

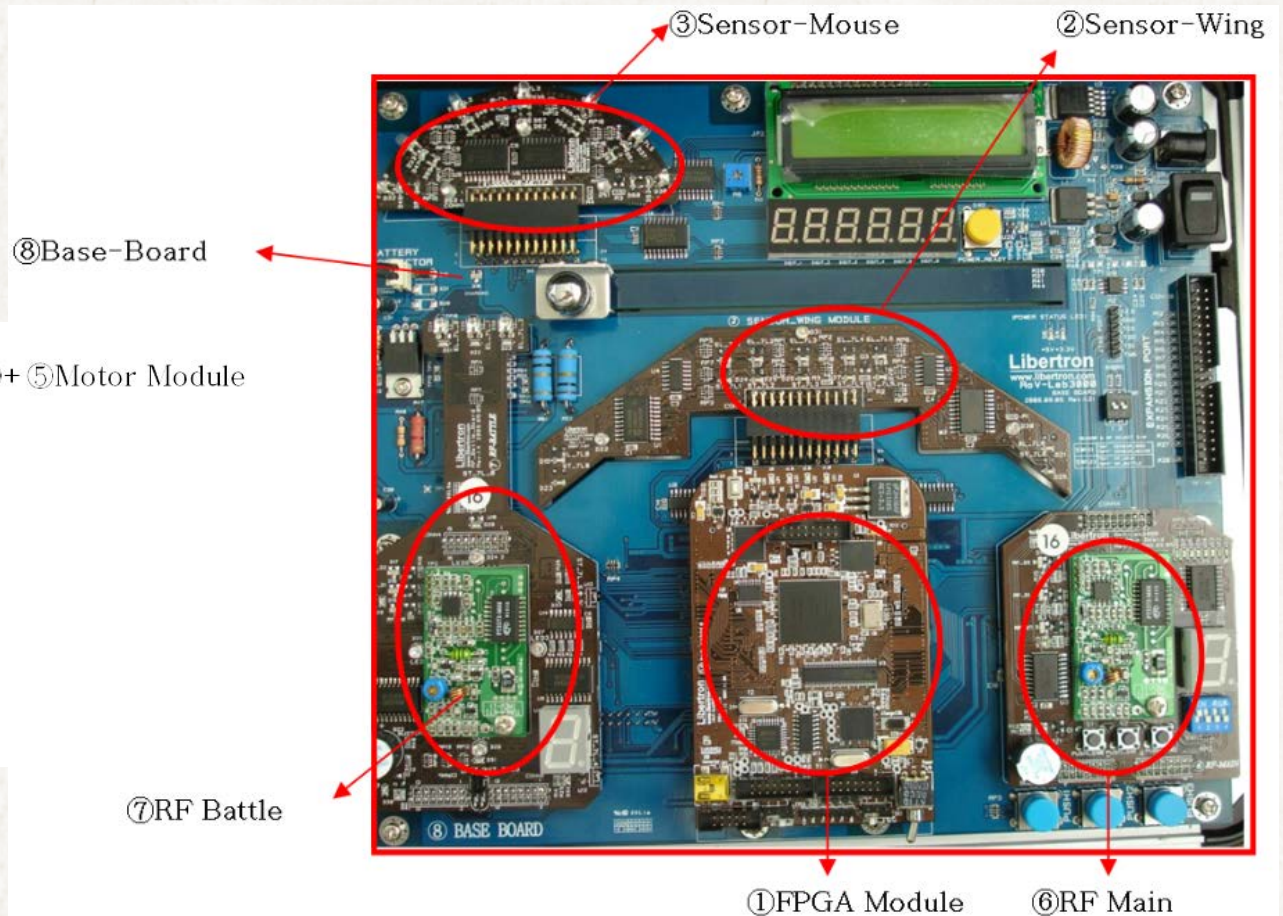
# Rov-Lab 사용자 설명

## *iRoV*Lab3000의 회로 구성

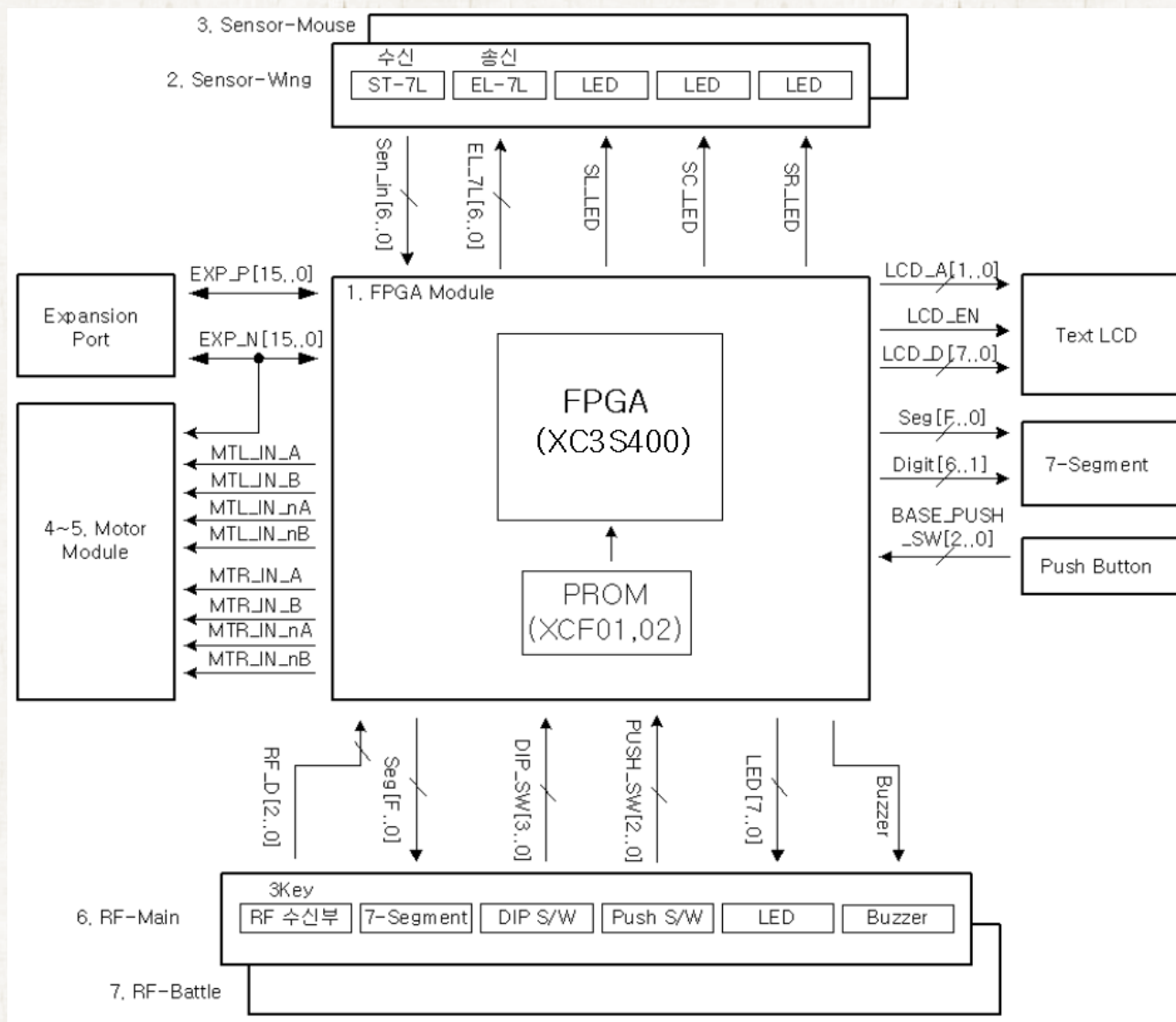
# iRoV Lab3000의 실물



④+⑤ Motor Module



# iRoV Lab3000의 블록도





# 실험시 주의사항

- 전원 연결한채로 절대로 기판을 올리거나 두껍을 닫지 말아야 함
- J-TAG케이블 연결시 단자 반드시 확인후 연결(좌측이 12V)
- 교재의 핀 할당 표는 맞지 않으니 수업시 제공하는 핀 번호 파일 사용할 것

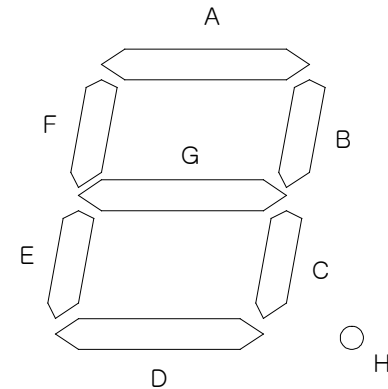
# 응용회로 설계

# 8-1. 7-Segment 제어기의 설계

## ★ 7-Segment

- 7개의 조각으로 나뉘어진 LED
- 각 세그먼트마다 A ~ G까지 이름이 붙어 있다.
- 각 A ~ G의 Segment들이 조합으로 숫자 표시필요 (Decoder가 필요하다.)

입 력				출 력						
d3	d2	d1	d0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

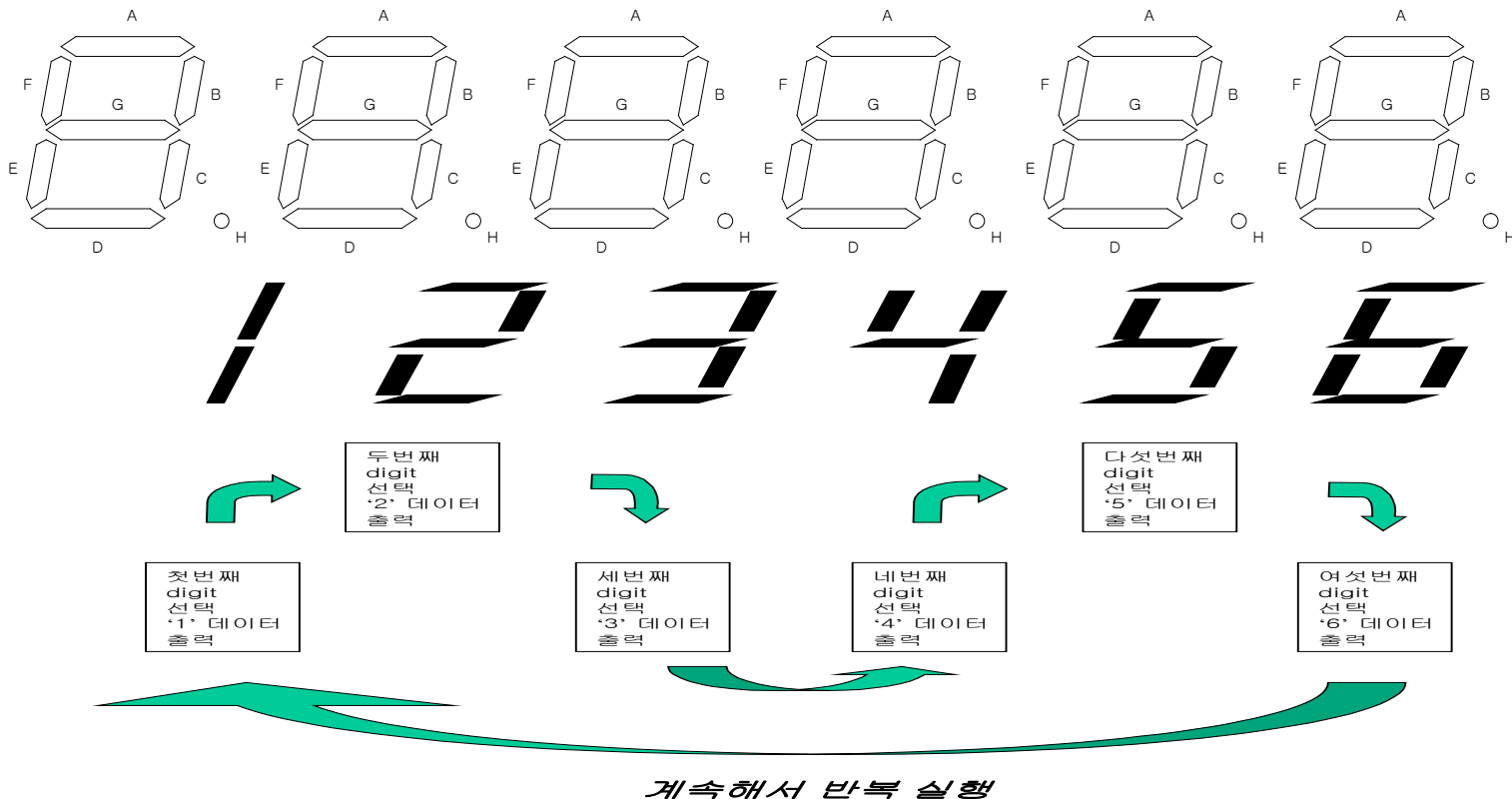


7Segment Decoder 및 LED

# 8-1. 7-Segment 제어기의 설계

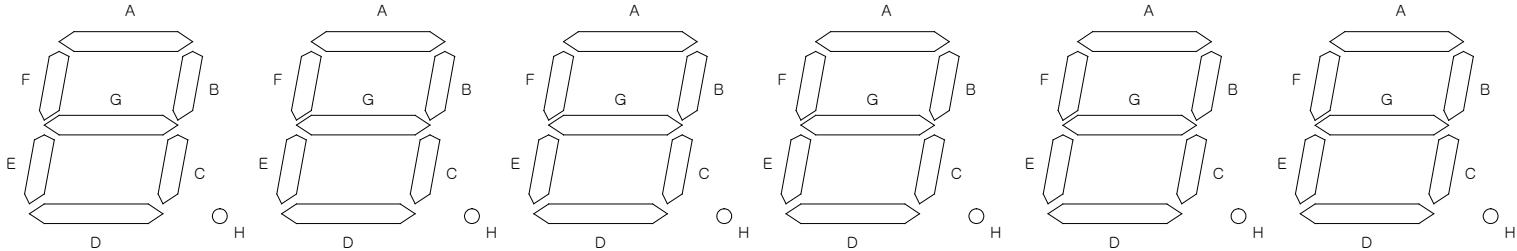
## ★ 6개의 7-Segment 동작원리

- 각 segment에 대한 LED신호가 6개 모두 공통으로 연결 되어짐
- 각 세그먼트들의 활성화 비활성화를 지정해주기 위한 Digit신호 6개가 필요
- Digit신호를 각 segment마다 일정주기로 전달하여 숫자표시





# 8-1. 7-Segment 제어기의 설계

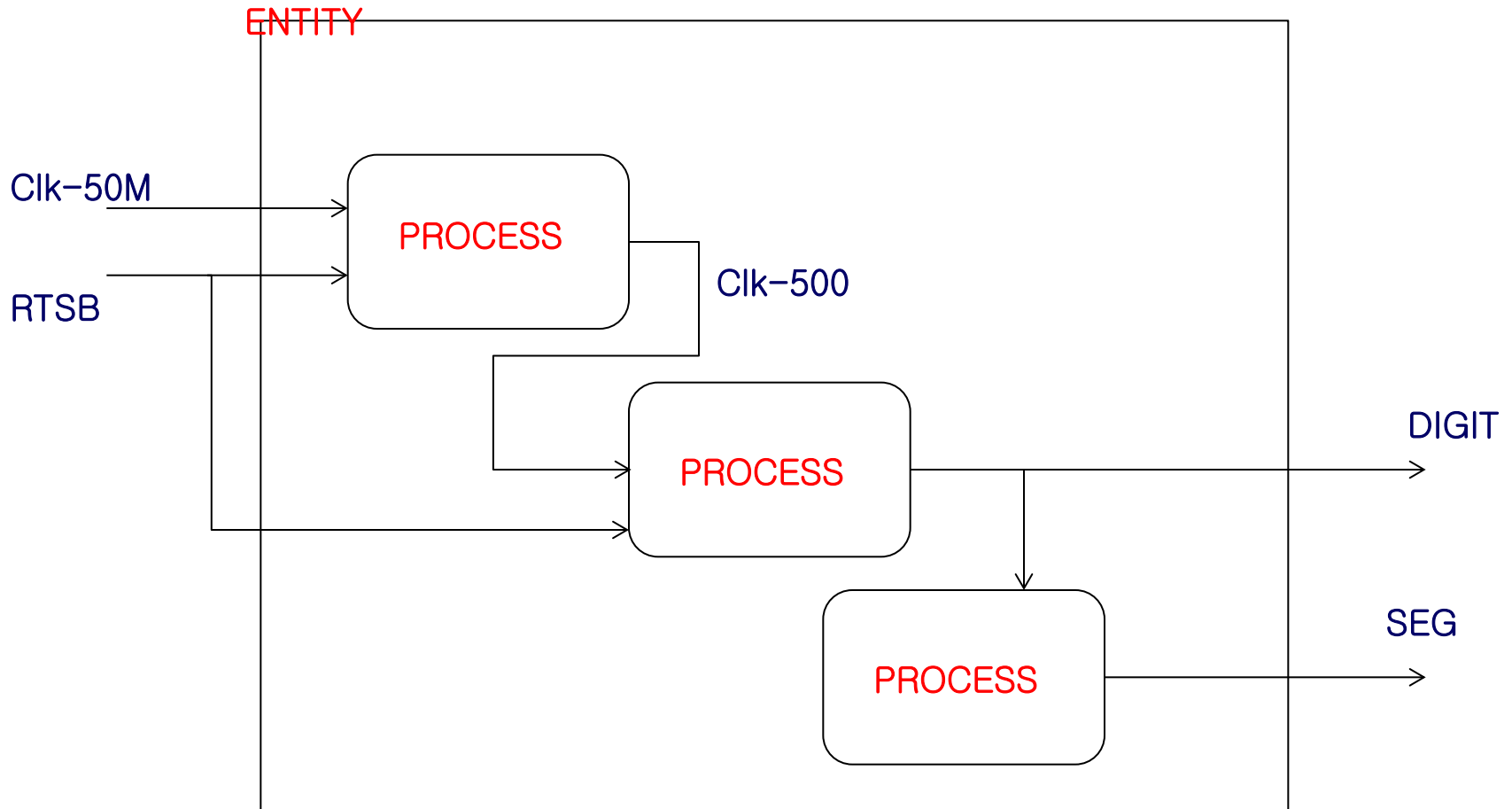


DIGIT1

DIGIT6

DIGIT1	G3	O	<p>•7-Segment Digit Value (Base Board)왼쪽1에서 오른쪽 6</p>
DIGIT2	G4	O	
DIGIT3	H3	O	
DIGIT4	H4	O	
DIGIT5	H1	O	
DIGIT6	G1	O	

## 8-1. 7-Segment 제어기의 설계- Design



# 8-1. 7-Segment 제어기의 설계- Design

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity seg7 is
    Port ( RSTB    : in std_logic;
          CLK_50M  : in std_logic ;
          DIGIT    : inout std_logic_vector(5 downto 0);
          SEG      : out std_logic_vector(6 downto 0));
end seg7;
architecture Behavioral of seg7 is
    signal clk_500 : std_logic;
begin
    ===== 자리 선택 Clock(500Hz) Generator =====
    process(RSTB,CLK_50M)
        variable cnt : integer range 0 to 50000; -- 50M/50K=1KHZ and 1KHZ/2=500 HZ
                                                -- 5000000 50M/5M=10HZ and 10HZ/2=5 HZ
    begin
        if RSTB = '0' then
            cnt := 0;          -- variable 인 경우에는 := 를 사용함
            clk_500 <= '0';    -- signal 인 경우에는 <= 를 사용함
        elsif rising_edge (CLK_50M) then
            if cnt >= 49999 then -- 정상동작시
                -- if cnt >= 1 then -- 시뮬레이션시
                    cnt := 0;
                    clk_500 <= not clk_500;
                else
                    cnt := cnt + 1;
                end if;
            end if;
        end process;
    end process;
```

# 8-1. 7-Segment 제어기의 설계- Design

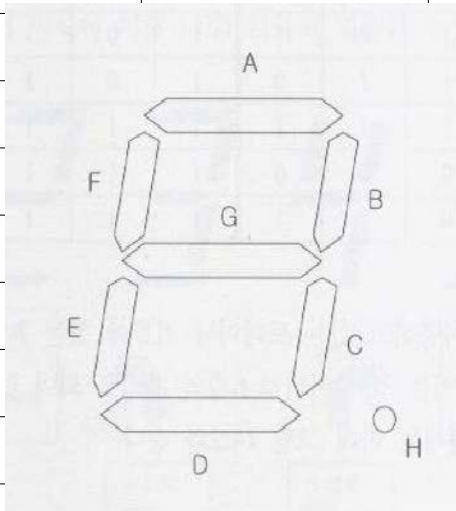
```
----- Digit selection -----
process(RSTB,clk_500)
begin
    if RSTB = '0' then
        DIGIT <= "100000";
    elsif rising_edge (clk_500) then
        DIGIT <= DIGIT(0) & DIGIT (5 downto 1); -- 선택 자리 이동
    end if;
end process;
----- 각 자리마다 숫자 표시 -----
process(DIGIT)
begin
    case DIGIT is
        when "100000" => SEG <="0000110"; --가장 오른쪽 첫번째 1 표시, GFEDCBA
        when "010000" => SEG <="1011011"; --두번째 2표시
        when "001000" => SEG <="1001111"; --세번째 3표시
        when "000100" => SEG <="0110011"; --네번째 4표시
        when "000010" => SEG <="1101101"; --다섯번째 5표시
        when "000001" => SEG <="1111101"; --여섯번째 6표시
        when others    => SEG <="0000000"; --표시 없음
    end case;
end process;
-----
end Behavioral;
```

# 8-1. 7-Segment 제어기의 설계- 각 신호기능

신호 이름	입출력	설 명
<b>CLK_50M</b>	입력	트레이닝 키트에 있는 클록 신호. <b>50 MHz</b> <b>T9</b> 번으로 연결되어 있음.
<b>RSTB</b>	입력	시스템 <b>Reset</b> 신호 <b>F4</b> 번으로 연결되어 있음. <b>Active 'L'</b>
<b>DIGIT(5:0)</b>	출력	6개의 자리 중 하나를 선택하기 위해 사용되는 신호. 6비트의 값으로 한 비트의 값만 <b>1</b> 인데 <b>500 Hz</b> 속도로 <b>1</b> 의 값이 오른쪽으로 이동한다. 6번째 비트 다음에는 다시 첫 번째 비트가 <b>1</b> 이 된다. <b>Active 'H'</b> ‘ <b>DIGIT(0) = DIGIT 1: 맨 왼쪽</b>
<b>SEG(6:0)</b>	출력	<b>7</b> 세그먼트로 연결되는 신호 7개의 세그먼트 <b>A ~ G</b> 로 연결됨. <b>Active 'H'</b> <b>SEG(0) = A</b>

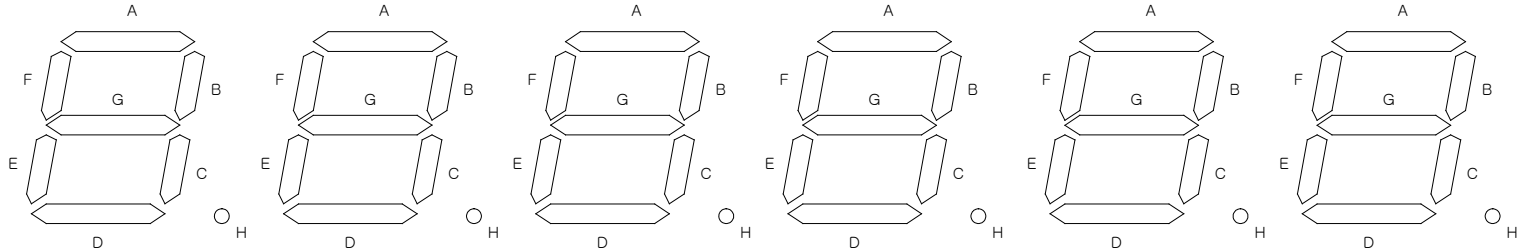
# 8-1. 7-Segment 제어기의 설계-Pin배치

입 력			출 력		
신호 이름	키트 이름	핀 번호	신호 이름	키트 이름	핀 번호
<b>CLK_50M</b>	<b>FPGA_clock</b>	<b>T9</b>	<b>DIGIT(5)</b>	<b>DIGIT6</b>	<b>G1</b>
<b>RSTB</b>	<b>Reset Pin</b>	<b>F4</b>	<b>DIGIT(4)</b>	<b>DIGIT5</b>	<b>H1</b>
			<b>DIGIT(3)</b>	<b>DIGIT4</b>	<b>H4</b>
			<b>DIGIT(2)</b>	<b>DIGIT3</b>	<b>H3</b>
			<b>DIGIT(1)</b>	<b>DIGIT2</b>	<b>G4</b>
			<b>DIGIT(0)</b>	<b>DIGIT1</b>	<b>G3</b>
			<b>SEG(6)</b>	<b>SEG_G</b>	<b>D2</b>
			<b>SEG(5)</b>	<b>SEG_F</b>	<b>D1</b>
			<b>SEG(4)</b>	<b>SEG_E</b>	<b>C3</b>
			<b>SEG(3)</b>	<b>SEG_D</b>	<b>C2</b>
			<b>SEG(2)</b>	<b>SEG_C</b>	<b>B1</b>
			<b>SEG(1)</b>	<b>SEG_B</b>	<b>C2</b>
			<b>SEG(0)</b>	<b>SEG_A</b>	<b>G2</b>





# 8-1. 7-Segment 제어기의 설계



DIGIT1(DIGIT(0))

DIGIT6(DIGIT(5))

DIGIT1	G3	O	<p>•7-Segment Digit Value (Base Board)왼쪽1에서 오른쪽 6</p>
DIGIT2	G4	O	
DIGIT3	H3	O	
DIGIT4	H4	O	
DIGIT5	H1	O	
DIGIT6	G1	O	

## 8-1. 7-Segment 제어기의 설계- Design

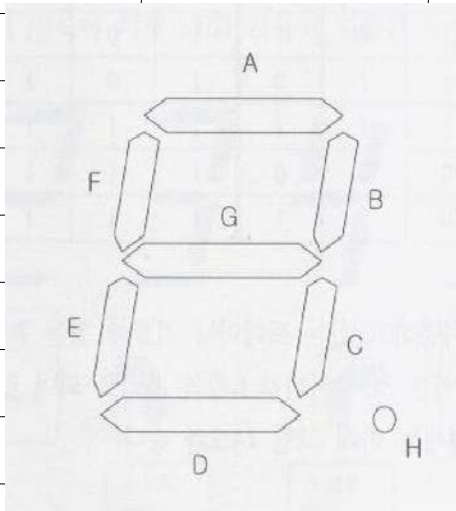
```
----- Digit selection -----
process(RSTB,clk_500)
begin
    if RSTB = '0' then
        DIGIT <= "100000";
    elsif rising_edge (clk_500) then
        DIGIT <= DIGIT(0) & DIGIT (5 downto 1); -- 선택 자리 이동
    end if;
end process;
----- 각 자리마다 숫자 표시 -----
process(DIGIT)
begin
    case DIGIT is
        when "100000" => SEG <="0000110"; --가장 왼쪽 첫번째 1 표시, GFEDCBA
        when "010000" => SEG <="1011011"; --두번째 2표시
        when "001000" => SEG <="1001111"; --세번째 3표시
        when "000100" => SEG <="0110011"; --네번째 4표시
        when "000010" => SEG <="1101101"; --다섯번째 5표시
        when "000001" => SEG <="1111101"; --여섯번째 6표시
        when others    => SEG <="0000000"; --표시 없음
    end case;
end process;
-----
end Behavioral;
```

# 8-1. 7-Segment 제어기의 설계- 각 신호기능

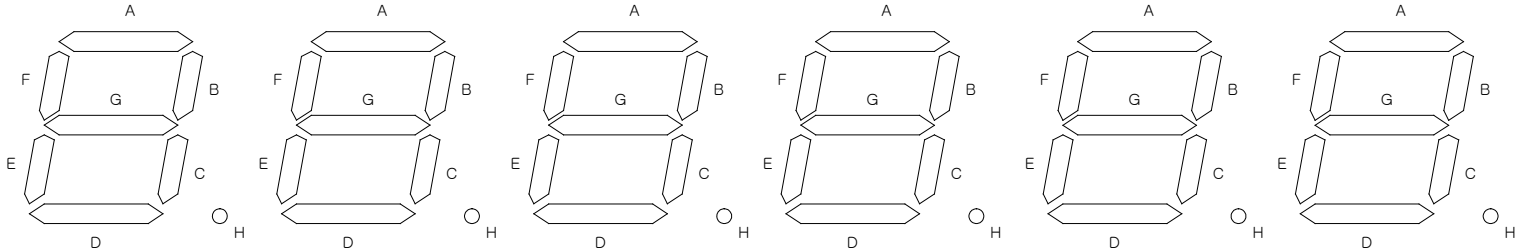
신호 이름	입출력	설 명
<b>CLK_50M</b>	입력	트레이닝 키트에 있는 클록 신호. <b>50 MHz</b> <b>T9</b> 번으로 연결되어 있음.
<b>RSTB</b>	입력	시스템 <b>Reset</b> 신호 <b>F4</b> 번으로 연결되어 있음. <b>Active 'L'</b>
<b>DIGIT(5:0)</b>	출력	<b>6</b> 개의 자리 중 하나를 선택하기 위해 사용되는 신호. <b>6</b> 비트의 값으로 한 비트의 값만 <b>1</b> 인데 <b>500 Hz</b> 속도로 <b>1</b> 의 값이 오른쪽으로 이동한다. <b>6</b> 번째 비트 다음에는 다시 첫 번째 비트가 <b>1</b> 이 된다. <b>Active 'H'</b> ‘ <b>DIGIT(5) = DIGIT 1 : 맨 왼쪽</b>
<b>SEG(6:0)</b>	출력	<b>7</b> 세그먼트로 연결되는 신호 <b>7</b> 개의 세그먼트 <b>A ~ G</b> 로 연결됨. <b>Active 'H'</b> <b>SEG(0) = A</b>

# 8-1. 7-Segment 제어기의 설계-Pin배치

입 력			출 력		
신호 이름	키트 이름	핀 번호	신호 이름	키트 이름	핀 번호
<b>CLK_50M</b>	<b>FPGA_clock</b>	<b>T9</b>	<b>DIGIT(5)</b>	<b>DIGIT1</b>	<b>G3</b>
<b>RSTB</b>	<b>Reset Pin</b>	<b>F4</b>	<b>DIGIT(4)</b>	<b>DIGIT2</b>	<b>G4</b>
			<b>DIGIT(3)</b>	<b>DIGIT3</b>	<b>H3</b>
			<b>DIGIT(2)</b>	<b>DIGIT4</b>	<b>H4</b>
			<b>DIGIT(1)</b>	<b>DIGIT5</b>	<b>H1</b>
			<b>DIGIT(0)</b>	<b>DIGIT6</b>	<b>G1</b>
			<b>SEG(6)</b>	<b>SEG_G</b>	<b>D2</b>
			<b>SEG(5)</b>	<b>SEG_F</b>	<b>D1</b>
			<b>SEG(4)</b>	<b>SEG_E</b>	<b>C3</b>
			<b>SEG(3)</b>	<b>SEG_D</b>	<b>C2</b>
			<b>SEG(2)</b>	<b>SEG_C</b>	<b>B1</b>
			<b>SEG(1)</b>	<b>SEG_B</b>	<b>C2</b>
			<b>SEG(0)</b>	<b>SEG_A</b>	<b>G2</b>



# 8-1. 7-Segment 제어기의 설계



DIGIT1(DIGIT(5))

DIGIT6(DIGIT(0))

DIGIT1	G3	O	<p>•7-Segment Digit Value (Base Board)왼쪽1에서 오른쪽 6</p>
DIGIT2	G4	O	
DIGIT3	H3	O	
DIGIT4	H4	O	
DIGIT5	H1	O	
DIGIT6	G1	O	



# The END

수고하셨습니다!!!