LCD Controller Using MAX II CPLDs

December 2007, version 1.0 Application Note 497

Introduction

This document details the implementation of an LCD controller in an Altera® MAX® II CPLD. The example display uses an Optrex 16 x 2 Dot Matrix LCD module, such as the popular SC1602D device. The controller receives its display signal from a microprocessor or a micro controller and translates it into commands that are understood by the LCD module. MAX II devices possess the industry's only user flash memory (UFM), which is used by the controller to display a message upon power on. This eliminates the need for an external memory. Also, no external clock signal is required by the controller as it utilizes the CPLD's internal UFM oscillator for its clocking needs. These unique features make the MAX II CPLD the perfect target device for such a controller.

LCDs

Liquid crystal displays (LCDs) are thin, flat display devices made up of any number of color or monochrome pixels arranged in front of a light source or reflector. These displays have become an indispensable part of most of electronic devices. Some features that have made LCDs popular are:

- LCDs provide applications with a useful interface that allows debugging capabilities and provides the application with a more professional look
- LCDs use very small amounts of electric power; they are very suitable for battery powered devices
- LCDs are slim and use small amounts of space compared to other display types

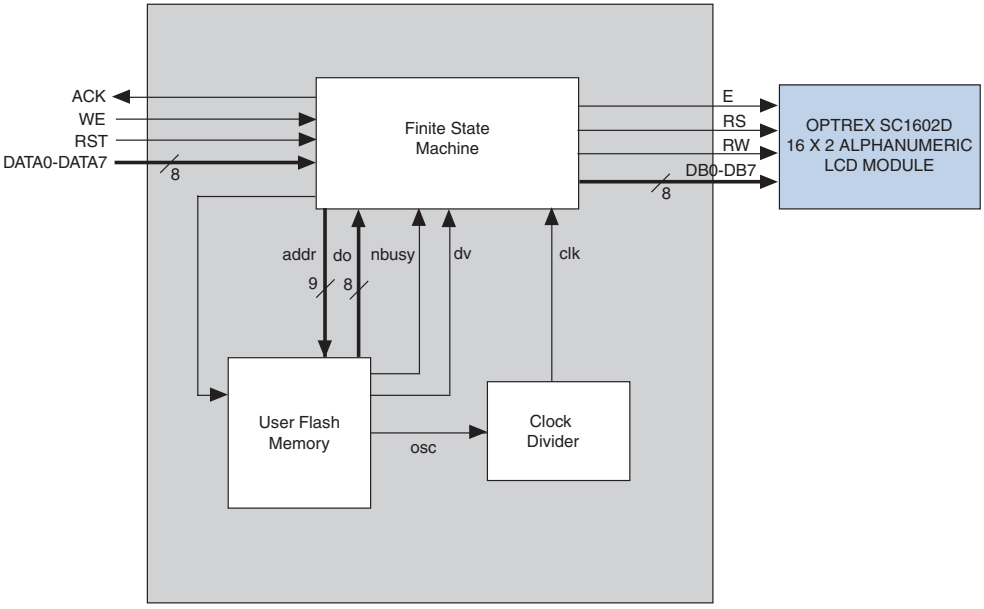
Most LCD modules are equipped with a generic interface, on-board controllers, and an on-board driver. Thus, using an LCD to enhance the visual appeal of any application has become a lot simpler. Some devices that use LCD displays are wrist watches, calculators, laptops, PDAs, cellular phones, home electronics, and toys.

LCD Controller

You can easily integrate this controller design into any application to provide a suitable display unit. It meets the timing constraints of the LCD by generating the necessary timing signals and simultaneously off loads the microprocessor and controller from the details of the LCD, thereby allowing the processor and the controller to carry out their other operations efficiently.

Figure 1 shows a block diagram for the LCD controller. The three main modules inside the LCD controller are the finite state machine (FSM), the user flash memory (UFM), and the clock divider module. The interfacing signals to the LCD module are E, RS, RW, and DB0-DB7. This design successfully interfaces the processor and the LCD module through the controller. For more information on the signals, refer to Table 1.

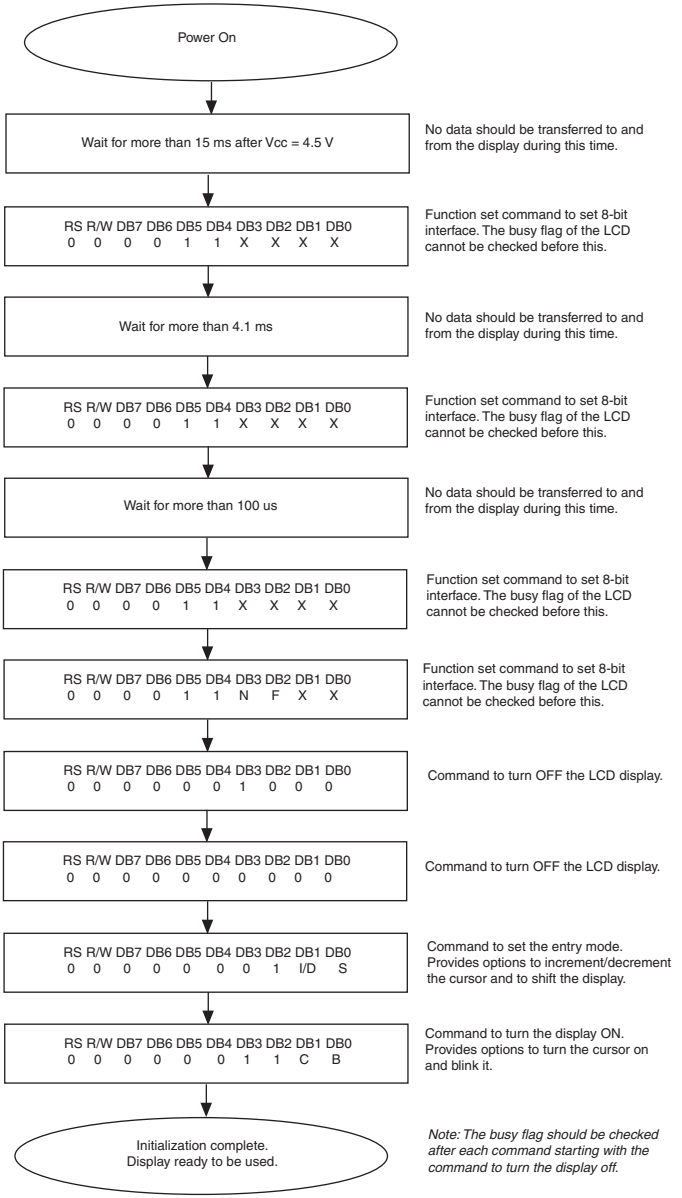
Figure 1. LCD Controller



Initializing the LCD Module and the Finite State Machine

The FSM is made of eight different states. It is used to initialize the LCD, display a message after initialization is complete, and to write to and read from the LCD. The initialization steps for the LCD module are shown in Figure 2.

Figure 2. LCD Module Initialization Flow Chart



To simplify initialization, a delay of 15 ms is provided after each command is sent (instead of checking the status of the busy flag). However, while writing data into the LCD module, the status of the busy flag is checked after each data write. This makes the writing procedure faster.

For more information about initializing the LCD module and the various commands supported by it, see the Optrex LCD Manual at the following link:

http://www.apollodisplays.com/pdf/dmcman.pdf

User Flash Memory

The UFM inside the MAX II CPLD is used to store the ASCII values of the characters to be displayed as soon as the controller is turned on. It is instantiated using the UFM megafunction. Parallel interface programs the UFM with the help of a hexadecimal (Intel_Format) file (**.hex**). For more information on creating a **.hex** file using the Quartus® II software, refer to the following link:

http://www.altera.com/support/software/nativelink/quartus2/glossary/def_rif.html

For more information about instantiating the UFM using the MegaWizard® Plug-In Manager, refer to the *Using User Flash Memory in MAX II Devices* chapter in the *MAX II Device Handbook*.

Clock Divider

The clock divider module reduces the frequency of the internal oscillator output so that the timing requirements of the LCD module are met and the controller performance is satisfactory. Table 1 lists signal descriptions for the LCD controller (MAX II CPLD) block diagram shown in Figure 1.

Table 1. Signal Descriptions for the LCD Controller (MAX II CPLD) (Part 1 of 2)

Signal	Description
DB0-DB7	8-bit bi-directional bus that carries data and commands to and from the LCD module.
RW	Read and write signal to distinguish between a read and a write operation. • RW = 0: Write Operation • RW = 1: Read Operation

Table 1. Signal Descriptions for the LCD Controller (MAX II CPLD) (Part 2 of 2)

Signal	Description
RS	Register select signal. This is used to choose between the data and the command registers of the LCD module. • RS = 0: Command Register • RS = 1: Data Register
E	Enable strobe signal. A negative edge on this signal latches the data on the bi-directional bus into the data or command register, and vice versa.
DATA0-DATA7	8-bit bus coming from the processor and controller that carries the ASCII value of the character to be displayed on the LCD module.
RST	Reset signal. This signal is used to reset the controller and initialize the LCD module again.
WE	Write enable signal. This signal indicates the beginning of a write cycle. It should be set to **low** to enable writing to the LCD module.
ACK	Acknowledge signal. This signal indicates that the data has been successfully sent to the LCD module and is ready to receive the next data. Any data send before this signal goes low will be ignored by the controller.
addr	9-bit address bus used to select a particular memory location of the UFM.
nread	Set this signal to **0** in order to read the contents of the memory location pointed to by the address bus.
do	Data output signal. An 8-bit data bus which holds the higher order 8-bits of the 16-bit memory location pointed to by the address bus.
dv	Data valid signal. Signal that indicates the data on the 8-bit data bus is valid and can be read.
osc	Output of the internal oscillator of the UFM block. Outputs a signal frequency of 5.5 MHz.
clk	Output of the clock divider block. Reduces the frequency of the oscillator output signal to approximately 43 kHz.

Implementation

You can implement this design with an EPM240G or any other MAX II CPLD and an Optrex SC1602D 16 x 2 alphanumeric LCD module. Use this design source code and allocate the data bus and control lines (input and output) to the general purpose I/O (GPIO) lines of the MAX II CPLD. Use the internal user flash memory to store any default message you wish to display on the LCD module upon power on. Use the memory editor in the Quartus II software to create a memory initialization file to store the default message in the LCD module.

The following details the implementation of this design on the MDN-B2 demo board. Table 2 lists the EPM240G pin assignments for this design example.

Table 2. Design Implementation Using the MDN-B2 Demo Board

EPM240G Pin Assignments			
Signal	**Pin**	**Signal**	**Pin**
DB[0]	pin 53	DB[4]	pin 1
DB[1]	pin 98	DB[5]	pin 96
DB[2]	pin 52	DB[6]	pin 92
DB[3]	pin 97	DB[7]	pin 95
data[0]	pin 55	data[4]	pin 61
data[1]	pin 56	data[5]	pin 66
data[2]	pin 57	data[6]	pin 67
data[3]	pin 58	data[7]	pin 68
E	pin 99	rst	pin 77
RS	pin 100	ack	pin 69
RW	pin 54	we	pin 81

Unused pins are assigned as **input tri-stated** in the Quartus II software's device and pin option settings prior to compilation.

Design Notes

To demonstrate this design on the MDN-B2 demo board, complete the following:

1. Download the design on to the MAX II CPLD through the JTAG header JP5 and a conventional cable (ByteBlaster™ II, USB-Blaster™). Programming requires the demo board to be switched on (slide switch SW1). After programming, turn off the power.

2. Ensure that the memory file **lcd_new.hex** (included with the source code files) is placed in the Quartus II software's project library.

3. Connect the LCD module to the board using the supplied interconnect cable. Connect the 14-pin socket of the cable to CN1 on the LCD module (the red mark meets pin1 on CN1 of the demo board) and connect the 20-pin socket to JP8 on demo board (the red mark on socket meets pin 1 on JP8 of the demo board).

4. Use the Optrex SC1602D LCD module (which is also supplied with Altera's NIOS® II development kit 2C35N [Cyclone II version]).

5. Supply the LCD module with the required 5 V power supply through the DC socket on the interconnect cable. This DC socket has conventional polarity (the inner pin is positive). You can also use 3xAA or AAA batteries to power up the LCD module.

6. Connect the 5 V power supply to the LCD module first. Then connect the interconnect cables as mentioned in step 4.

7. Turn on the power supply to the MDN-B2 demo board using the slide switch SW1.

8. Observe the default message on the LCD module. This message is pre-programmed in the UFM of the MAX II CPLDs using the **lcd_new.hex** file. The controller reads this message only once upon power-on.

9. Use the reset push button SW6 on the demo board to reset the LCD module. Note that after reset, the cursor on the LCD is in its first character position on the first row and is blinking.

10. For the purpose of this exercise, you can enter any ASCII data corresponding to a character you wish to display using the SW5 switch (the 8-way dual in-line package (DIP) GPIO input switch on the demo board). The switch sets a **logic 0** when ON, and a **logic 1** when OFF. Switch #1 is the LSB on this ASCII data byte.

11. Press button SW8 on the demo board to write enable the controller, which enables it to take in fresh data set by the SW5 switch. The demo board takes in new data as long as the SW8 button is pressed, and at a rate of approximately 4.25 kHz. It updates all characters on both rows almost immediately after pressing the SW8 button.

This design is made to work specifically with Optrex's SC1602D LCD module. This LCD module has non-consecutive addresses on the last character of the first row and first character of the second row on the LCD. This design makes suitable alterations to match this arrangement.

Source Code

This design has been implemented in Verilog and successful operation has been demonstrated using the MDN-B2 demo board, as referenced in the documentation. The source code, test bench, and complete Quartus II project are available at:

www.altera.com/literature/an/an497_design_example.zip

Conclusion

MAX II CPLDs are an excellent choice to implement LCD controllers. Their low power, easy power-on feature, and built-in user flash memory make them ideal programmable logic devices to implement LCD controller designs.

Referenced Documents

This application note references the following documents:

- http://www.apollodisplays.com/pdf/dmcman.pdf
- http://www.altera.com/support/software/nativelink/quartus2/glossary/def_rif.html
- *Using User Flash Memory in MAX II Devices* chapter in the *MAX II Device Handbook*.

Additional Resources

The following are additional resources for this application note:

- MAX II CPLD home page:
  http://www.altera.com/products/devices/cpld/max2/mx2-index.jsp
- MAX II Device Literature page:
  http://www.altera.-com/literature/lit-max2.jsp
- MAX II Power-Down Designs:
  http://www.altera.com/support/examples/max/exm-power-down.html
- MAX II Application Notes:
  AN 428: MAX II CPLD Design Guidelines
  AN 422: Power Management in Portable Systems Using MAX II CPLDs

Document Revision History

Table 3 shows the revision history for this application note.

Table 3. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
December 2007, v1.0	Initial release.	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support:
www.altera.com/support
Literature Services:
literature@altera.com