مرکز آموزش نیـــرا سیستم

nirasystem.com

Alphanumeric Liquid-Crystal Display (ALCD)

Ali Mirghasemi

Introduction

- An Alphanumeric Liquid-Crystal Display (ALCD) is a type of electronic visual display that utilizes liquid crystals to display text, numbers, and simple graphics.
- ALCDs are commonly used in various electronic devices due to their low power consumption, readability in various lighting conditions, and cost-effectiveness

Application

- **Embedded Systems**
  - ALCDs are often integrated into microcontroller-based projects for displaying real-time data, user interfaces, and system status information.
- **Consumer Electronics**
  - They are used in digital watches, calculators, home appliances, and various gadgets requiring text and numeric displays.
- **Information Panels**
  - ALCDs are employed in information panels at airports, train stations, and public places to display schedules, announcements, and directions.

HD44780 Driver

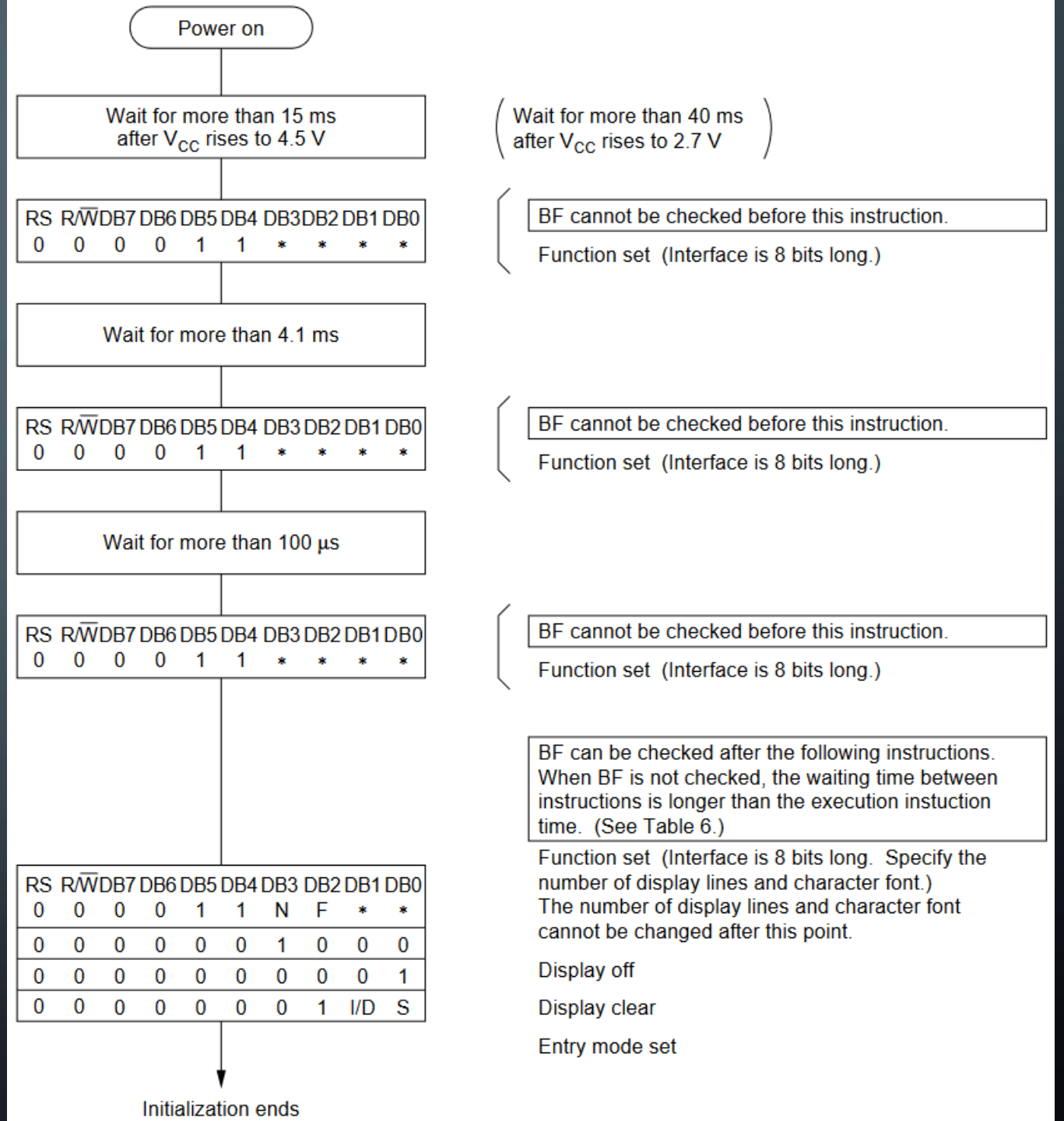
- The HD44780 is a widely used controller chip that simplifies interfacing ALCDs with microcontrollers.
- It supports standard character LCDs, making it compatible with a broad range of display sizes and manufacturers.
- The HD44780 driver provides easy-to-use commands for configuring and updating the display, such as clearing the screen, positioning the cursor, and writing characters.
- It typically requires minimal pins for communication (e.g., data, enable, read/write, and control lines) and can be connected to microcontrollers using various interface modes (e.g., 4-bit or 8-bit).
- This driver's popularity stems from its ease of use, well-documented protocols, and the availability of libraries for different microcontroller platforms.

Commands

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)
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

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

Initialize 8-bit



Initialize 4-Bit