

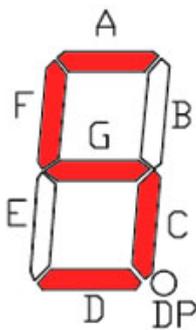
NTC FPGA 강좌 7. 7-Segment 사용하기

(주) 뉴티씨 (NewTC)

<http://www.NewTC.co.kr>

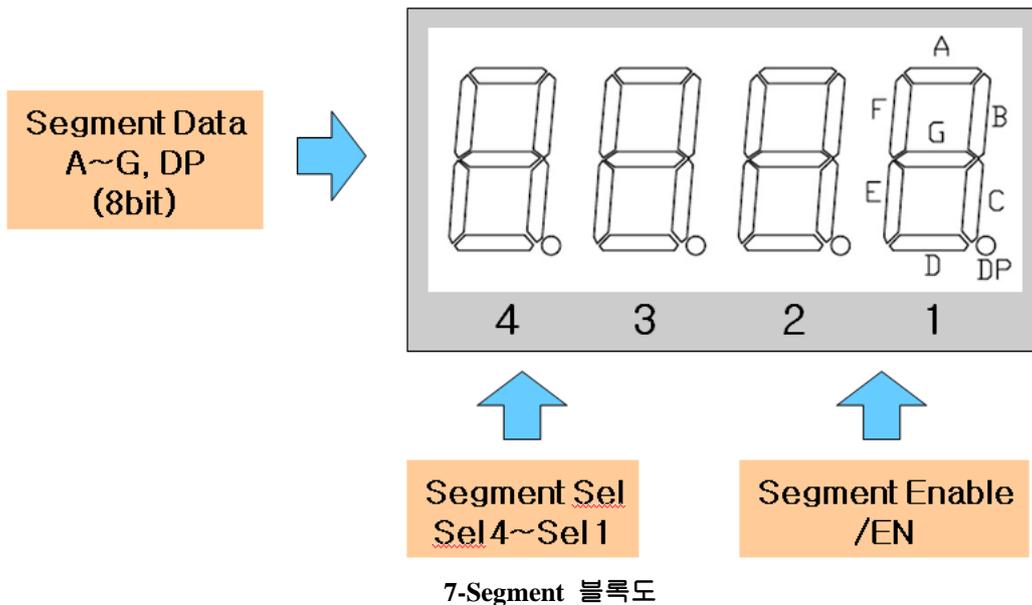
1 7-Segment 구조

7-Segment 는 7개의 세그먼트 LED로 숫자를 표현할 수 있습니다. 아래 그림과 같이 각 세그먼트는 A~G 와 DP 세그먼트로 구성되어 있습니다.



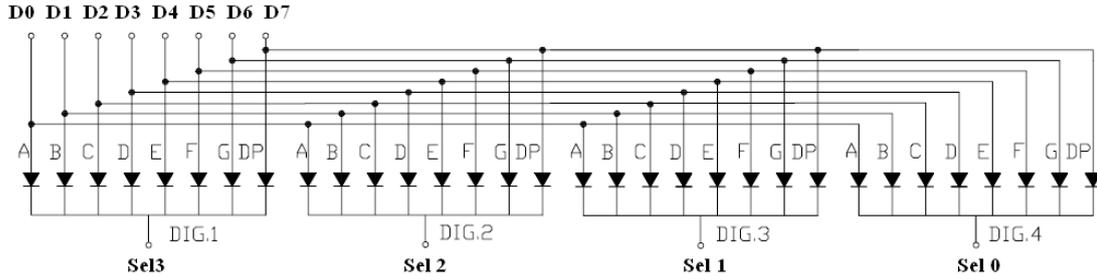
각 세그먼트를 켜고 끄으로써 0~9 숫자를 표현할 수 있습니다. 예를 들어 숫자 5는 왼쪽 그림과 같이 A, F, G, C, D 5개 세그먼트를 켜서 숫자 1을 표현할 수 있습니다. 각 세그먼트는 LED로 구성되어 있기 때문에 극성에 맞추어 전원을 공급하면 해당 세그먼트는 켜지게 됩니다. 7-Segment 는 LED 어레이로 한쪽이 Common 단자로 묶여있고 반대쪽으로 데이터를 인가하여 켜고 끄는 것을 제어하게 됩니다.

FM-CY6S에 사용된 7-Segment 는 4개가 결합되어 있는 것으로 A~G, DP 세그먼트는 공유를 하고 각 자리를 선택하는 SEL0~3 신호가 있습니다. 잔상 효과를 이용한 것으로 4개의 7-Segment가 순차적으로 빠르게 깜빡이면 사람은 4개의 숫자가 모두 켜진 것으로 인식을 하게 되는 원리입니다.



사용된 7-Segment 는 Common Cathode 형으로 각 세그먼트의 Cathode가 공통 핀으로 여기에 GND 가 연결되면 해당되는 자리 숫자가 활성화 됩니다. 이때 A~G, DP 입력이 ‘1’인

세그먼트에 불이 켜지게 됩니다.



7-Segment 출력 데이터

	D7	D6	D5	D4	D3	D2	D1	D0	Data
	DP	G	F	E	D	C	B	A	16진수
	0	0	1	1	1	1	1	1	0x3f
	0	0	0	0	0	1	1	0	0x06
	0	1	0	1	1	0	1	1	0x5b
	0	1	0	0	1	1	1	1	0x4f
	0	1	1	0	0	1	1	0	0x66
	0	1	1	0	1	1	0	1	0x6d
	0	1	1	1	1	1	0	0	0x7c
	0	0	1	0	0	1	1	1	0x27
	0	1	1	1	1	1	1	1	0x7f
	0	1	1	0	0	1	1	1	0x67
	1	0	0	0	0	0	0	0	0x80

※ Common Anode 형 7-Segment 는 위 데이터를 반전하여 출력하면 됩니다.

2 7-Segment 동작 시키기

7-Segment 에 4개의 숫자가 모두 켜진 것 처럼 보이려면 아래와 같이 4개의 숫자를 순차적으로 켜주는 동작을 빠르게 반복해야 합니다. 아래 그림과 같이 “0123” 숫자를 표현하는 것입니다. 해당 신호를 인가한 후 적당한 시간 지연 후 다음 신호를 인가하게 됩니다. (/EN 신호에는 항상 “0”신호를 인가해야 합니다.)

	D7	D6	D5	D4	D3	D2	D1	D0	S3	S2	S1	S0
	0	0	1	1	1	1	1	1	0	1	1	1
	0x3f											

	0	0	0	0	0	1	1	0	1	0	1	1
	0x06											

	0	1	0	1	1	0	1	1	1	1	0	1
	0x5b											

	0	1	0	0	1	1	1	1	1	1	1	0
	0x4f											

	위 동작을 빠르게 반복하면 왼쪽 그림과 같이 “0123”이 켜진 것 처럼 보이게 됩니다.											
--	---	--	--	--	--	--	--	--	--	--	--	--

※ 설계 Tip

SEL 신호는 0,1,2,3 으로 4가지 경우가 있습니다. 또한 중간에 정확한 타이밍을 필요로 하지 않는 시간 지연이 필요합니다. 이럴 경우 카운터를 사용하면서 최상위 또는 하위의 연속된 2비트로 SEL 신호를 발생 시키면 효과적으로 설계할 수 있습니다.

```

reg [31:0] counter
always @(posedge clk)
    if(reset == 0)
        counter <= 0;
    else
        counter <= counter + 1;
assign sel[1:0] = counter[31:30];
    
```

위 카운터에서 counter[31:30] 2비트를 SEL 신호를 발생시키는데 사용합니다. 이 경우 counter[29:0]가 증가하는 동안 자동으로 시간 지연이 발생하게 됩니다. 시간 지연을 짧게 하려면 하위 비트 counter[10:9]와 같이 변경해서 사용합니다.

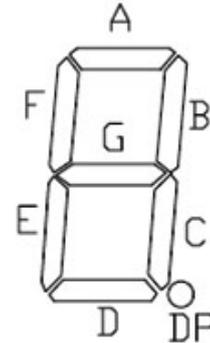
3 7-Segment 하드웨어 구성

3.1 7-Segment 컨트롤 핀 번호

Seven-Segment Data 출력 핀 번호

Data[7]	Data[6]	Data[5]	Data[4]	Data[3]	Data[2]	Data[1]	Data[0]
187	188	195	196	197	198	199	200

Data[7] Seven Segment – A Segment
 Data[6] Seven Segment – B Segment
 Data[5] Seven Segment – C Segment
 Data[4] Seven Segment – D Segment
 Data[3] Seven Segment – E Segment
 Data[2] Seven Segment – F Segment
 Data[1] Seven Segment – G Segment
 Data[0] Seven Segment – Dot



Seven-Segment 컨트롤 출력 핀 번호

SEL[4]	SEL[3]	SEL[2]	SEL[1]	/EN
203	204	205	206	201

SEL[n] n 번째 자릿수 Enable (H:Disable, L:Enable)
 /EN 7-Segment Enable (H:Disable, L:Enable)

실습 과제

1. 7-Segment 1개에 숫자를 켜고 0부터 9까지 반복하여 카운트 하도록 설계합니다.
2. 카운터를 확장하여 0에서 1000까지 카운트 하여 7-Segment에 값을 출력합니다. 먼저 임의의 4자리 숫자를 출력하는 로직을 설계합니다. 그리고 값을 1초 단위로 증가하도록 설계합니다. (아래 소스는 정수를 4개의 BCD(Binary-Coded Decimal)로 표현하는 것입니다. 나누기 연산은 로직을 많이 사용하지만 이번 강좌에서는 나누기 연산을 이용하여 간단히 구현하도록 합니다.)

```
ss_num_0 <= (ss_int_buff/1000)%100;
ss_num_1 <= (ss_int_buff/100)%10;
ss_num_2 <= (ss_int_buff/10)%10;
ss_num_3 <= (ss_int_buff%10);
```