

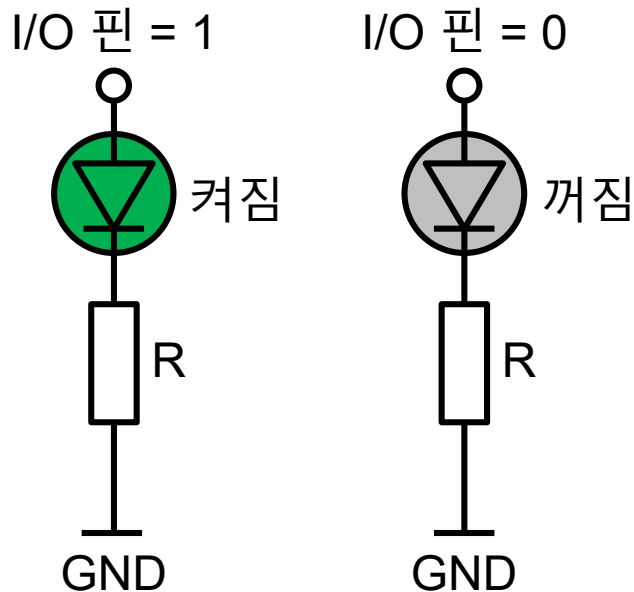
Lecture 04

FND 제어

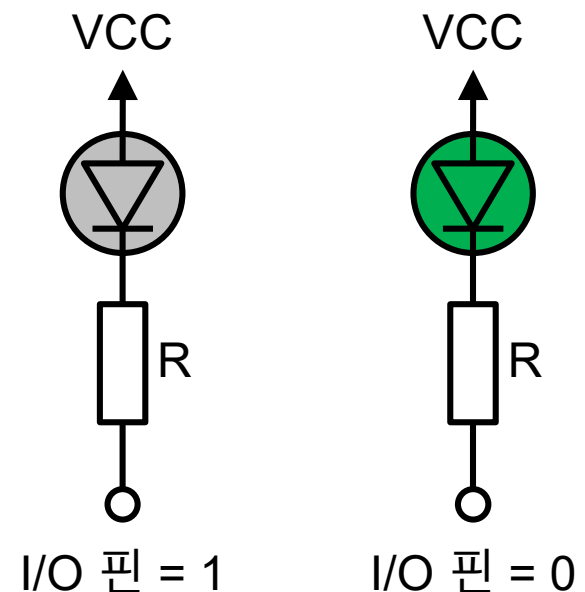
LED 제어

LED 제어 방법

Active high

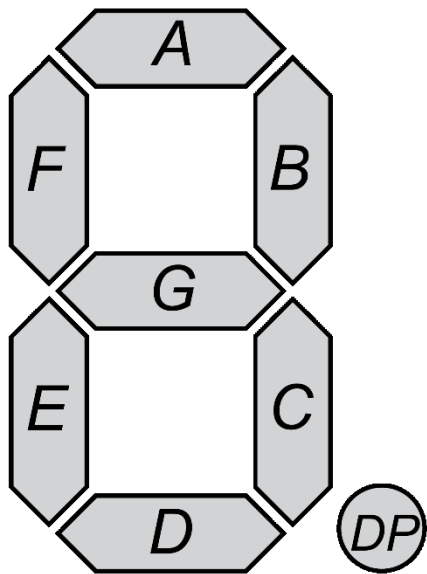


Active low

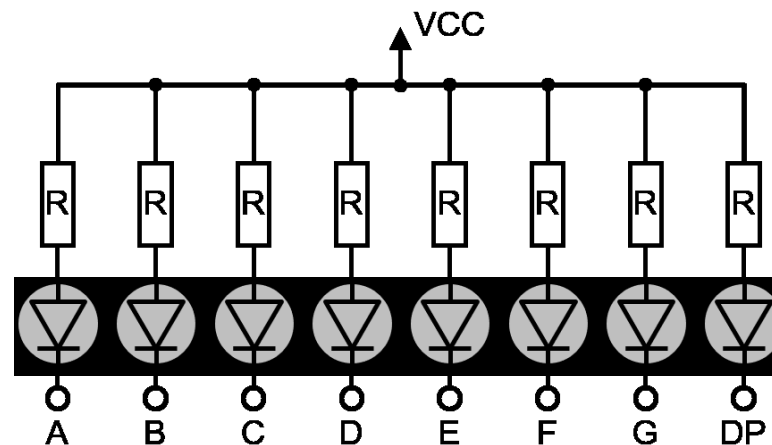
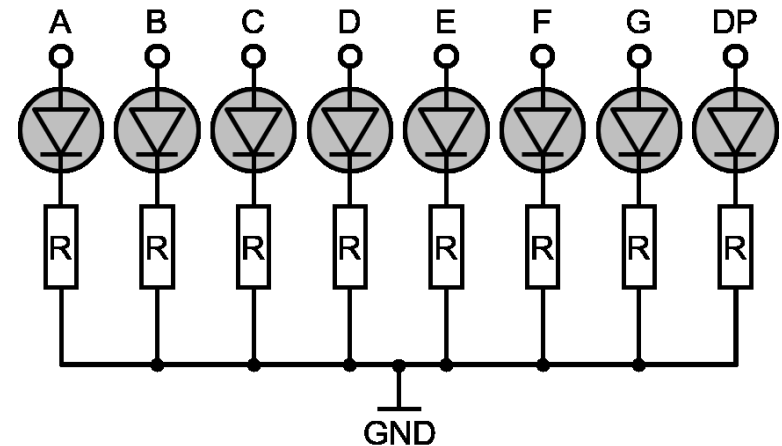


FND (Flexible Numeric Display)

- FND (또는 7-segment LED)



**Active high 또는
common cathode (CC)**



**Active low 또는
common anode (CA)**

FND (Flexible Numeric Display)

■ FND 제어 방법

숫자	Active high (또는 CC)								16진수
	DP	G	F	E	D	C	B	A	
0	0	0	1	1	1	1	1	1	0x3f
1	0	0	0	0	0	1	1	0	0x06
2	0	1	0	1	1	0	1	1	0x5b
3	0	1	0	0	1	1	1	1	0x4f
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6d
6	0	1	1	1	1	1	0	1	0x7d
7	0	0	1	0	0	1	1	1	0x27
8	0	1	1	1	1	1	1	1	0x7f
9	0	1	1	0	0	1	1	1	0x67

숫자	Active high (또는 CC)								16진수
	DP	G	F	E	D	C	B	A	
a	0	1	1	1	0	1	1	1	0x77
b	0	1	1	1	1	1	0	0	0x7c
c	0	1	0	1	1	0	0	0	0x58
d	0	1	0	1	1	1	1	0	0x5e
e	0	1	1	1	1	0	0	1	0x79
f	0	1	1	1	0	0	0	1	0x71

FND (Flexible Numeric Display)

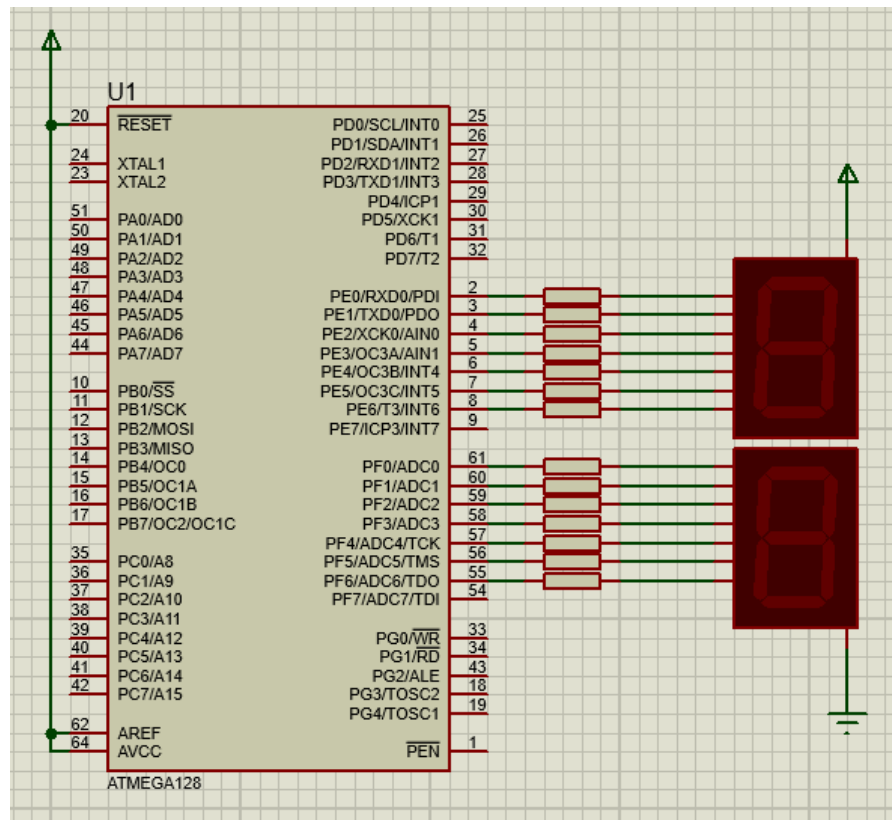
■ FND 제어 방법

숫자	Active low (또는 CA)								16진수
	DP	G	F	E	D	C	B	A	
0	1	1	0	0	0	0	0	0	0xc0
1	1	1	1	1	1	0	0	1	0xf9
2	1	0	1	0	0	1	0	0	0xa4
3	1	0	1	1	0	0	0	0	0xb0
4	1	0	0	1	1	0	0	1	0x99
5	1	0	0	1	0	0	1	0	0x92
6	1	0	0	0	0	0	1	0	0x82
7	1	1	0	1	1	0	0	0	0xd8
8	1	0	0	0	0	0	0	0	0x80
9	1	0	0	1	1	0	0	0	0x98

숫자	Active low (또는 CA)								16진수
	DP	G	F	E	D	C	B	A	
a	1	0	0	0	1	0	0	0	0x88
b	1	0	0	0	0	0	1	1	0x83
c	1	0	1	0	0	1	1	1	0xa7
d	1	0	1	0	0	0	0	1	0xa1
e	1	0	0	0	0	1	1	0	0x86
f	1	0	0	0	1	1	1	0	0x8e

FND (Flexible Numeric Display)

■ CC타입 및 CA타입 FND 실습



FND (Flexible Numeric Display)

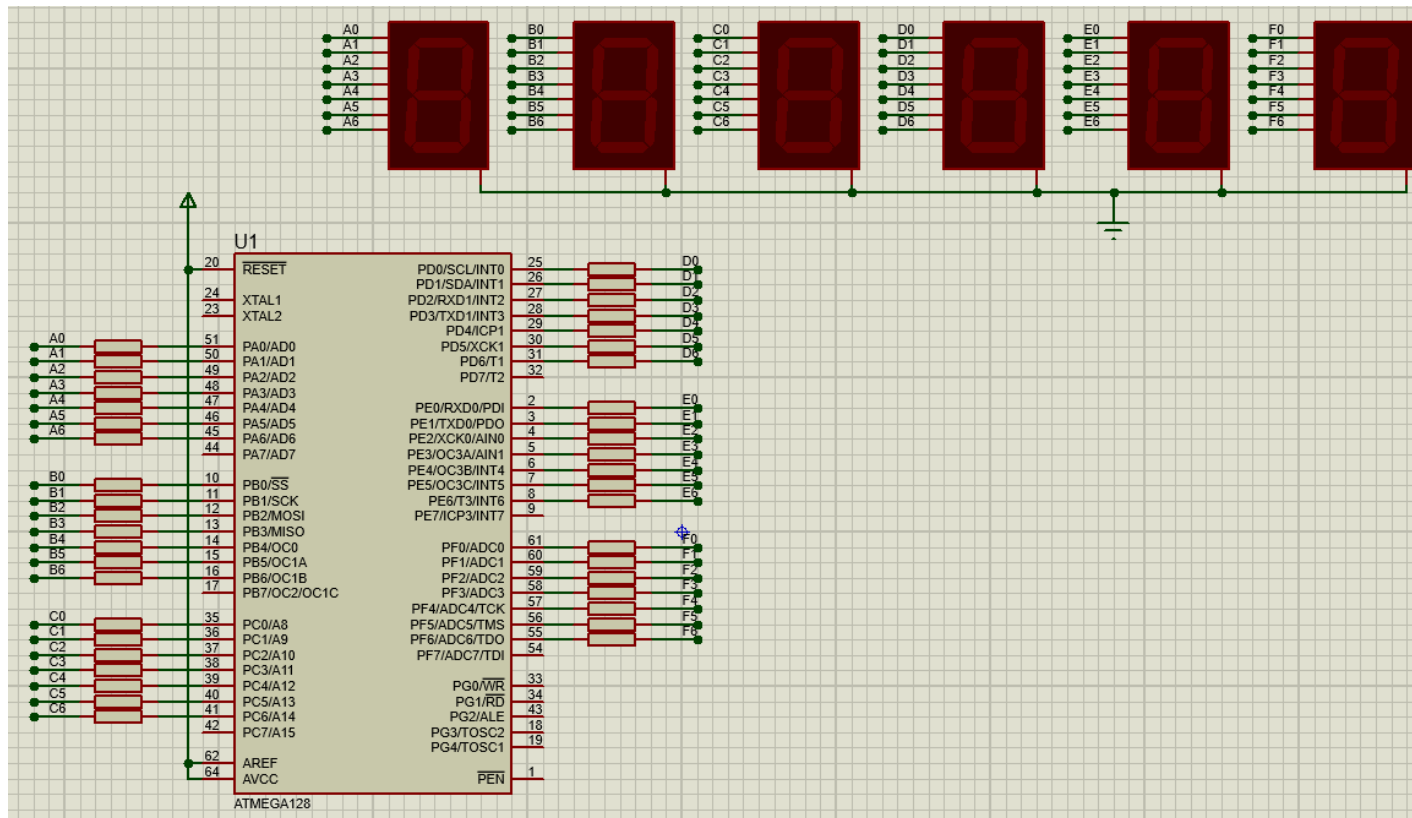
■ CC타입 및 CA타입 FND 실습

```
unsigned char CC[] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d,  
0x27, 0x7f, 0x67, 0x77, 0x7c, 0x58, 0x5e, 0x79, 0x71, 0x80};  
unsigned char CA[] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,  
0xd8, 0x80, 0x98, 0x88, 0x83, 0xa7, 0xa1, 0x86, 0x8e, 0x7f};
```

```
int main(void) {  
    DDRE = DDRF = 0xff;  
    int i = 0;  
    while(1) {  
        PORTE = CA[i];  
        PORTF = CC[i];  
        i++;  
        if (i>15) i=0;  
        _delay_ms(300);  
    }  
}
```

FND (Flexible Numeric Display)

■ 디지털 시계 실습



FND (Flexible Numeric Display)

■ 디지털 시계 실습

```
void display(void);
void time_process(void);

unsigned char CC[] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d,
0x27, 0x7f, 0x67, 0x77, 0x7c, 0x58, 0x5e, 0x79, 0x71, 0x80};
unsigned char CA[] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,
0xd8, 0x80, 0x98, 0x88, 0x83, 0xa7, 0xa1, 0x86, 0x8e, 0x7f};
unsigned char h, m, s;

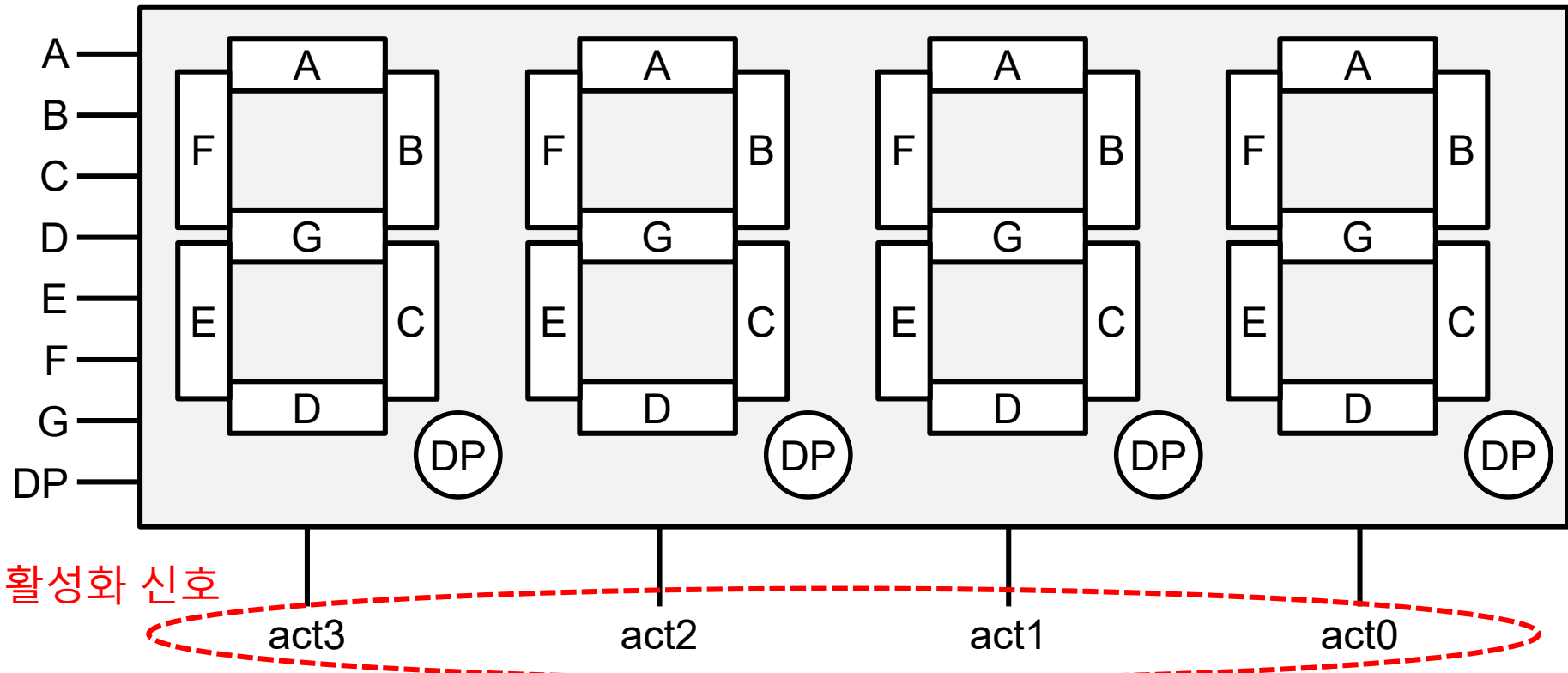
int main(void) {
    DDRA = DDRB = DDRC = DDRD = DDRE = DDRF = 0xff;
    h = m = s = 0;
    while(1) {
        display();
        _delay_ms(1000);
        time_process();
    }
}
```

```
void display(void) {
    PORTA = CC[h/10];
    PORTB = CC[h%10];
    PORTC = CC[m/10];
    PORTD = CC[m%10];
    PORTE = CC[s/10];
    PORTF = CC[s%10];
}

void time_process(void) {
    s++;
    if (s>59) {
        s = 0;
        m++;
        if (m>59) {
            m = 0;
            h++;
            if (h>12) h = 1;
        }
    }
}
```

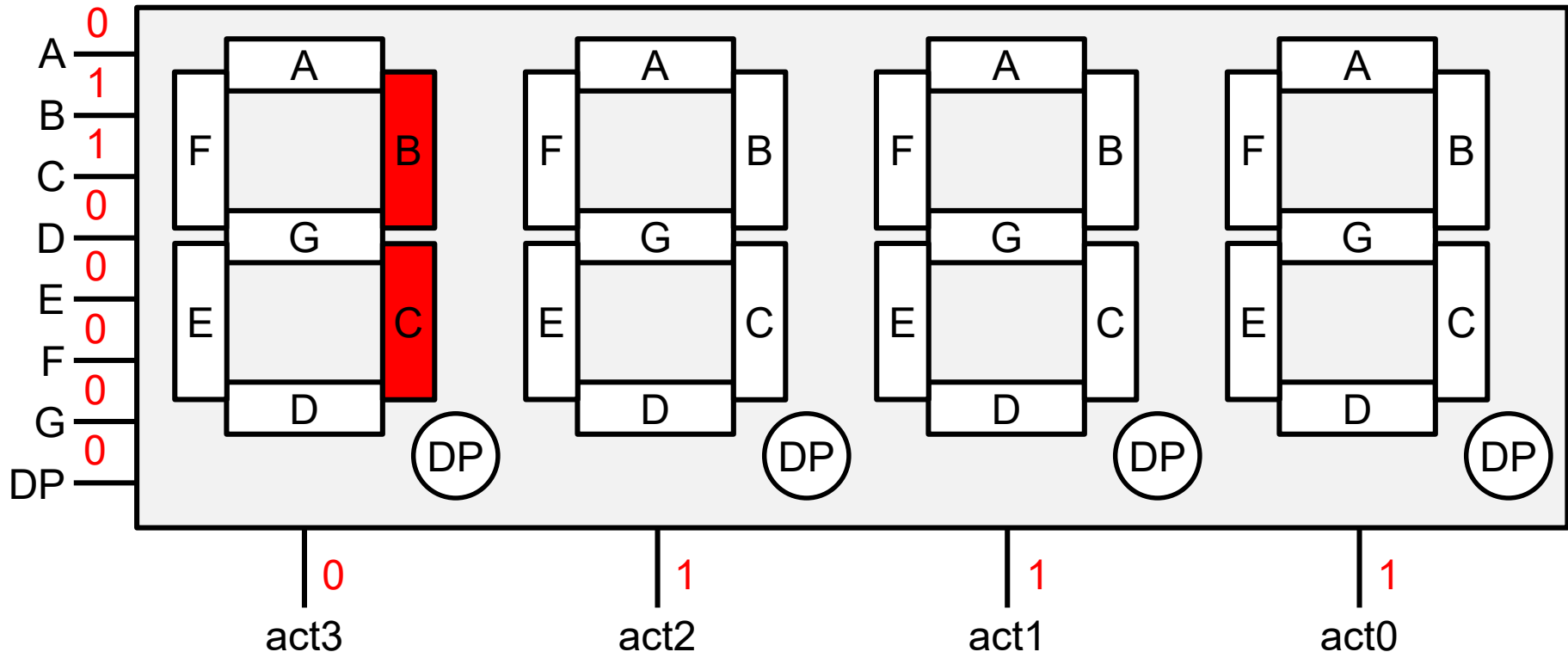
FND (Flexible Numeric Display)

- FND 제어 시 핀 개수를 줄이는 방법(CC타입 사용)



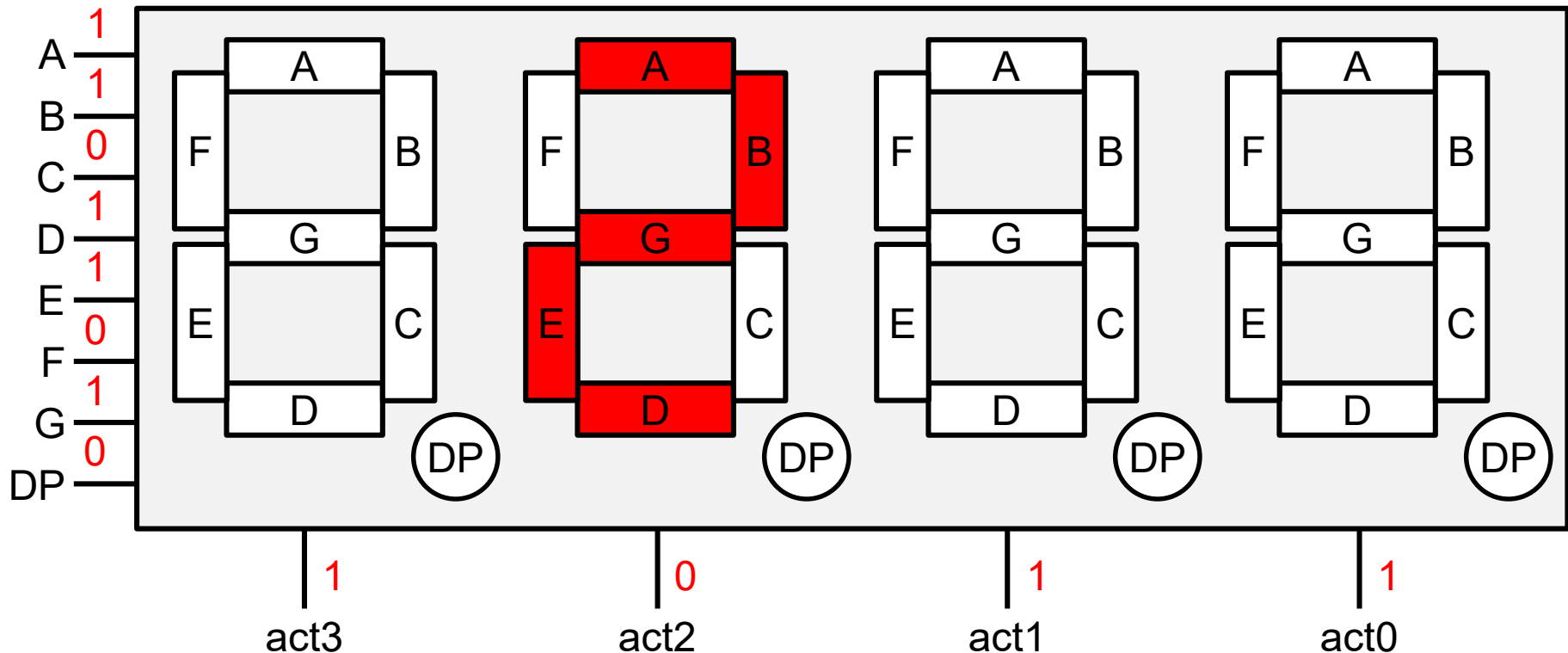
FND (Flexible Numeric Display)

- FND 제어 시 핀 개수를 줄이는 방법(CC타입 사용)



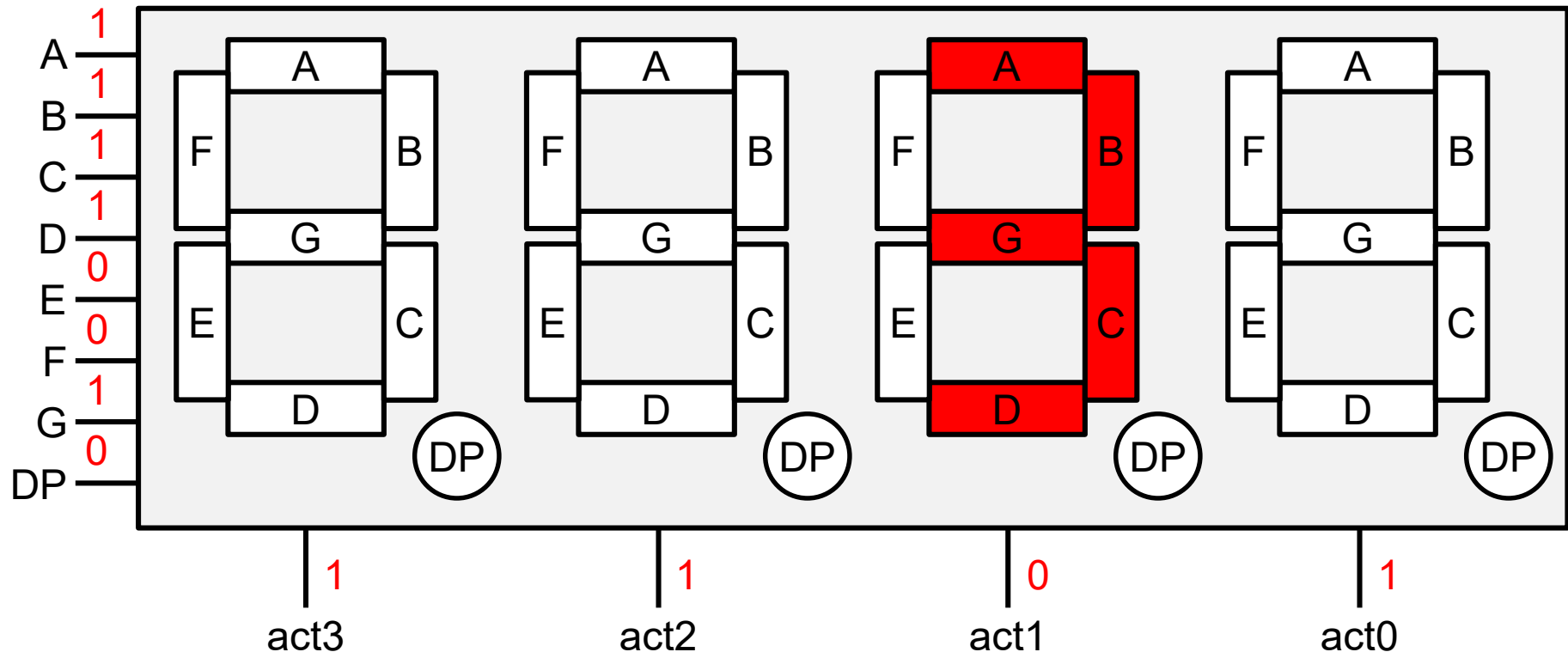
FND (Flexible Numeric Display)

- FND 제어 시 핀 개수를 줄이는 방법(CC타입 사용)



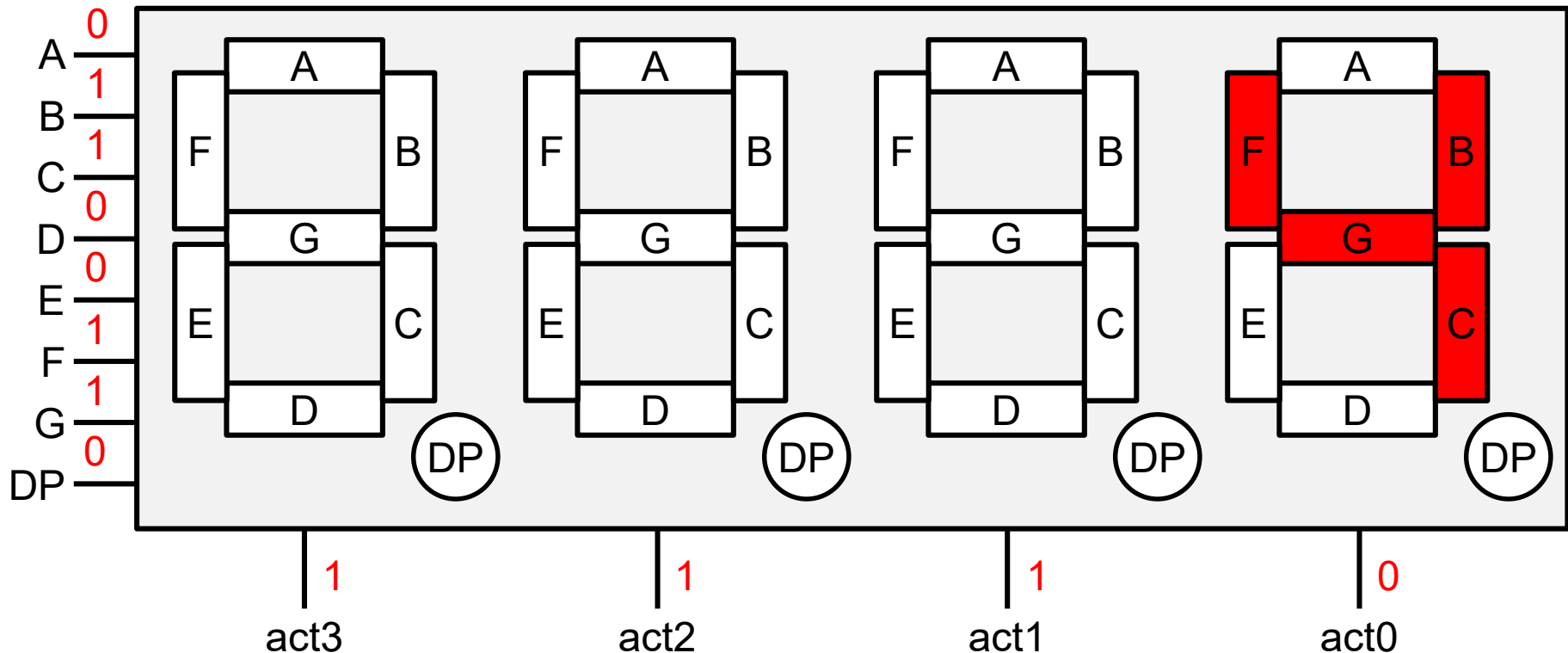
FND (Flexible Numeric Display)

- FND 제어 시 핀 개수를 줄이는 방법(CC타입 사용)



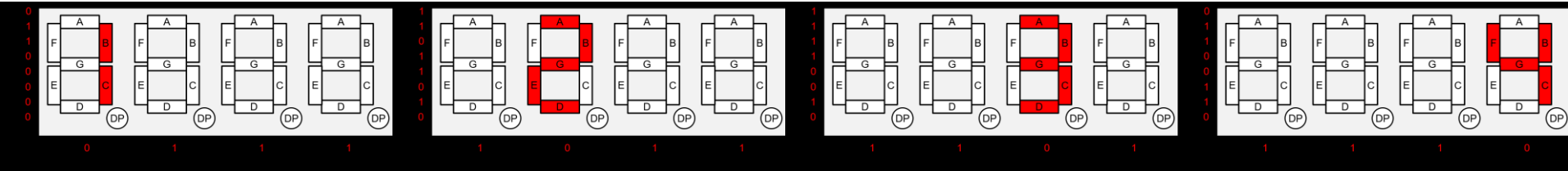
FND (Flexible Numeric Display)

- FND 제어 시 핀 개수를 줄이는 방법(CC타입 사용)



FND (Flexible Numeric Display)

- 활성화 신호 빠르게 변환시킴 → 동시에 켜진 것처럼 보임

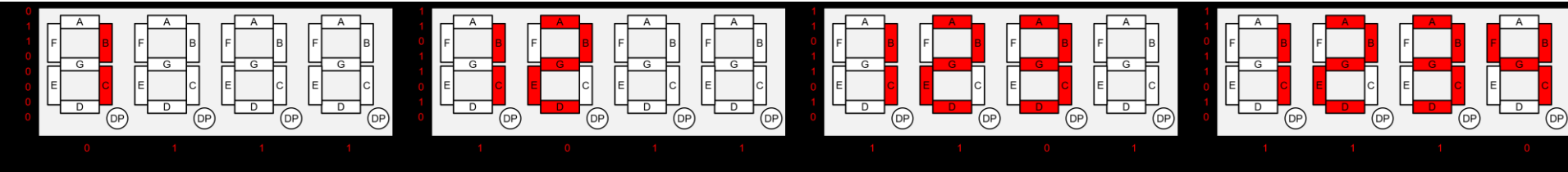


현실



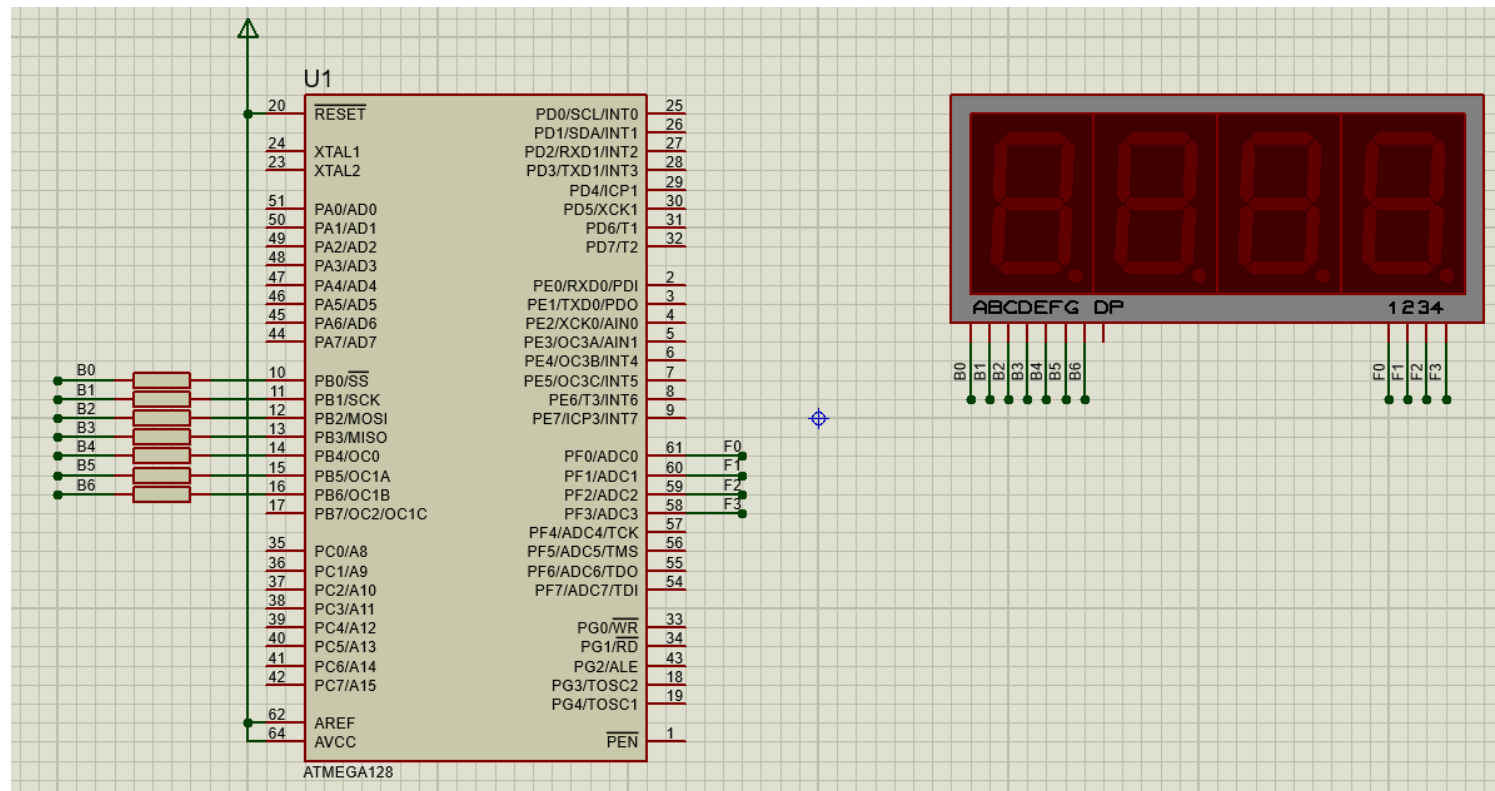
활성화 신호 바꾸는 속도 $\geq 50\text{Hz}$

Persistence of Vision (POV)



FND (Flexible Numeric Display)

■ FND 4개 실습



FND (Flexible Numeric Display)

■ FND 4개 실습

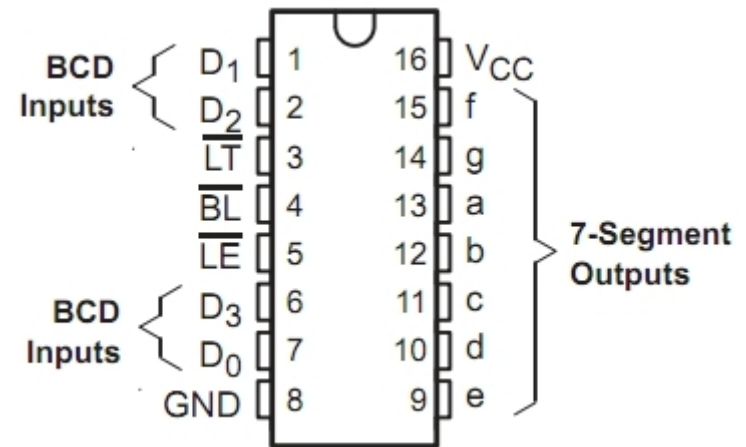
```
unsigned char CC[] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x27, 0x7f, 0x67, 0x77, 0x7c, 0x58, 0x5e, 0x79, 0x71, 0x80};
unsigned char act[4] = {0x0e, 0x0d, 0x0b, 0x07};

int main(void) {
    DDRB = DDRF = 0xff;
    int i, j=0, r=0;
    while(1) {
        for (i=0; i<4; i++) {
            PORTF = act[i];
            if (i==0) PORTB = CC[j/1000];
            if (i==1) PORTB = CC[(j%1000)/100];
            if (i==2) PORTB = CC[(j%100)/10];
            if (i==3) PORTB = CC[j%10];
            _delay_ms(1);
        }
        r++;
        if (r==20) { // update j every 20*20 = 400ms
            r=0;
            j++;
        }
        if (j>9999) j=0;
        _delay_ms(16); // total delay time is 1*4 + 16 = 20ms
    }
}
```

FND (Flexible Numeric Display)

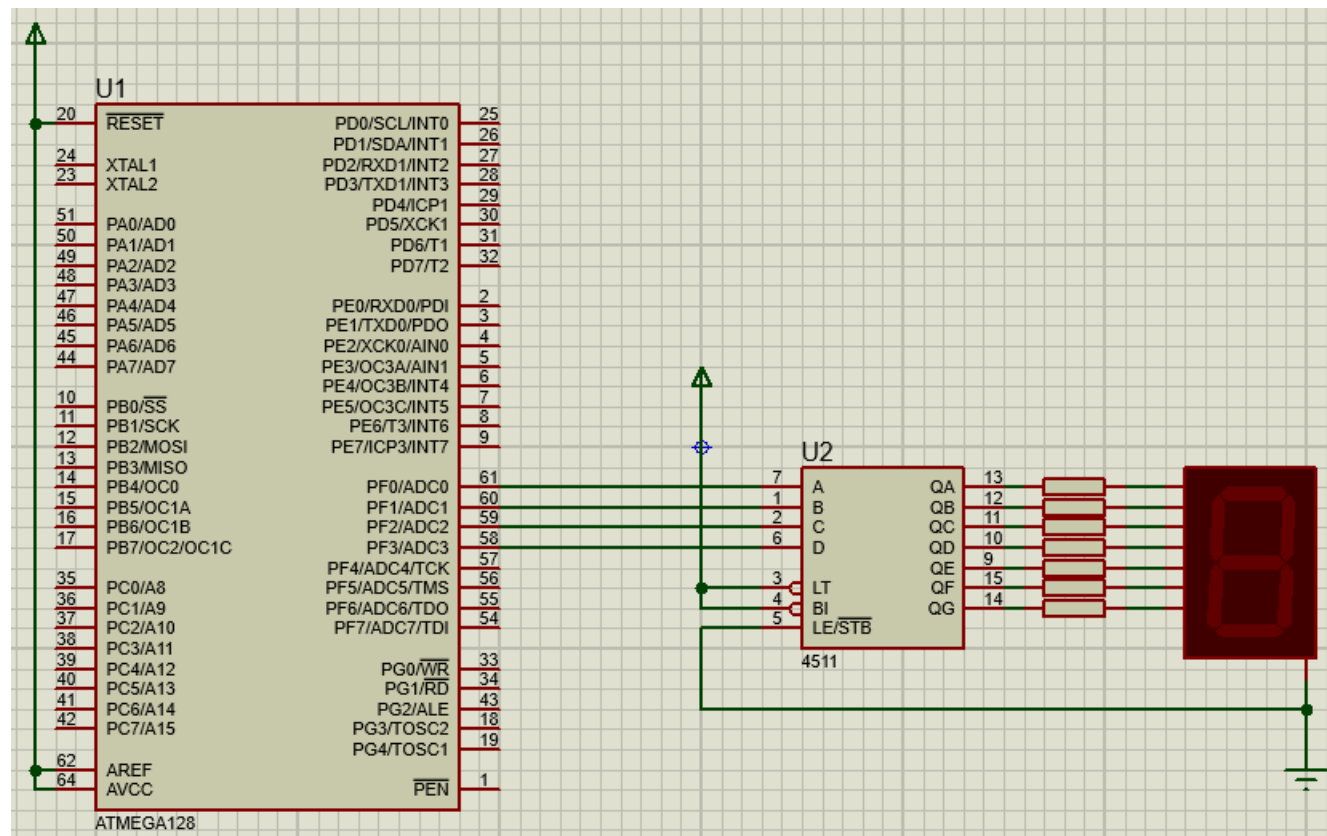
■ CD4511: BCD-to-7-Segment Decoder

숫자	Active high (또는 CC)								16진수
	DP	G	F	E	D	C	B	A	
0	0	0	1	1	1	1	1	1	0x3f
1	0	0	0	0	0	1	1	0	0x06
2	0	1	0	1	1	0	1	1	0x5b
3	0	1	0	0	1	1	1	1	0x4f
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6d
6	0	1	1	1	1	1	0	1	0x7d
7	0	0	1	0	0	1	1	1	0x27
8	0	1	1	1	1	1	1	1	0x7f
9	0	1	1	0	0	1	1	1	0x67



FND (Flexible Numeric Display)

- CD4511: BCD-to-7-Segment Decoder



FND (Flexible Numeric Display)

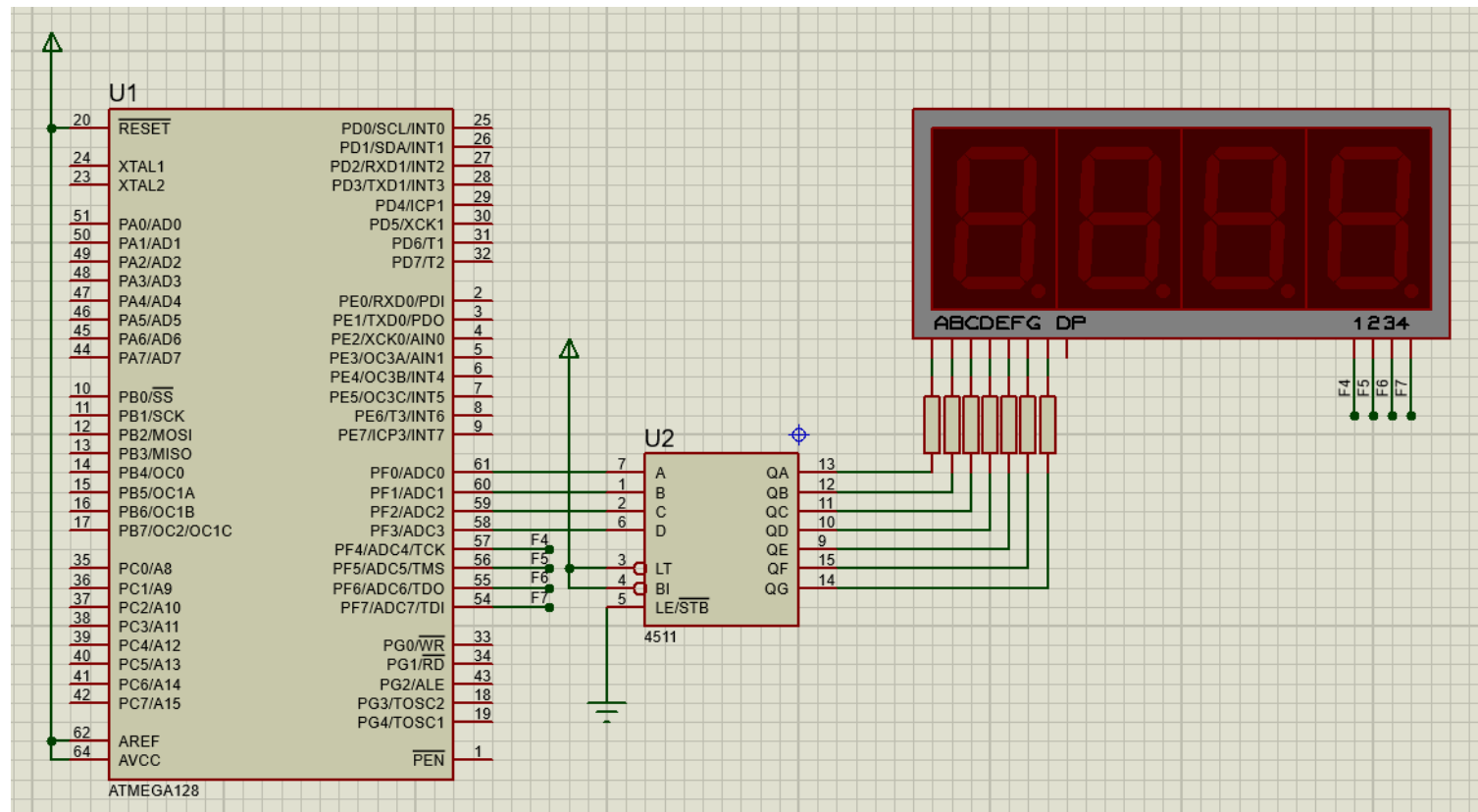


- CD4511: BCD-to-7-Segment Decoder

```
int main(void) {  
    int i = 0;  
    DDRF = 0xff;  
    while(1) {  
        PORTF = i;  
        _delay_ms(500);  
        i++;  
        if (i>9) i = 0;  
    }  
}
```

FND (Flexible Numeric Display)

■ CD4511: BCD-to-7-Segment Decoder



FND (Flexible Numeric Display)

■ CD4511: BCD-to-7-Segment Decoder

```
unsigned char act[4] = {0xe0, 0xd0, 0xb0, 0x70};

int main(void) {
    DDRF = 0xff;
    int i, j=0, r=0;
    while(1) {
        for (i=0; i<4; i++) {
            if (i==0) PORTF = (j/1000)|act[i];
            if (i==1) PORTF = ((j%1000)/100)|act[i];
            if (i==2) PORTF = ((j%100)/10)|act[i];
            if (i==3) PORTF = (j%10)|act[i];
            _delay_ms(1);
        }
        r++;
        if (r==20) { // update j every 20*20 = 400ms
            r=0;
            j++;
        }
        if (j>9999) j=0;
        _delay_ms(16); // total delay time is 1*4 + 16 = 20ms
    }
}
```