LCD16X2A - LCD Controller

Summary

This document provides detailed reference information with respect to the non-Wishbone LCD Controller peripheral component, LCD16X2A.

The LCD Controller component (LCD16X2A) provides an interface between a host processor and an LCD panel that is equipped with a KS0066U-compatible Controller. The Controller provides a high-level interface, without direct access to the LCD panel's associated controller chip.

The Controller performs initialization of the LCD panel and has a maximum operating frequency of 80MHz.

Important Notice: Supply of this soft core under the terms and conditions of the Altium End-User License Agreement does not convey nor imply any patent rights to the supplied technologies. Users are cautioned that a license may be required for any use covered by such patent rights.

Features

- Fully compatible with LCD panels equipped with KS0066U Controller
- Fully synchronous design
- Hardware-based LCD panel initialization:

Availability

From a schematic document, the LCD16X2A component can be found in the FPGA Peripherals integrated library (`FPGA Peripherals.IntLib`), located in the `\Library\Fpga` folder of the installation.

Functional Description

Symbol

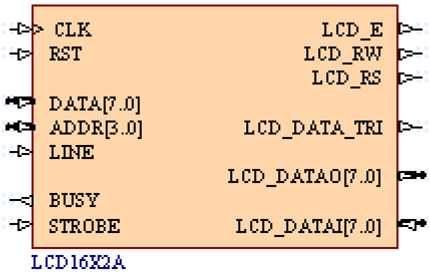

Figure 1. LCD16X2A symbol.

Pin Description

Table 1. LCD16X2A Pin description

Name	Type	Polarity/ Bus size	Description
Control Signals			
CLK	I	Rise	External system clock
RST	I	High	External system reset
Microcontroller Interface Signals			
DATA	I	8	The 8-bit ASCII code for the character to be displayed on the LCD panel.[1]
ADDR	I	4	Specifies the position on the LCD panel to be used to display a character. The LCD panel has two lines, each capable of displaying 16 characters. The 4-bit address specifies the character position along a line, where 0h represents the left-most character and Fh the right-most.
LINE	I	Level	Specifies which line on the LCD panel to be used for displaying a character, as follows: 0 – line 1 1 – line 2.
BUSY	O	High	Active when the LCD panel is busy performing the requested action. All control inputs, with the exception of RST, are ignored when the LCD panel is busy.
STROBE	I	High	Triggers the display operation (transfer of data to the LCD panel). ADDR, DATA and LINE must all be stable when STROBE goes active.
LCD Panel Interface Signals			
LCD_E	O	High	Enable signal to the LCD panel (corresponds to E pin on most LCD panels).

[1] You should consult the manufacturer's specification for the LCD panel you are using for information on the particular ASCII codes supported by the device.

Name	Type	Polarity/ Bus size	Description
LCD_RW	O	Level	Read/Write select signal to the LCD panel (corresponds to the RW pin on most LCD panels). 0 – Write 1 - Read
LCD_RS	O	Level	Data/Instruction select signal to the LCD panel (corresponds to the RS pin on most LCD panels). 0 – Instruction 1 - Data
LCD_DATA_TRI	O	Low	Tri-state enable signal for the *LCD_DATA* bidirectional bus
LCD_DATAO	O	8	LCD data output (data from the LCD Controller to the LCD panel)
LCD_DATAI	I	8	LCD data input (data from the LCD panel to the LCD Controller) Bidirectional data bus to/from the LCD panel (corresponds to the DB bus on most LCD panels).

Note: To simplify using the bidirectional LCD_DATA bus (which corresponds to the DB bus on most LCD panels), the schematic symbol includes a bus pin for each direction, allowing them to be wired independently. Configuration of bus direction is performed under program control.

Hardware Description

Block Diagram

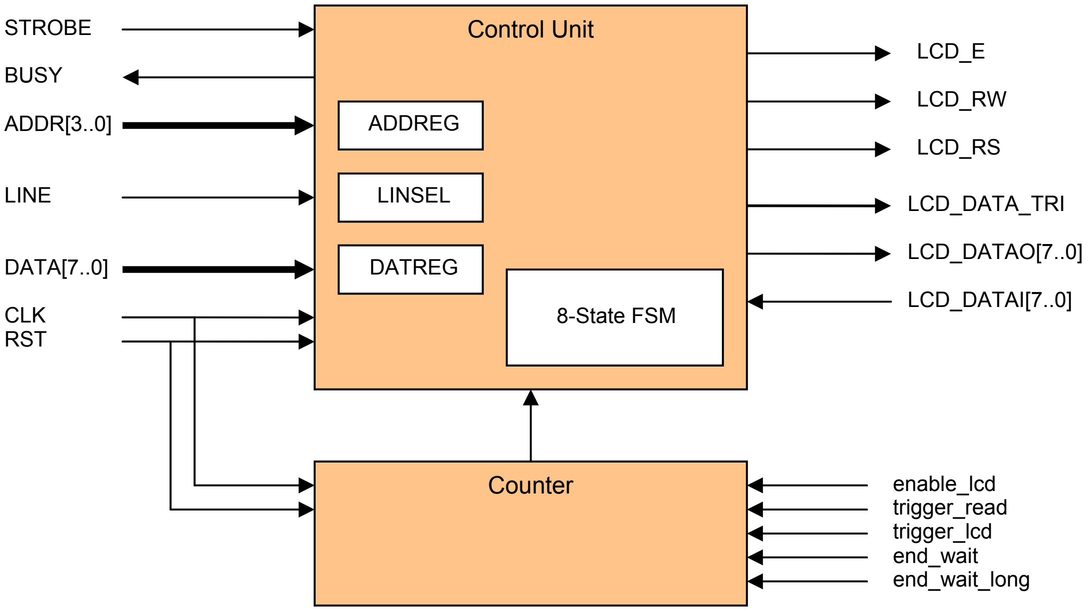

Figure 2. LCD16X2A block diagram

In the diagram, the five timing-related signals used by the Counter block are actually constants, whose values are determined by the operational frequency, the maximum of which is 80 MHz.

Placing the LCD16X2A Controller in a Design

The Controller provides a simple interface between a host microcontroller and an LCD panel. Figure 3 shows an example of how the Controller is wired into a design, in this case interfacing to the LCD panel located on the NanoBoard-NB1.

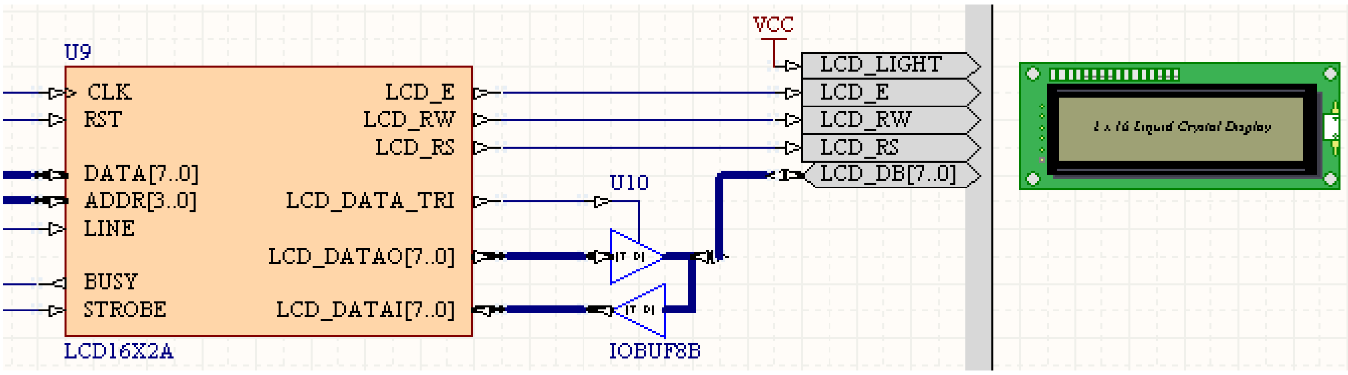

Figure 3. Providing the control interface between a microcontroller and the on-board LCD panel.

The LCD Controller is implemented as a Finite State Machine, with an LCD initialization sequence that is performed after a device reset.

The Controller's maximum operating frequency is 80 MHz.

Operation

The following sections provide information with respect to the Controller’s operation, including resetting the device, LCD panel initialization and how to display information on the LCD panel, via the Controller.

Resetting the Controller

When the RST signal goes active, the BUSY signal goes active as well, to indicate that the Controller is in the reset state and cannot be used at the current time.

When the RST signal goes inactive, the Controller will execute the standard initialization sequence for KS0066U-compatible LCD panels.

LCD panel Initialization Sequence

The typical initialization sequence is as follows[2]. At all times during the initialization period, the BUSY signal will be active. Consequently, all data written to the Controller will be ignored.

1. **Define Function Set** – in this state the LCD Controller is programmed to set the data length, the number of lines to be used and what type of fonts to be used. The Controller sets these values to 8-bit, 2 lines and 5x8 pixels respectively.
2. Wait for 39us.
3. **Display ON/OFF Control** – in this state, the LCD Controller is programmed to set display, cursor and blink of cursor settings, as follows:

   Display – ON

   Cursor – OFF

   Blink of cursor – OFF.
4. Wait for 39us.

[2] Timing values are with respect to the 5.0V version of the LCD panel that comes with the NanoBoard.

5. **Display Clear** – in this state, the ASCII code 20h (space character) is written to the LCD panels' DDRAM (data memory) and the character address is changed to 0h (top-left-most character on the panel). After this stage, there is a space character at all addresses, so the LCD panel gets cleared.
6. Wait for more than 1.53ms.
7. **Entry Mode Set** – in this state, the Controller sets the cursor to decrement (move to the right after each character insertion) and disables shift of the entire display.
8. Wait for 39us.
9. **Initialization Done** – the BUSY signal goes inactive. The LCD panel is now ready to use.

Displaying Data on the LCD Panel

When the BUSY signal is inactive, the LCD Controller is awaiting data and is ready to display a new character on the LCD panel. This is the 'Wait on Data' state.

To initiate a display operation, the ADDR, DATA and LINE signals must be stable and the STROBE port then used to trigger the operation.

When the STROBE signal goes active, the following transfers take place:

- The character position within the LCD panel, defined by the content of the 4-bit ADDR bus, is loaded into the Controller's Address register (ADDREG).
- The line on the panel in which the character is to be displayed, defined by the level of the LINE signal, is loaded into the Controller's Line Select register (LINESEL).
- The actual character to be displayed, defined by the 8-bit content of the DATA bus, is loaded into the Controller's Data register (DATREG).
- The Controller's FSM then enters the 'Set Address' state and the BUSY signal changes to active as well. The contents of the ADDREG and LINSEL registers are passed to the LCD panel.
- The Controller then waits for 39us, after which time, the FSM enters the 'Show Char' state. Whatever is in the DATREG register at this time is written to the LCD character memory.
- The Controller then waits for the LCD panel to finish its operation and then returns to the 'Wait on Data' state, ready for the next character to be displayed. The BUSY signal is changed to inactive at this time. Typically, this takes about 43us.

Useful Tips

It is required that the ADDR, DATA and LINE signals are stable during the time when the STROBE signal is active. The reason for this is that these signals are latched on the rising edge of the CLK signal when STROBE is active, so if they are changed while keeping STROBE active, the behavior might be undefined.

To avoid such a situation, you could strobe the Controller with one clock cycle pulse. This will latch data on the clocks rising edge and trigger the Controller to perform a display operation.

If the STROBE signal goes active and the Controller is in an internal state (BUSY signal is active), you can change the values on the ADDR, DATA and LINE signals, since the Controller will use the internal data that was latched when the STROBE signal was last active and BUSY was inactive.

Revision History

Date	Version No.	Revision
12-Feb-2009	1.0	Initial document release
30-Aug-2011	-	Updated template.

Software, hardware, documentation and related materials:

Copyright © 2011 Altium Limited.

All rights reserved. You are permitted to print this document provided that (1) the use of such is for personal use only and will not be copied or posted on any network computer or broadcast in any media, and (2) no modifications of the document is made. Unauthorized duplication, in whole or part, of this document by any means, mechanical or electronic, including translation into another language, except for brief excerpts in published reviews, is prohibited without the express written permission of Altium Limited. Unauthorized duplication of this work may also be prohibited by local statute. Violators may be subject to both criminal and civil penalties, including fines and/or imprisonment.

Altium, Altium Designer, Board Insight, DXP, Innovation Station, LiveDesign, NanoBoard, NanoTalk, OpenBus, P-CAD, SimCode, Situs, TASKING, and Topological Autorouting and their respective logos are trademarks or registered trademarks of Altium Limited or its subsidiaries. All other registered or unregistered trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.