



ADEETC

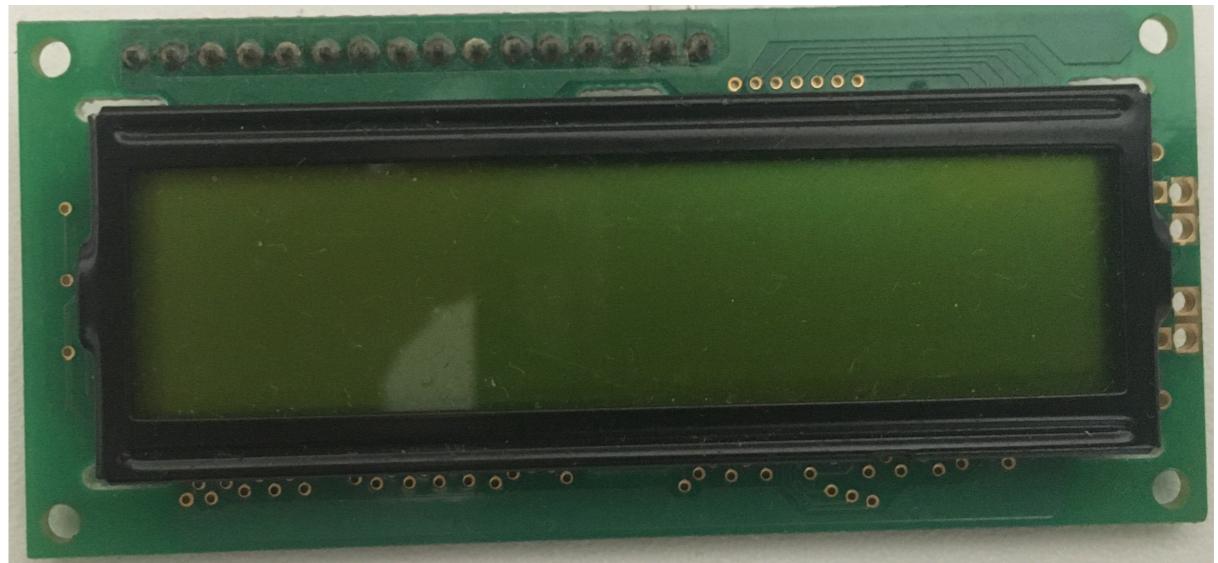
Área Departamental de
Engenharia Electrónica e
de Telecomunicações e
de Computadores

Laboratório de Informática e Computadores

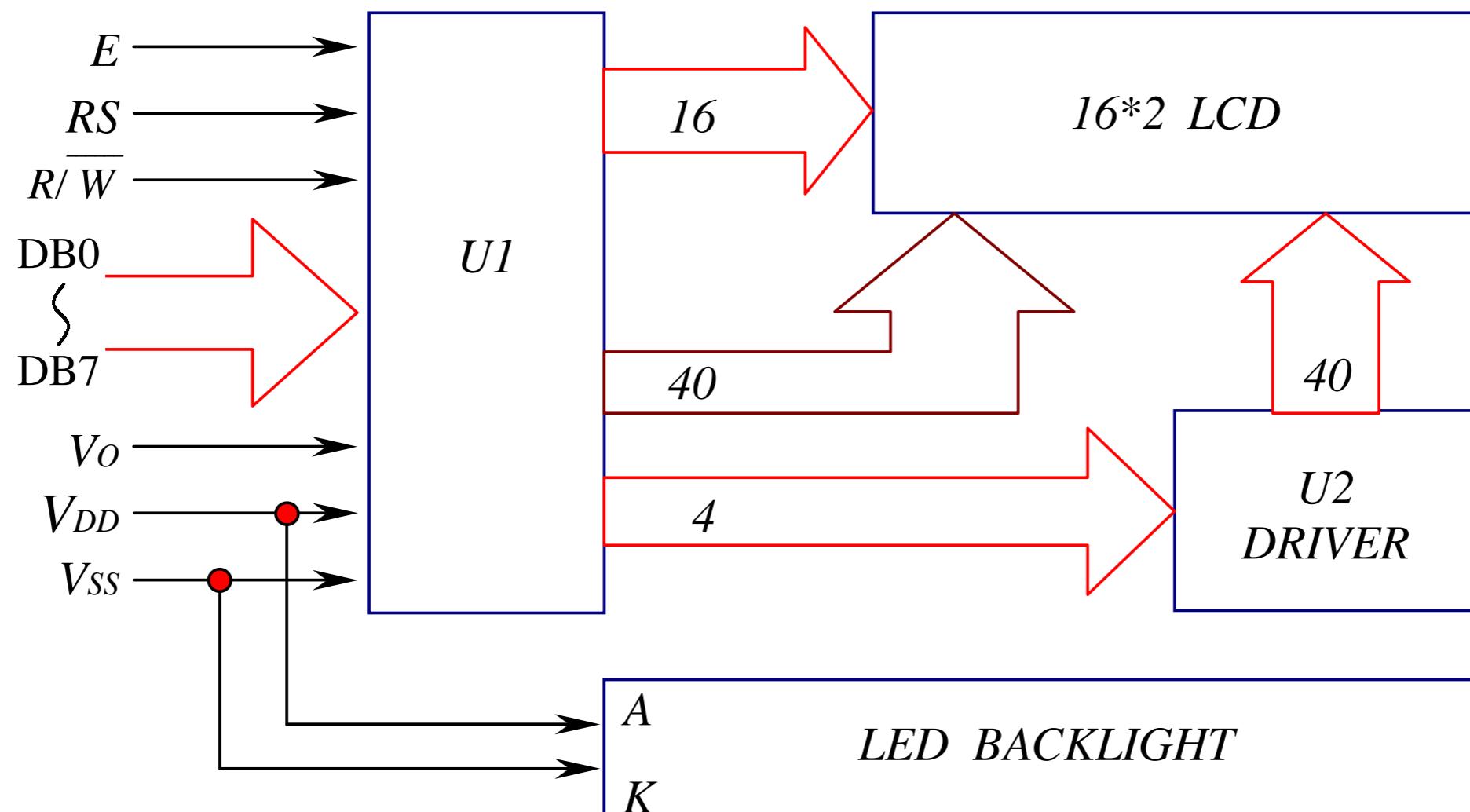
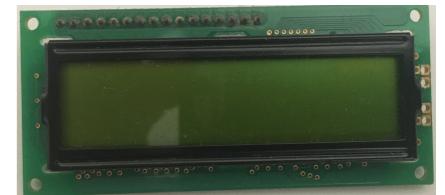
2020 / 2021 inverno
LI2ID

Liquid Cristal Display

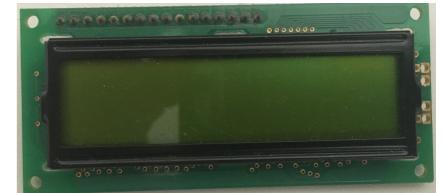
Pin No	Name	I/O	Description
1	Vss	Power	GND
2	Vdd	Power	+5v
3	Vo	Analog	Contrast Control
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	Enable (<i>Strobe</i>)
7	D0	I/O	Data <i>LSB</i>
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data <i>MSB</i>



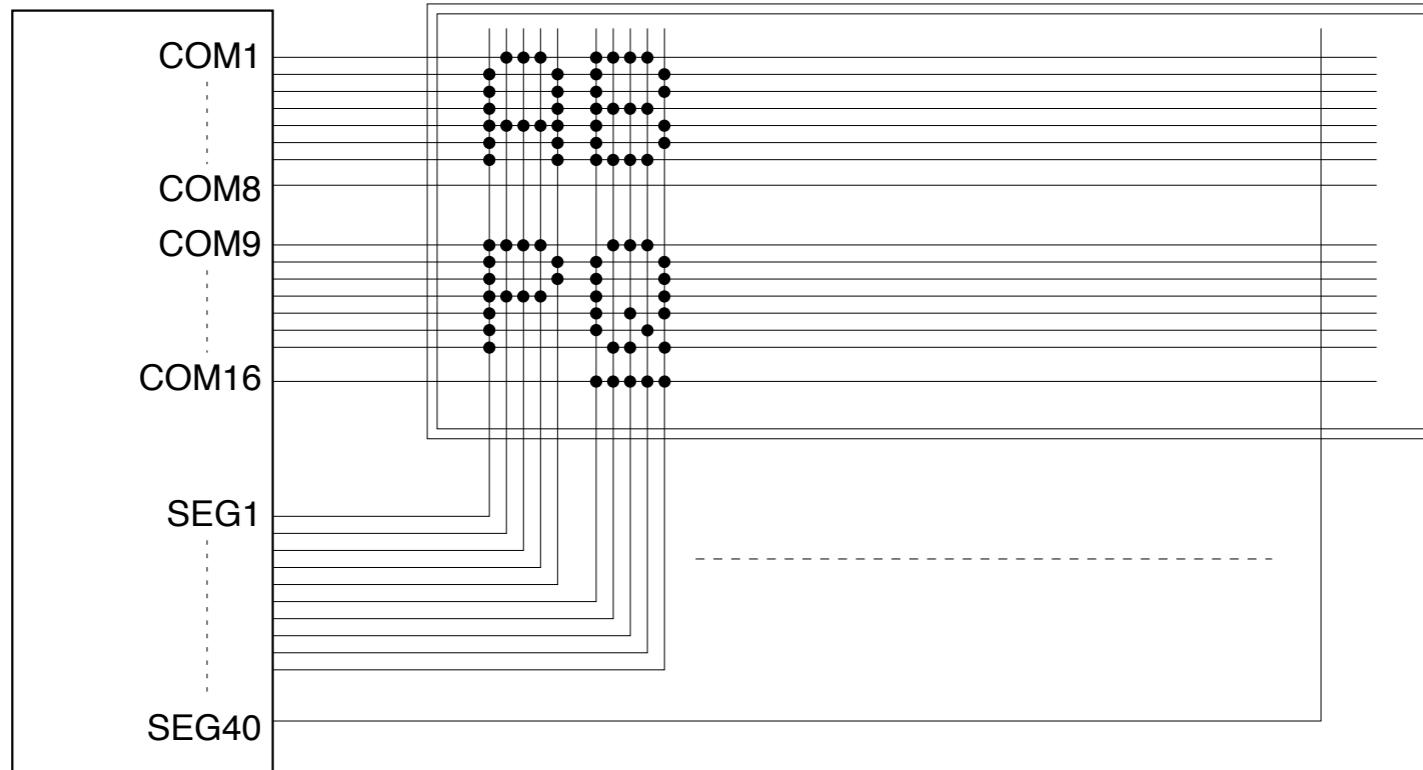
Liquid Cristal Display



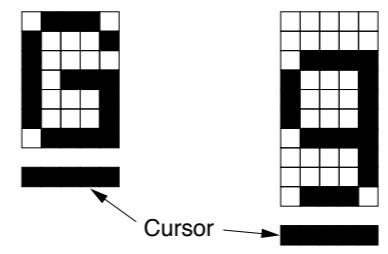
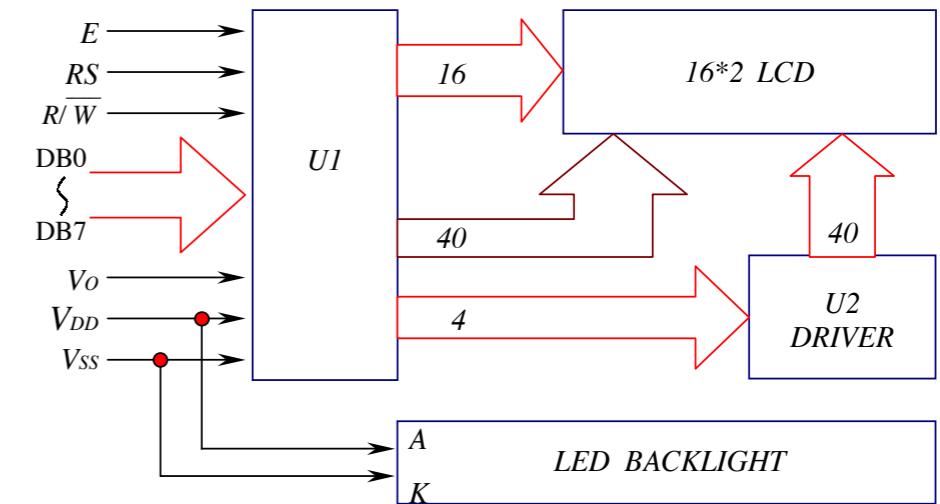
Liquid Cristal Display



HD44780

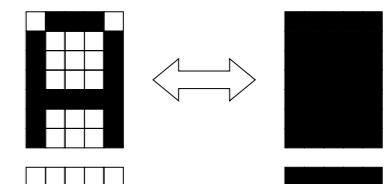


Example of a 5×8 dot, 8-character \times 2-line display (1/5 bias, 1/16 duty cycle)



5 \times 8 dot character font
5 \times 10 dot character font

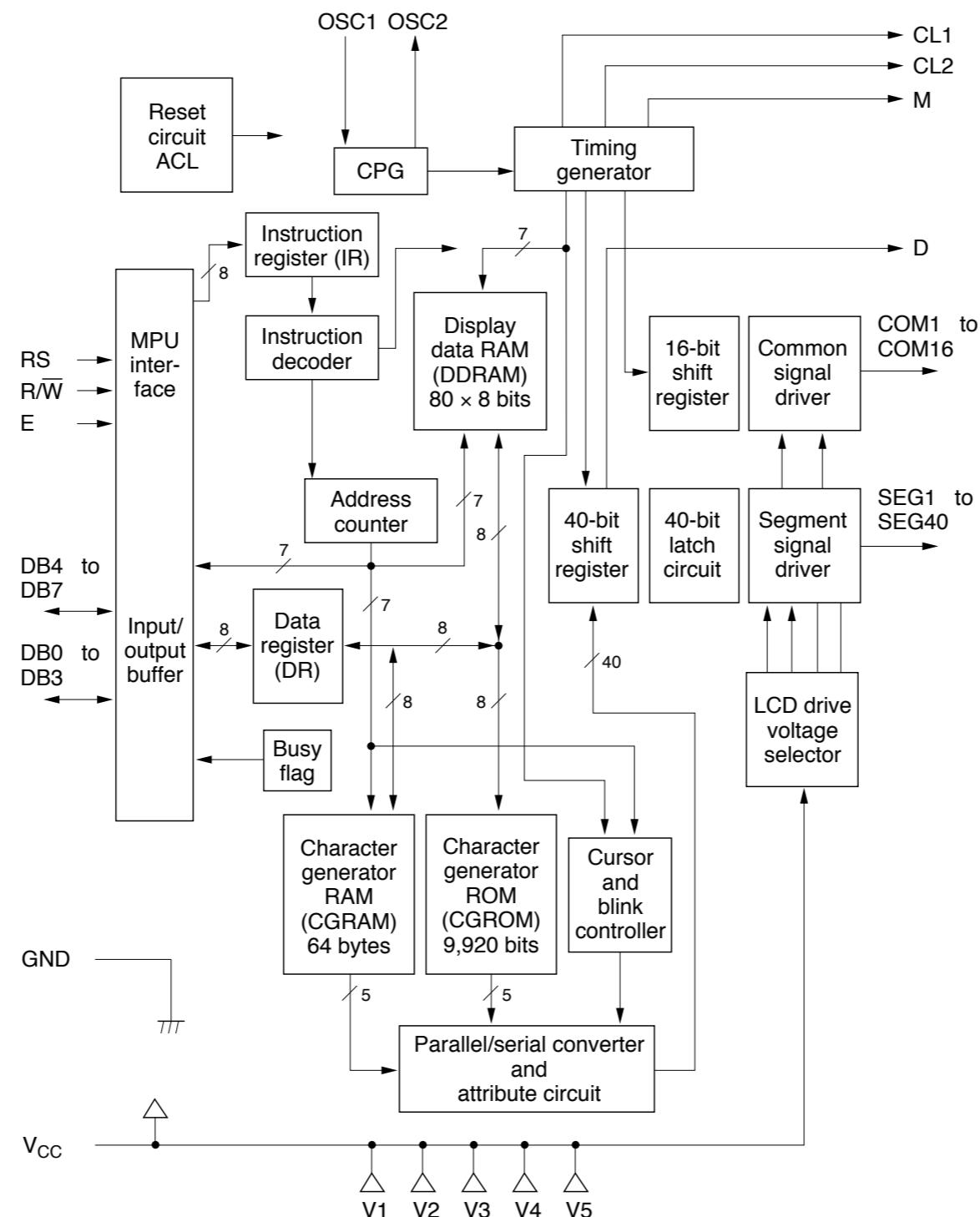
Cursor display example



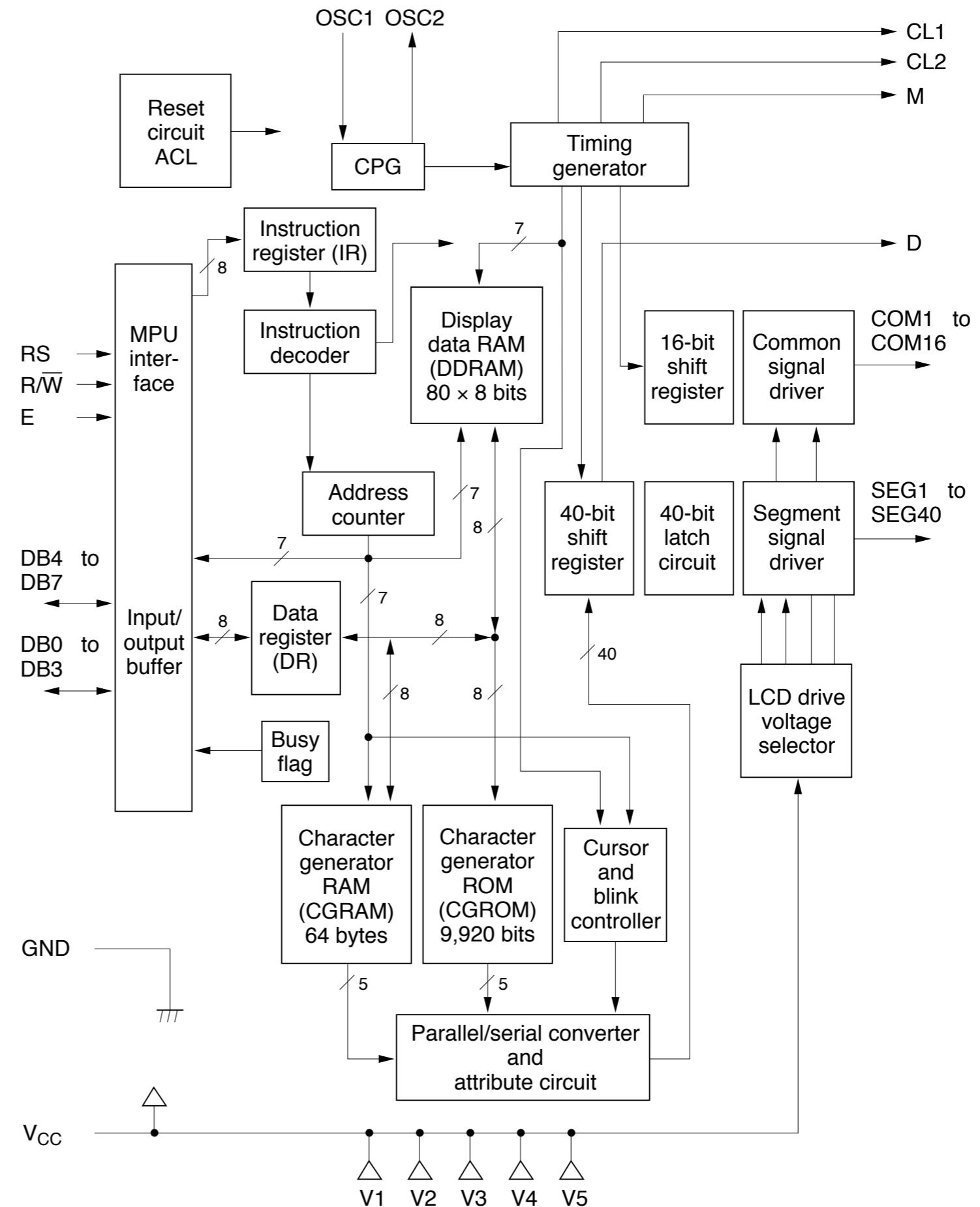
Alternating display

Blink display example

Liquid Cristal Display



LCD



Liquid Cristal Display

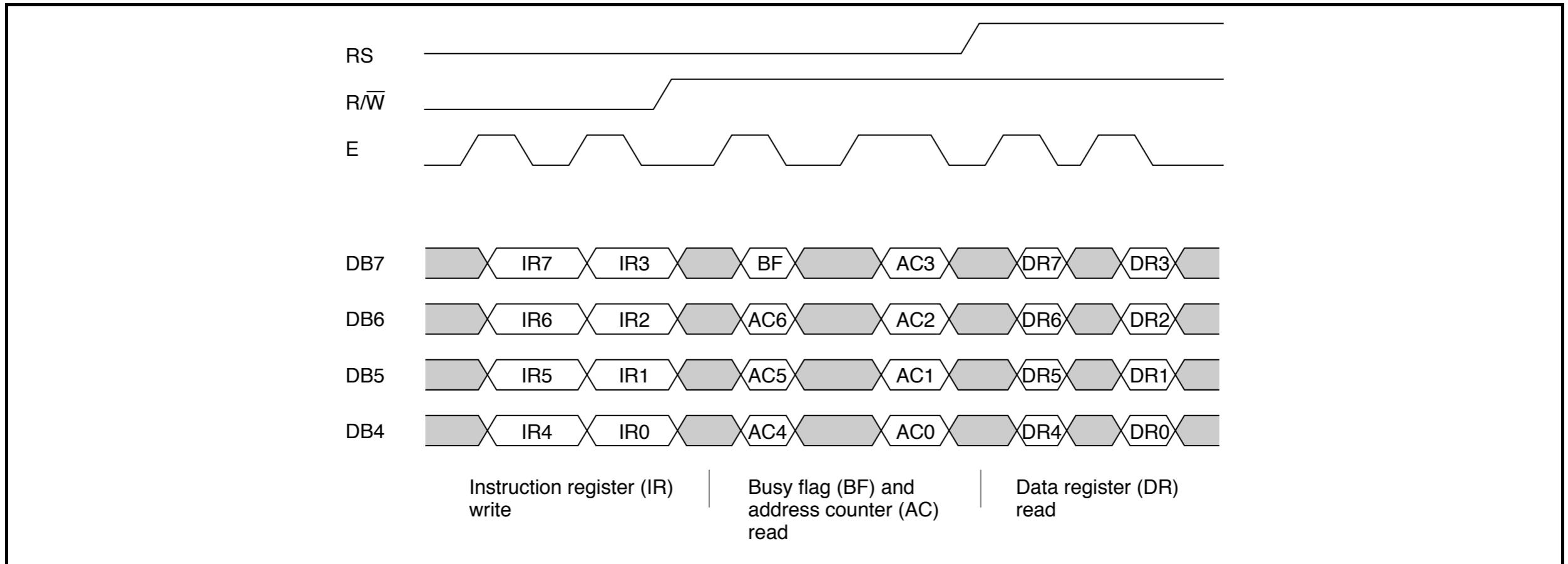
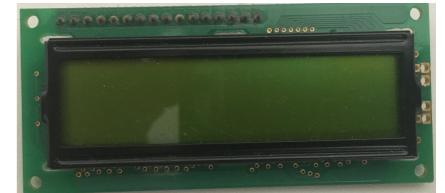
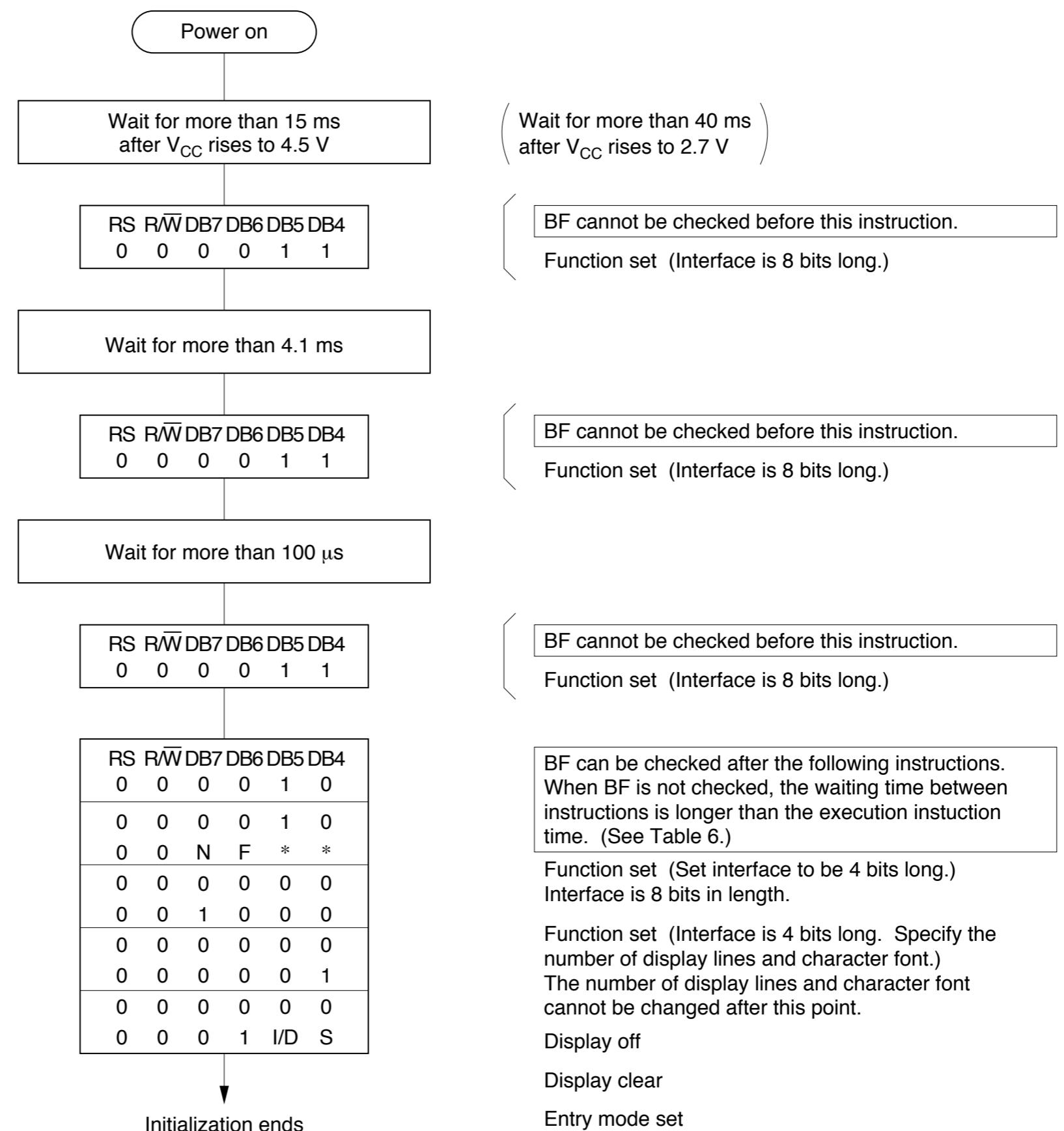
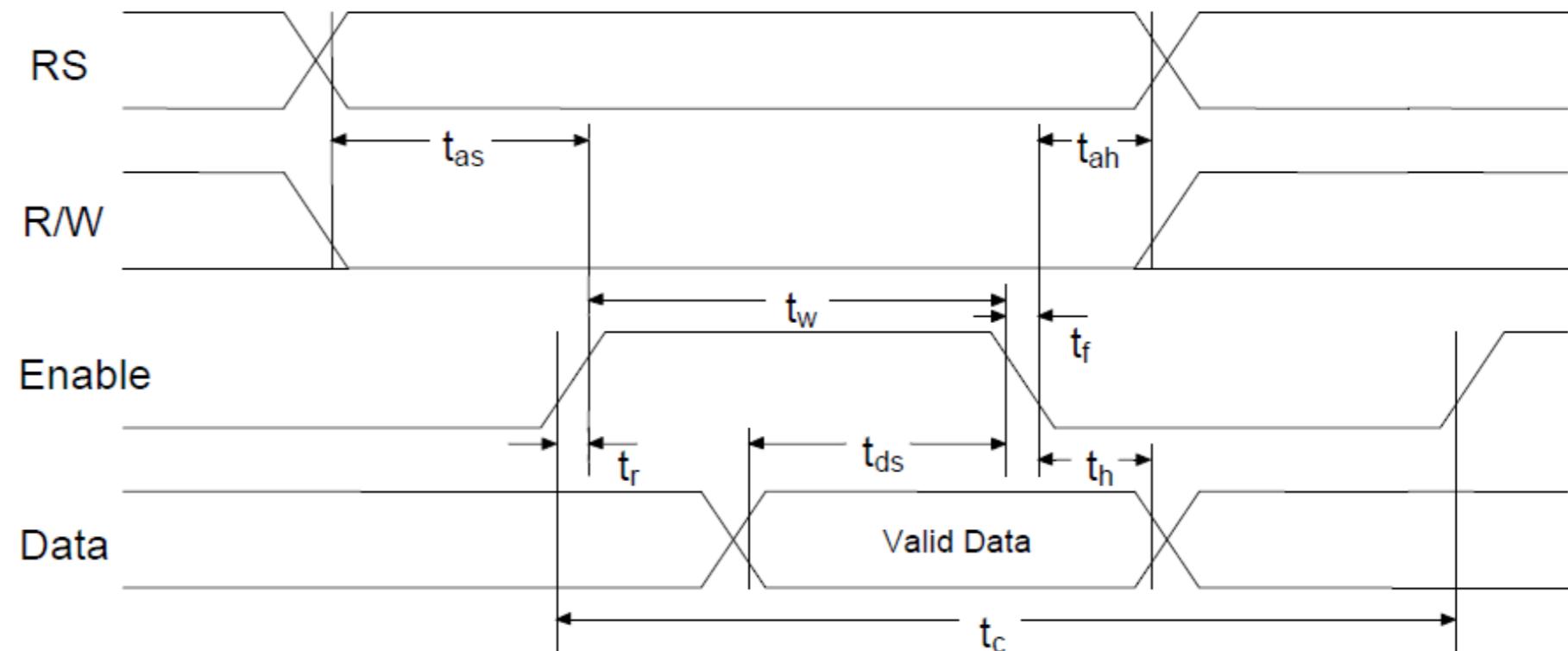


Figure 9 4-Bit Transfer Example

LCD



LCD



Write cycle

Parameter	Symbol	Min ⁽¹⁾	Typ ⁽¹⁾	Max ⁽¹⁾	Unit
Enable Cycle Time	t_c	500	-	-	ns
Enable Pulse Width (High)	t_w	230	-	-	ns
Enable Rise/Fall Time	t_r, t_f	-	-	20	ns
Address Setup Time	t_{as}	40	-	-	ns
Address Hold Time	t_{ah}	10	-	-	ns
Data Setup Time	t_{ds}	80	-	-	ns
Data Hold Time	t_h	10	-	-	ns

Note ¹ The above specifications are a indication only. Timing will vary from manufacturer to manufacturer.

LCD



Display position

1	2	3	4	5		39	40
00	01	02	03	04	26	27
40	41	42	43	44	66	67

DDRAM address
(hexadecimal)

Instruction	Code										Execution Time (max) (when f_{cp} or f_{osc} is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
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.
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.
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).
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).
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.
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.
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.
Write data to CG or DDRAM	1	0	Write data						Writes data into DDRAM or CGRAM.		37 µs $t_{ADD} = 4 \mu s^*$
Read data from CG or DDRAM	1	1	Read data						Reads data from DDRAM or CGRAM.		37 µs $t_{ADD} = 4 \mu s^*$
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, $37 \mu s \times \frac{270}{250} = 40 \mu s$

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.

LCD



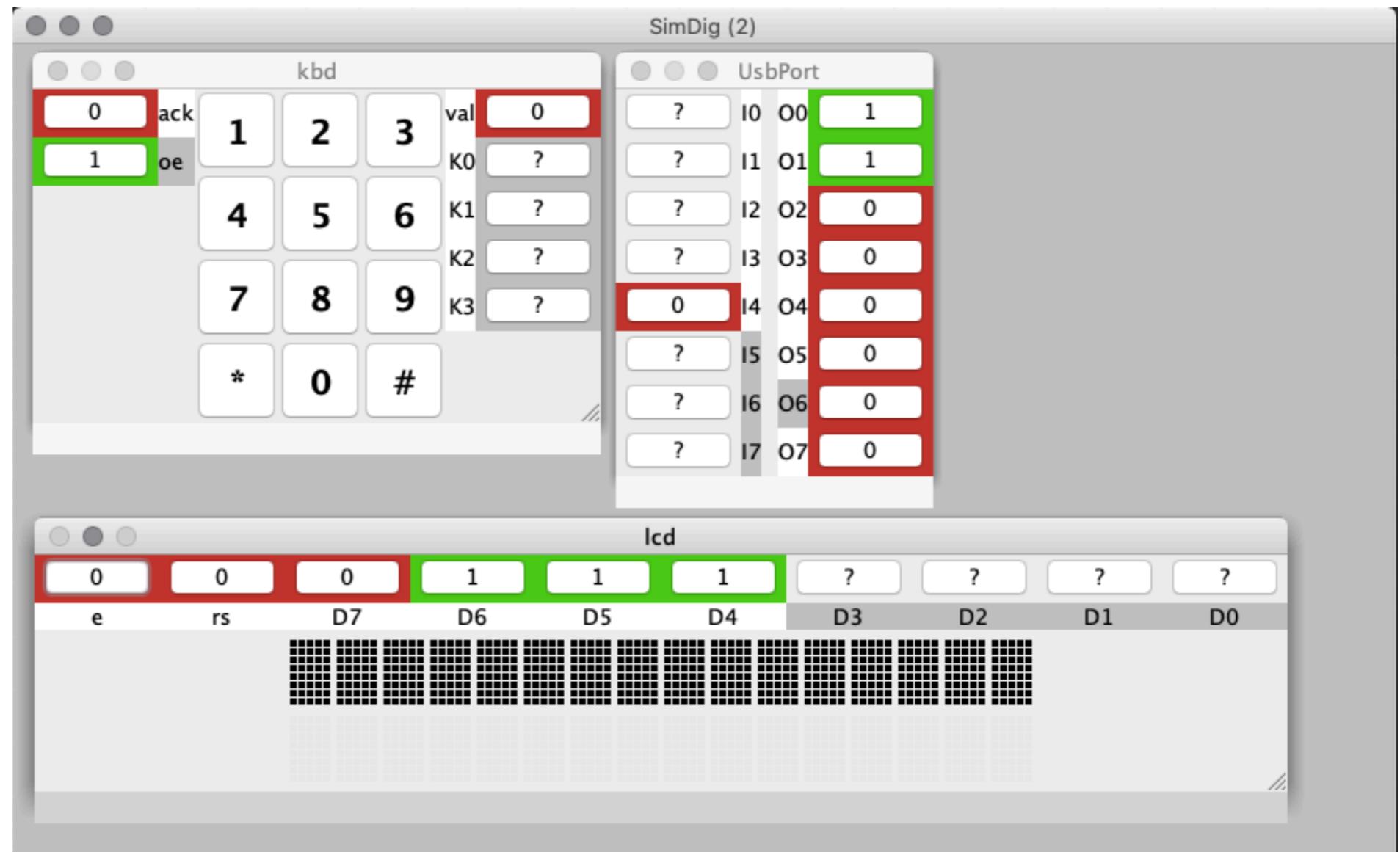
```
public class LCD {    // Escreve no LCD usando a interface a 4 bits.  
    public static final int LINES = 2, COLS = 16; // Dimensão do display.  
    // Define se a interface com o LCD é série ou paralela  
    private static final boolean SERIAL_INTERFACE = false;  
    // Escreve um nibble de comando/dados no LCD em paralelo  
    private static void writeNibbleParallel(boolean rs, int data) ...  
    // Escreve um nibble de comando/dados no LCD em série  
    private static void writeNibbleSerial(boolean rs, int data) ...  
    // Escreve um nibble de comando/dados no LCD  
    private static void writeNibble(boolean rs, int data) ...  
    // Escreve um byte de comando/dados no LCD  
    private static void writeByte(boolean rs, int data) ...  
    // Escreve um comando no LCD  
    private static void writeCMD(int data) ...  
    // Escreve um dado no LCD  
    private static void writeDATA(int data) ...  
    // Envia a sequência de iniciação para comunicação a 4 bits.  
    public static void init() ...  
    // Escreve um carácter na posição corrente.  
    public static void write(char c) ...  
    // Escreve uma string na posição corrente.  
    public static void write(String txt) ...  
    // Envia comando para posicionar cursor ('lin':0..LINES-1 , 'col':0..COLS-  
    public static void cursor(int lin, int col) ...  
    // Envia comando para limpar o ecrã e posicionar o cursor em (0,0)  
    public static void clear() ...  
}
```

LCD



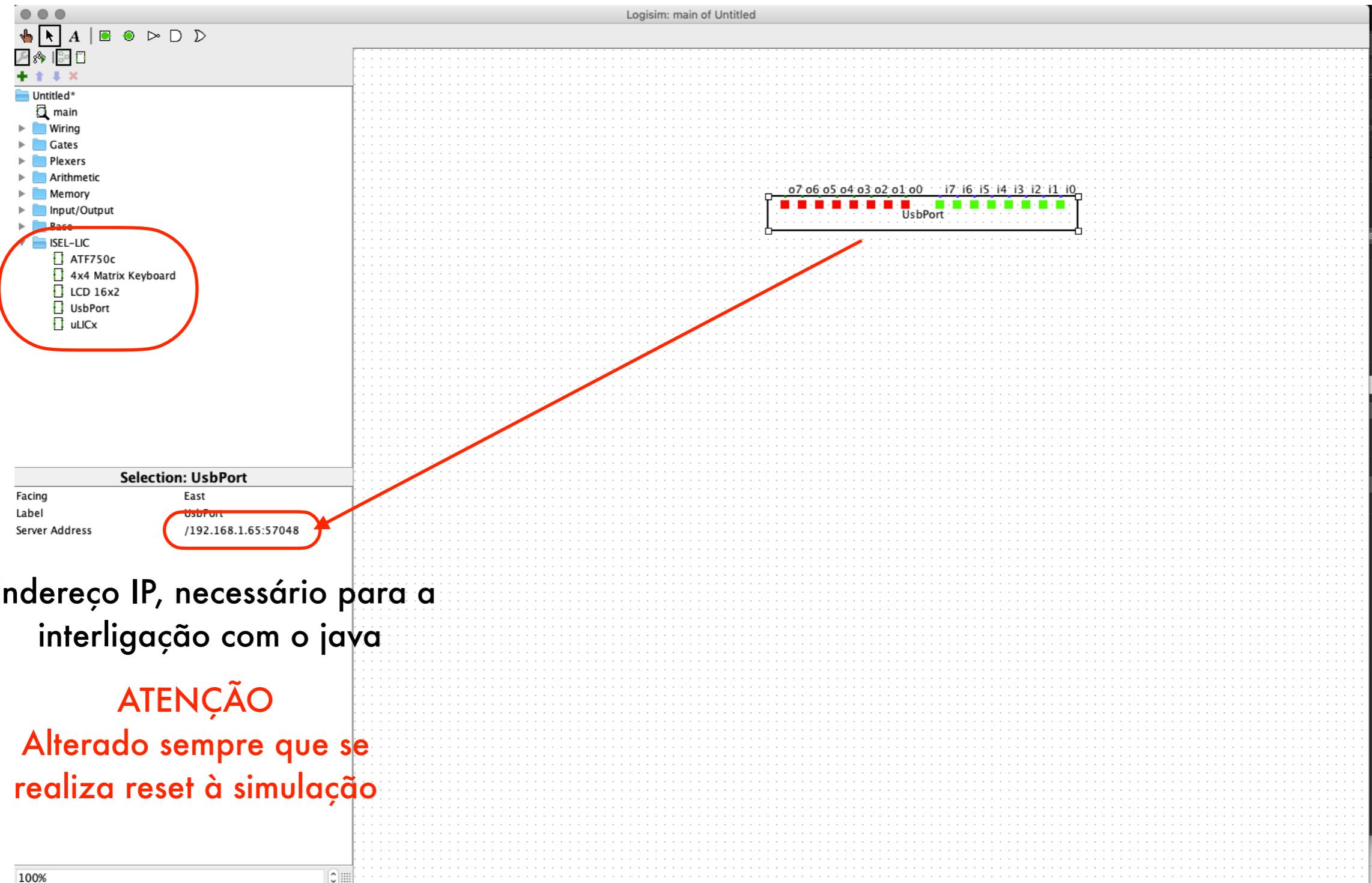
```
1 ## --- Additional packages with Hardware to Simulator
2 package isel.leic.simul
3
4 UsbPort = UsbPort
5
6 kbd = Keyboard("123456789*0#",4,3,0)
7 #lcd = LCD
8
9 # keyboard
10 kbd.val -> UsbPort.I4
11 UsbPort.07 -> kbd.ack
12 kbd.K[0-3] -> UsbPort.I[0-3]
13 1 -> kbd.oe
14
15 # LCD
16 #UsbPort.0[0-3] -> lcd.D[4-7]
17 #UsbPort.04 -> lcd.rs
18 #UsbPort.05 -> lcd.e
19
```

LCD



LCD

Biblioteca Logisim UsbPort



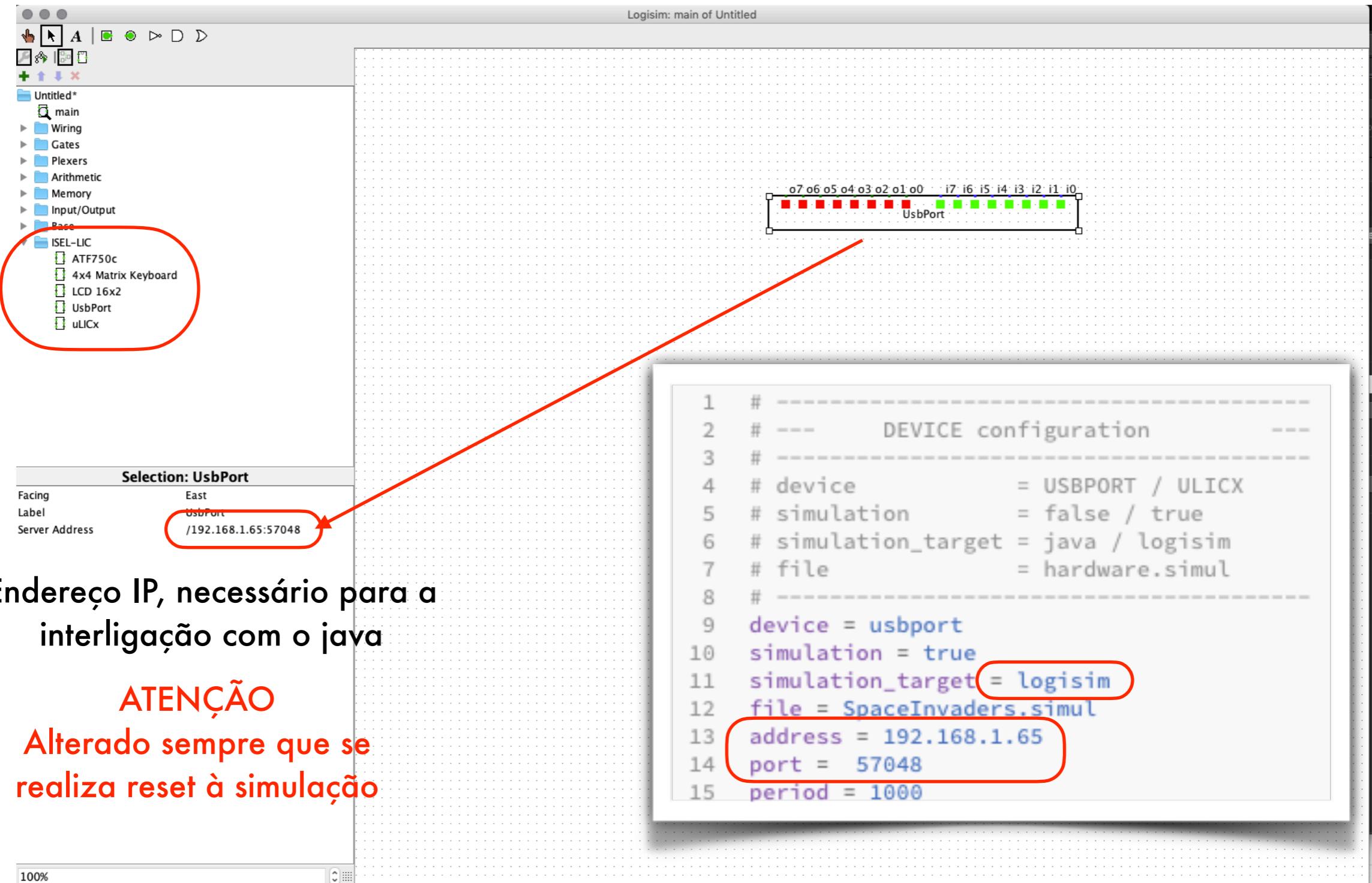
Endereço IP, necessário para a
interligação com o java

ATENÇÃO

Alterado sempre que se
realiza reset à simulação

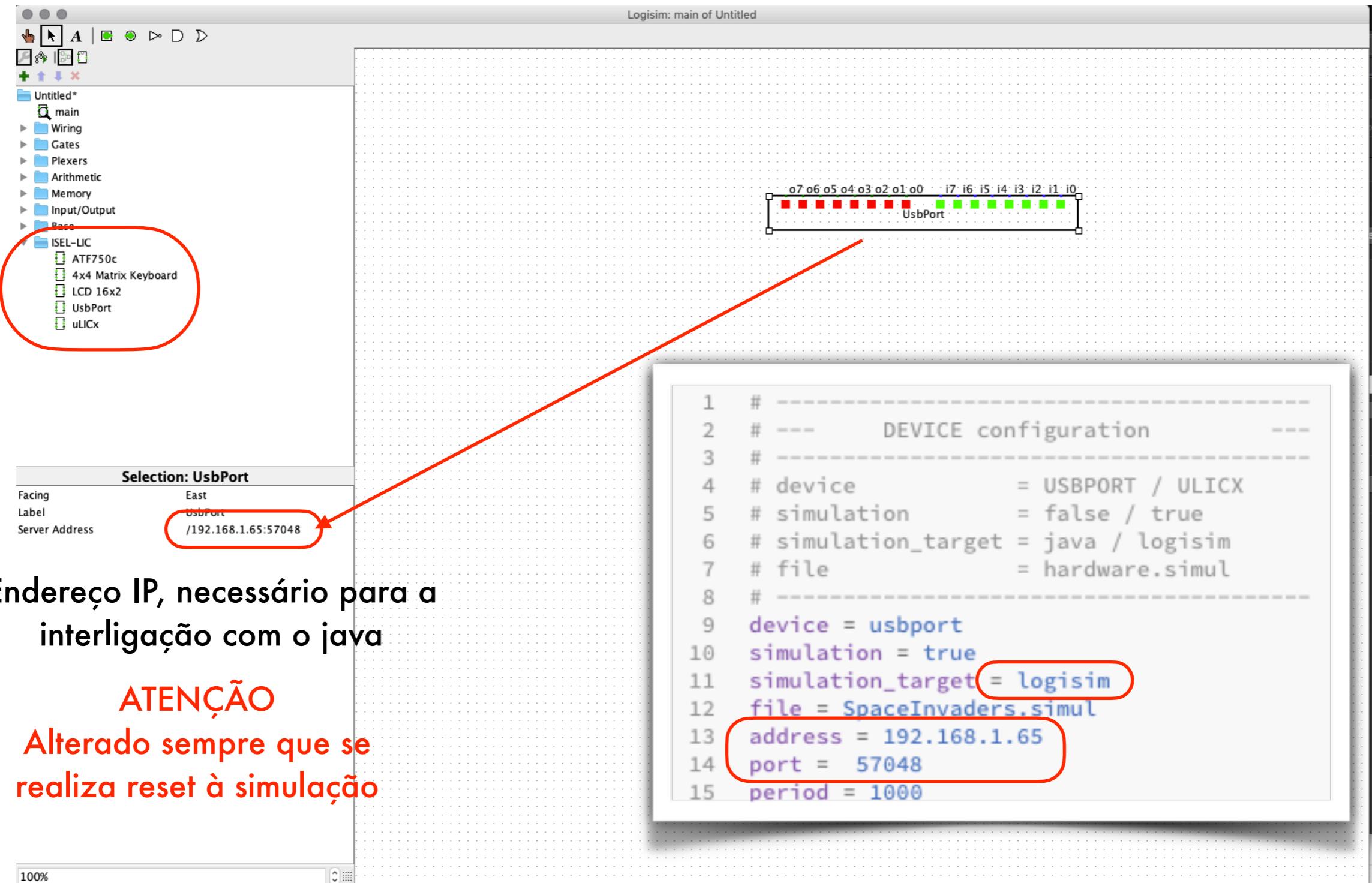
LCD

Biblioteca Logisim UsbPort



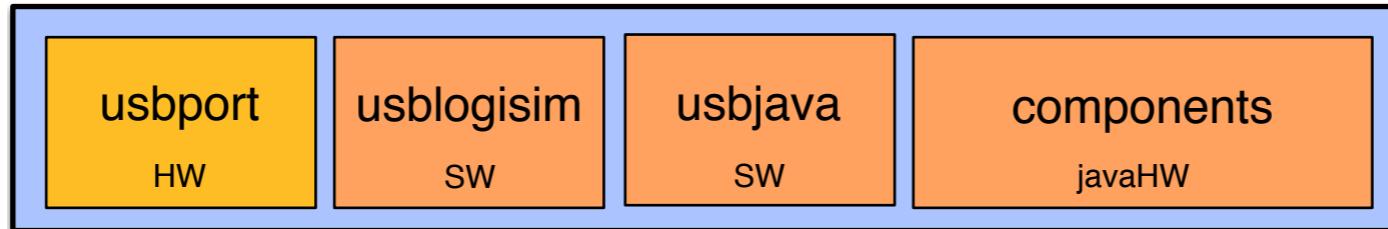
LCD

Biblioteca Logisim UsbPort



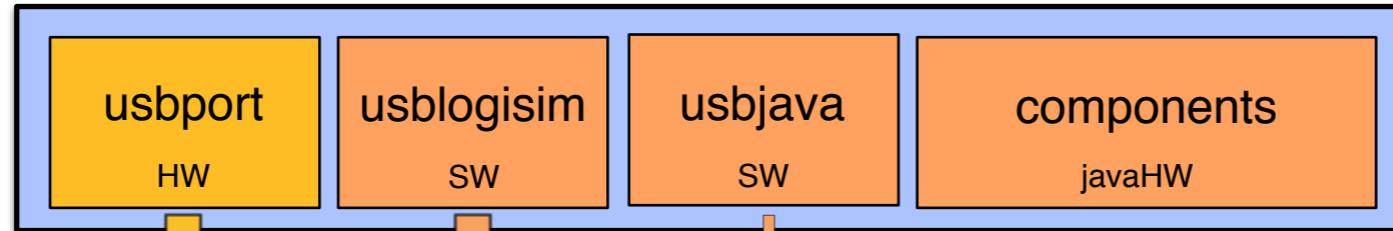
Biblioteca USBPORT

UsbPort.jar



Biblioteca USBPORT

UsbPort.jar



USB_PORT.properties

```
1 # ----- DEVICE configuration -----
2 # --- USBPORT / ULICX
3 #
4 # device = USBPORT / ULICX
5 # simulation = false / true
6 # simulation_target = java / logisim
7 # file = hardware.simul
8 #
9 device = usbport
10 simulation = true
11 simulation_target = logisim
12 file = SpaceInvaders.simul
13 address = 192.168.1.65
14 port = 55304
15 period = 10
```

