TCL without Left Turn Signals

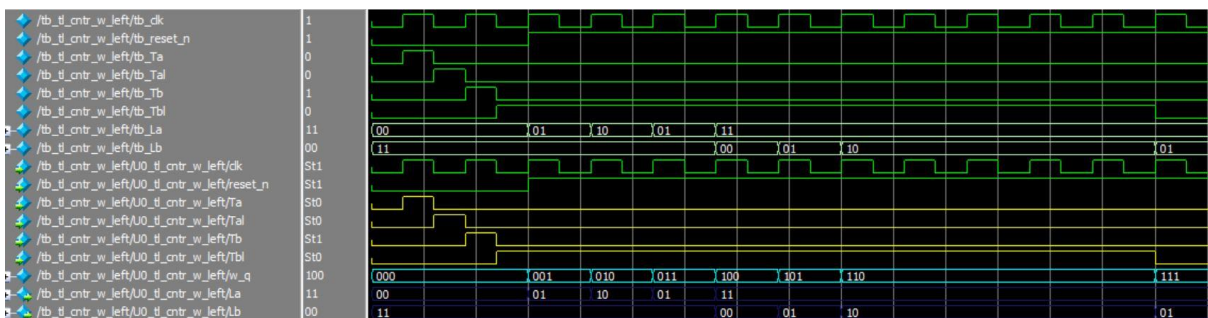


Reset_n이 0이면 출력은 입력에 관계없이 유지한다. Reset_1이면 clock의 rising edge에서 입력 Ta, Tb에 따라 La, Lb가 출력된다. La가 초록색(00) -> 노란색(01) -> 빨간색(10) -> 초록색(00) 순서로 변하는 것을 확인하였다. 마지막으로 reset_n이 0이되면 결과가 바뀌는 것을 확인하였다. State을 확인 할 수 있도록 waveform을 설정해야 transition을 확인할 수 있다.

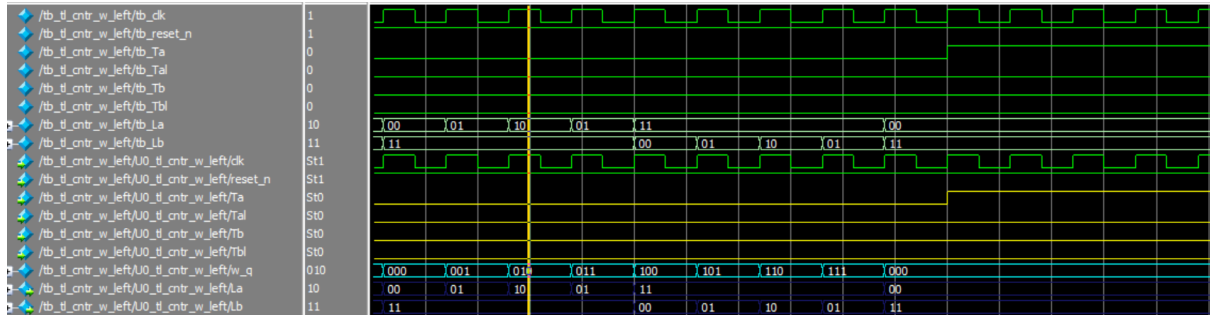
i. TCL with Left Turn Signals



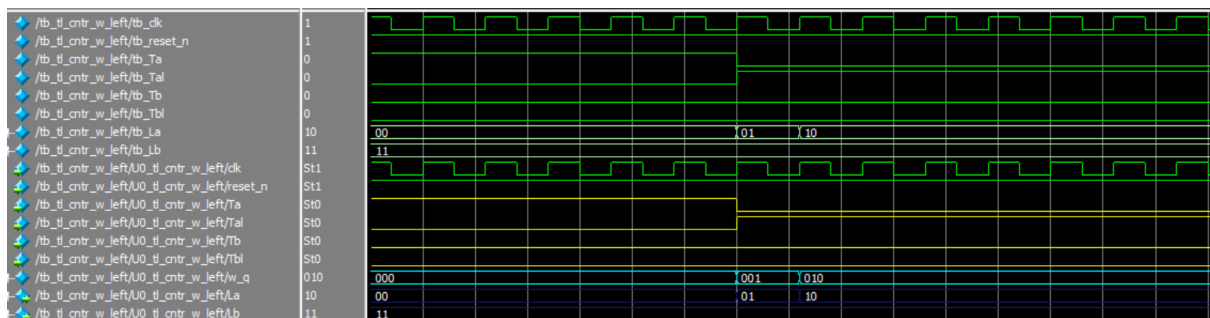
Current State이 보이지 않아 state transition 관찰이 어렵다. Waveform에 instance를 추가하여 current state이 보이도록 하겠다.



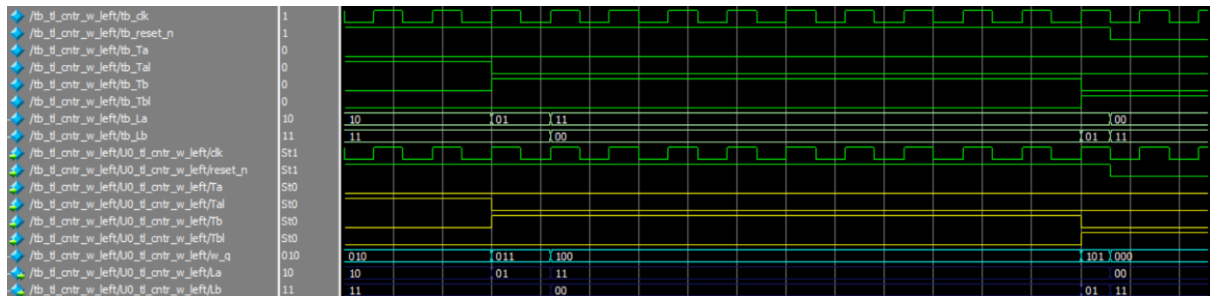
Reset_n이 0일 때는 Ta, Tal, Tb, Tbl의 값에 관계없이 값이 일정하다. 좌회전 신호를 조절하는 sensor Tal, Tbl이 추가되었다. Ta가 1이면, La는 녹색(00)이 되고 Lb는 빨간색(11)이 된다. Tb가 1이면, La는 빨간색(11)이 되고 Lb는 녹색(00)이 된다. 좌회전 -> 노란불 -> 빨간불 순서로 결과가 바뀌는 것을 확인하였다. 마지막으로 reset_n이 0이되면 결과가 바뀌는 것을 확인하였다.



Ta만 1이면 La가 초록색(00) Lb가 빨간색(11)이 될 때까지 결과 La, Lb가 clock edge에서 지속적으로 바뀐다.



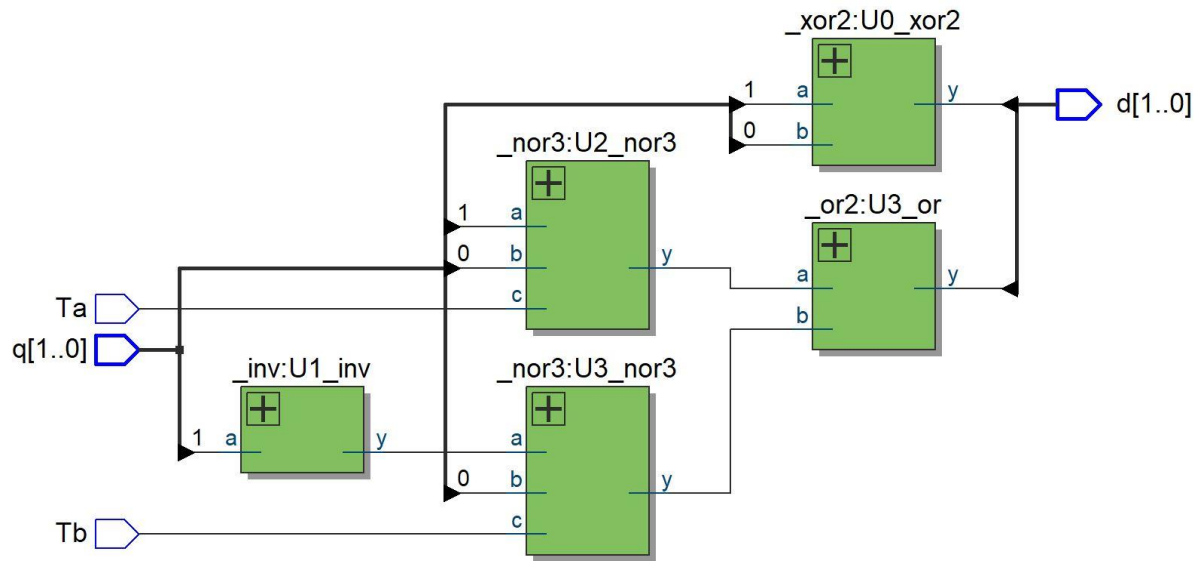
Tal이 1이라면 S2(010), La는 좌회전(10), Lb는 빨간색(11) 상태를 유지한다.



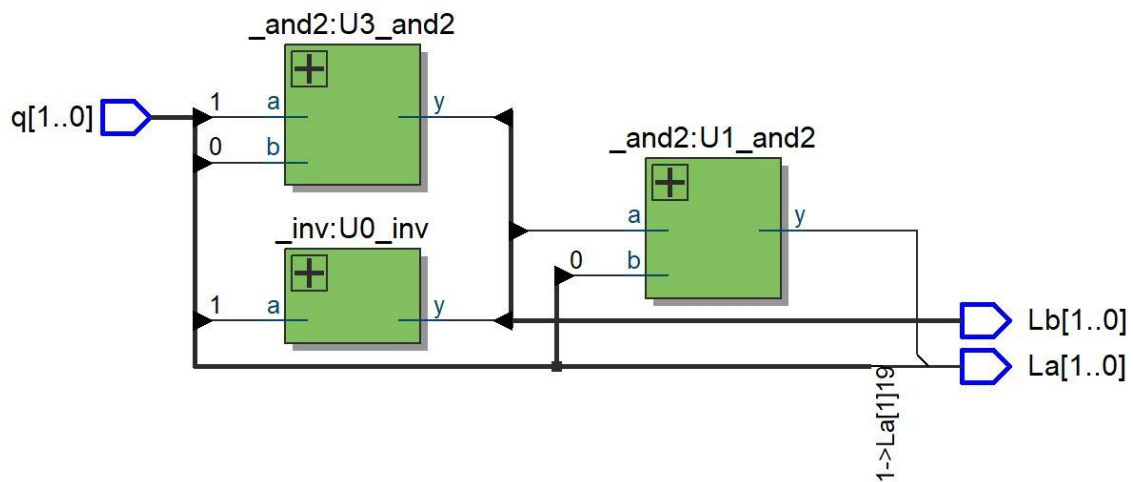
B. 합성(synthesis) 결과

i. TCL without Left Turn Signals

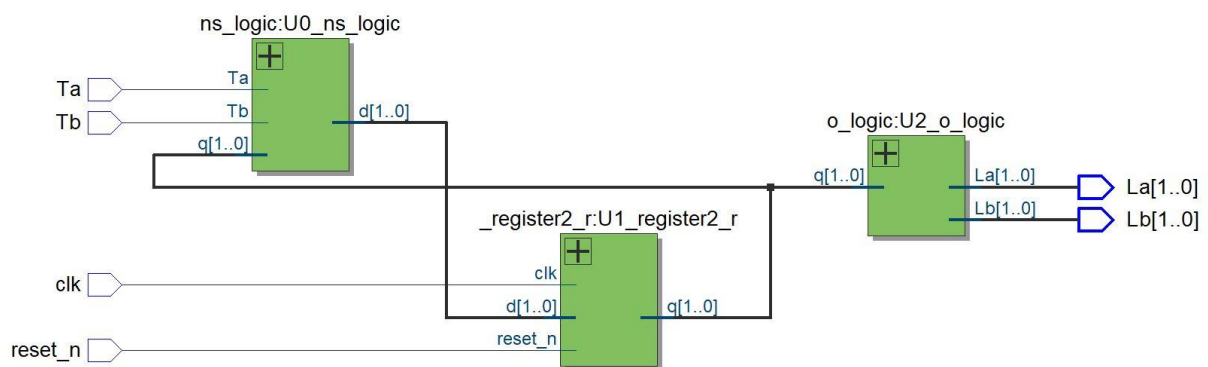
1. Next State Logic



2. Output Logic



3. TCL

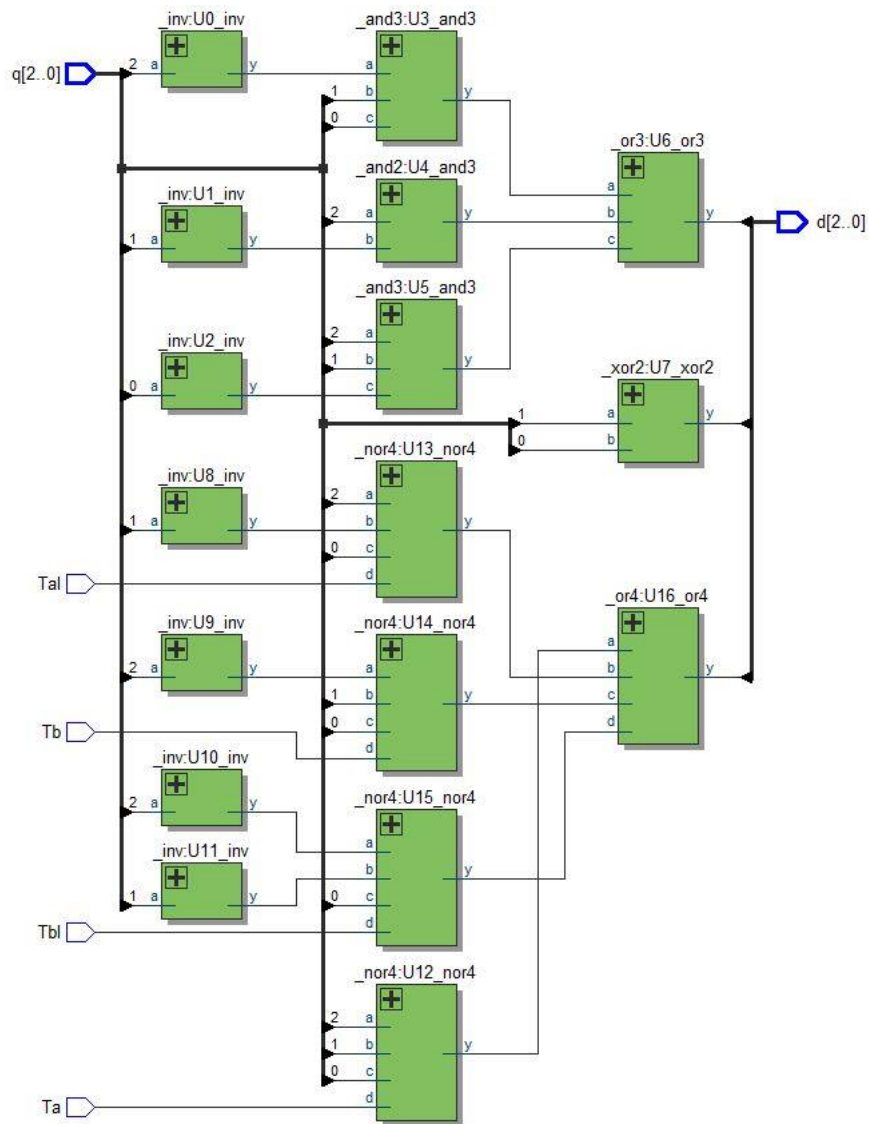


4. Flow Summary

Flow Summary	
Flow Status	Successful - Mon Sep 23 08:59:37 2019
Quartus Prime Version	15.1.0 Build 185 10/21/2015 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)	N/A
Total registers	2
Total pins	8
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

ii. TCL with Left Turn Signals

1. Next State Logic



2. Output Logic

5. 고찰 및 결론

A. 고찰

불 대수 기본 법칙을 이용하여 수식을 먼저 최소화 해준 이후에 Karnaugh Map을 적용하면 좋다. Case문을 이용하여, 그리고 gate을 이용하여 next state logic, output logic을 구현하였는데 logic utilization number가 다를 것으로 예상했지만 같게 나왔다. 검색 결과, logic utilization number는 percentage로 표현된다고 한다.

B. 결론

Logic Utilization Number는 percentage로 표현되어 간단한 circuit의 경우 구성방식이 달라도 결과값이 같게 나올 수 있다. Karnaugh Map으로 수식을 간단화 할 때, 결과는 다양하다. 따라서, transistor의 개수를 고려하면서 구현해야 할 경우 이를 응용하면 되겠다. 또한 드모르간 법칙을 이용하여 and, or를 nand, nor로 바꾸는 것도 좋다.

6. 참고문헌

이준환/ 디지털논리회로2 강의자료/ 광운대학교/ 컴퓨터공학과/ 2019

D. M. Harris and S. L. Harris/ Digital Design and Computer Architecture/ Morgan Kaufmann/ 2007

Logic Utilization Number/

https://www.intel.com/content/www/us/en/programmable/support/support-resources/knowledge-base/solutions/rd05172012_146.html