# CP316
# Liquid Crystal Displays (LCDs)

Terry Sturtevant

Wilfrid Laurier University

February 13, 2018

# HD44780 Display Controller

# HD44780 Display Controller

used in many different displays

# HD44780 Display Controller

used in many different displays

pins and instructions

# HD44780 Display Controller

used in many different displays

pins and instructions

→ Section 7.7

# HD44780 Display Controller

used in many different displays

pins and instructions

→ Section 7.7

→ page 178 - 2 line x 8 character display

# HD44780 Display Controller

used in many different displays

pins and instructions

→ Section 7.7

→ page 178 - 2 line x 8 character display

internal memory

# HD44780 Display Controller

used in many different displays

pins and instructions

→ Section 7.7

→ page 178 - 2 line x 8 character display

internal memory

→ Sections 7.7.1 to 7.7.3

# HD44780 Display Controller

used in many different displays

pins and instructions

→ Section 7.7

→ page 178 - 2 line x 8 character display

internal memory

→ Sections 7.7.1 to 7.7.3

internal registers

# HD44780 Display Controller

used in many different displays

pins and instructions

→ Section 7.7

→ page 178 - 2 line x 8 character display

internal memory

→ Sections 7.7.1 to 7.7.3

internal registers

→ Section 7.7.4

# HD44780 Display Controller (continued)

# HD44780 Display Controller (continued)

4 bit and 8 bit modes

# HD44780 Display Controller (continued)

4 bit and 8 bit modes

(including instruction timing)

# HD44780 Display Controller (continued)

4 bit and 8 bit modes

(including instruction timing)

→ Section 7.7.6

# HD44780 Display Controller (continued)

4 bit and 8 bit modes

(including instruction timing)

→ Section 7.7.6

→ page 200 - 4 bit interface

# HD44780 Display Controller (continued)

4 bit and 8 bit modes

(including instruction timing)

→ Section 7.7.6

→ page 200 - 4 bit interface

QwikFlash connections

# HD44780 Display Controller (continued)

4 bit and 8 bit modes

(including instruction timing)

→ Section 7.7.6

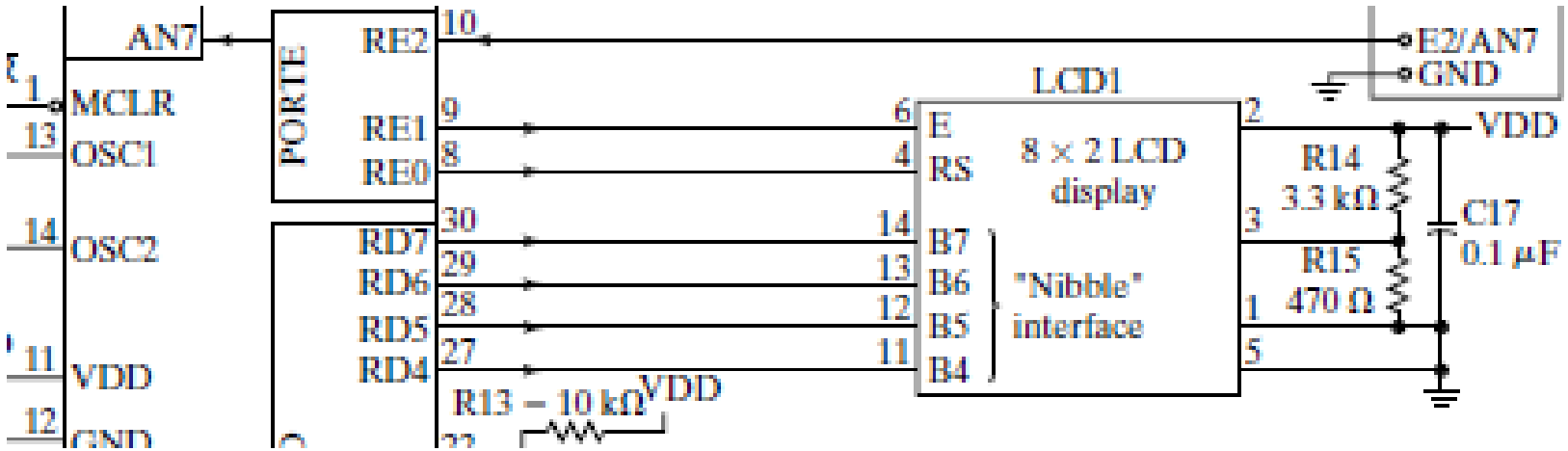
→ page 200 - 4 bit interface

QwikFlash connections

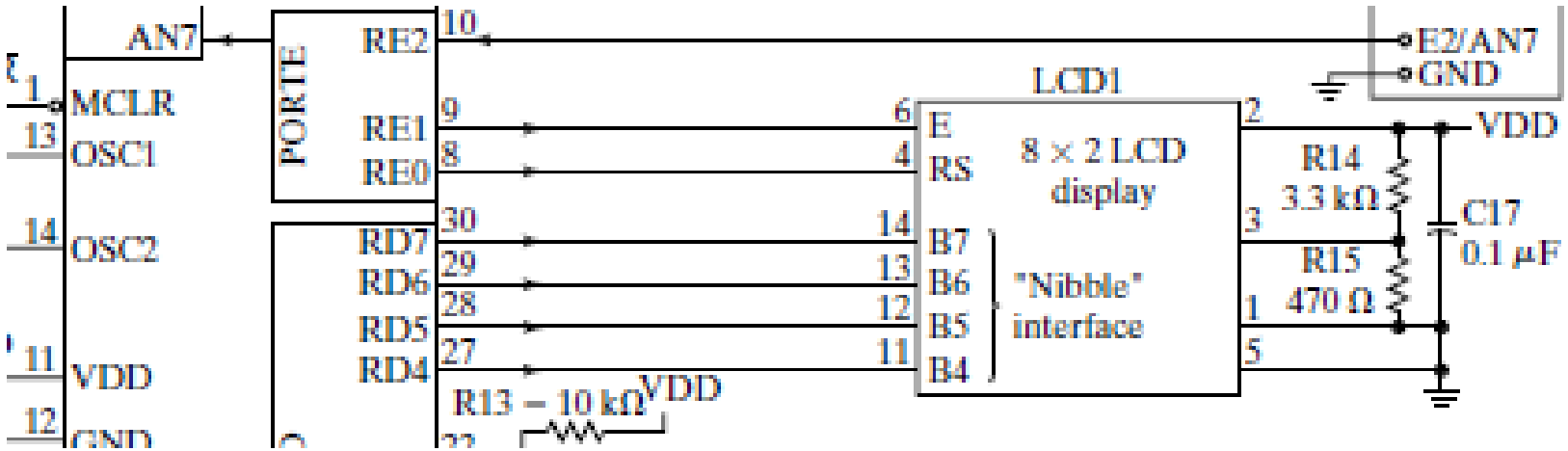
ramifications???

# QwikFlash LCD connections

# QwikFlash LCD connections

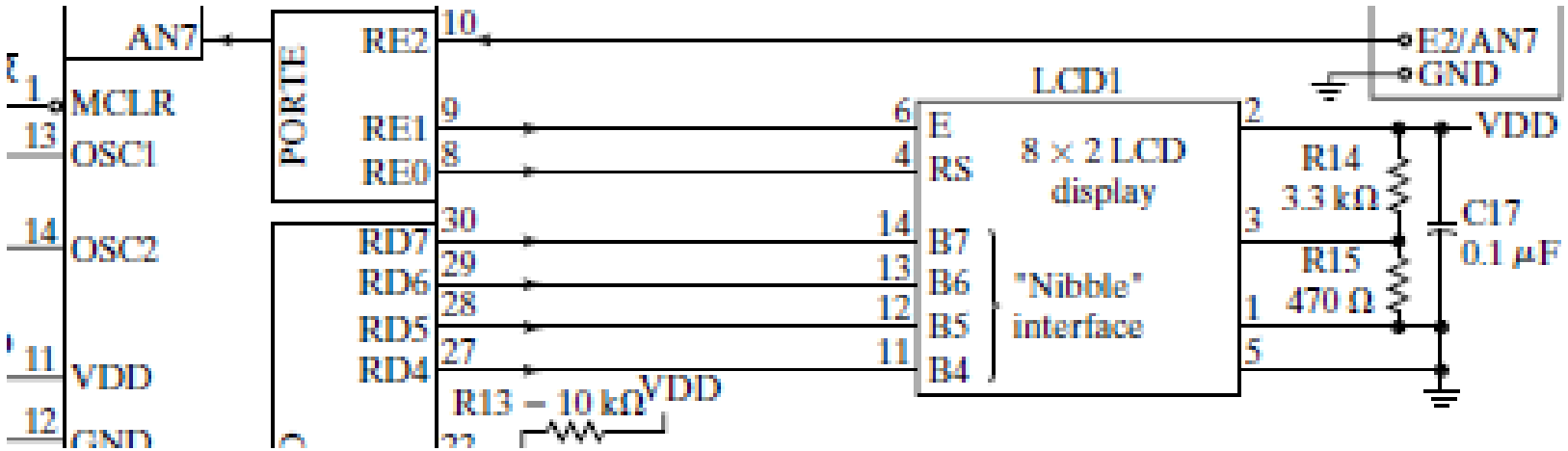


# QwikFlash LCD connections




What are pins 1, 2, 3, and 5?

# QwikFlash LCD connections




What are pins 1, 2, 3, and 5? (See page 7)

# Initialization

# Initialization

Two methods of resetting device:

# Initialization

Two methods of resetting device:

1. internal reset (on power-up)

# Initialization

Two methods of resetting device:

1. internal reset (on power-up)

→ page 190 - see note

# Initialization

Two methods of resetting device:

1. internal reset (on power-up)

   → page 190 - see note

2. by instruction (NOT in textbook)

# Initialization

Two methods of resetting device:

1. internal reset (on power-up)

   → page 190 - see note

2. by instruction (NOT in textbook)

   → page 213 - 4 bit mode reset

# Initialization

Two methods of resetting device:

1. internal reset (on power-up)

   → page 190 - see note

2. by instruction (NOT in textbook)

   → page 213 - 4 bit mode reset

   Example 7.6 problems

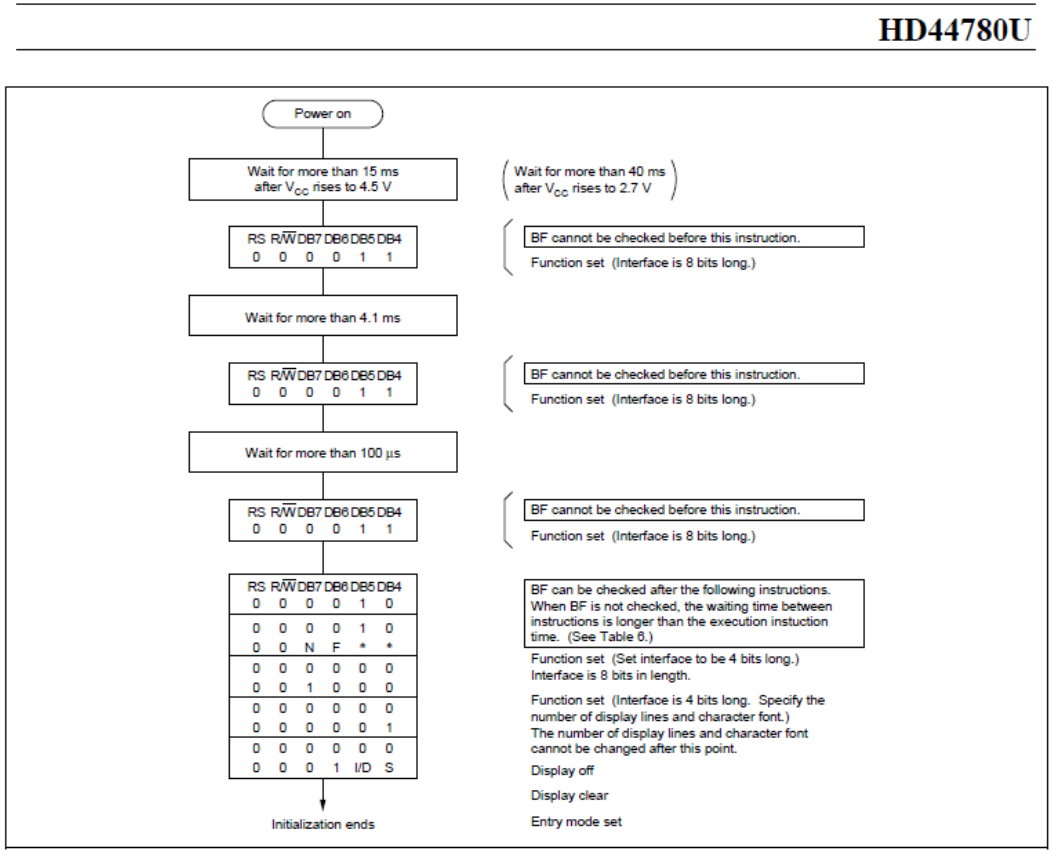



**Figure 26 4-Bit Interface**

# Initializing in 4 bit mode

HD44780U

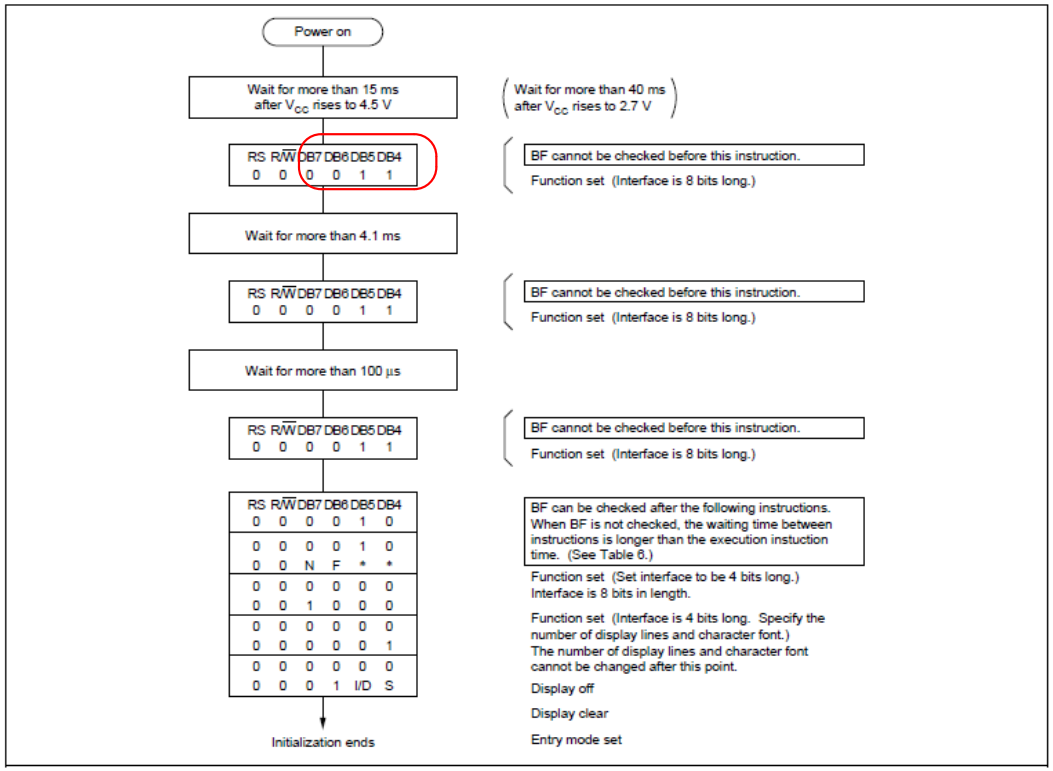



**Figure 26 4-Bit Interface**

Step 1; *Function Set* 8 bit mode

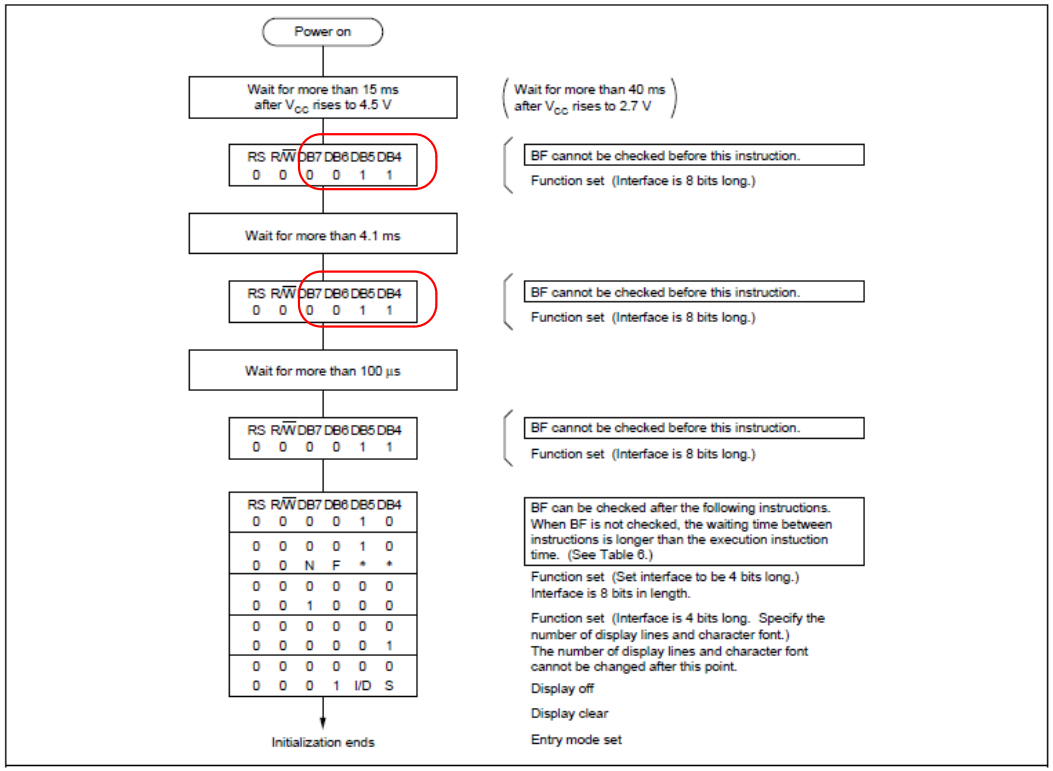



**Figure 26 4-Bit Interface**

Step 2; *Function Set* 8 bit mode

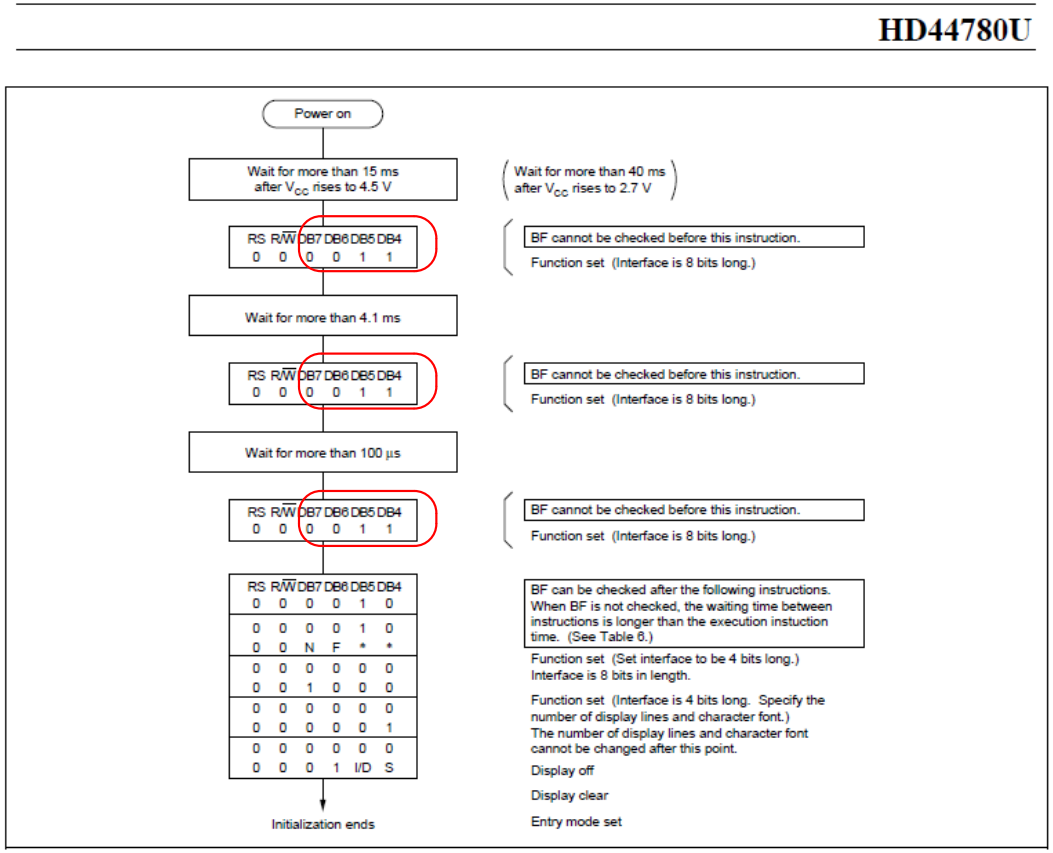



**Figure 26 4-Bit Interface**

Step 3; *Function Set* 8 bit mode

HD44780U

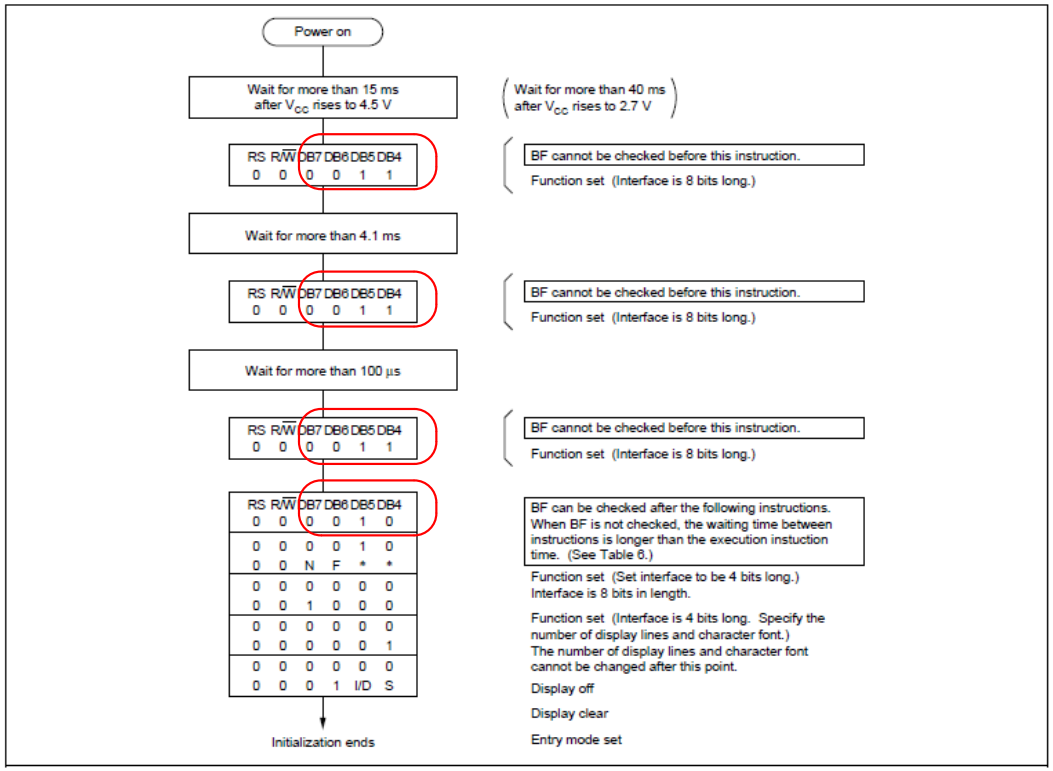



**Figure 26 4-Bit Interface**

*Function Set* interface data length **DL** (4 bit)

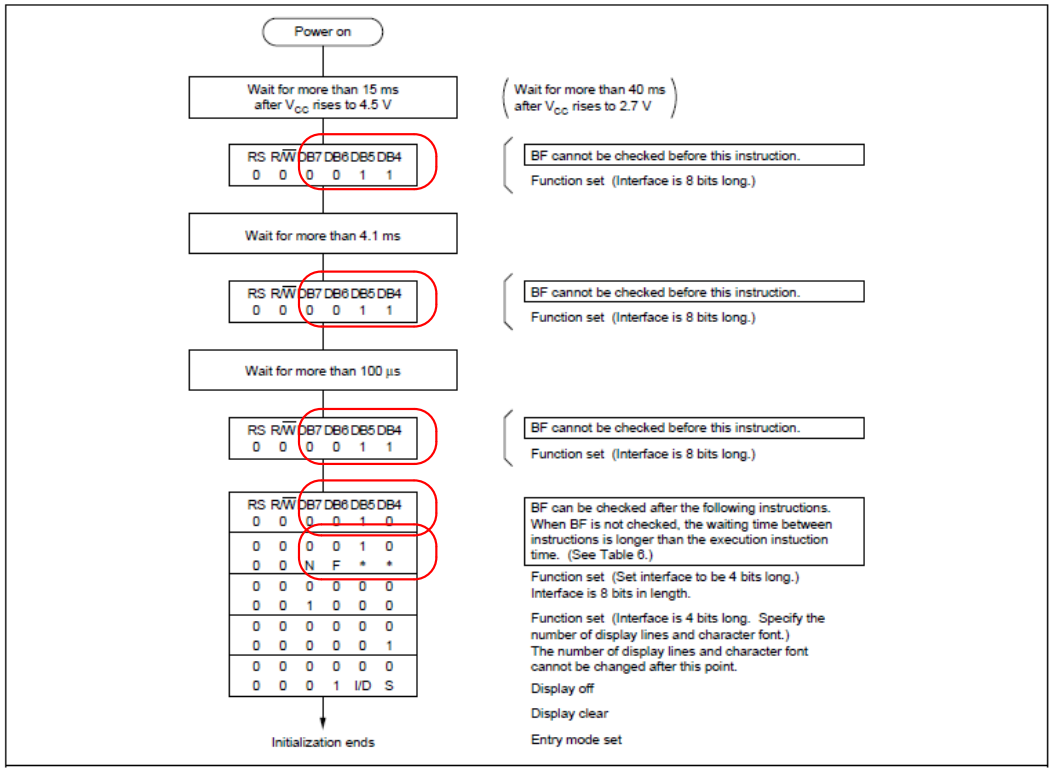



**Figure 26 4-Bit Interface**

*Function Set* display lines **N**

HD44780U

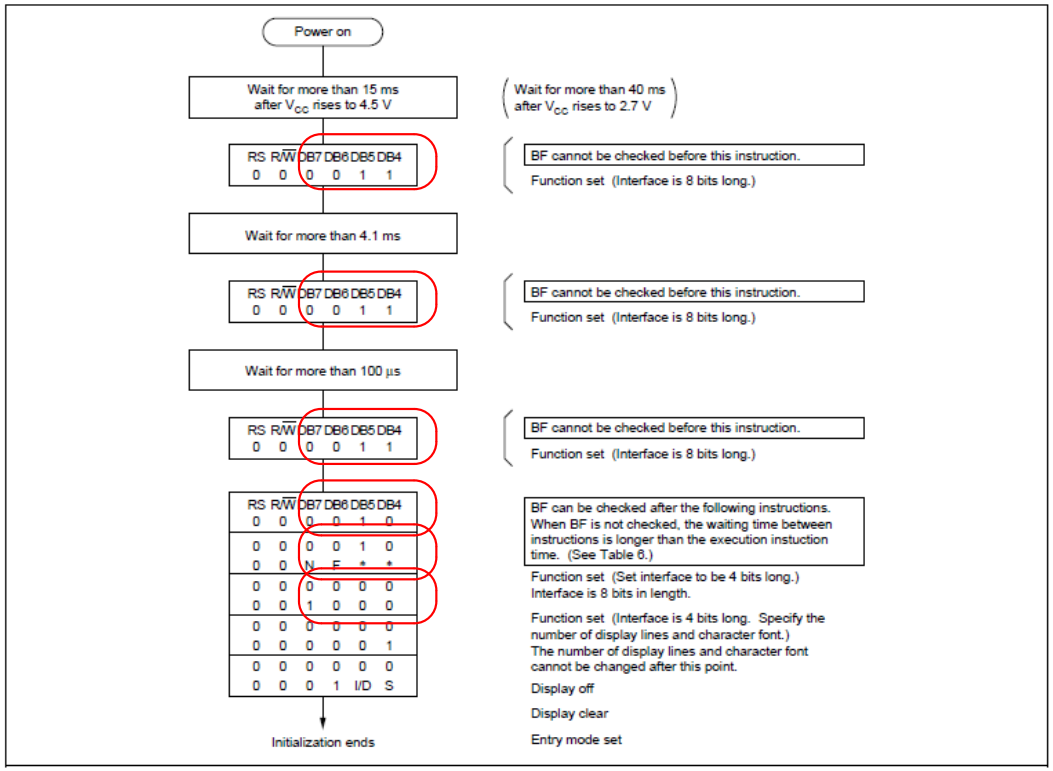



**Figure 26 4-Bit Interface**

*Display On/Off control* **D** (off)

HD44780U

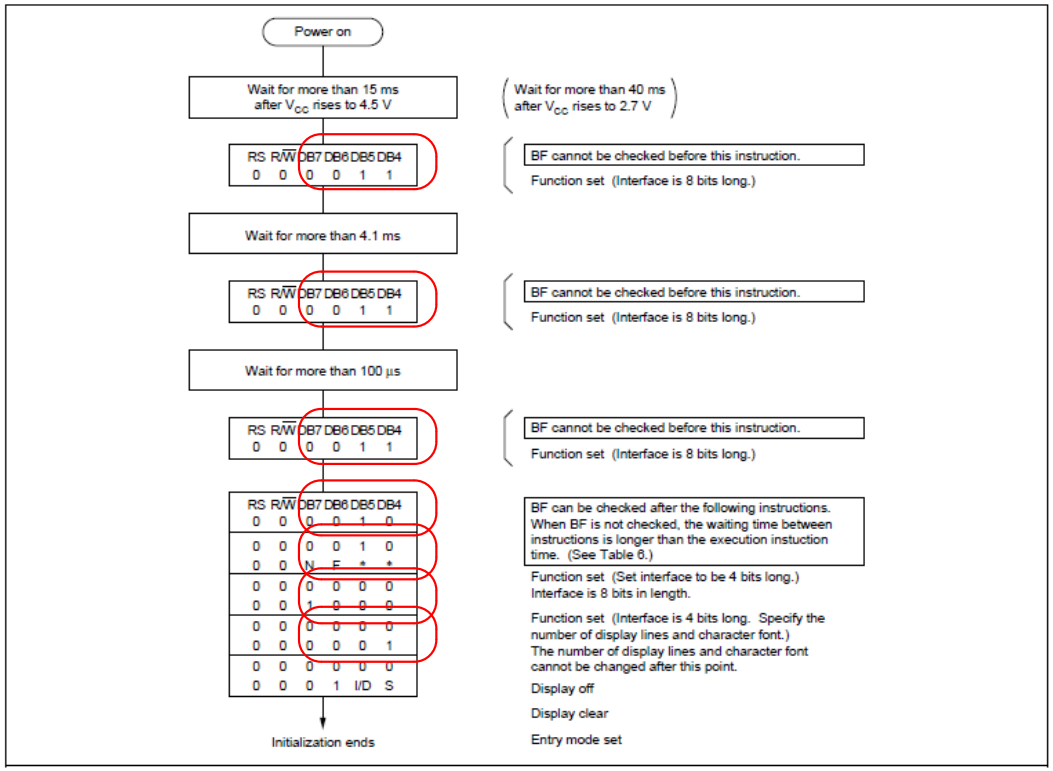



**Figure 26 4-Bit Interface**

*Clear Display*

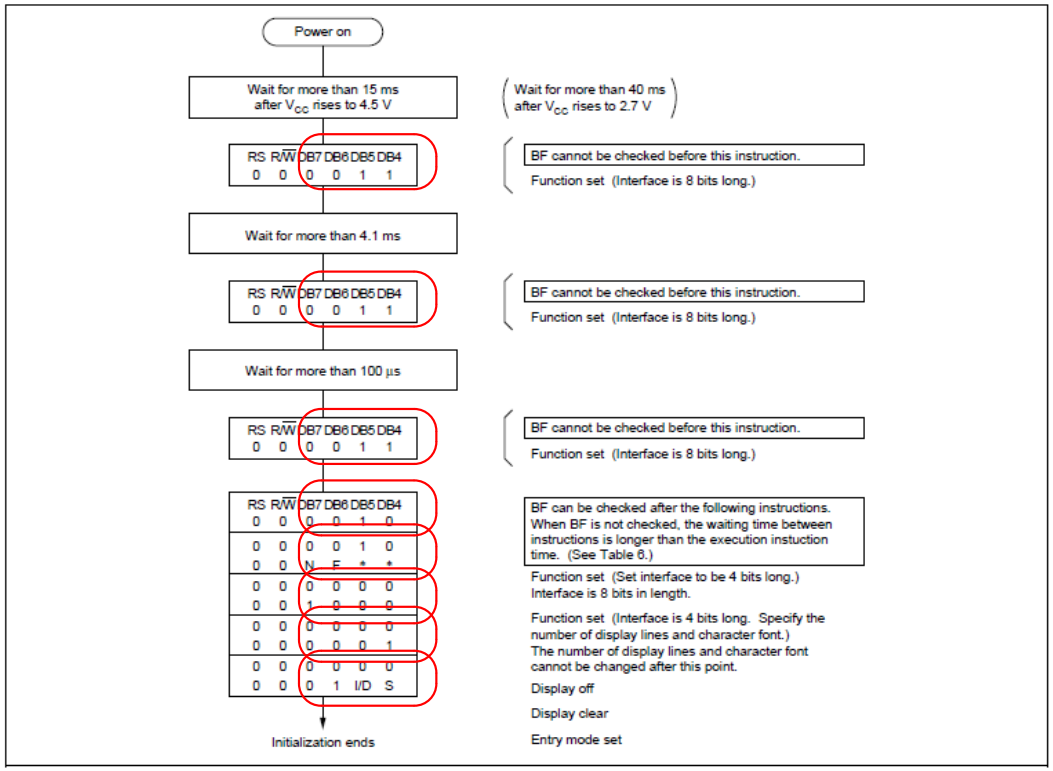



**Figure 26 4-Bit Interface**

*Entry Mode* **I/D S**

# Instructions

# Instructions

| Instruction | Code RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | Description | Execution Time (max) (when $f_{op}$ or $f_{osc}$ is 270 kHz) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Clear display | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Clears entire display and sets DDRAM address 0 in address counter. | |
| Return home | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | — | Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged. | 1.52 ms |
| Entry mode set | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S | Sets cursor move direction and specifies display shift. These operations are performed during data write and read. | 37 µs |
| Display on/off control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B | Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B). | 37 µs |
| Cursor or display shift | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | — | — | Moves cursor and shifts display without changing DDRAM contents. | 37 µs |
| Function set | 0 | 0 | 0 | 0 | 1 | DL | N | F | — | — | Sets interface data length (DL), number of display lines (N), and character font (F). | 37 µs |
| Set CGRAM address | 0 | 0 | 0 | 1 | ACG | ACG | ACG | ACG | ACG | ACG | Sets CGRAM address. CGRAM data is sent and received after this setting. | 37 µs |
| Set DDRAM address | 0 | 0 | 1 | ADD | ADD | ADD | ADD | ADD | ADD | ADD | Sets DDRAM address. DDRAM data is sent and received after this setting. | 37 µs |
| Read busy flag & address | 0 | 1 | BF | AC | AC | AC | AC | AC | AC | AC | Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents. | 0 µs |

# Instructions (continued)

# Instructions (continued)

| Instruction | Code RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | Description | Execution Time (max) (when $f_{cp}$ or $f_{OSC}$ is 270 kHz) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Write data to CG or DDRAM | 1 | 0 | Write data | | | | | | | | Writes data into DDRAM or CGRAM. | 37 μs<br>$t_{ADD}$ = 4 μs* |
| Read data from CG or DDRAM | 1 | 1 | Read data | | | | | | | | Reads data from DDRAM or CGRAM. | 37 μs<br>$t_{ADD}$ = 4 μs* |
| | I/D = 1: Increment<br>I/D = 0: Decrement<br>S = 1: Accompanies display shift<br>S/C = 1: Display shift<br>S/C = 0: Cursor move<br>R/L = 1: Shift to the right<br>R/L = 0: Shift to the left<br>DL = 1: 8 bits, DL = 0: 4 bits<br>N = 1: 2 lines, N = 0: 1 line<br>F = 1: 5 × 10 dots, F = 0: 5 × 8 dots<br>BF = 1: Internally operating<br>BF = 0: Instructions acceptable | | | | | | | | | | DDRAM: Display data RAM<br>CGRAM: Character generator RAM<br>ACG: CGRAM address<br>ADD: DDRAM address (corresponds to cursor address)<br>AC: Address counter used for both DD and CGRAM addresses | Execution time changes when frequency changes<br>Example:<br>When $f_{cp}$ or $f_{OSC}$ is 250 kHz,<br>$37\ \mu s \times \frac{270}{250} = 40\ \mu s$ |

Note: — indicates no effect.

* After execution of the CGRAM/DDRAM data write or read instruction, the RAM address counter is incremented or decremented by 1. The RAM address counter is updated after the busy flag turns off. In Figure 10, $t_{ADD}$ is the time elapsed after the busy flag turns off until the address counter is updated.

# Shift

# Shift

**Table 7 Shift Function**

| S/C | R/L | |
|---|---|---|
| 0 | 0 | Shifts the cursor position to the left. (AC is decremented by one.) |
| 0 | 1 | Shifts the cursor position to the right. (AC is incremented by one.) |
| 1 | 0 | Shifts the entire display to the left. The cursor follows the display shift. |
| 1 | 1 | Shifts the entire display to the right. The cursor follows the display shift. |

**Table 8 Function Set**

| N | F | No. of Display Lines | Character Font | Duty Factor | Remarks |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 5 × 8 dots | 1/8 | |
| 0 | 1 | 1 | 5 × 10 dots | 1/11 | |
| 1 | * | 2 | 5 × 8 dots | 1/16 | Cannot display two lines for 5 × 10 dot character font |

Note: * Indicates don't care.

# Shift

**Table 7 Shift Function**

| S/C | R/L | |
|---|---|---|
| 0 | 0 | Shifts the cursor position to the left. (AC is decremented by one.) |
| 0 | 1 | Shifts the cursor position to the right. (AC is incremented by one.) |
| 1 | 0 | Shifts the entire display to the left. The cursor follows the display shift. |
| 1 | 1 | Shifts the entire display to the right. The cursor follows the display shift. |

**Table 8 Function Set**

| N | F | No. of Display Lines | Character Font | Duty Factor | Remarks |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 5 × 8 dots | 1/8 | |
| 0 | 1 | 1 | 5 × 10 dots | 1/11 | |
| 1 | * | 2 | 5 × 8 dots | 1/16 | Cannot display two lines for 5 × 10 dot character font |

Note: * Indicates don't care.

You can *move the cursor* or *shift the display*

# DD RAM

# DD RAM

**HD44780U**

- 2-line display (N = 1) (Figure 4)
  - Case 1: When the number of display characters is less than 40 × 2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters × 2 lines are displayed. See Figure 5.

    When display shift operation is performed, the DDRAM address shifts. See Figure 5.

| Display position | 1 | 2 | 3 | 4 | 5 | ..... | 39 | 40 |
|---|---|---|---|---|---|---|---|---|
| DDRAM address (hexadecimal) | 00 | 01 | 02 | 03 | 04 | ..... | 26 | 27 |
| | 40 | 41 | 42 | 43 | 44 | ..... | 66 | 67 |

**Figure 4 2-Line Display**

| Display position | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| DDRAM address | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| For shift left | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 |
| | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| For shift right | 27 | 00 | 01 | 02 | 03 | 04 | 05 | 06 |
| | 67 | 40 | 41 | 42 | 43 | 44 | 45 | 46 |

**Figure 5 2-Line by 8-Character Display Example**

# DD RAM

**HD44780U**

- 2-line display (N = 1) (Figure 4)
  - Case 1: When the number of display characters is less than 40 × 2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters × 2 lines are displayed. See Figure 5.

    When display shift operation is performed, the DDRAM address shifts. See Figure 5.

| Display position | 1 | 2 | 3 | 4 | 5 | ..... | 39 | 40 |
|---|---|---|---|---|---|---|---|---|
| DDRAM address (hexadecimal) | 00 | 01 | 02 | 03 | 04 | ..... | 26 | 27 |
| | 40 | 41 | 42 | 43 | 44 | ..... | 66 | 67 |

**Figure 4 2-Line Display**

| Display position | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| DDRAM address | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 |
| | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| For shift left | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 |
| | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| For shift right | 27 | 00 | 01 | 02 | 03 | 04 | 05 | 06 |
| | 67 | 40 | 41 | 42 | 43 | 44 | 45 | 46 |

**Figure 5 2-Line by 8-Character Display Example**

Note shifts keep addresses within original lines

# Display RAM

# Display RAM

| Instruction | Code RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | Description | Execution Time (max) (when $f_{op}$ or $f_{osc}$ is 270 kHz) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Clear display | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Clears entire display and sets DDRAM address 0 in address counter. | |
| Return home | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | — | Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged. | 1.52 ms |
| Entry mode set | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S | Sets cursor move direction and specifies display shift. These operations are performed during data write and read. | 37 μs |
| Display on/off control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B | Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B). | 37 μs |
| Cursor or display shift | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | — | — | Moves cursor and shifts display without changing DDRAM contents. | 37 μs |
| Function set | 0 | 0 | 0 | 0 | 1 | DL | N | F | — | — | Sets interface data length (DL), number of display lines (N), and character font (F). | 37 μs |
| Set CGRAM address | 0 | 0 | 0 | 1 | ACG | ACG | ACG | ACG | ACG | ACG | Sets CGRAM address. CGRAM data is sent and received after this setting. | 37 μs |
| Set DDRAM address | 0 | 0 | 1 | ADD | ADD | ADD | ADD | ADD | ADD | ADD | Sets DDRAM address. DDRAM data is sent and received after this setting. | 37 μs |
| Read busy flag & address | 0 | 1 | BF | AC | AC | AC | AC | AC | AC | AC | Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents. | 0 μs |

DDRAM addresses are 7-bit; CGRAM addresses are 6-bit

# DD RAM

# DD RAM

**HD44780U**

- 2-line display (N = 1) (Figure 4)
  - Case 1: When the number of display characters is less than 40 × 2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters × 2 lines are displayed. See Figure 5.

    When display shift operation is performed, the DDRAM address shifts. See Figure 5.

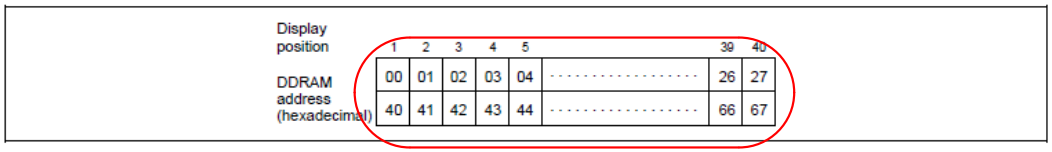



**Figure 4 2-Line Display**

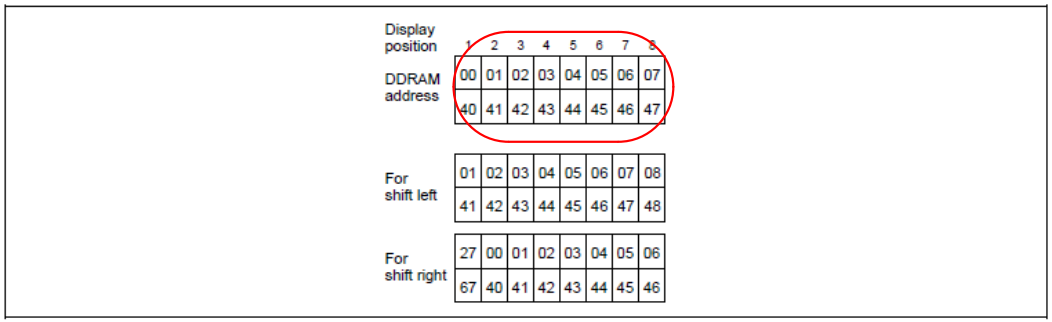



**Figure 5 2-Line by 8-Character Display Example**

# DD RAM

**HD44780U**

- 2-line display (N = 1) (Figure 4)
  - — Case 1: When the number of display characters is less than 40 × 2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters × 2 lines are displayed. See Figure 5.

    When display shift operation is performed, the DDRAM address shifts. See Figure 5.

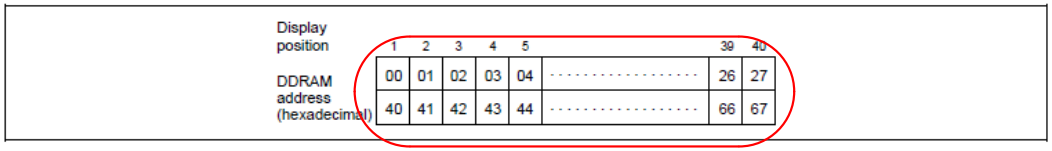



**Figure 4 2-Line Display**

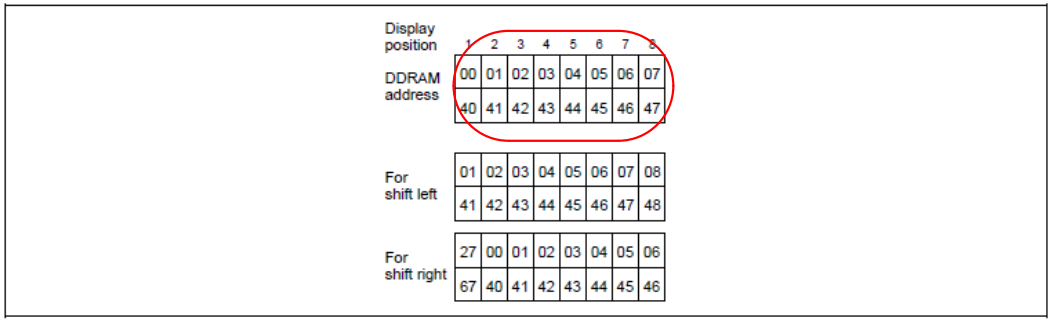



**Figure 5 2-Line by 8-Character Display Example**

Addresses 0x00 → 0x27 on top row; 0x40 → 0x67 on bottom row;

# Code

# Code

PORT configuration

# Code

PORT configuration

$\rightarrow$ macro or subroutine?

# Code

PORT configuration

$\rightarrow$ macro or subroutine?

Initialization

# Code

PORT configuration

→ macro or subroutine?

Initialization

→ macro or subroutine?

# Code

PORT configuration

→ macro or subroutine?

Initialization

→ macro or subroutine?

Write character

# Code

PORT configuration

→ macro or subroutine?

Initialization

→ macro or subroutine?

Write character

→ macro or subroutine?

# Code

PORT configuration

→ macro or subroutine?

Initialization

→ macro or subroutine?

Write character

→ macro or subroutine?

Write ???

# Code

PORT configuration

→ macro or subroutine?

Initialization

→ macro or subroutine?

Write character

→ macro or subroutine?

Write ???

→ macro or subroutine?

# Code

PORT configuration

→ macro or subroutine?

Initialization

→ macro or subroutine?

Write character

→ macro or subroutine?

Write ???

→ macro or subroutine?

*How to create delays??*

# Initialization

# Initialization

1. configure ports (**TRISD** , **TRISE**)

# Initialization

1. configure ports (**TRISD** , **TRISE**)

   don't inadvertently alter other bits

# Initialization

1. configure ports (**TRISD** , **TRISE**)
   don't inadvertently alter other bits
2. perform initialization sequence

# Initialization

1. configure ports (**TRISD** , **TRISE**)

   don't inadvertently alter other bits
2. perform initialization sequence

   make sure to observe all delays

# Initialization

1. configure ports (**TRISD** , **TRISE**)

   don't inadvertently alter other bits
2. perform initialization sequence

   make sure to observe all delays

   slow down if necessary

# Initialization

1. configure ports (**TRISD** , **TRISE**)

   don't inadvertently alter other bits
2. perform initialization sequence

   make sure to observe all delays

   slow down if necessary

   single nibble transfers until 4-bit mode selected

# Initialization

1. configure ports (**TRISD** , **TRISE**)

   don't inadvertently alter other bits

2. perform initialization sequence

   make sure to observe all delays

   slow down if necessary

   single nibble transfers until 4-bit mode selected

   after that, each transfer is 2 nibbles

# Sending a single nibble

# Sending a single nibble

1. Set or clear RS as needed

# Sending a single nibble

1. Set or clear RS as needed
2. Wait the necessary time

# Sending a single nibble

1. Set or clear RS as needed
2. Wait the necessary time
3. Write the nibble to the port

# Sending a single nibble

1. Set or clear RS as needed
2. Wait the necessary time
3. Write the nibble to the port
4. Wait the necessary time

# Sending a single nibble

1. Set or clear RS as needed
2. Wait the necessary time
3. Write the nibble to the port
4. Wait the necessary time
5. Assert the Enable

# Sending a single nibble

1. Set or clear RS as needed
2. Wait the necessary time
3. Write the nibble to the port
4. Wait the necessary time
5. Assert the Enable
6. Wait the necessary time

# Sending a single nibble

1. Set or clear RS as needed
2. Wait the necessary time
3. Write the nibble to the port
4. Wait the necessary time
5. Assert the Enable
6. Wait the necessary time
7. De-assert the Enable

# Sending a single nibble

1. Set or clear RS as needed
2. Wait the necessary time
3. Write the nibble to the port
4. Wait the necessary time
5. Assert the Enable
6. Wait the necessary time
7. De-assert the Enable
8. Wait the necessary time

# Sending a single nibble

1. Set or clear RS as needed
2. Wait the necessary time
3. Write the nibble to the port
4. Wait the necessary time
5. Assert the Enable
6. Wait the necessary time
7. De-assert the Enable
8. Wait the necessary time

*The necessary delays aren't the same for each step.*

# Sending a byte in 4-bit mode

# Sending a byte in 4-bit mode

1. Send upper nibble

# Sending a byte in 4-bit mode

1. Send upper nibble

   assuming code is designed to send upper nibble

# Sending a byte in 4-bit mode

1. Send upper nibble

   assuming code is designed to send upper nibble
2. Swap nibbles

# Sending a byte in 4-bit mode

1. Send upper nibble

   assuming code is designed to send upper nibble
2. Swap nibbles

   assuming code is designed to send upper nibble

# Sending a byte in 4-bit mode

1. Send upper nibble

   assuming code is designed to send upper nibble
2. Swap nibbles

   assuming code is designed to send upper nibble
3. Send lower nibble

# Sending commands

# Sending commands

Sending commands is simple

# Sending commands

Sending commands is simple

1. make sure RS is low

# Sending commands

Sending commands is simple

1. make sure RS is low

   it can remain low until all commands are completed

# Sending commands

Sending commands is simple

1. make sure RS is low

   it can remain low until all commands are completed
2. set cursor position (if desired)

# Sending commands

Sending commands is simple

1. make sure RS is low

   it can remain low until all commands are completed
2. set cursor position (if desired)

   DDRAM address will determine whether characters are seen or not

# Sending commands

Sending commands is simple

1. make sure RS is low

   it can remain low until all commands are completed
2. set cursor position (if desired)

   DDRAM address will determine whether characters are seen or not

   *in 4-bit mode, every transfer is 2 nibbles*

# Sending data

# Sending data

Sending data is simple

# Sending data

Sending data is simple

1. make sure RS is high

# Sending data

Sending data is simple

1. make sure RS is high

   it can remain high until all characters are sent

# Sending data

Sending data is simple

1. make sure RS is high

   it can remain high until all characters are sent

   if you wish to change DDRAM address, you need to send a command

# Sending data

Sending data is simple

1. make sure RS is high

   it can remain high until all characters are sent

   if you wish to change DDRAM address, you need to send a command

   *in 4-bit mode, every transfer is 2 nibbles*