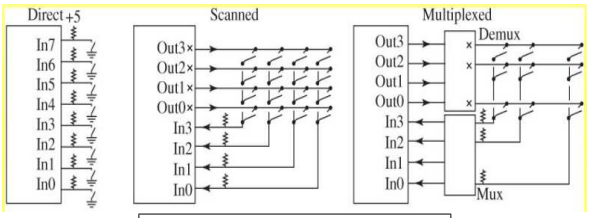
ECE/CS 5780/6780: Embedded System Design

Chris J. Myers

Lecture 16: Keyboards and Displays

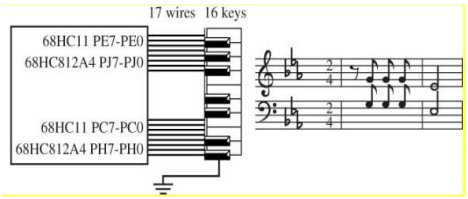
Basic Approaches to Interfacing Multiple Keys

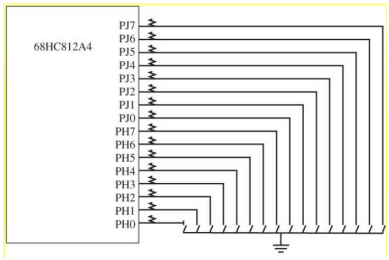


Row	Out3	Out2	Out1	Out0
3	0	HiZ	HiZ	HiZ
2	HiZ	0	HiZ	HiZ
1	HiZ	HiZ	0	HiZ
0	HiZ	HiZ	HiZ	0

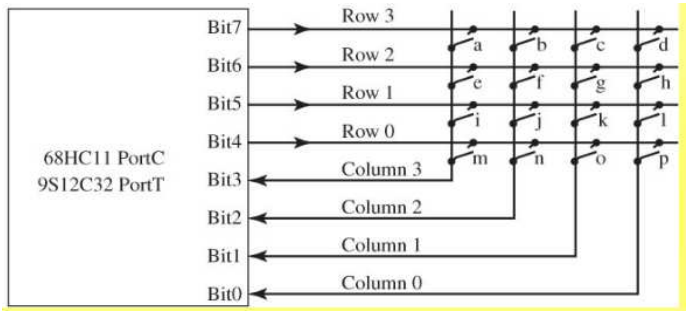
Row	Out3	Out2	Out1	Out0	15	14	...	0
15	1	1	1	1	0	HiZ	...	HiZ
14	1	1	1	0	HiZ	0	...	HiZ
...	...	...	...	...	...	...	...	...
0	0	0	0	0	HiZ	HiZ	...	0

Sixteen-Key Electronic Piano

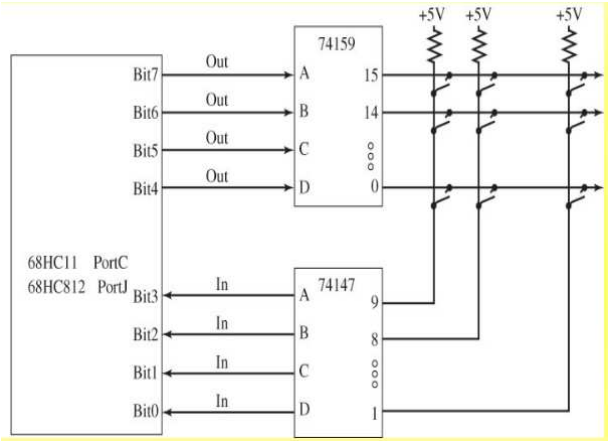




Hardware for Generating Interrupts



Multiplexed/Demultiplexed Scanned Keyboard



Software for Multiplexed Keyboard

```
unsigned char Key;          // current pattern
unsigned char PreviousKey;  // 10 ms ago
#define period 20000        // 10 ms
unsigned char KeyScan(void){
    unsigned char key,row;
    key=0;                  // means no key pressed
    for(row=0;row<16;row++){
          PORTJ=row<<4;     // Select row
          if((PORTJ&0x0F)!=0x0F){
               key=PORTJ^0x0F; }}
    return(key);}
```

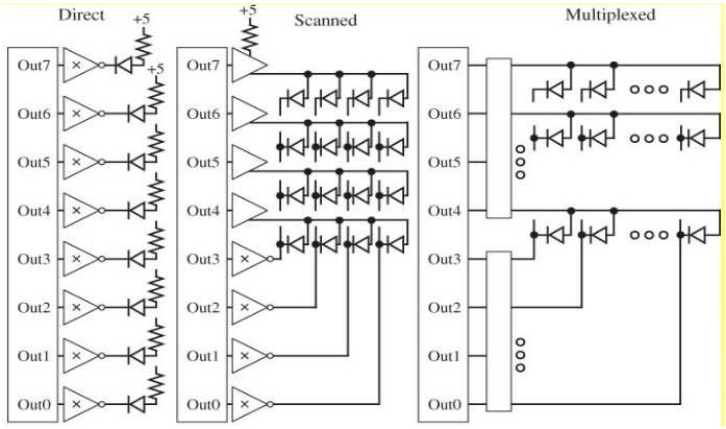
Software for Multiplexed Keyboard (cont)

```
void Ritual(void){
     asm(" sei");               // make atomic
     DDRJ=0xF0;
     PreviousKey=Key=KeyScan(); // read
     TMSK1|=0x20;               // Arm OC5
     TIOS|=OC5;                 // enable OC5
     TSCR|=0x80;                // enable
     TMSK2=0x32;                // 500 ns clock
     TC5=TCNT+wait;
     TFLG1=0x20;                // clear OC5F
     asm(" cli"); }
```

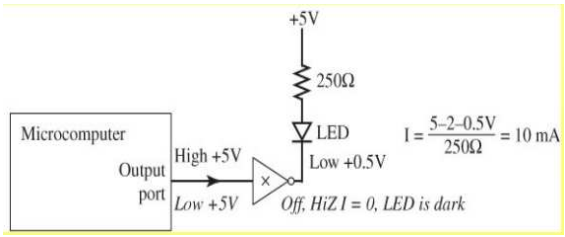

Software for Multiplexed Keyboard (cont)

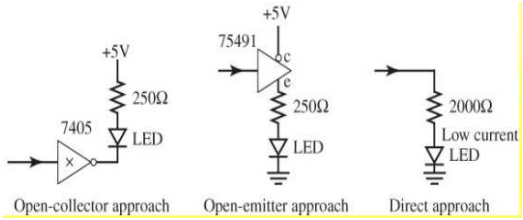
```
#pragma interrupt_handler TOC5handler()
void TOC5handler(void){
unsigned char NewKey;
     NewKey=KeyScan();  // Current pattern
     if(NewKey==PreviousKey) Key=NewKey;
     PreviousKey=NewKey;
     TOC5=TOC5+period;
     TFLG1=0x20;}       // ack OC5F
```


Interfacing Multiple LEDs

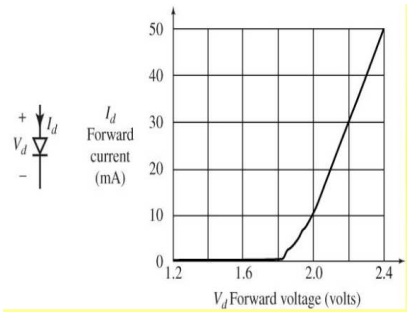


Single LED Interface



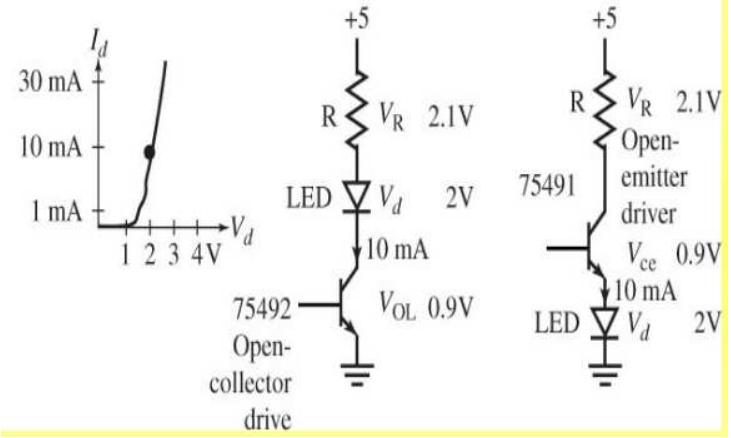


Typical Voltage/Current Response of a LED

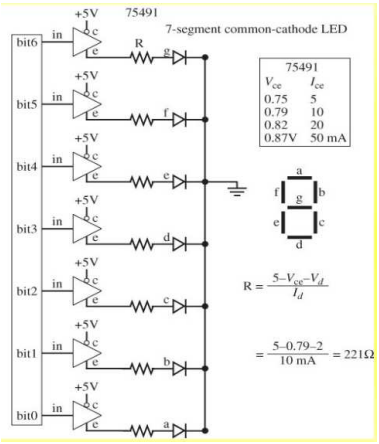


Parameter	red	green	yellow	orange	units
Max power	55	75	60	75	mW
Peak current	160	100	80	100	mA
Max current	25	25	20	25	mA

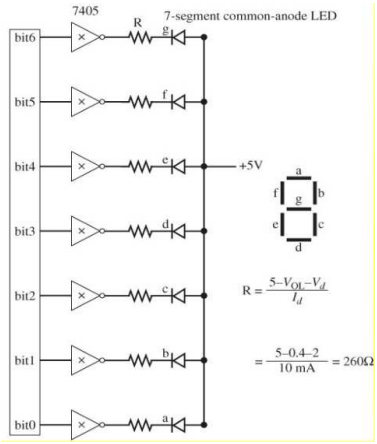
Calculating the Resistor Value



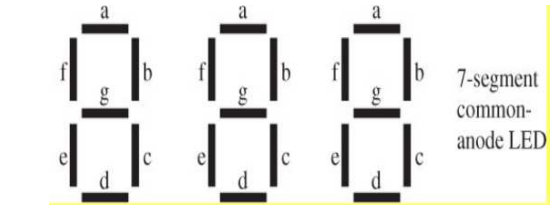
Seven-Segment LED Interfaces (Common-Cathode)

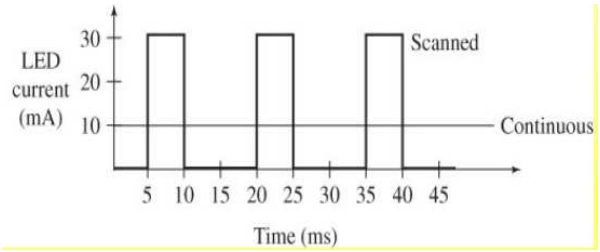


Seven-Segment LED Interfaces (Common-Anode)

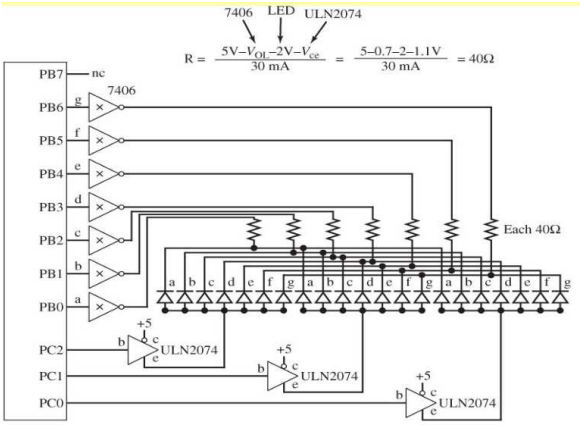


Scanned Seven-Segment LED Interface





Circuit Used to Scan a LED Interface



For MC9S12C32, replace PB with PT and PC with PM.

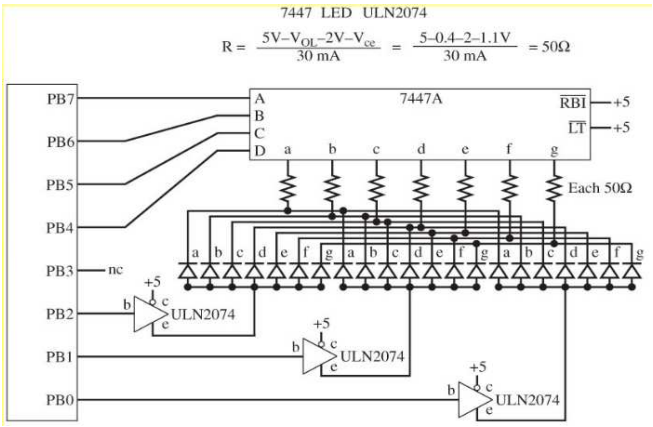
Software for Scanned LED Display

```
// PT7-PT0 output, 7 bit pattern
// PM2-PM0 output, selects LED digit
unsigned char code[3]; // binary codes
static unsigned char select[3]={4,2,1};
unsigned short index;  // 0,1,2
void LED_Init(void) {
asm sei          // make atomic
  index = 0;
  DDRT = 0xFF;   // outputs 7 segment code
  DDRM |= 0x03;  // outputs select LED
  TIE |=0x20;     // Arm OC5
  TIOS |=0x20;    // enable OC5
  TSCR1 =0x80;    // enable
  TSCR2 =0x01;    // 500 ns clock
  TC5 = TCNT+10000;
asm cli }
```


Software for Scanned LED Display

```
void interrupt 13 TC5handler(void){
  TFLG1 = 0x20;        // Acknowledge
  TC5 = TC5+10000;     // every 5 ms
  PTM = select[index]; // which LED?
  PTT = code[index];   // enable
  if(++index==3) index=0;}
asm(" cli"); }
```

Scanned LED Interface Using Decoder



For MC9S12C32, replace PB with PT.

Software for Multiplexed LED Display

```
unsigned short Global; // 12-bit packed BCD
const struct LED
{ unsigned char enable; // select
  unsigned char shift;  // bits to shift
  const struct LED *Next; };  // Link
typedef const struct LED LEDType;
typedef LEDType * LEDPtr;
LEDType LEDTab[3]={
{ 0x04, 8, &LEDTab[1] },  // Most sig
{ 0x02, 4, &LEDTab[2] },
{ 0x01, 0, &LEDTab[0] }}; // least sig
LEDPtr Pt;   // Points to current digit
```


Software for Multiplexed LED Display (cont)

```
void LED_Init(void) {
asm sei         // make atomic
  DDRT = 0xFF;  // outputs to LED's
  Global = 0;
  Pt=&LEDTab[0];
  TIE |= 0x20;     // Arm OC5
  TIOS |= 0x20;    // enable OC5
  TSCR1 = 0x80;    // enable
  TSCR2 = 0x01;    // 500 ns clock
  TC5 = TCNT+10000;
asm cli }
```

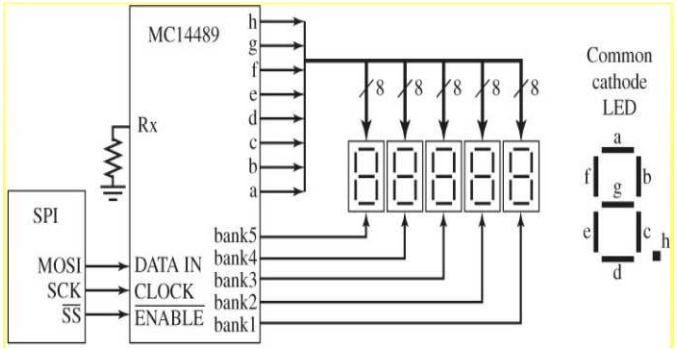

Software for Multiplexed LED Display (cont)

```
void interrupt 13 TC5handler(void){
  TFLG1 = 0x20;      // Acknowledge
  TC5 = TC5+10000;  // every 5 ms
  PTT = (Pt->enable)
      +(Global>>(pt->shift))<<4);
  Pt = Pt->Next; }
```

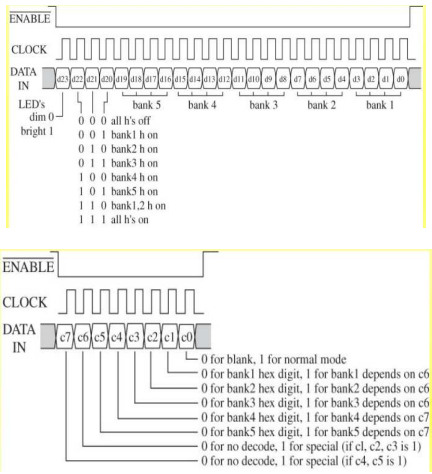

Extensions to Multiple Digits

- Two issues to consider as number of digits is increased:
  1. Scan frequency - for display to "look" continuous, each digit must be updated faster than 60 Hz.
     - interrupt rate = 60 Hz × #digits
  2. Duty cycle - this decreases as digits added, so must increase instantaneous current.
     - instantaneous current = desired current × #digits
- Ratio of maximum instantaneous current to desired LED current determines maximum number of digits.

Integrated IC Interface for LED Digits



Data Timing of Integrated LED Controller



Software for Integrated LED Display

```
// PM4/MOSI = MC14489 DATA IN
// PM5/SCLK = MC14489 CLOCK IN
// PM3 (simple output) = MC14489 ENABLE
void LED_Init(void) {
  DDRM |= 0x38;  // outputs to MC14489
  SPICR1 = 0x50;
  SPICR2 = 0x00; // regular drive
  SPIBR = 0x01;  // 1MHz SCLK
  PTM |= 0x08;   // ENABLE=1
  PTM &=~0x08;   // ENABLE=0
  SPIDR= 0x01;   // hex format
  while(SPISR&0x80)==0){};
  PTM |=0x08;}   // ENABLE=1
```

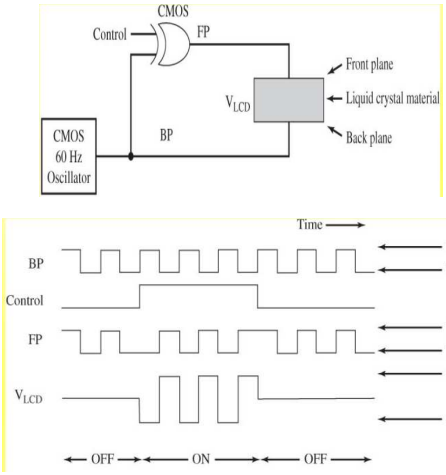

Software for Integrated LED Display

```
void LED_out(unsigned char data[3]){
unsigned char dummy;
  PTM &=~0x08;      // ENABLE=0
  while((SPISR&SPTEF)==0); // wait for transmit empty
  SPIDR = data[2];  // send MSbyte
  dummy = SPIDR;    // clear SPIF
  while((SPISR&SPTEF)==0); // wait for transmit empty
  SPIDR = data[1];  // send middle byte
  dummy = SPIDR;    // clear SPIF
  while((SPISR&SPTEF)==0); // wait for transmit empty
  SPIDR = data[0];  // send LSbyte
  dummy = SPIDR;    // clear SPIF
  Timer_Wait(10);   // wait for SPI output completion
  PTM |=0x08;}      // ENABLE=1
```

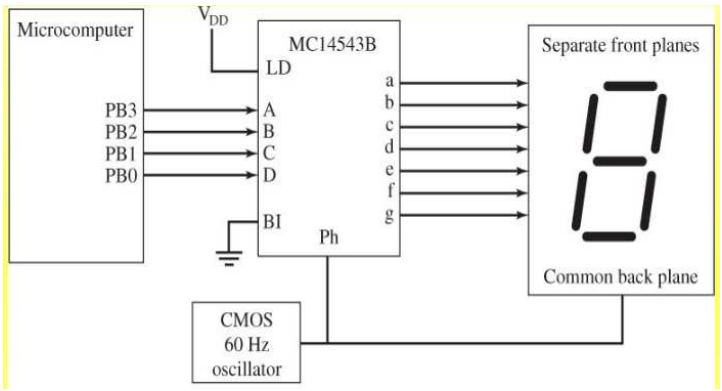

LCD Fundamentals

- LCD display consume less power than LED displays.
- LCDs are more flexible in sizes and shapes, allowing for combination of numbers, letters, words, and graphics.
- Uses liquid-crystal material that behaves as a capacitor.
- While LED converts electric power to emitted optical power, LCD uses AC voltage to change light reflectivity.
- Light energy is supplied by room or separate back light.
- Display controlled by altering reflectivity of each segment.
- Disadvantage is that they have slow response time, but typically fast enough for human perception.

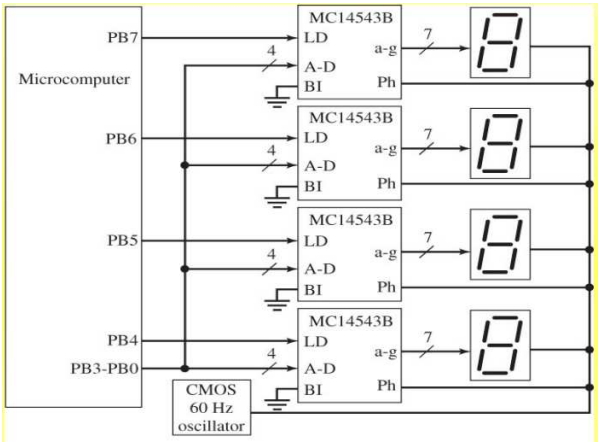
Basic Idea of a Liquid Crystal Interface



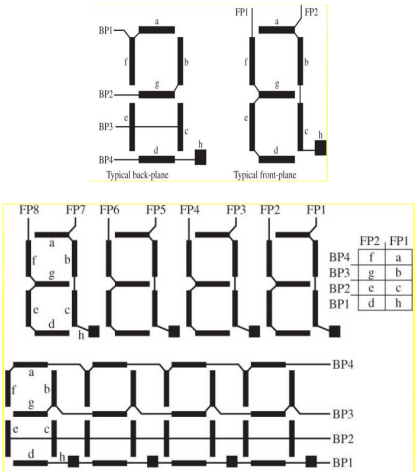
Direct Interface of a LCD



Latched Interface of a LCD



Artwork for 8-Segment LCD Digits



Interface of a 48-Segment LCD Display