

Week 10 – 7-Segment, Keypad, and Dot Matrix

Logic Circuit Design Laboratory

Yoonmyung Lee

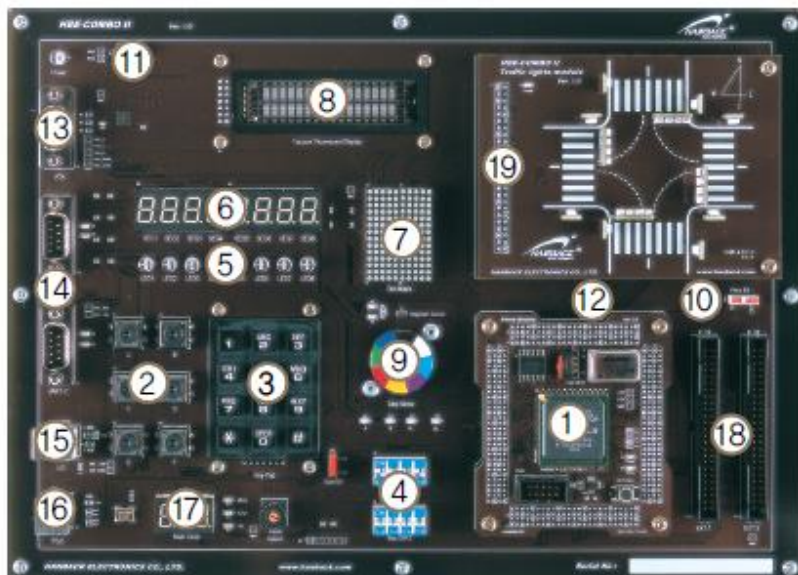
yoonyung@skku.edu

Dept. of Semiconductor Systems Engineering

Sungkyunkwan University

HBE-COMBO II

구성 및 명칭



- | | |
|------------------------------------|-------------------------|
| 1. FPGA 모듈 | 11. IrDA |
| 2. Button S/W | 12. Piezo |
| 3. KEYPAD | 13. VGA Port |
| 4. Dip S/W | 14. UART Port |
| 5. High Brightness LED | 15. USB to Serial Port |
| 6. 7-Segment | 16. PS/2 Port |
| 7. DOT Matrix | 17. Clock Control Block |
| 8. VFD(Vacuum Fluorescent Display) | 18. Expansion Port 1, 2 |
| 9. Step Motor | 19. Expansion Board |
| 10. SRAM | |

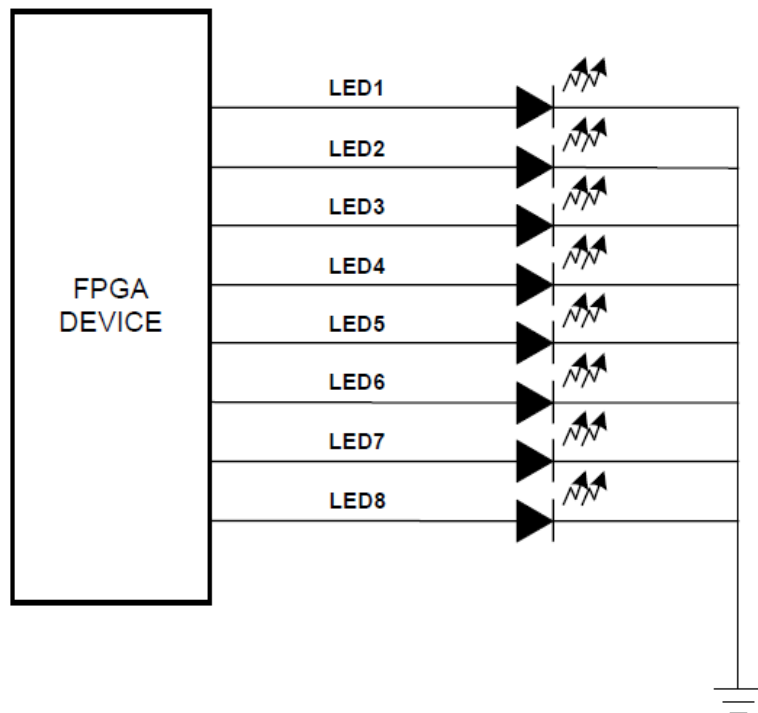
CLOCK

- 장비를 구동하는 clock은 Base board에 있는 50 MHz를 기본적으로 사용
- Clock Control Block의 7-segment에 현재 FPGA 디바이스 모듈로 공급되는 클럭 값이 표시되고 오른 쪽에 있는 LED에 현재 주파수 대역이 표시됨.
- Clock SW를 조절하여 0 Hz ~ 50 MHz 의 분주된 클럭의 값을 FPGA 디바이스 모듈로 전달



Clock SW	클럭 입력	Clock SW	클럭 입력	Clock SW	클럭 입력	Clock SW	클럭 입력
0	0Hz	4	100Hz	8	10kHz	C	1MHz
1	1Hz	5	500Hz	9	50kHz	D	5MHz
2	10Hz	6	1kHz	A	100kHz	E	25MHz
3	50Hz	7	5kHz	B	500kHz	F	50MHz

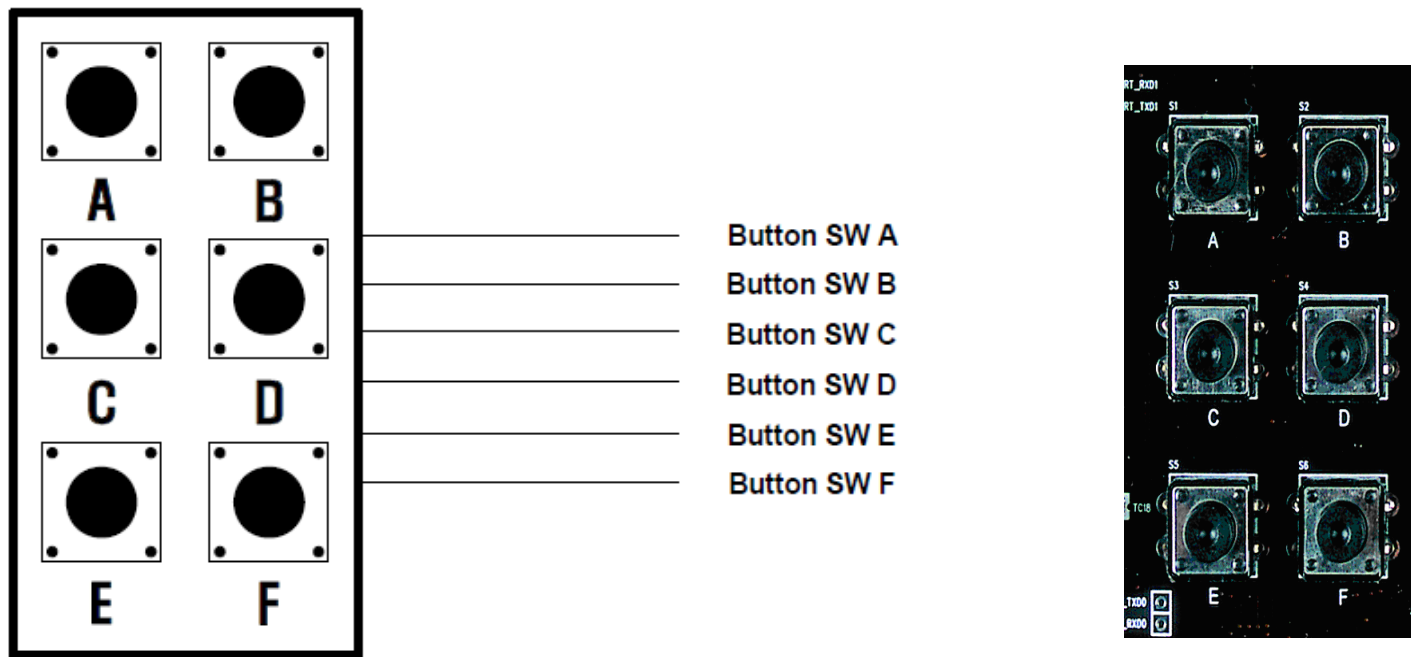
High Brightness LED



- LED는 회로도 에서 보는 것과 같이 8개의 LED가 각각의 FPGA의 I/O 핀에 연결 됨
- FPGA에 연결된 부분에 '1' 혹은 '0' 신호를 인가하여 LED를 제어 할 수 있도록 구성



BUTTON SWITCH

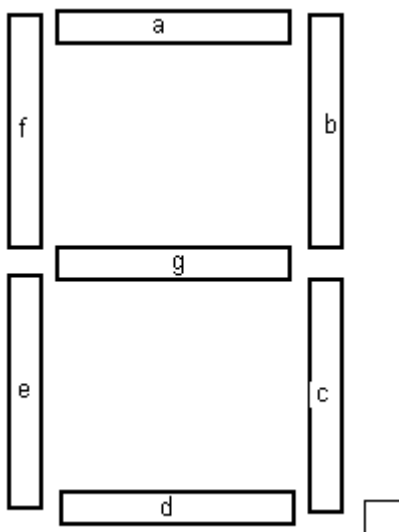


- 버튼 스위치마다 별개의 I/O로 직접 연결이 되어 있으므로 Keypad 보다 쉽게 제어 가능
- 버튼 스위치는 키가 눌러 질 때 발생하는 채터링 현상을 방지하기 위한 회로가 내장되어 정확한 입력이 가능하도록 구성

* 채터링 : 계전기(릴레이)의 접점이 닫힐 때 한 번에 닫히지 않고 여러 번 단속(斷續)을 반복하는 것

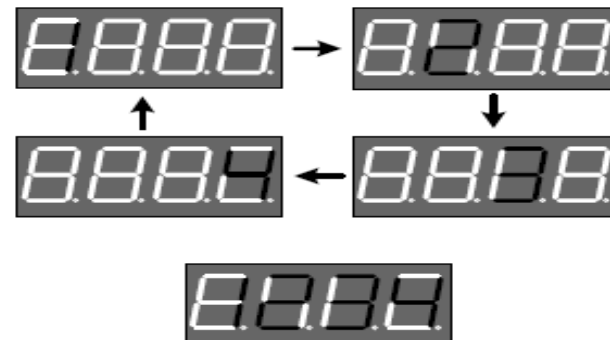
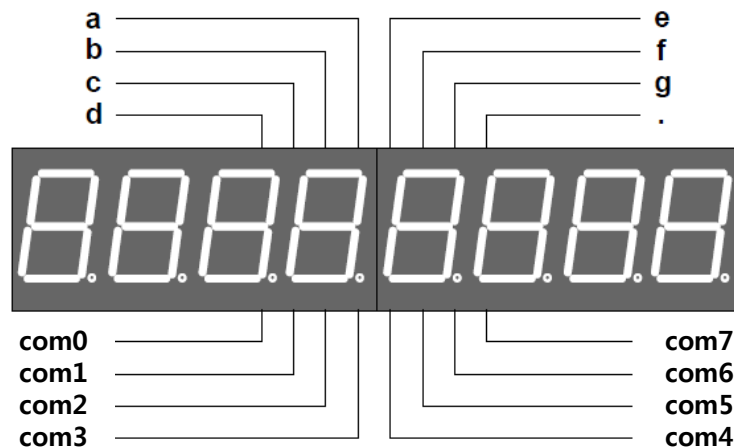
7-Segment

- 7-segment : 숫자를 표시하기 위해 만들어진 소자로 아래와 같은 모양을 가짐



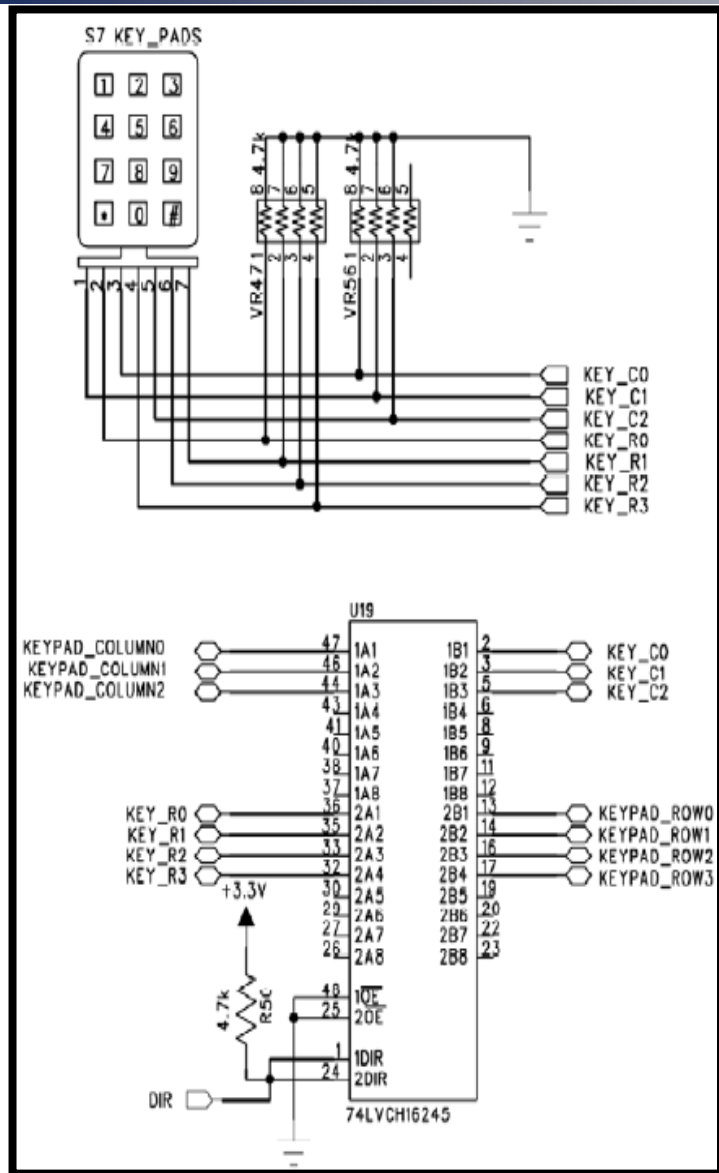
16진수	A	B	C	D	E	F	G	dg
0	1	1	1	1	1	1	0	0
1	0	1	1	0	0	0	0	0
2	1	1	0	1	1	0	1	0
3	1	1	1	1	0	0	1	0
4	0	1	1	0	0	1	1	0
5	1	0	1	1	0	1	1	0
6	0	0	1	1	1	1	1	0
7	1	1	1	0	0	0	0	0
8	1	1	1	1	1	1	1	0
9	1	1	1	0	0	1	1	0
A	1	1	1	0	1	1	1	0
B	0	0	1	1	1	1	1	0
C	1	0	0	1	1	1	0	0
D	0	1	1	1	1	0	1	0
E	1	0	0	1	1	1	1	0
F	1	0	0	0	1	1	1	0

7-Segment



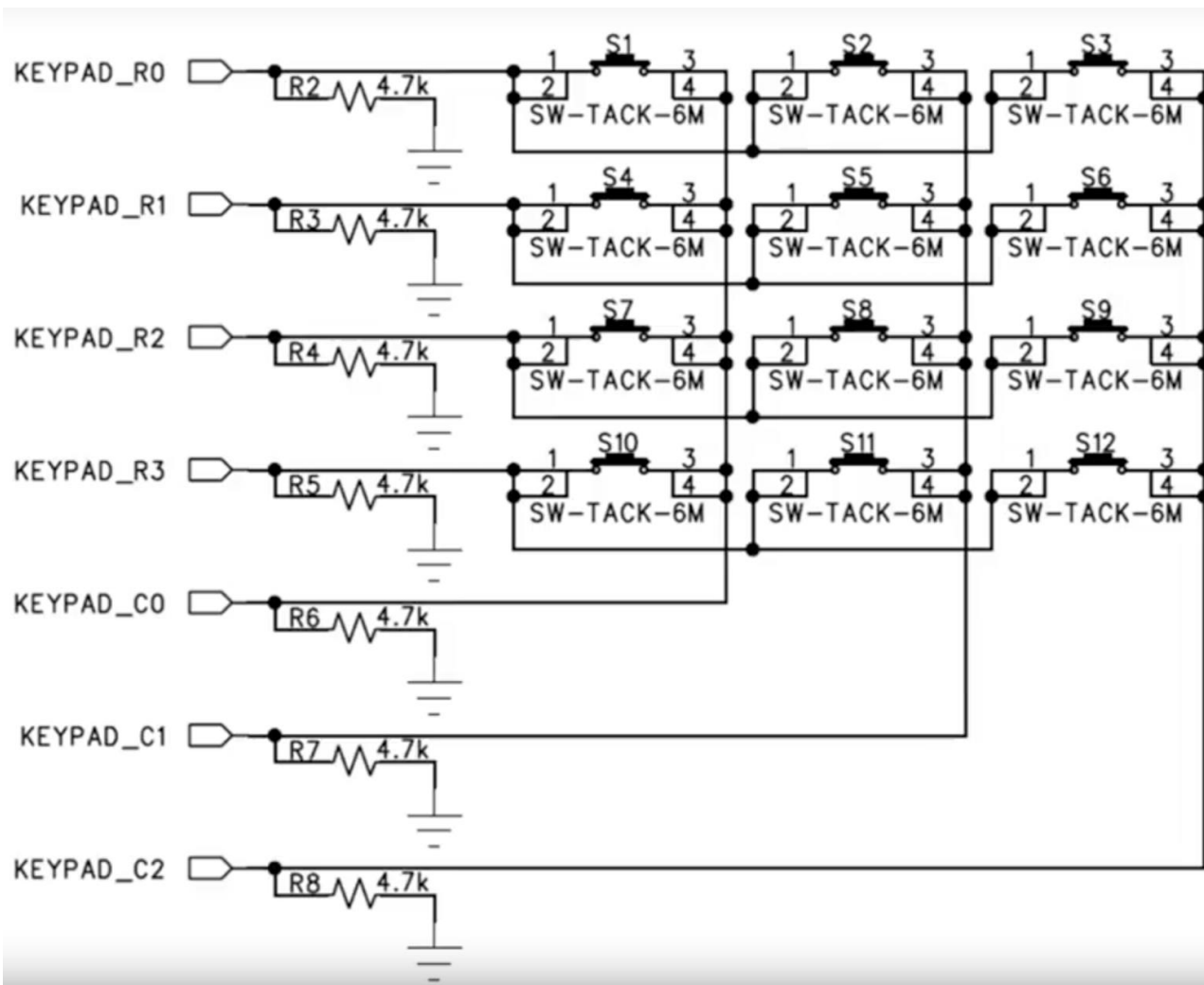
- 16개의 핀을 가지고 8개의 7-Segment를 제어
- A~G + dg pin 은 16진수의 숫자 중 하나를 표현
- 정해진 16진수의 숫자를 출력할 자리를 지정하기 위해 Segment common(com0~7) 핀을 선택(com0 : MSB, com7 : LSB)
- 8개의 digit 중 display될 특정 digit을 com0~7 을 이용하여 지정, A~G + dg 로 data 를 인가하여 Display
- Segment common 핀을 여러 개 선택하여 동시에 출력하는 것도 가능
- 잔상효과를 주는 방법
 - 7-segment 데이터 값을 "01100000"로 주어 '1'을 표시하는 값을 주고 com0에 '0'을 그리고 나머지 com1~3는 '1'의 값을 주면 첫째 7-Segment에 '1'이 표시
 - 데이터에 "11011010"을 주어 '2'의 값을 주고 com1에 '0'과 com0, com2~3에 '1'을 주면 둘째 7-Segment에 2가 표시
 - 4까지의 숫자를 표시하고 다시 처음으로 돌아가 위의 내용을 반복
 - 약 1ms 이상의 주기로 반복하면 잔상효과에 의해 "1234"의 숫자가 모두 켜져 있는 것처럼 표시됨

KEYPAD



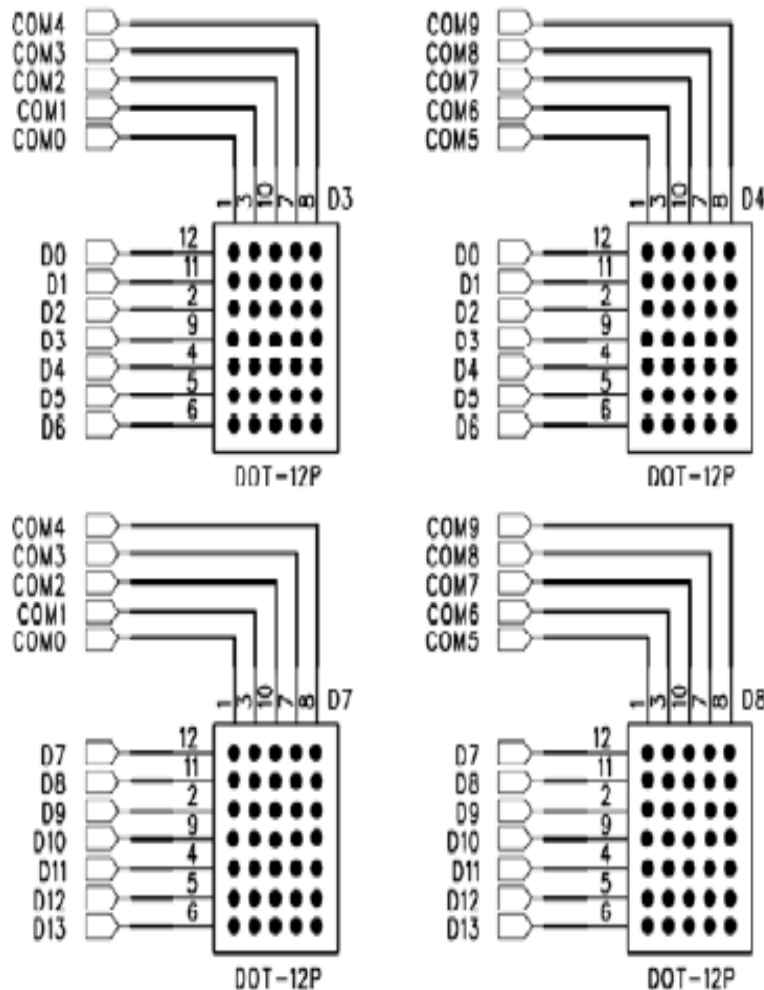
- 4x3의 12개의 키를 입력받아서 사용할 수 있도록 구성
- Column을 스캔할 때는 Row의 값을 읽어들이고 Row를 스캔할 때는 Column의 값을 읽어들이는다.
- Keypad의 스캔 방향은 Scan Dir의 헤더핀을 이용하여 바꿀 수 있음.
- 현재 예비보고서는 Keypad의 Column 을 스캔하는 코드임.
- Column의 1의 값을 shift 시키면서 1이 되는 Row를 감지하여 어떤 Keypad가 눌렸는지 판단

KEYPAD



- (Column을 스캔하는 경우)
 - KEYPAD_C#에 값 할당 (100, 010, 001)
 - 키패드의 눌린 스위치 (S1~S12)에 따라 KEYPAD_R# 값 정해짐
 - 1의 값을 갖는 COL과 ROW로 키패드숫자 판별

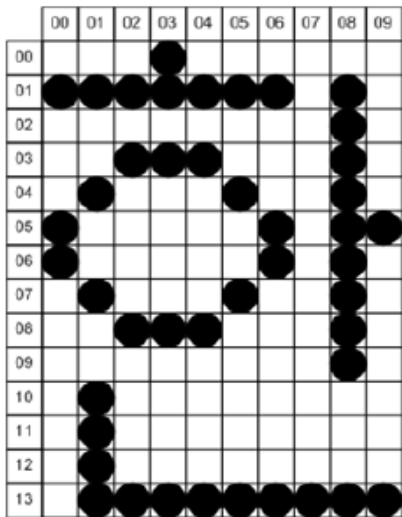
DOT MATRIX



< 10x14 Dot-matrix (5x7 Dot-matrix 4ea) >

- dot_d0~dot_d13 해당 열에 표시될 데이터 입력.
(데이터 값은 '1'의 값을 가질 경우에 켜짐)
- dot_scan0~dot_scan9는 dot_d0~dot_d13의 데이터를 표시할 열의 위치를 지정하는 입력.
('1'의 값을 가질 때 해당 열이 선택)
ex) 1행 1열의 dot LED를 켜기 위해서는 dot_d0의 값을 '1'로 입력하고 dot_scan0의 값을 '1'으로 할 때 켜짐.
- dot의 scan라인을 선택하고, 해당되는 라인에 data값을 주어서 dot를 display함.
- 1ms이상의 주기로 반복해서 display하게 되면 눈의 잔상효과로 인해서 모든 dot가 display 되어 있는 것처럼 느끼게 됨.
- 7-Segment의 Scan 방식과 비슷한 구조를 가짐.

DOT MATRIX



구분	Dot_d13~dot_d0	Dot_scan9~dot_scan0
0열	"00000001100010"	"0000000001"
1열	"11110010010010"	"0000000010"
2열	"10000100001010"	"0000000100"
3열	"10000100001011"	"0000001000"
4열	"10000100001010"	"0000010000"
5열	"10000010010010"	"0000100000"
6열	"10000001100010"	"0001000000"
7열	"10000000000000"	"0010000000"
8열	"10001111111110"	"0100000000"
9열	"10000000100000"	"1000000000"

- dot_d13 ~ dot_d0 "00000001100010"인 첫째 열의 데이터를 입력하고, dot_scan9~dot_scan0 "0000000001"의 값을 입력하면 첫째 열의 원하는 dot LED가 켜짐.
- dot_d13 ~ dot_d0 "11110010010010"인 둘째 열의 데이터를 입력하고, dot_scan9~dot_scan0 "0000000010"의 값을 입력하면 둘째 열의 원하는 dot LED가 켜짐.
- 열 번째 열까지 반복하며, 반복되는 주기를 약 1ms 이하로 해서 반복하게 되면 눈의 잔상효과에 의해서 '한'이라는 글자가 dot matrix LED 모듈에 표시.

실습 내용

- 1.예비 보고서 실습
 - 1.1) 7-segments를 이용한 디지털 시계 구현 : watch.v
 - 1.2) button과 7-segment를 이용하여 keypad 동작 확인 : keypad.v
 - 1.3) dot matrix를 이용하여 파형 출력 : dot_matrix.v
- 2.간단한 덧셈(+), 뺄셈(-) 계산기 설계하기

계산기 설계

- 1) 7-segment에 {부호, 10의 자릿수, 1의 자릿수}를 출력
 - Ex) +02, -97, +00등을 7-segment에 표시.
- 2) keypad 로 연산을 위한 한 자리 숫자 입력
- 3-1) Button Switch A → RESET : 계산 초기화 (initial state)
 - 7-segment에 '+00'표시
- 3-2) Button Switch B → = : 계산 결과를 7-segment에 표시
- 3-3) Button Switch C → 더하기 (+) : 덧셈 연산 수행
- 3-4) Button Switch D → 빼기 (-) : 뺄셈 연산 수행
 - Dot matrix : 더하기(+), 빼기(-) 연산 기호 입력 시 dot matrix 에 표시.
- 2자리까지 연산 가능하도록 구현할 것.
- 연속 연산이 가능해야 함 (Ex. $7-1-3-5 = -2$)

INPUT/OUTPUT

- Input
 - clock, button switch, keypad
- Output
 - LED, 7-segment, dot-matrix