

Traffic Light Controller

with/without Left Turn Signals

이번 실습에서는 FSM의 기법 중 하나인 Moore FSM을 적용하여 traffic light controller를 설계한다.

1. Traffic Light Controller 설명

A. Description

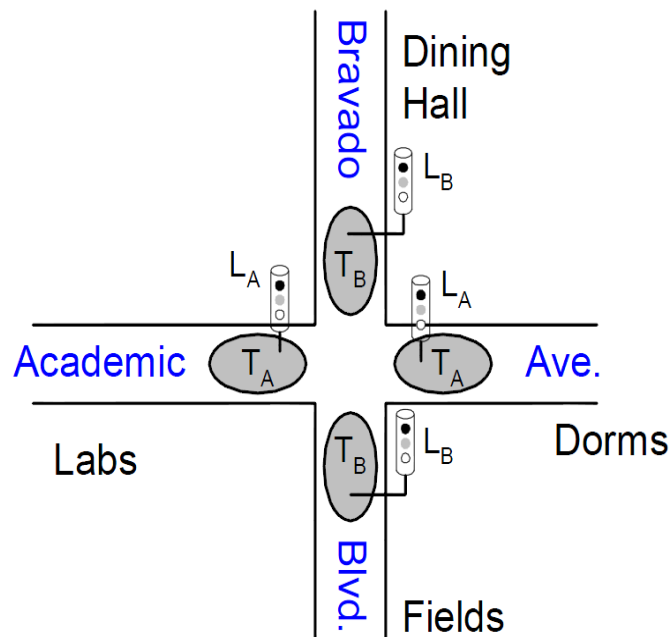


Figure 1 - Traffic light controller

이번 실습에서는 위의 그림 1과 같은 신호등을 제어하는 logic을 구현한다. 신호등 L_A 는 'Academic Ave.'의 차량 통행을 제어하는 신호등이고, L_B 는 'Bravado Blvd.'의 차량 통행을 제어하는 신호등이다. 일반적인 신호등의 경우는 시간에 따라 변하지만, 이번 실습을 통해 구현하려는 신호등은 거리에 차량이 있을 때 신호등이 초록색이 되고, 없을 때는 빨간색이 된다. 차량이 있음을 감지하기 위하여 'Academic Ave.'에 traffic sensor인 T_A 를, 'Bravado Blvd.'에 traffic sensor인 T_B 를 설치하였다.

신호등의 구현을 위해 다음 규칙을 만족해야 한다.

- 1) Traffic light는 교통이 없을 때 초록색에서 노란색을 거쳐 빨간색으로 변한다.
- 2) 만약 traffic light L_A 가 초록색이거나 노란색이면, traffic light L_B 는 빨간색이다. 반대의 경우도 마찬가지이다.

B. Structural description

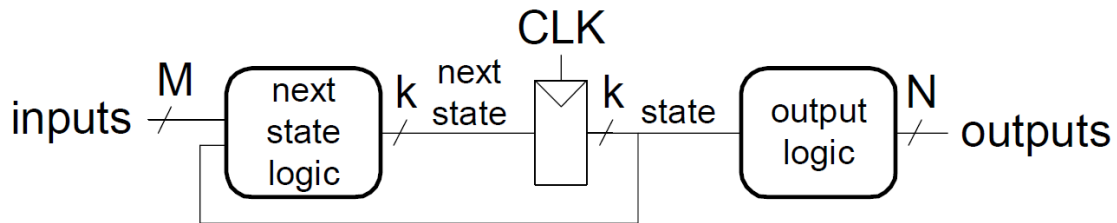


Figure 2 - Moore FSM

이번에 구현할 traffic light controller는 그림 2의 Moore FSM처럼 구현하여 본다.

2. Traffic Light Controller 실습

A. Design 단계

- i. 앞서 Lab 6에서 하였던 것처럼 sequential logic 설계 단계에 맞추어서 구현하도록 한다.

(각각의 설계 단계에 대하여 report에 작성하고 이에 대하여 설명한다.)

- ii. Sequential logic의 설계 단계는 다음과 같다.

(1) Drawing the finite state diagram

- (1-1) Define states
- (1-2) Define inputs
- (1-3) Define outputs
- (1-4) Draw the diagram

Diagram 작성 후 state transition table과 output table을 작성하고, 이에 대하여 Boolean equation으로 report에 정리한다.

- (2) Encoding states
- (3) Coding the module header
- (4) Coding state registers(flip-flops) – sequential circuits
- (5) Coding combinational circuits

B. Implementation

Traffic light controller를 구현할 때, 구조적으로 구현한다. 다음의 표는 최소한으로 포함하여야 하는 module configuration을 나타낸다.

(이들을 report에 정리하고, 또한 추가적인 모듈이 필요하다면 추가한 후 이에 대한 기능을 설명하도록 한다.)

구분	이름	설명
Top module	tl_cntr	Traffic light controller의 top module
Sub module	ns_logic	Traffic light controller의 next state를 결정하는 combinational logic
Sub module	_register2_r_async	2-bit resettable register with active low asynchronous reset module (내부에 _dff_r_async를 instance) - 현재 state의 값을 저장하고 있다.
Sub module	_dff_r_async	Resettable D flip-flop with active low asynchronous reset
Sub module	o_logic	현재 state의 값에 기반하여 output 값을 결정하는 combinational logic

Table 1 - Module configuration

(기타 필요한 gate들에 대하여서는 gates.v를 이용한다.)

다음은 top module의 I/O configuration을 나타낸다. Top module을 제외한 port의 이름과 비트는 자유롭게 해도 무관하다.

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
		Ta	1-bit	Traffic sensor A('Academic Ave.'에 위치)
		Tb	1-bit	Traffic sensor B('Bravado Blvd.'에 위치)
	output	La	2-bit	신호등 값 출력 A('Academic Ave.'에 위치)
		Lb	2-bit	신호등 값 출력 B('Bravado Blvd.'에 위치)

Table 2 – I/O configuration of top module

3. Traffic Light Controller with Left Turn Signals

A. Description

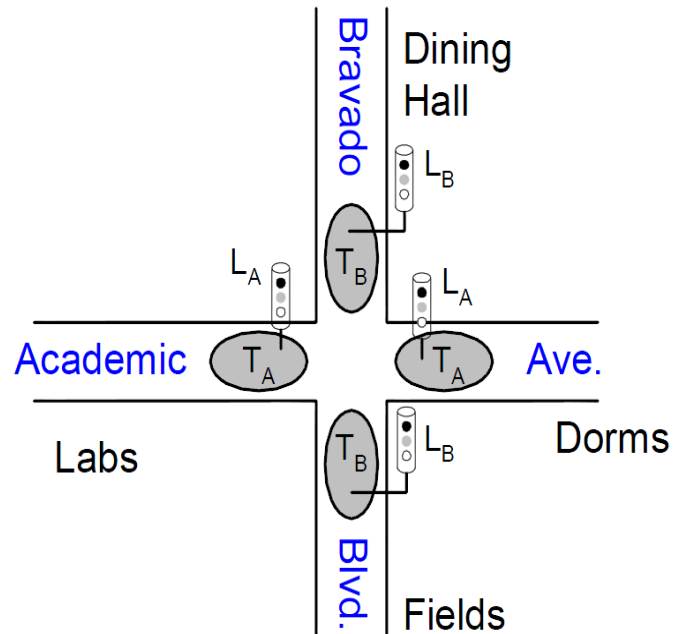


Figure 3 - Traffic light controller

앞선 실습에서 위의 그림 3과 같은 신호등을 제어하는 logic을 구조적으로 구현하였다. 이번 실습에서는 지난 신호등에 좌회전 신호를 추가하여 구현하여 보도록 한다.

지난 실습과 마찬가지로 신호등 L_A 는 'Academic Ave.'의 차량 통행을 제어하는 신호등이고, L_B 는 'Bravado Blvd.'의 차량 통행을 제어하는 신호등이다. 일반적인 신호등의 경우는 시간에 따라 변하지만, 이번 실습을 통해 구현하려는 신호등은 거리에 차량이 있을 때 신호등이 초록색(또는 좌회전)이 되고, 없을 때는 빨간색이 된다. 차량이 있음을 감지하기 위하여 'Academic Ave.'에 traffic sensor인 T_A, T_{AL} 를, 'Bravado Blvd.'에 traffic sensor인 T_B, T_{BL} 를 설치하였다. Traffic sensor인 T_A, T_B 는 직진에 대한 차량이 있음을 감지하고, traffic sensor인 T_{AL}, T_{BL} 은 좌회전에 대하여 차량이 있음을 감지한다.

신호등의 구현을 위해 다음의 규칙을 지켜야 한다.

- 1) Traffic light는 교통이 없을 때 초록색에서 노란색을 거쳐 좌회전으로 변한다.
- 2) Traffic light는 교통이 없을 때 좌회전에서 노란색을 거쳐 빨간색으로 변한다.
- 3) Traffic light는 비록 좌회전하는 교통이 없더라도 초록색에서 좌회전으로 우선 변해야 한다.
- 4) 만약 traffic light L_A 가 초록색, 노란색, 좌회전일 동안에 L_B 는 빨간색이어 한다. 반대의 경우도 마찬가지이다.

4. Traffic Light Controller with Left Turn signals 실습

A. Design 단계

- i. 앞선 실습과 마찬가지로 sequential logic 설계 단계에 맞추어서 구현하도록 한다.

(각각의 설계 단계에 대하여 report에 작성하고 이에 대하여 설명한다.)

- ii. Sequential logic의 설계 단계는 다음과 같다.

(1) Drawing the finite state diagram

(1-5) Define states

(1-6) Define inputs

(1-7) Define outputs

(1-8) Draw the diagram

Diagram 작성 후 state transition table과 output table을 작성하고, 이에 대하여 Boolean equation으로 report에 정리한다.

(2) Encoding states

(3) Coding the module header

(4) Coding state registers(flip-flops) – sequential circuits

(5) Coding combinational circuits

B. Implementation

Traffic light controller를 구현할 때, 구조적으로 구현한다. 다음 표는 최소한으로 포함하여야 하는 module configuration을 나타낸다.

(이들을 report에 정리하고, 또한 추가적인 모듈이 필요하다면 추가한 후 이에 대한 기능을 설명하도록 한다.)

구분	이름	설명
Top module	tl_cntr_w_left	Traffic light controller의 top module
Sub module	ns_logic	Traffic light controller의 next state를 결정하는 combinational logic
Sub module	_register3_r	3-bit resettable register with active low asynchronous reset module(내부에 _dff_r_async를 instance) - 현재의 state 값을 저장
Sub module	dff_r_async	Resettable D flip-flop with active low asynchronous reset
Sub module	o_logic	현재 state의 값에 기반하여 output 값을 결정하는 combinational logic

Table 3 - Module configuration

(기타 필요한 gate들에 대하여서는 gates.v를 이용한다.)

다음은 top module의 I/O configuration을 나타낸다. Top module을 제외한 port의 이름과 비트는 자유롭게 해도 무관하다.

구분	이름	비트 수	설명
input	clk	1-bit	Clock
	reset_n	1-bit	Active low에 동작하는 reset 신호로 state를 초기화
	Ta	1-bit	Traffic sensor A('Academic Ave.'에 위치하여 직진 감지)
	Tal	1-bit	Traffic sensor AL('Academic Ave.'에 위치하여 좌회전 감지)
	Tb	1-bit	Traffic sensor B('Bravado Blvd.'에 위치하여 직진 감지)
	Tbl	1-bit	Traffic sensor BL('Bravado Blvd.'에 위치하여 좌회전 감지)
output	La	2-bit	신호등 값 출력 A('Academic Ave'에 위치)
	Lb	2-bit	신호등 값 출력 B('Bravado Blvd.'에 위치)

Table 4 - I/O configuration of top module

Submission

- 제출기한
 - 자세한 제출기한은 KLAS 와 일정을 참고
- 과제 업로드
 - ✓ Source code 와 report 를 같이 ZIP 파일로 압축하여 KLAS(종합정보서비스) 과제 제출에 해당 과제 upload
 - ✓ 업로드 파일명은 (요일#)_(학번)_Assignment_#.zip
 - 요일번호
 - 실습 미수강은 0
 - 월요일 0, 1, 2 교시 1
 - 화요일 0, 1, 2 교시 2
 - 수요일 0, 1, 2 교시 3
 - Ex) 월요일 반 수강, 2019110609, Assignment 1 제출 시
1_2019110609_Assignment_01.zip 으로 제출
 - ✓ Report 명은 (요일#)_(학번)_Assignment_#.pdf
 - 요일 번호는 위의 업로드 파일명과 동일하게 진행
 - ✓ Ex) 수요일 반 수강, 2019110609, Assignment 1 제출 시
3_2019110609_Assignment_01.pdf 으로 제출
 - ✓ Report 는 PDF 로 변환해 제출 (미수행시 감점)
- Source code 압축 시 db, incremental_db, simulation 폴더는 삭제 (미수행시 감점)
- Source code 압축 시 ~.bak 파일 삭제 (미수행시 감점)

➤ 채점기준

세부사항		점수	최상	상	중	하	최하
소스코드	Source code 가 잘 작성 되었는가? (Structural design 으로 작성되었는가?)	10	10	8	5	3	0
	주석을 적절히 달았는가? (반드시 영어로 주석 작성)	20	20	15	10	5	0
설계검증 (보고서)	보고서를 성실히 작성하였는가? (보고서 형식에 맞추어 작성)	30	30	20	10	5	0
	합성결과를 설명하였는가?	10	10	8	5	3	0
	검증을 제대로 수행하였는가? (모든 입력 조합, waveform 설명)	30	30	20	10	5	0
총점		100					