

## 실험

### 실험 목표

카운터와 비교기로 구성된 펄스폭 변조기를 설계 및 구현한 후, 이의 출력을 사용하여 LED의 밝기 및 모터의 속도를 제어함으로써 펄스폭 변조의 동작을 이해한다.

### 실험 부품

FPGA 실험 보드(EP4CE6)

USB Blaster

Quartus II

DC motor (3-volts)

LED / 저항 430 ohm

### 관련 이론

#### 카운터

카운터는 외부에서 입력되는 클럭 펄스의 수를 세는 장치이다. 카운터는 입력된 클럭 펄스의 수를 저장할 수 있는 레지스터를 가지고 있으며, 하나의 클럭 펄스가 입력될 때마다 그 값이 1씩 증가 혹은 감소하게 된다. 상향 카운터의 경우, 입력되는 클럭 펄스의 수가 계속 증가하여 이 레지스터가 가질 수 있는 값의 최대 범위를 넘어가면 오버플로우(overflow)가 발생하며 레지스터에 저장된 값은 다시 '0'으로 돌아간다. 아래의 그림 9-1은 3-비트 상향 카운터 레지스터의 값의 변화와 오버플로우를 나타낸다.

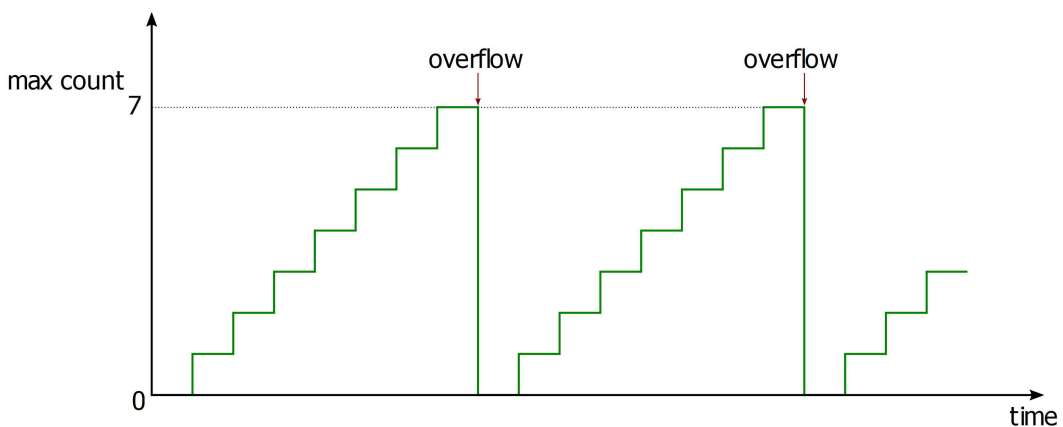


그림 9-1 3-비트 상향 카운터/타이머의 동작

### 펄스폭 변조

그림 9-2에 나타난 바와 같이 디지털 비교기(Comparator)에서는 듀티비(Duty cycle)를 결정하는 레지스터(Register)의 출력과 카운터 레지스터의 출력이 비교되고, 그 결과에 따라 비교기의 출력이 0 또는 1로 된다.

카운터에 클럭 신호를 공급하면 카운터의 값은 0에서 시작하여 최대값(max count)까지 순차적으로 증가된다. 듀티비를 결정하는 레지스터에 임의의 값을 미리 저장해 두면, 카운터의 값이 듀티비를 결정하는 레지스터의 값보다 작은 동안에는 비교기의 출력이 0이 되고, 카운터의 값이 듀티비를 결정하는 레지스터의 값보다 같거나 큰 동안에는 비교기의 출력이 1이 되어 그림 9-3과 같은 비교기 출력 파형을 얻게 된다.

비교기 출력의 한 주기에 대해 출력이 1이 되는 시간의 비율을 듀티비(Duty cycle)라 하는데, 카운터의 클럭(Clock) 신호가 일정한 주파수를 가지는 신호라면 그림 9-3에 나타난 것처럼 Register에 써넣는 값에 따라 원하는 듀티비를 가지는 비교기 출력 신호를 얻을 수 있다.

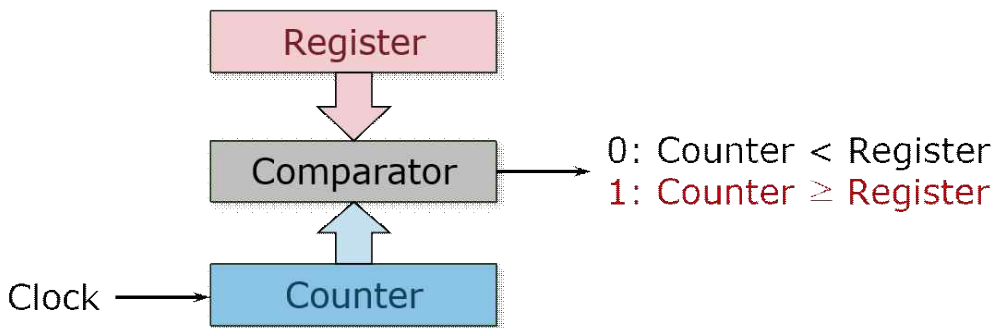


그림 9-2 펄스폭 변조기의 구성도

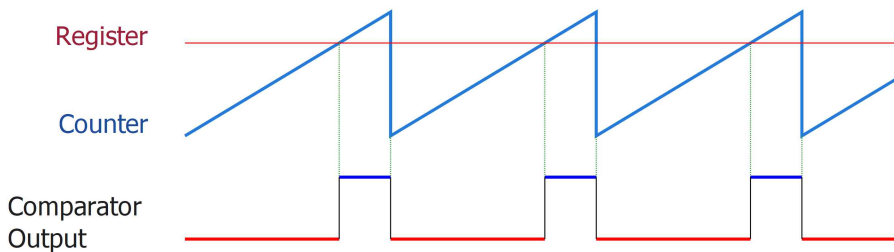


그림 9-3 펄스폭 변조기의 동작

결국 그림 9-4에 나타낸 것처럼 듀티비를 바꾸면 비교기 출력의 평균값이 바뀌는 효과를 얻을 수 있으며, 이러한 방식을 펄스폭 변조(Pulse Width Modulation, PWM) 방식이라 하고, 부하에 공급하는 평균 전류를 제어하는 곳에 많이 사용되고 있다.

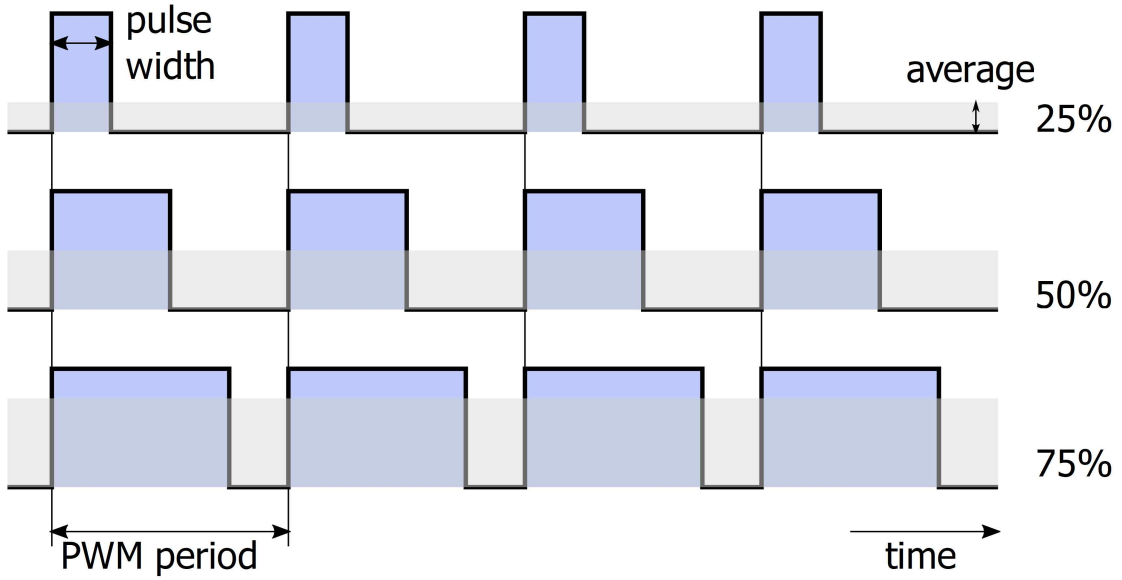


그림 9-4 펄스폭 변조에 있어서 듀티비(duty cycle)와 평균값(average)의 관계

## 실험 순서

### 실험                      신호 발생 장치 설계

아래의 그림 9-5에 나타낸 개념도를 참조하여 FPGA 실험 보드(EP4CE6)를 사용하여 4-bit PWM 신호 발생장치를 설계 및 구현하시오.

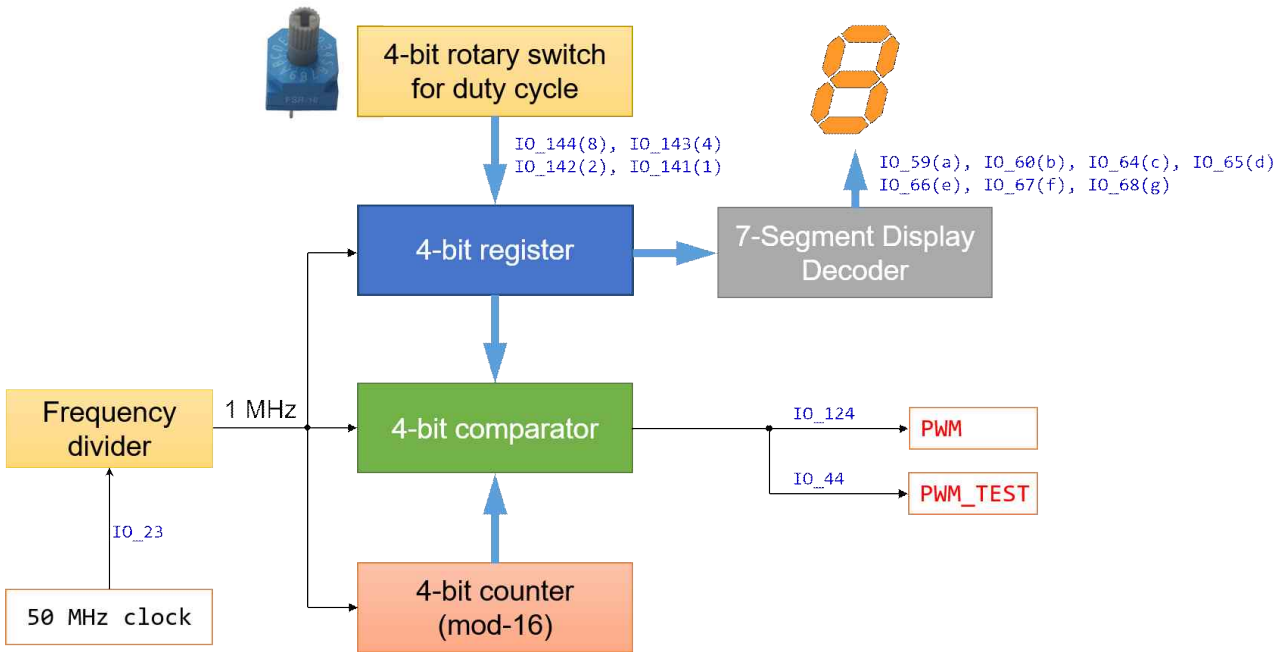


그림 9-5 4-bit PWM 신호 발생 장치

1. 외부에서 공급되는 50 MHz의 클럭 신호를 분주하여 1 MHz의 클럭 신호를 발생시킨다. 이 클럭 신호는 이후에 설명하는 모든 회로의 클럭 신호로 사용된다.
2. 분주된 1 MHz의 클럭 신호에 따라 0부터 15까지 계수하는 mod\_16\_counter를 설계.
3. FPGA 실험 보드(EP4CE6)에 장착되어 있는 FSR-16 rotary switch를 통해 duty cycle을 조정.
  - 1) FSR-16 rotary switch의 4-bit 설정값을 읽어 내부의 4-bit duty\_register에 저장.
  - 2) duty\_register에 저장된 값을 SSD(seven segment display)에 '0' ~ 'F'로 표시.
4. mod\_16\_counter와 duty\_register의 값을 비교하여 PWM 신호를 생성하고 이를 PWM(IO\_124) 핀과 PWM\_TEST (IO\_44) 핀으로 출력한다.

- 구현된 PWM 신호 발생장치의 출력 단자(PWM)에 그림 9-6과 같이 LED 혹은 직류 모터 등과 같은 적절한 부하를 연결한 후, FSR-16 rotary switch의 설정값에 따라 LED의 밝기 변화 혹은 모터의 속도 변화를 관찰한다.
- PWM 신호 발생장치의 시험용 출력 단자(PWM\_TEST)에 오실로스코프를 연결하여 출력되는 PWM 신호의 주기(주파수)와 듀티비(duty cycle)를 관찰하여 기록한다.

### 참고

FPGA 실험 보드(EP4CE6)에 장착되어 있는 그림 9-6의 LED1의 밝기도 PWM의 duty cycle에 따라 변한다.

### 주의 사항

실험시간에 주어지는 모터는 3V 용이므로 J9 Jumper Switch를 3번 핀과 4번 핀을 연결하여 모터에 3V가 공급되도록 한다.

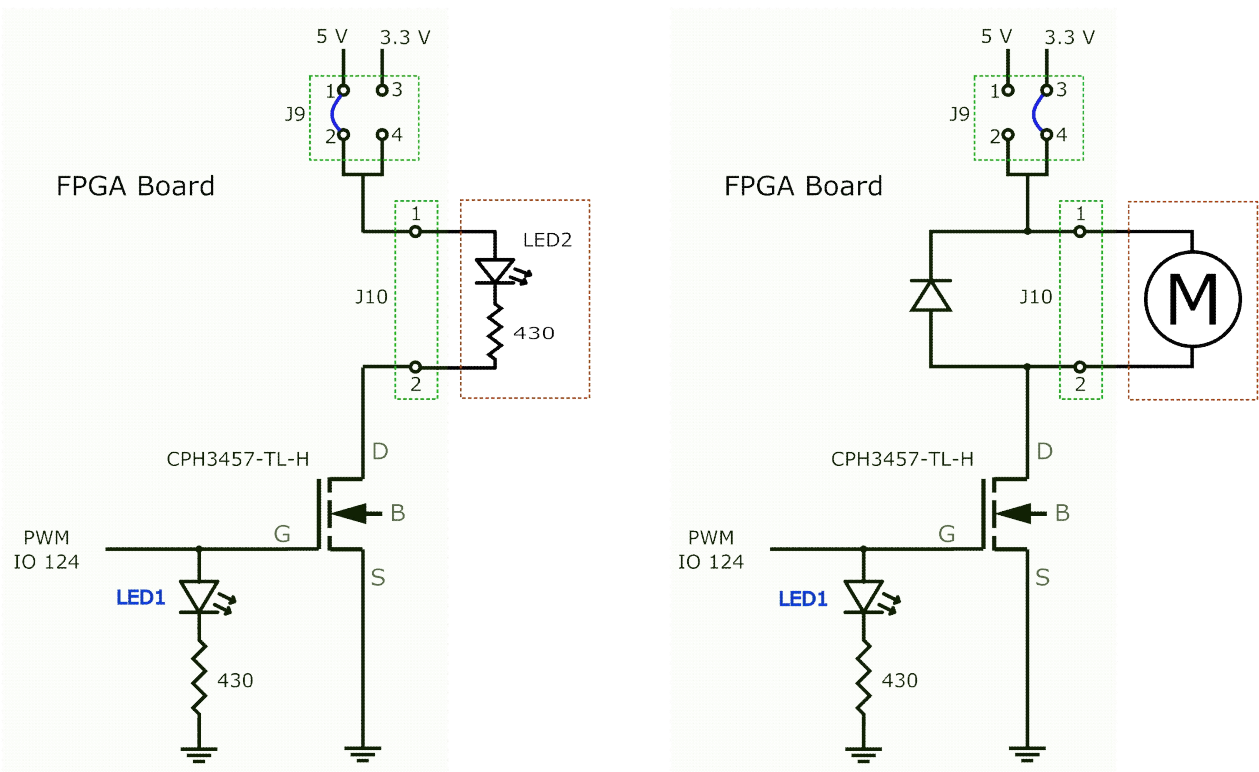


그림 9-6. PWM 장치의 시험 회로

```

-- <4-bit PWM generator>
-- use 50MHz clock without frequency division

library IEEE;
use ieee.std_logic_1164.all;

-- entity
entity pwm_4_bit_free_running is
    port( clk50MHz: in std_logic;
          duty_switch: in integer range 0 to 15;
          pwm_out: out std_logic;
          segment7: out std_logic_vector(6 downto 0));
end entity;

-- architecture
architecture my_pwm of pwm_4_bit_free_running is

    signal mod16_counter: integer range 0 to 15;
    signal duty_register: integer range 0 to 15;           -- for storing of BCD switch

begin
    -- mod-16 counter process (4-bit counter)
    process(clk50MHz)
    begin
        if (clk50MHz'event and clk50MHz = '1') then
            _____ -- update mod16_counter
        end if;
    end process;

    -- reading FSR-16 rotary switch process
    process(clk50MHz)
    begin
        if (clk50MHz'event and clk50MHz = '1') then
            _____ -- read duty cycle switch
        end if;
    end process;
end architecture;

```

```

-- compare process
process(clk50MHz)
begin
    if (clk50MHz'event and clk50MHz = '1') then
        if (mod16_counter < duty_register) then
            pwm_out    <= ____;
        else
            pwm_out    <= ____;
        end if;
    end if;
end process;

-- seven segment display process
process(duty_register)
begin
    case duty_register is
        when 0=> segment7 <=_____; -- '0'
        when 1=> segment7 <=_____; -- '1'
        when 2=> segment7 <=_____; -- '2'
        when 3=> segment7 <=_____; -- '3'
        when 4=> segment7 <=_____; -- '4'
        when 5=> segment7 <=_____; -- '5'
        when 6=> segment7 <=_____; -- '6'
        when 7=> segment7 <=_____; -- '7'
        when 8=> segment7 <=_____; -- '8'
        when 9=> segment7 <=_____; -- '9'
        when 10=> segment7 <=_____; -- 'A'
        when 11=> segment7 <=_____; -- 'b'
        when 12=> segment7 <=_____; -- 'C'
        when 13=> segment7 <=_____; -- 'd'
        when 14=> segment7 <=_____; -- 'E'
        when 15=> segment7 <=_____; -- 'F'
        when others => segment7 <= null;
    end case;
end process;
end architecture;
-- end of file

```

## 실험 결과

### 실험

1) 4-bit rotary switch의 입력에 따른 듀티비(duty cycle, %)를 계산하여 표 9-1에 작성하시오.

Rotary switch 입력	Duty cycle (%)
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

표 9-1

- 2) Rotary switch의 값을 '0'부터 '15'까지 바꾸어 가면서 각각의 값에 대한 PWM 출력의 Simulation 결과를 검토하시오.
- 3) Simulation 결과에 이상이 없으면 FPGA 소자에 프로그램한 후, 두 개의 PWM 출력에 각각 LED와 전류 제한 저항을 사용하여 회로를 구성하고, DIP switch의 설정값에 따른 LED의 밝기 변화를 관찰하고 보고서에 기록하시오.



## 결론 및 고찰

### 숙제

아래의 두 조건을 동시에 만족시키도록 PWM 발생장치의 VHDL code를 수정하고, simulation을 통해 확인하시오.

- 1) Duty cycle: 0%부터 100%
- 2) PWM 출력 신호의 주파수: 500Hz (그림 9-4 참조)