Traffic Light Controller

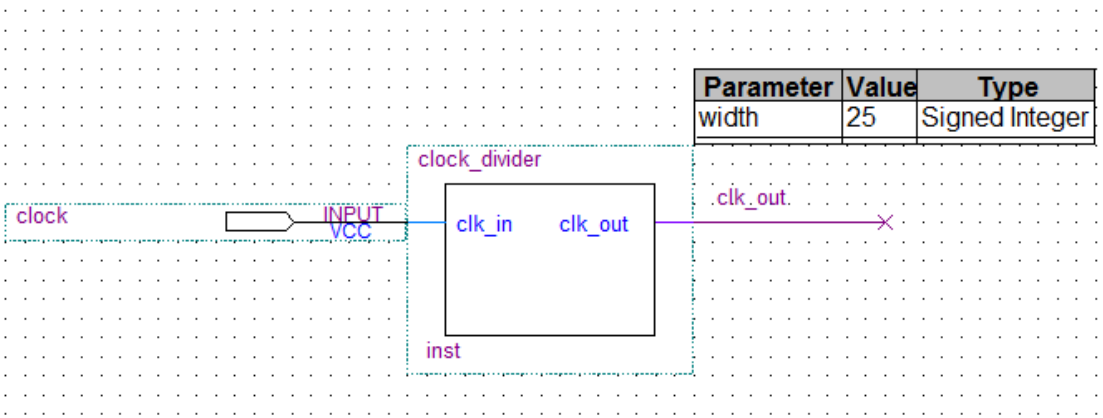
2018.10.24-31

석사 : 박제창

❖ Traffic Light Controller

➢ 클럭 분주기

신호등의 신호 제어는 DE2 보드의 내부 클럭 신호에 의존한다. 따라서, 시스템에 원하는 시간 만큼 클럭을 분주해 사용할 필요가 있다.



❖ Traffic Light Controller

➢ 클럭 분주기

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY clock_divider IS
    GENERIC(width : POSITIVE := 25);
    PORT(clk_in : IN STD_LOGIC;
         clk_out : OUT STD_LOGIC);
END clock_divider;

ARCHITECTURE divider OF clock_divider IS
    SIGNAL count : STD_LOGIC_VECTOR(width-1 downto 0);
BEGIN
    PROCESS(clk_in)
        BEGIN
        IF(clk_in'EVENT and clk_in = '1')THEN
            count <= count + 1;
        END IF;

        clk_out <= count(width-1);
    END PROCESS;
END divider;
```

클럭분주 관할

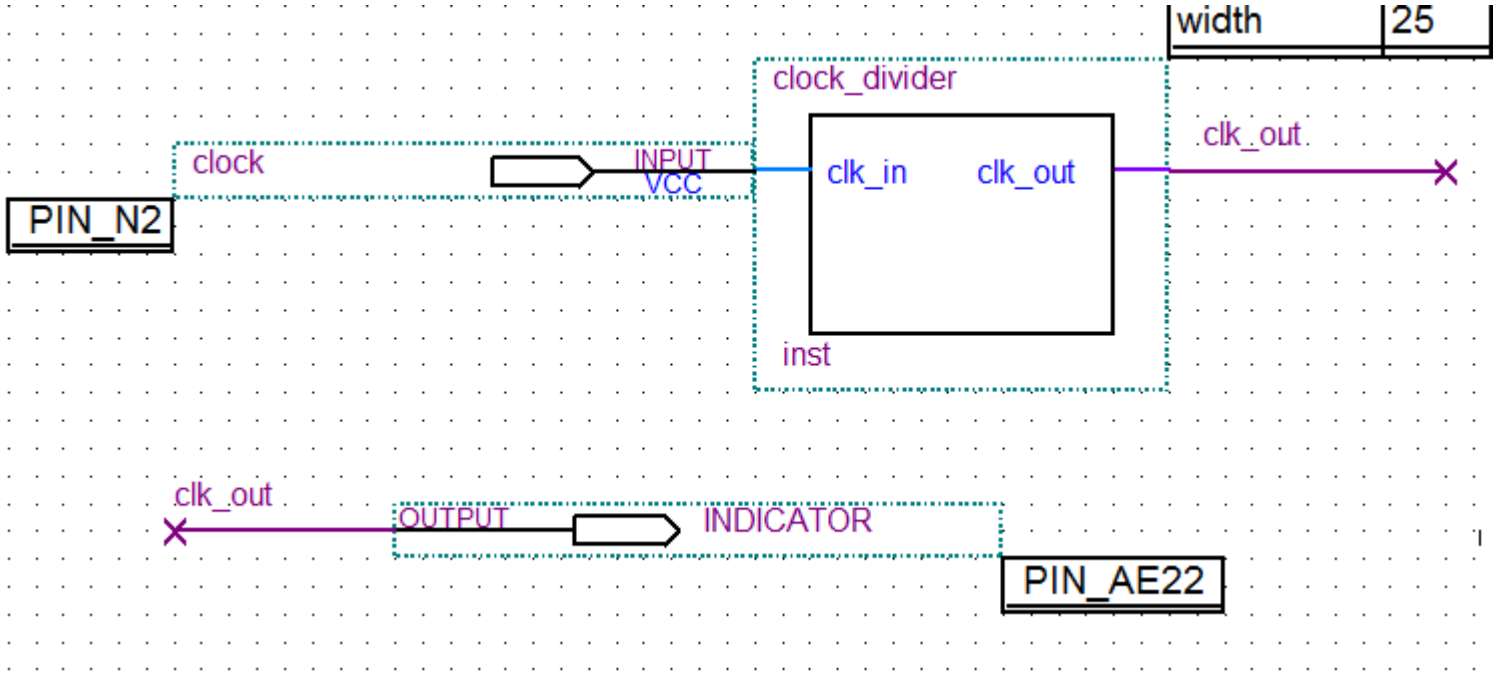
클럭 분주 카운터는 여러 보드에서 상호 호환해 사용할 수 있도록 제작하는 게 좋다.

따라서 다음과 같이 적용할 시스템에 맞는 클럭 출력을 위해 분주 비 변수를 변경만 하면 되도록 작성하는게 효율적이다.

❖ Traffic Light Controller

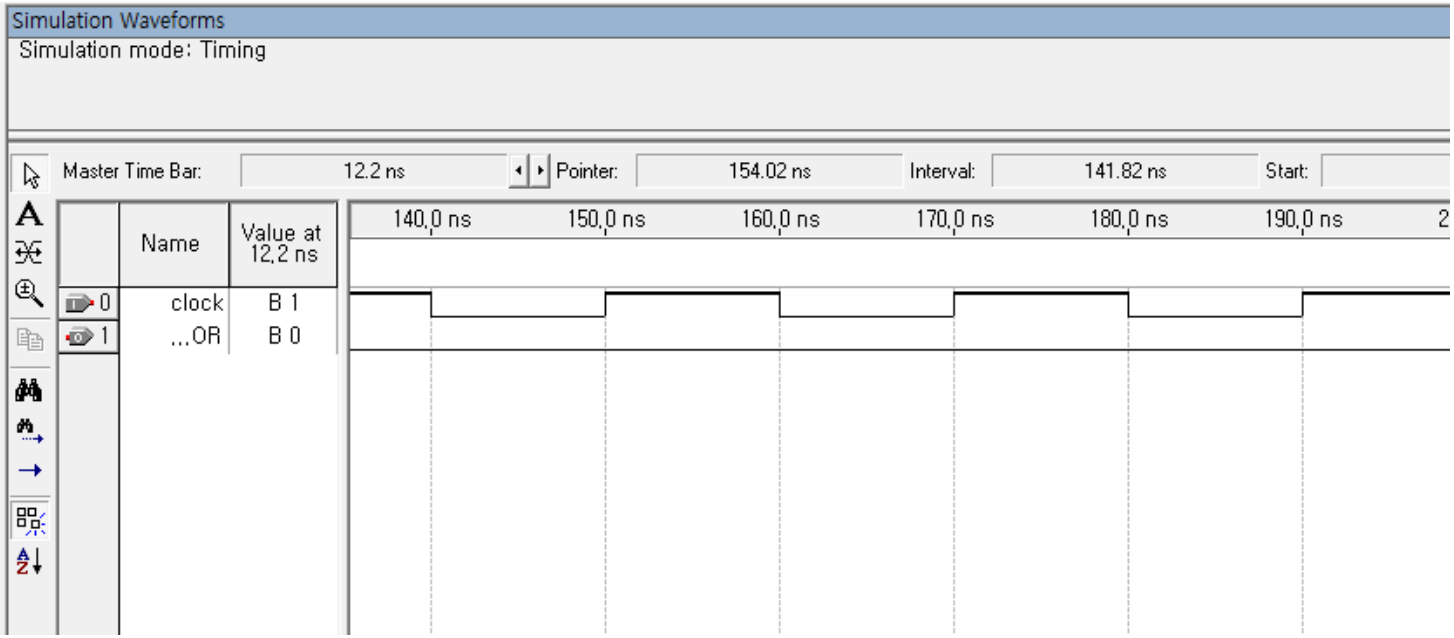
➢ 클럭 분주기

출력 결과를 확인하고자 DE2보드의 LED를 활용해 점멸을 확인한다.



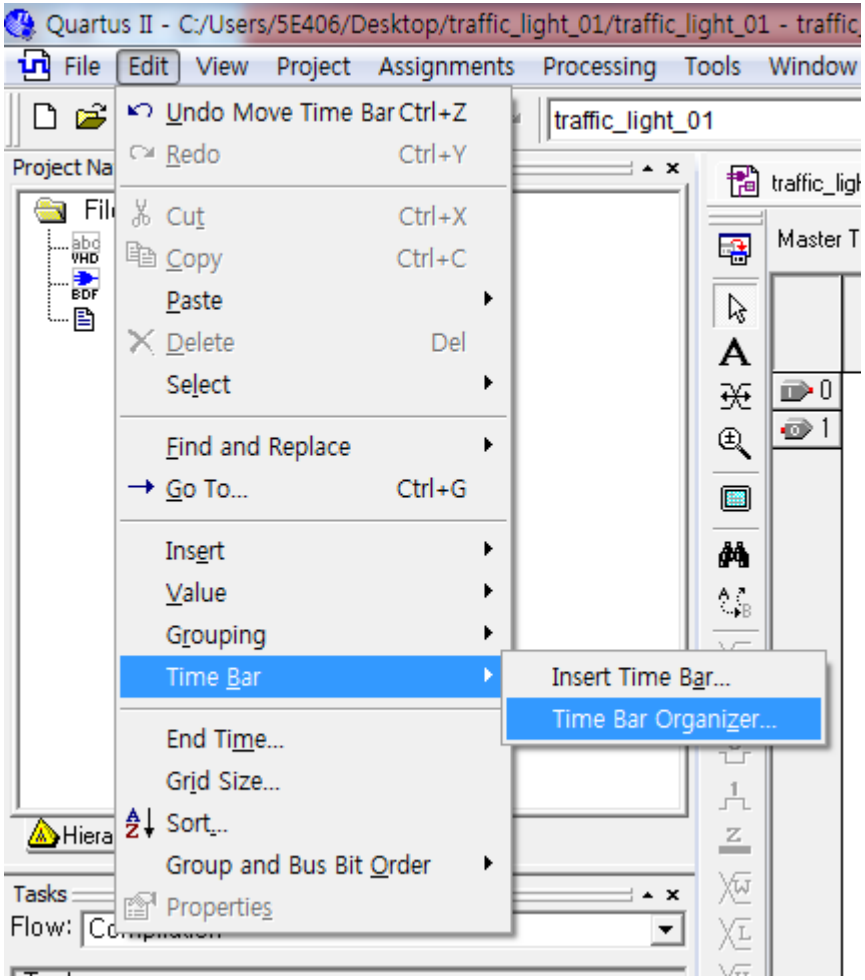
❖ Traffic Light Controller

➢ 클럭 분주기



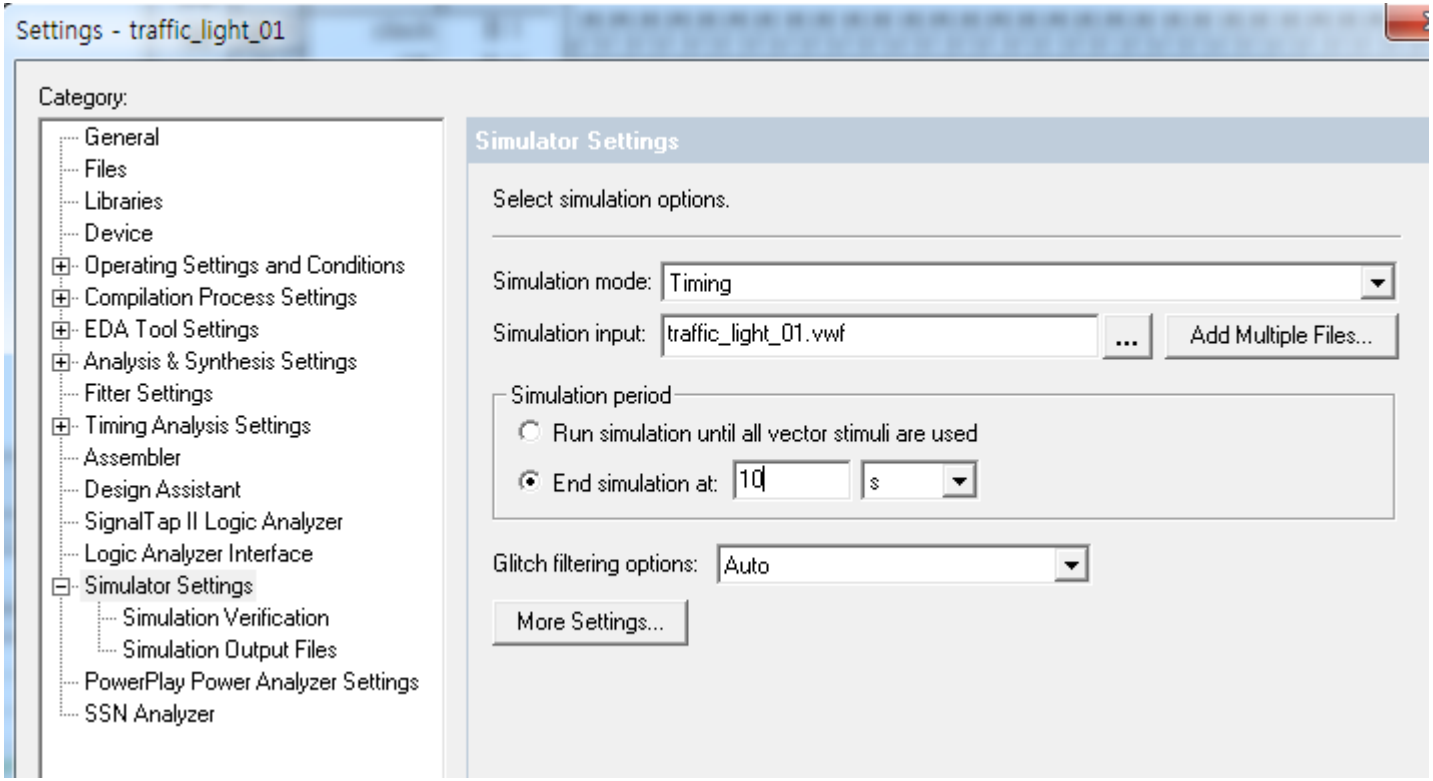
❖ Traffic Light Controller

➢ 클럭 분주기



❖ Traffic Light Controller

➢ 클럭 분주기

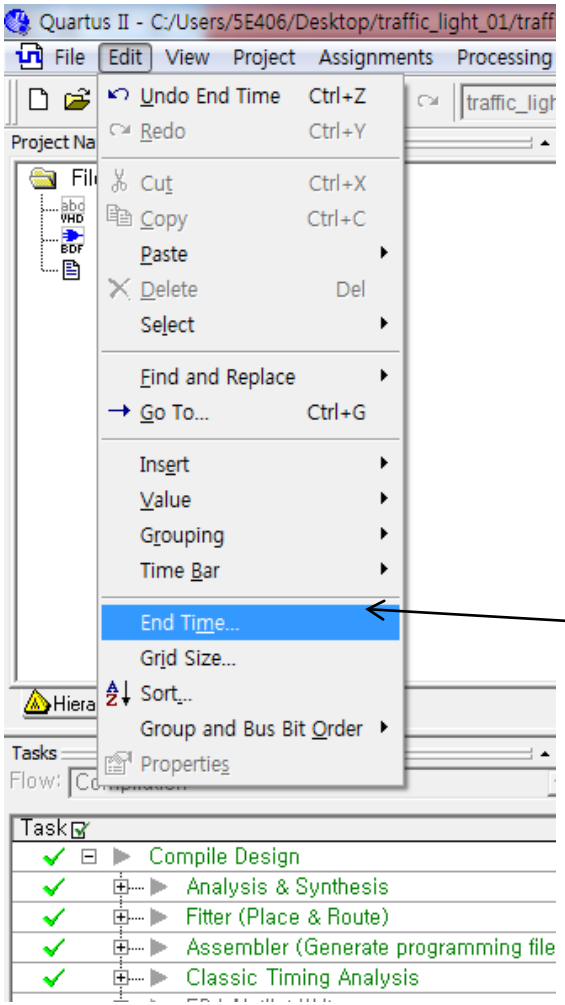


시뮬레이터 시간 설정 방법

❖ Traffic Light Controller

➢ 클럭 분주기

시뮬레이터 시간 설정 방법

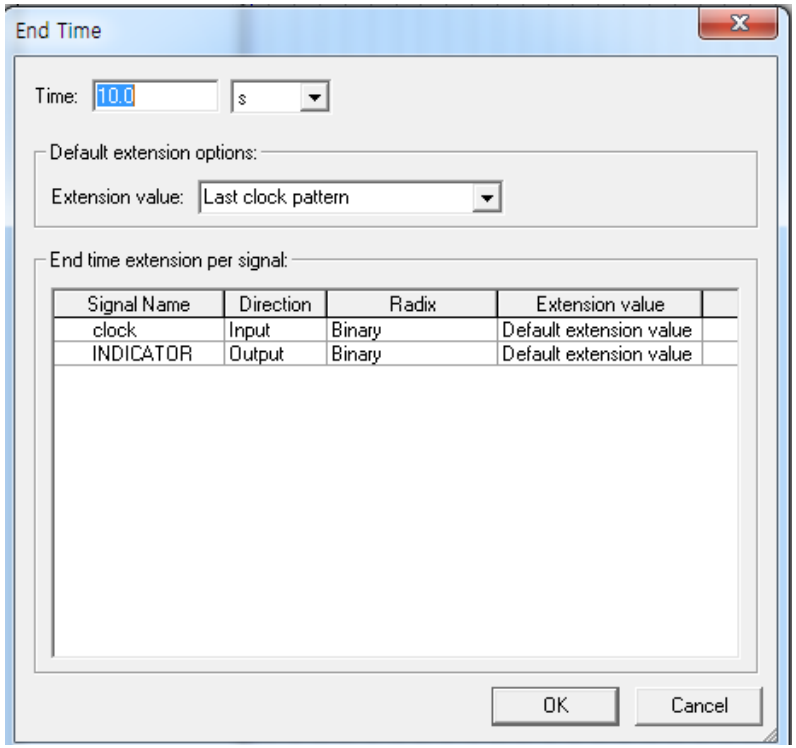


시뮬레이션 폼을 누르고 Edit 메뉴에 가면 End Time을 조정할 수 있다.

❖ Traffic Light Controller

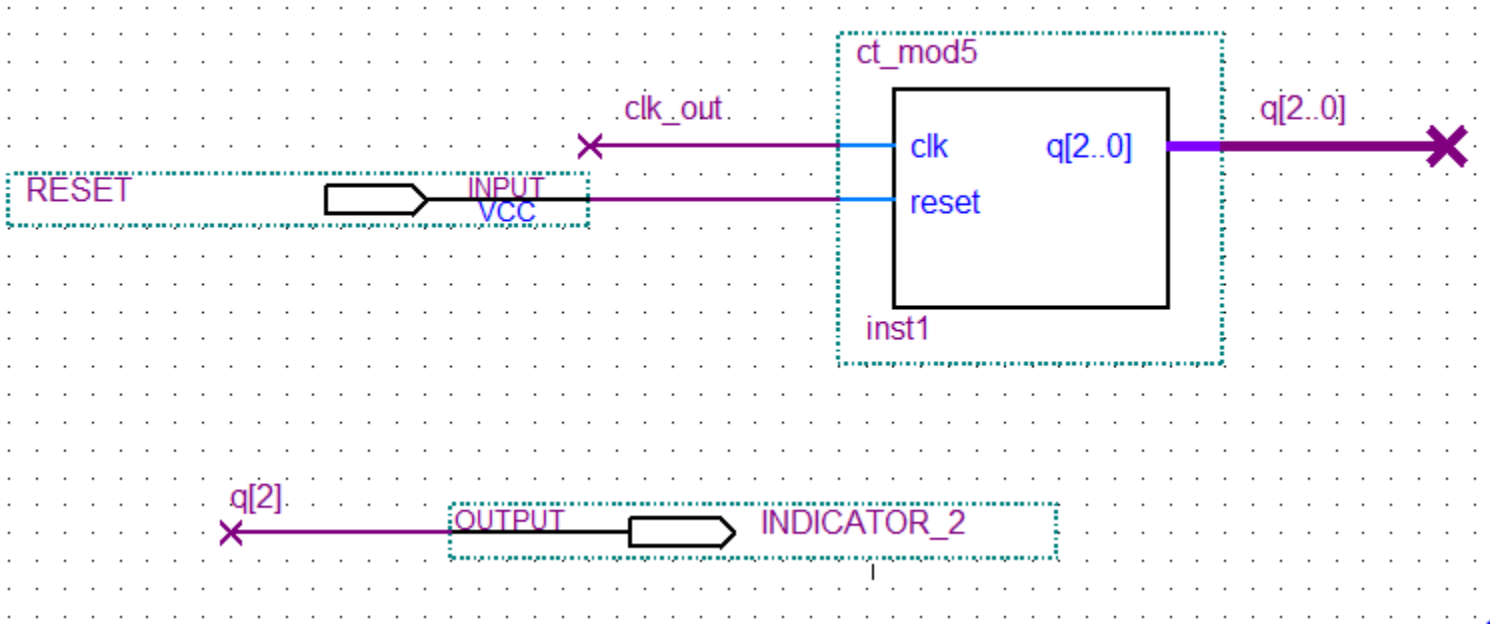
➢ 클럭 분주기

시뮬레이터 시간 설정 방법
다음 화면이 뜨면 성공, 원하는 시간을 설정해준다.



❖ Traffic Light Controller

➢ Control div 5 만들기 : 클럭 재 분주



❖ Traffic Light Controller

➢ Control div 5 만들기 : 클럭 재 분주

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY ct_mod5 IS
    PORT(
        clk, reset  : IN  STD_LOGIC;
        q           : OUT INTEGER RANGE 0 to 5);
END ct_mod5;

ARCHITECTURE a OF ct_mod5 IS
BEGIN
    PROCESS (clk, reset)
        VARIABLE count : INTEGER RANGE 0 to 5;
    BEGIN
        IF (reset = '0') THEN
            count   := 0;
        ELSE
            IF (clk'EVENT and clk = '1') THEN
                IF (count = 4) THEN
                    count := 0;
                ELSE
                    count := count + 1;
                END IF;
            END IF;
        END IF;
        q <= count;
    END PROCESS;
END a;
```

속성설정
핀, 등등

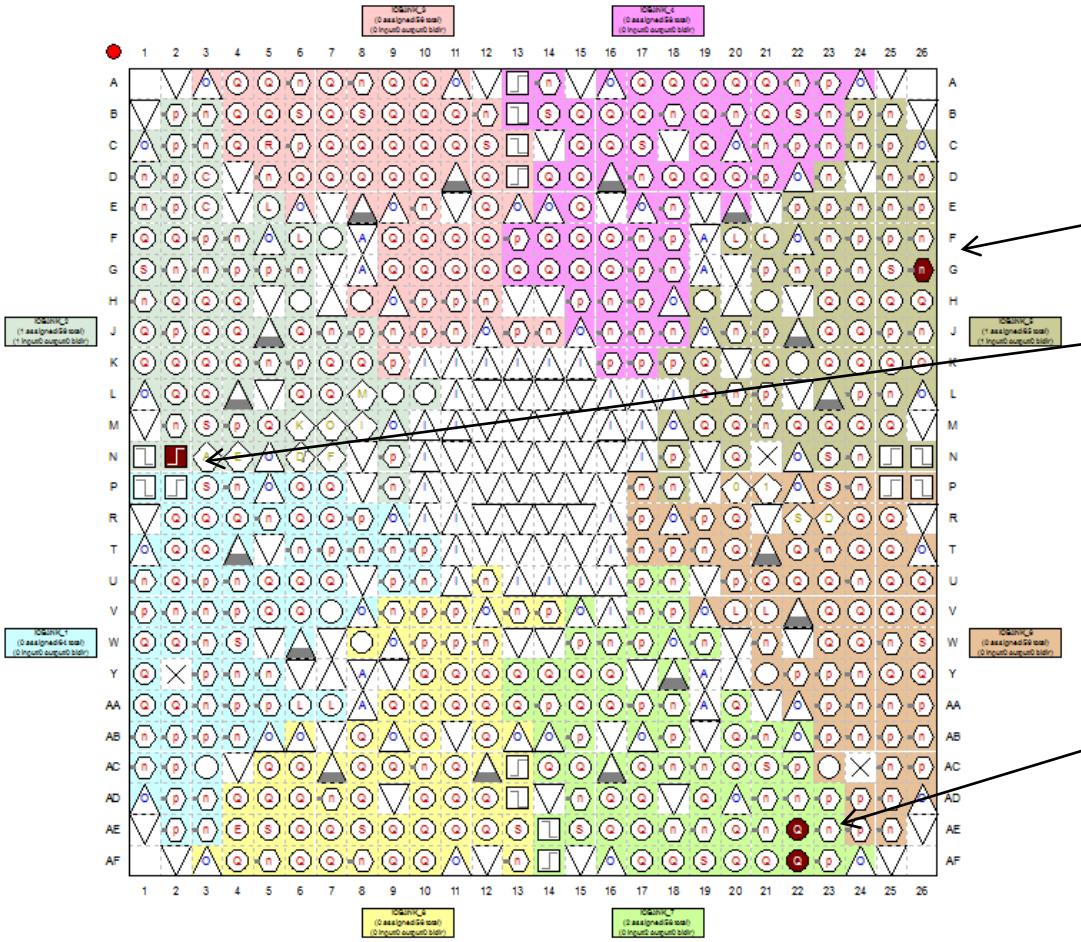
Ct_mot5활용

내부 카운터 변수

4일 때 0으로 초기화

아닌 경우 1씩 증가

Top View - Wire Bond
Cyclone II - EP2C35F672C6



스위치

내부 클럭신호

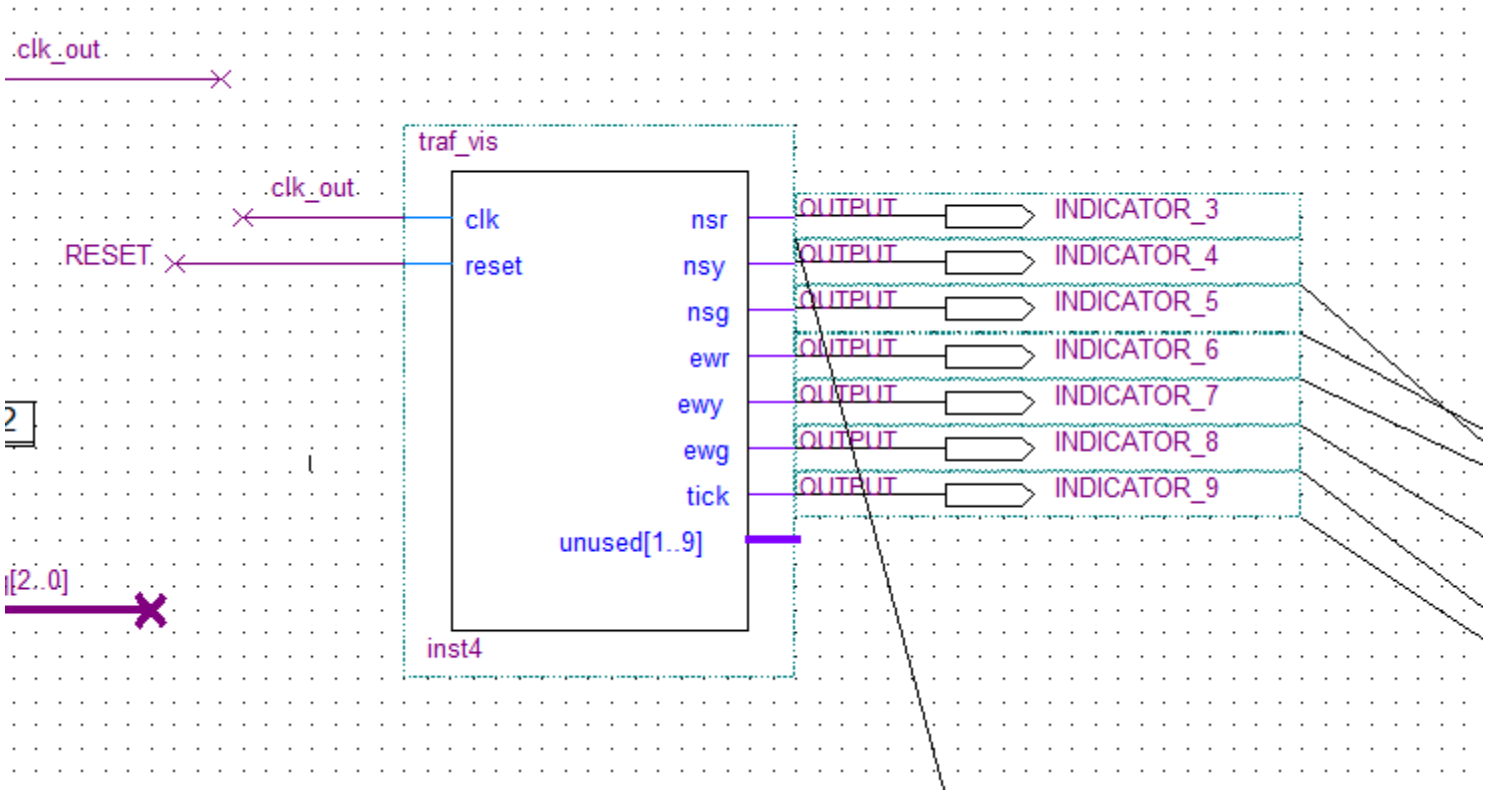
LEd

Named: Edit: Filter:

	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Group	Current Strength
1	clock	Input	PIN_N2	2	B2_N1	3.3-V LVTTL (default)			24mA (default)
2	INDICATOR	Output	PIN_AE22	7	B7_N0	3.3-V LVTTL (default)			24mA (default)
3	INDICATOR_2	Output	PIN_AF22	7	B7_N0	3.3-V LVTTL (default)			24mA (default)
4	RESET	Input	PIN_G26	5	B5_N0	3.3-V LVTTL (default)			24mA (default)
5	<<new node>>								

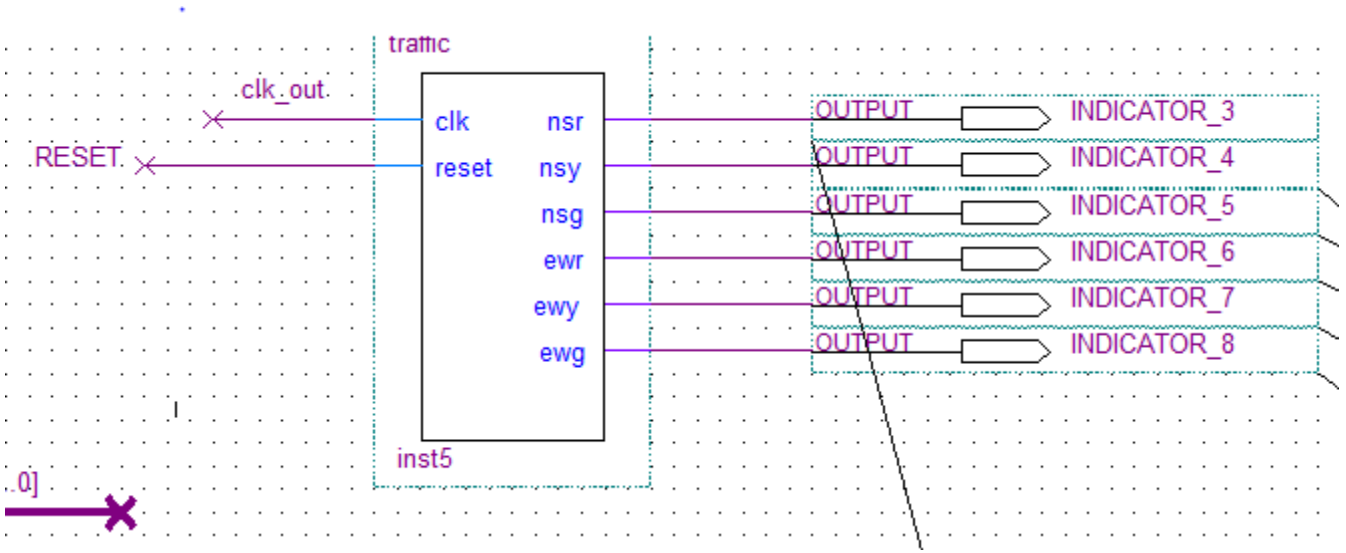
❖ Traffic Light Controller

➢ Control div 5 만들기 : 클럭 재 분주



❖ Traffic Light Controller

➢ 신호등 제어 블록 설계



❖ Traffic Light Controller

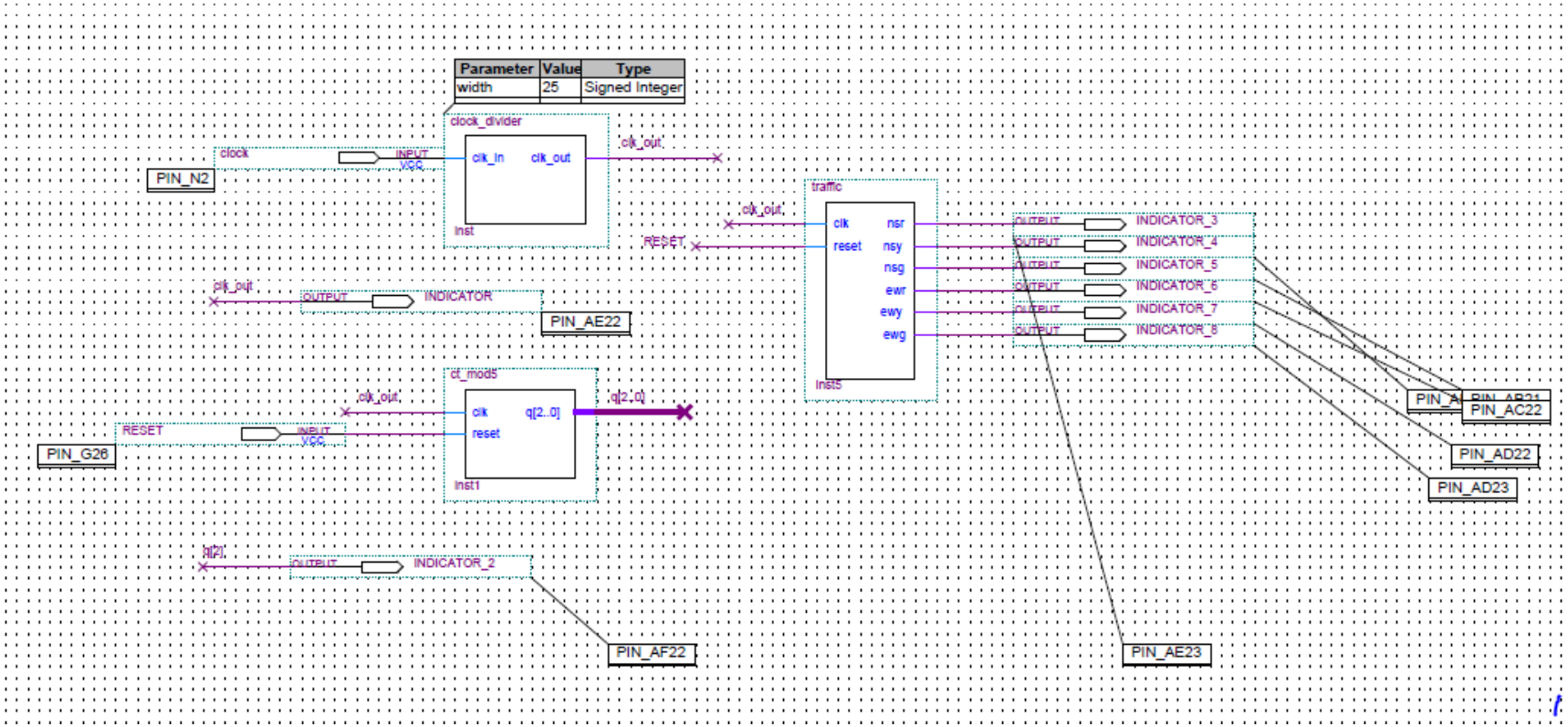
➢ 신호등 블록 설계

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY traffic IS
    PORT(
        clk, reset                    : IN  STD_LOGIC;
        nsr, nsy, nsg, ewr, ewy, ewg : OUT STD_LOGIC);
END traffic;

ARCHITECTURE a OF traffic IS
    COMPONENT ct_mod5
        PORT(
            clk, reset      : IN    STD_LOGIC;
            q               : OUT   INTEGER RANGE 0 TO 5);
    END COMPONENT;
    TYPE STATES IS (s0, s1, s2, s3);
    SIGNAL sequence : STATES;
    SIGNAL lights : STD_LOGIC_VECTOR (5 downto 0);
    SIGNAL timer : INTEGER RANGE 0 to 5;
BEGIN
light_timer: ct_mod5
    PORT MAP (  clk     =>  clk,
                reset   =>  reset,
                q       =>  timer);
    PROCESS (clk)
    BEGIN
        IF (reset = '0') THEN
            sequence <= s0;
            lights <= "011110";
        ELSIF (clk'EVENT and clk = '1') THEN
            CASE sequence IS
                WHEN s0 =>  IF timer < 4 THEN
                                sequence <= s0;
                                lights <= "011110";
                            ELSE
                                sequence <= s1;
                                lights <= "011101";
                            END IF;
                WHEN s1 =>  sequence <= s2;
                            lights <= "110011";
                WHEN s2 =>  IF timer < 4 THEN
                                sequence <= s2;
                                lights <= "110011";
                            ELSE
                                sequence <= s3;
                                lights <= "101011";
                            END IF;
                WHEN s3 =>  sequence <= s0;
                            lights <= "011110";
                WHEN others =>  sequence <= s0;
                                lights <= "011110";
            END CASE;
        END IF;
        nsr <=  lights(5);
        nsy <=  lights(4);
        nsg <=  lights(3);
        ewr <=  lights(2);
        ewy <=  lights(1);
        ewg <=  lights(0);
    END PROCESS;
END a;
```

❖ Traffic Light Controller

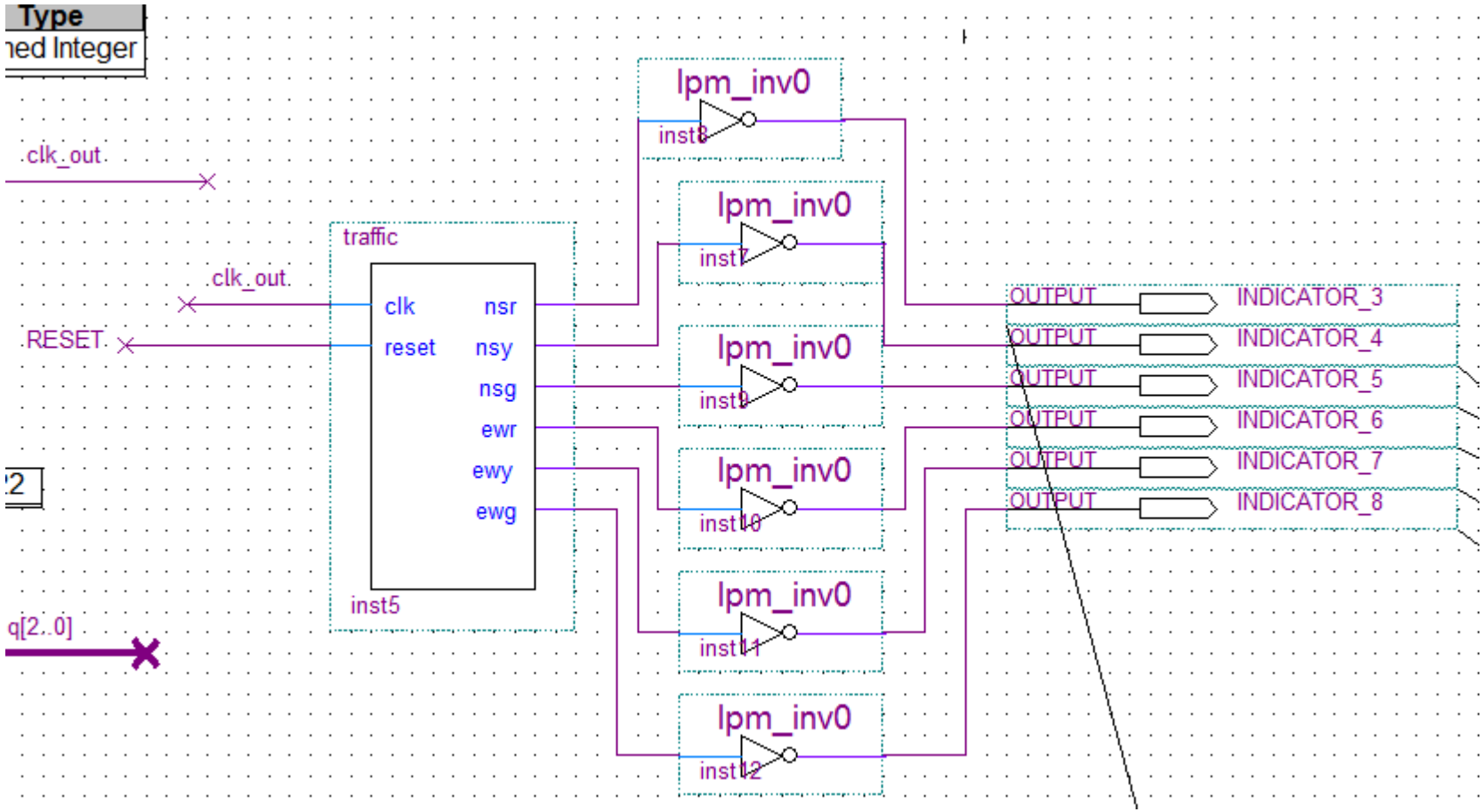
➢ 1차 기본 신호 설계

Date: October 24, 2018 traffic_light_01.bdf* Project: traffic_light_01



❖ Traffic Light Controller

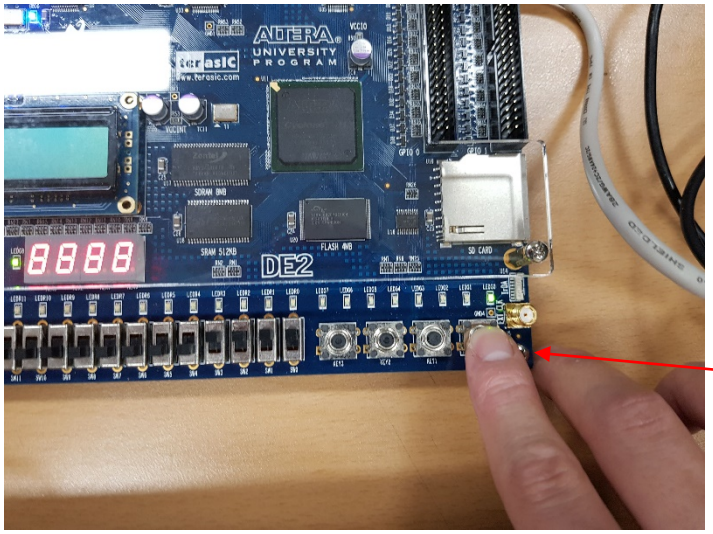
➢ 2차 신호등 블록 설계 : 회로도 수정



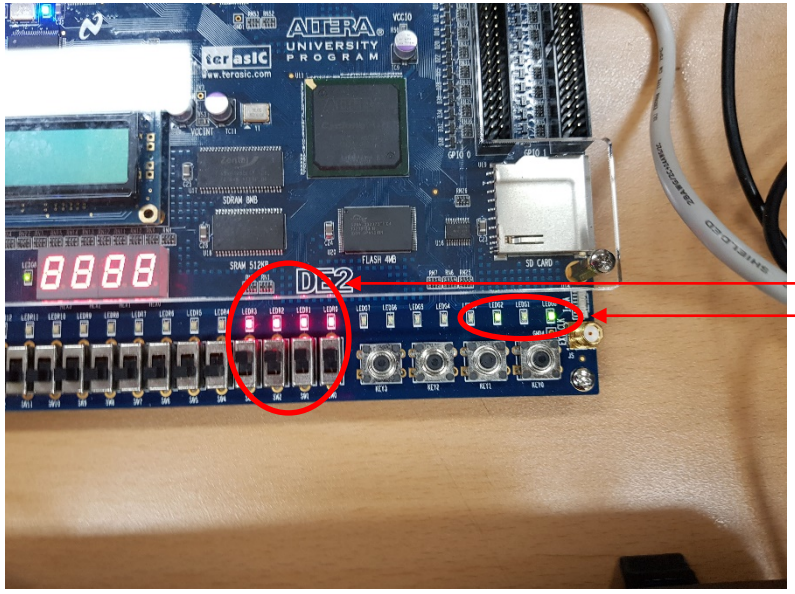
발광다이오드가 반전되어 출력되는 결과로 NOT 반전 논리 소자를적용해 올바른 신호등 출력으로 변경

❖ Traffic Light Controller

➢ 실행 결과



KEY0 리셋



	녹	주	빨	녹	주	빨