PmodCLP™ Reference Project

Revision: January 12, 2009

This document was produced by Digilent Romania. For questions, contact support@digilent.ro

Overview

This document describes the VHDL implementation of the PmodCLP (Character LCD Parallel) reference project. The PmodCLP reference project file is CLPRefProj.vhd.

The Character LCD reference project displays the message “Hello from Digilent” on the first row of the LCD screen. The Character LCD component generates the necessary signals for the LCD to display a text message.

Functional Description

The character LCD allows the user to display character messages on two rows. The PmodCLP receives data or commands through the data bus. The message is written one letter at a time. When the message fills the first row it starts shifting left.

The controller sends the *LCD_CMDS_T* array to the LCD. The array contains data and commands. In the array each byte is preceded by two bits. The first bit is used to select between data and instructions registers (*RS*). The second bit is used to select between read and write modes (*RW*).

This reference project not only writes data to the LCD module. Data can also be read from the LCD, for example, to check when the LCD is busy.

For simplicity, instead of reading back the status of the LCD, a counter (*count*) is used to create the necessary delay for different processes (e.g., 98.3ms is the maximum delay for character writes and shifts; 1.6ms is the delay for clearing the display; 20ms is the delay for powering on, etc.) The state machine moves from one state to another only after the minimum delay period for the previous operation has passed.

The RS signal (Register Selection) is used to select between the data register and the instruction register. To select the data register, RS must be set high; to select the instruction register, RS must be set low.

The RW signal (Read / Write) is used to select between read and write mode. To select the read mode, RW must be set high; to select the write mode, RW must be set low.

The E signal (Enable) is used to start reading / writing the data.

The Character LCD Component needs a 50MHz clock on the clk input.

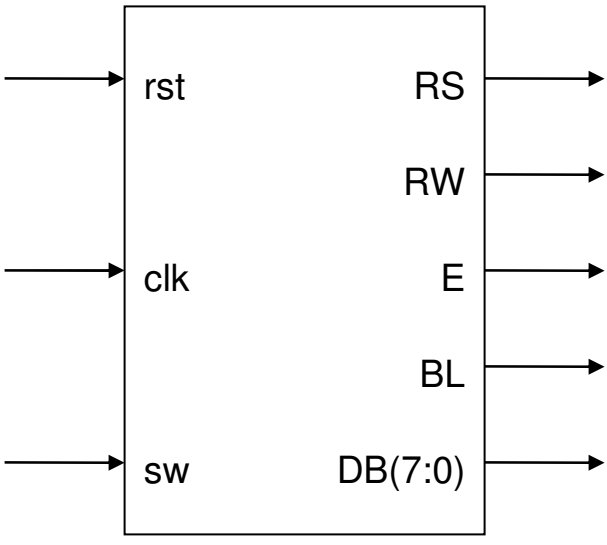


Figure 1 ***Character LCD Reference Project***

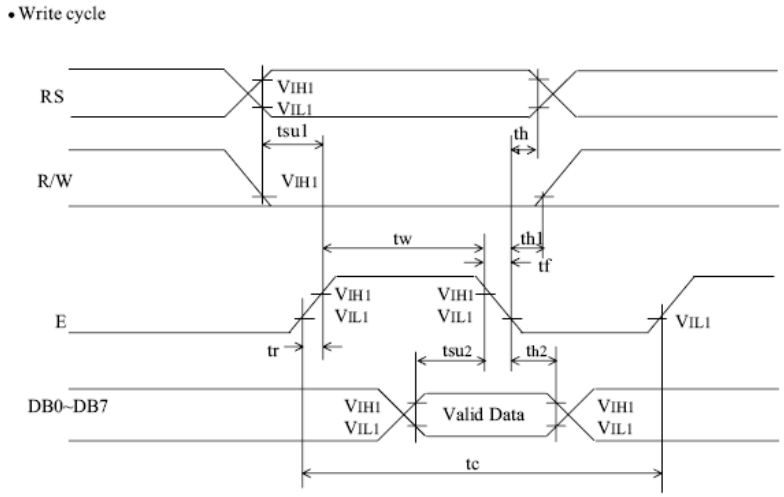
The sw (switch) entrance is internally connected to the BL (backlight) output. This output controls the backlight of the LCD (on or off).

When activated the rst (reset) input signal restarts the controller from the initial state.

Port Definitions

rst	input pin, global reset signal
clk	input pin, clock signal that has a 50MHz frequency
sw	input pin, switch input for controlling the backlight of the LCD
RS	output pin, register selection
RW	output pin, read/write used to select between read and write mode
E	output pin, start enable signal to read or write data
BL	output pin, backlight control pin
DB	output bus, eight data bus lines used for data transfer

Write Cycle Example



Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Pin
Enable cycle time	tc	500	-	-	ns	E
Enable "H" level pulse width	tw	220	-	-	ns	E
Enable rise /fall time	tr,tf	-	-	25	ns	E
RS,R/W setup time	tsu	40	-	-	ns	RS,R/W
RS,R/W address hold time	th	10	-	-	ns	RS,R/W
Read data output delay time	tD	60	-	-	ns	DB0~DB7
Read data hold time	tDH	10	-	-	ns	DB0~DB7

Figure 2 ***Write Cycle Timing***

To write data in the data register of the LCD:

- The R/W signal must be set low.
- The RS signal must be set high.
- DB0-DB7 must contain valid data.
- E must go high and after at least 220*ns* period low again. The minimum enable cycle time is 500*ns*.

Set-Up

The functionality of this project can be demonstrated using an FPGA system board which has at least one 2x6-pin connector, one 6-pin connector, and a 50MHz clock.

To set up the hardware:

1. Make sure that the clock frequency select jumper on the board (if there is one) is set to the 50MHz position.
2. Connect port J1 on the PmodCLP to a 2x6-pin connector and port J2 to a 6-pin connector.

To set up the software:

1. Create a new Xilinx ISE 9.2 project.
2. Place the VHDL file into the project.
3. Create an UCF file and make the following connections in it:

clk	connect to a 50MHz clock pin
rst	connect to a button pin
sw	connect to a switch
Db(7:0)	connect the 8 lines to the 8 data pins of the 2x6-pin connector in which port J1 of the PmodCLP is connected. Db(0) should match pin 1, Db(1) should match pin 2, Db(2) should match pin 3, Db(3) should match pin 4, Db(4) should match pin 7, Db(5) should match pin 8, Db(6) should match pin 9, and Db(7) should match pin 10
Rs	connect to pin 1 of the 6-pin connector in which port J2 of the PmodCLP is connected
Rw	connect to pin 2 of the 6-pin connector in which port J2 of the PmodCLP is connected
E	connect to pin 3 of the 6-pin connector in which port J2 of the PmodCLP is connected
Bl	connect to pin 4 of the 6-pin connector in which port J2 of the PmodCLP is connected

4. Synthesize the project and generate the programming file (*.bit).
5. Use Digilent's Adept software to program the resulting *.bit file into the FPGA. Please refer to the *Digilent Adept Reference Manual* for more detailed information.