Today

- Interfacing with LCD
The basic construction of an LCD is illustrated in Figure 7.34.

The most common type of LCD allows the light to pass through when activated.

An LCD segment is activated when a low frequency bipolar signal in the range of 30 Hz to 1KHz is applied to it.

LCD can display characters and graphics.

LCDs are often sold in a module with LCDs and controller unit built in.

The Hitachi HD44780 is the most popular LCD controller being used today.
A HD44780-Based LCD Kit (1 of 3)

- Display capability: 4 x 20
- Uses the HD44780 as the controller as shown in Figure 7.35.
- Pins DB7~DB0 are used to exchange data with the CPU.
- E input should be connected to one of the address decoder output or I/O pin.
- The RS signal selects instruction register (0) or data register (1).
- The VEE signal allows the user to adjust the LCD contrast.
- The HD44780 can be configured to display 1-line, 2-line, and 4-line information.
- The pin assignment for character-based LCD module with less than and more than 80 characters are shown in Table 7.7 and 7.8.

Figure 7.35 Block diagram of a HD44780-based LCD kit
Table 7.7 Pin assignment for displays with less than 80 characters

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>symbol</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VSS</td>
<td>-</td>
<td>Power supply (GND)</td>
</tr>
<tr>
<td>2</td>
<td>VCC</td>
<td>-</td>
<td>Power supply (+5V)</td>
</tr>
<tr>
<td>3</td>
<td>VEE</td>
<td>-</td>
<td>Contrast adjust</td>
</tr>
<tr>
<td>4</td>
<td>RS</td>
<td>I</td>
<td>0 = instruction input, 1 = data input</td>
</tr>
<tr>
<td>5</td>
<td>R/W</td>
<td>I</td>
<td>0 = write to LCD, 1 = read from LCD</td>
</tr>
<tr>
<td>6</td>
<td>E</td>
<td>I</td>
<td>enable signal</td>
</tr>
<tr>
<td>7</td>
<td>DB0</td>
<td>I/O</td>
<td>data bus line 0</td>
</tr>
<tr>
<td>8</td>
<td>DB1</td>
<td>I/O</td>
<td>data bus line 1</td>
</tr>
<tr>
<td>9</td>
<td>DB2</td>
<td>I/O</td>
<td>data bus line 2</td>
</tr>
<tr>
<td>10</td>
<td>DB3</td>
<td>I/O</td>
<td>data bus line 3</td>
</tr>
<tr>
<td>11</td>
<td>DB4</td>
<td>I/O</td>
<td>data bus line 4</td>
</tr>
<tr>
<td>12</td>
<td>DB5</td>
<td>I/O</td>
<td>data bus line 5</td>
</tr>
<tr>
<td>13</td>
<td>DB6</td>
<td>I/O</td>
<td>data bus line 6</td>
</tr>
<tr>
<td>14</td>
<td>DB7</td>
<td>I/O</td>
<td>data bus line 7</td>
</tr>
</tbody>
</table>
A HD44780-Based LCD Kit (3 of 3)

Table 7.8 Pin assignment for displays with more than 80 characters

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>symbol</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DB7</td>
<td>I/O</td>
<td>data bus line 7</td>
</tr>
<tr>
<td>2</td>
<td>DB6</td>
<td>I/O</td>
<td>data bus line 6</td>
</tr>
<tr>
<td>3</td>
<td>DB5</td>
<td>I/O</td>
<td>data bus line 5</td>
</tr>
<tr>
<td>4</td>
<td>DB4</td>
<td>I/O</td>
<td>data bus line 4</td>
</tr>
<tr>
<td>5</td>
<td>DB3</td>
<td>I/O</td>
<td>data bus line 3</td>
</tr>
<tr>
<td>6</td>
<td>DB2</td>
<td>I/O</td>
<td>data bus line 2</td>
</tr>
<tr>
<td>7</td>
<td>DB1</td>
<td>I/O</td>
<td>data bus line 1</td>
</tr>
<tr>
<td>8</td>
<td>DB0</td>
<td>I/O</td>
<td>data bus line 0</td>
</tr>
<tr>
<td>9</td>
<td>E1</td>
<td>I</td>
<td>enable signal row 0 &amp; 1</td>
</tr>
<tr>
<td>10</td>
<td>R/W</td>
<td>I</td>
<td>0 = write to LCD, 1 = read from LCD</td>
</tr>
<tr>
<td>11</td>
<td>RS</td>
<td>I</td>
<td>0 = instruction input, 1 = data input</td>
</tr>
<tr>
<td>12</td>
<td>VEE</td>
<td>-</td>
<td>Contrast adjust</td>
</tr>
<tr>
<td>13</td>
<td>VSS</td>
<td>-</td>
<td>Powersupply (GND)</td>
</tr>
<tr>
<td>14</td>
<td>VCC</td>
<td>-</td>
<td>Powersupply (+5V)</td>
</tr>
<tr>
<td>15</td>
<td>E2</td>
<td>I</td>
<td>Enable signal row 2 &amp; 3</td>
</tr>
<tr>
<td>16</td>
<td>N.C</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
### HD44780 Commands (1 of 4)

Table 7.9 HD44780U instruction set

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Code</th>
<th>Description</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear display</td>
<td>0 0 0 0 0 0 0 0 0 1</td>
<td>Clears display and returns cursor to the home position (address 0).</td>
<td>1.64 ms</td>
</tr>
<tr>
<td>Cursor home</td>
<td>0 0 0 0 0 0 0 0 0 1*</td>
<td>Returns cursor to home position (address 0). Also returns display being shifted to the original position. DDRAM contents remain unchanged.</td>
<td>1.64 ms</td>
</tr>
<tr>
<td>Entry mode set</td>
<td>0 0 0 0 0 0 0 0 1 I/D S</td>
<td>Set cursor move direction (I/D), specifies to shift the display (S). These operations are performed during data read/write.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Display on/off control</td>
<td>0 0 0 0 0 0 0 1 D C B</td>
<td>Sets on/off of all display (D), cursor on/off (C) and blink of cursor position character (B).</td>
<td>40 µs</td>
</tr>
<tr>
<td>Cursor/display shift</td>
<td>0 0 0 0 0 0 1 S/C R/L * *</td>
<td>Sets cursor-move or display-(S/C), shift direction (R/L). DDRAM contents remains unchanged.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Function set</td>
<td>0 0 0 0 1 DL N F * *</td>
<td>Sets interface data length (DL), number of display line (N) and character font (F).</td>
<td>40 µs</td>
</tr>
<tr>
<td>Set CGRAM address</td>
<td>0 0 0 1</td>
<td>Sets the CGRAM address. CGRAM data is sent and received after this setting.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Set DDRAM address</td>
<td>0 0 1</td>
<td>Sets the DDRAM address. DDRAM data is sent and received after this setting.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Read busy flag and address counter</td>
<td>0 1 BF</td>
<td>Reads busy flag (BF) indicating internal operation is being performed and reads CGRAM or DDRAM address counter contents (depending on previous instruction).</td>
<td>0 µs</td>
</tr>
<tr>
<td>Write CGRAM or DDRAM</td>
<td>1 0</td>
<td>Writes data to CGRAM or DDRAM.</td>
<td>40 µs</td>
</tr>
<tr>
<td>Read from CGRAM or DDRAM</td>
<td>1 1</td>
<td>Reads data from CGRAM or DDRAM.</td>
<td>40 µs</td>
</tr>
</tbody>
</table>
### HD44780 Commands (2 of 4)

Table 7.10 LCD instruction bit names

<table>
<thead>
<tr>
<th>Bit name</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/D</td>
<td>0 = decrement cursor position. 1 = increment cursor position</td>
</tr>
<tr>
<td>S</td>
<td>0 = no display shift. 1 = display shift</td>
</tr>
<tr>
<td>D</td>
<td>0 = display off 1 = display on</td>
</tr>
<tr>
<td>C</td>
<td>0 = cursor off 1 = cursor on</td>
</tr>
<tr>
<td>B</td>
<td>0 = cursor blink off 1 = cursor blink on</td>
</tr>
<tr>
<td>S/C</td>
<td>0 = move cursor 1 = shift display</td>
</tr>
<tr>
<td>R/L</td>
<td>0 = shift left 1 = shift right</td>
</tr>
<tr>
<td>DL</td>
<td>0 = 4-bit interface 1 = 8-bit interface</td>
</tr>
<tr>
<td>N</td>
<td>0 = 1/8 or 1/11 duty (1 line) 1 = 1/16 duty (2 lines)</td>
</tr>
<tr>
<td>F</td>
<td>0 = 5x8 dots 1 = 5 x 10 dots</td>
</tr>
<tr>
<td>BF</td>
<td>0 = can accept instruction 1 = internal operation in progress</td>
</tr>
</tbody>
</table>
The HD44780 has a display data RAM (DDRAM) to store data to be displayed on the LCD.

The address range of DDRAM for 1-line, 2-line, and 4-line LCDs are shown in Table 7.11a, 7.11b, and 7.11c.

The HD44780 has a character generator ROM that can generate $5 \times 8$ or $5 \times 10$ character patterns from a 8-bit code.

The user can rewrite character patterns into the character generator RAM (CGRAM).

Up to eight $5 \times 8$ patterns or four $5 \times 10$ patterns can be programmed.

<table>
<thead>
<tr>
<th>Display size</th>
<th>Visible character positions</th>
<th>DDRAM addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 * 8</td>
<td>00..07</td>
<td>0x00..0x07</td>
</tr>
<tr>
<td>1 * 16</td>
<td>00..15</td>
<td>0x00..0x0F</td>
</tr>
<tr>
<td>1 * 20</td>
<td>00..19</td>
<td>0x00..0x13</td>
</tr>
<tr>
<td>1 * 24</td>
<td>00..23</td>
<td>0x00..0x17</td>
</tr>
<tr>
<td>1 * 32</td>
<td>00..31</td>
<td>0x00..0x1F</td>
</tr>
<tr>
<td>1 * 40</td>
<td>00..39</td>
<td>0x00..0x27</td>
</tr>
</tbody>
</table>
Registers of HD44780

- The HD44780 has two 8-bit user accessible registers: instruction register (IR) and data register (DR).
- To write data into display data RAM or character generator RAM, the MCU writes into the DR register.
- The address of the data RAM should be set up with a previous instruction.
- The DR register is also used for data storage when reading data from DDRAM or CGRAM.
- The register selection is shown in Table 7.12.
- The HD44780 has a busy flag that is output from the DB7 pin.
- The HD44780 uses a 7-bit address counter to keep track of the address of the next DDRAM or CGRAM location to be accessed.

<table>
<thead>
<tr>
<th>RS</th>
<th>R/W</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>IR write as an internal operation (display clear, etc)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Read busy flag (DB7) and address counter (DB0 to DB6)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>DR write as an internal operation (DR to DDRAM or CGRAM)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>DR read as an internal operation (DDRAM or CGRAM to DR)</td>
</tr>
</tbody>
</table>
HD44780 Instructions (1 of 3)

- **Clear display**
  - Writes 0x20 (space character) to all DDRAM locations
  - Sets 0 to the address counter (return cursor to upper left corner of the LCD)
  - Sets increment mode

- **Return home**
  - Sets address counter to 0
  - DDRAM contents not changed

- **Entry mode set**
  - Sets incrementing or decrementing of the DDRAM address
  - Controls the shifting (shifts if S bit = 1) of the display

- **Display on/off control**
  - Turns on/off display
  - Turns on/off cursor
  - Turns on/off cursor blinking
HD44780 Instructions (2 of 3)

- **Cursor or display shift**
  - This function shifts the cursor position to the right or left without writing or reading display data.
  - The shifting is controlled by two bits as shown in Table 7.13.

<table>
<thead>
<tr>
<th>S/C</th>
<th>R/L</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Shifts the cursor position to the left. (AC is decremented by 1)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Shifts the cursor position to the right. (AC is incremented by 1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Shifts the entire display to the left. The cursor follows the display shift.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Shifts the entire display to the right. The cursor follows the display shift.</td>
</tr>
</tbody>
</table>

- **Function set**
  - Sets the interface length (DL bit) to be 4- or 8-bit
  - Selects the number of lines (N bit) to be one or two lines
  - Selects character font (F bit) to be $5 \times 8$ or $5 \times 10$
HD44780 Instructions (3 of 3)

- Set CGRAM address
  - This command contains the address to be written into the address counter.

- Set DDRAM address
  - This command allows the user to set the starting address to display information.

- Read busy flag and address
  - This command reads the busy flag and the address counter.
  - User can use this command to determine the LCD controller is ready to accept another command.
  - User can use this command to control where to start displaying information.
Interfacing the HD44780 with the HCS12

- One can treat the LCD kit as an I/O device and use an I/O port and several other I/O pins as control signals.
- The interface can be 4 bits or 8 bits.
- To read or write the LCD successfully, one must satisfy the timing requirements of the LCD. The timing diagrams for read and write are shown in Figure 7.37 and 7.38.

Figure 7.36a LCD interface example (8-bit bus, used in SSE256)

Figure 7.36b LCD interface example (4-bit bus, used in Dragon12)
Procedure to send a command to the IR register

- **Step 1**
  - Pull the RS and the E signals to low.

- **Step 2**
  - Pull the R/W signal to low.

- **Step 3**
  - Pull the E signal to high.

- **Step 4**
  - Output data to the output port attached to the LCD data bus. One needs to configure the I/O Port for output before writing data to the LCD kit.

- **Step 5**
  - Pull the E signal to low and make sure that the internal operation is complete.
The procedure for writing a byte to the LCD data register

- **Step 1**
  - Pull the RS signal to high.

- **Step 2**
  - Pull the R/W signal to low.

- **Step 3**
  - Pull the E signal to high.

- **Step 4**
  - Output data to the I/O port attached to the LCD data bus.

- **Step 5**
  - Pull the E signal to low and make sure that the internal operation is complete.

These procedures need to be repeated once for an LCD kit with 4-bit interface.
Write a function to send a command to the LCD kit

- Most LCD commands are completed in 40 ms.
- If the function waits for 40 ms after performing the specified operation, then most commands will be completed when the function returns.
- The assembly code for the 8-bit interface is as follows:

```assembly
lcdPort equ PTH ; LCD data port
lcdCtl equ PTK ; LCD control port
lcdE equ $80 ; E signal pin (PK7)
lcdRW equ $20 ; R/W signal pin (PK5)
lcdRS equ $10 ; RS signal pin (PK4)

; the command is contained in A

.cmd2lcd bclr lcdCtl,lcdRS+lcdRW ; select instruction register and Write
  bset lcdCtl,lcdE ; pull the E signal high
  staa lcdPort ; send the command, along with RS, E signals
  nop
  nop
  bclr lcdCtl,lcdE ; pull the E signal low
  bset lcdCtl,lcdRW ; pull R/W to high
  ldy #1 ; adding this delay will complete the internal
  jsr delayby50us ; operation for most instructions
  rts
```
The function to configure LCD sends four commands to the LCD kit:

- Entry mode set
- Display on/off
- Function set
- Clear display

```assembly
openlcd
    movb #$FF,lcdDIR ; configure port H for output
    bset lcdCtlDir,$B0 ; configure control pins for output
    ldy #5 ; wait for LCD to complete internal configuration
    jsr delayby100ms ; configuration
    ldaa #$38 ; set 8-bit data, 2-line display, 5x8 font
    jsr cmd2lcd ;
    ldaa #$0F ; turn on display, cursor, and blinking
    jsr cmd2lcd ;
    ldaa #$06 ; move cursor right (entry mode set instruction)
    jsr cmd2lcd ;
    ldaa #$01 ; clear LCD screen and return to home position
    jsr cmd2lcd ;
    ldy #2 ; wait until "clear display" command is complete
    jsr delayby1ms ;
    rts
```

† The function to configure LCD sends four commands to the LCD kit.
Function to output a character to the LCD

The character to be output is in accumulator A.

putc2lcd bset     lcdCtl,lcdRS ; select LCD Data register
                bclr     lcdCtl,lcdRW ; enable write to LCD
                bset     lcdCtl,lcdE  ; pull E to high
                staa     lcdPort     ; send data to LCD
                nop       ; provide enough length to E signal
                nop       ;
                bclr     lcdCtl,lcdE  ; pull E signal low
                bset     lcdCtl,lcdRW ; pull R/W high to complete the write cycle
                ldy      #1           ; wait until the write operation is
                jsr      delayby50us  ; complete
                rts
Function to output a string terminated by a NULL character

- The string to be output is pointed to by index register X.

```
puts2lcd ldaa 1,x+ ; get one character from the string
beq done_puts ; reach NULL character?
jsr putc2lcd
bra puts2lcd

done_puts rts
```

Example 7.7 Write an assembly program to test the previous four subroutines by displaying the following messages on two lines:

```
  hello world!
  I am ready!
```
```assembly
#include "hcs12.inc"

lcdPort equ PTH   ; LCD data pins (PH7~PH0)
lcdDIR equ DDRH    ; LCD data direction port
lcdCtl equ PTK     ; LCD control port
lcdCtlDir equ DDRK ; LCD control port direction
lcdE equ $80       ; E signal pin
lcdRW equ $20      ; R/W signal pin
lcdRS equ $10      ; RS signal pin

org $1500
lds #$1500          ; set up stack pointer
jsr openlcd         ; initialize the LCD
ldx #msg1lcd
jsr puts2lcd
ldaa #$C0           ; move to the second row
jsr cmd2lcd ;
ldx #msg2lcd
jsr puts2lcd
swi

msg1lcd fcc "hello world!"
dc.b 0

msg2lcd fcc "I am ready!"
dc.b 0

#include "c:\miniide\delay.asm"             ; include delay routines here
; include the previous four LCD functions
```