XAPP904 (v1.0) August 22, 2005

CoolRunner-II Character LCD Module Interface

Summary

CoolRunner^{TM}-II CPLDs can be used to control dot-matrix liquid crystal display (LCD) modules. The low-power characteristics of LCD modules make them an ideal complement to the CoolRunner-II family. These displays typically require 3.3V signals. However, this is not of concern because CoolRunner-II devices are 3.3V tolerant. Thus, it is possible to achieve ultra-low power at a 1.8V core voltage while using 3.3V at the I/O.

Introduction

Character LCD Module Background

There are many manufacturers of dot matrix LCD modules. However, most of these displays are similar. They all have on-board controllers and drivers capable of displaying alpha numerics and a wide variety of other symbols (including Japanese "Katakana" characters). The internal operation of LCD controller devices is determined by signals sent from a central processing unit (in this case, a CoolRunner-II CPLD).

These signals include:

1. Register Select (RS)
2. Read/Write (R/W)
3. Data Bus (DB7~DB0)
4. Enable Strobe (E)

Character LCD Instructions

Figure 1 shows the timing requirements for writing data to the LCD (this design does not perform read operations for the sake of simplicity). Notice that data is written with respect to the falling edge of the Enable Strobe (E) signal. The CoolRunner-II CPLD is responsible for making sure that LCD Module timing requirements are met. More specific timing requirements can be found in the LCD data sheets.

Figures 1 and 3 originated at Seiko, and were found at:

http://www.seiko-usa-ecd.com/lcd/products/char_mods/pdf/instructions.pdf

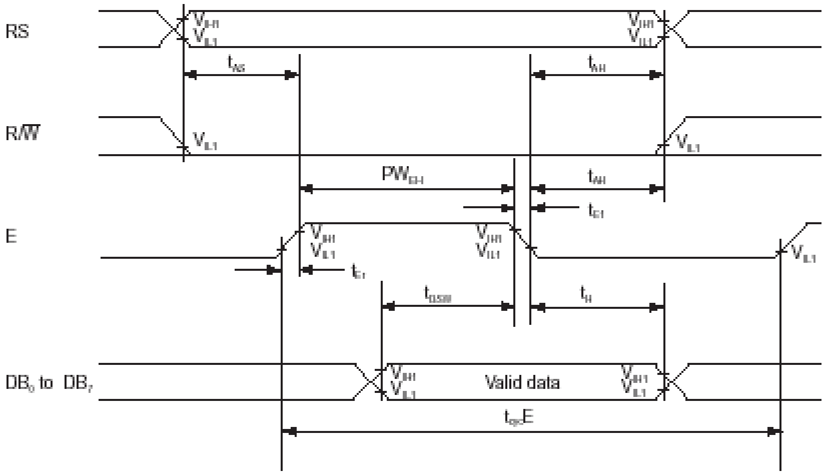

Figure 1: **Data Write from CPLD to LCD**

CPLD Implementation

Figure 2 shows a block diagram of an LCD Controller on a CoolRunner-II CPLD.

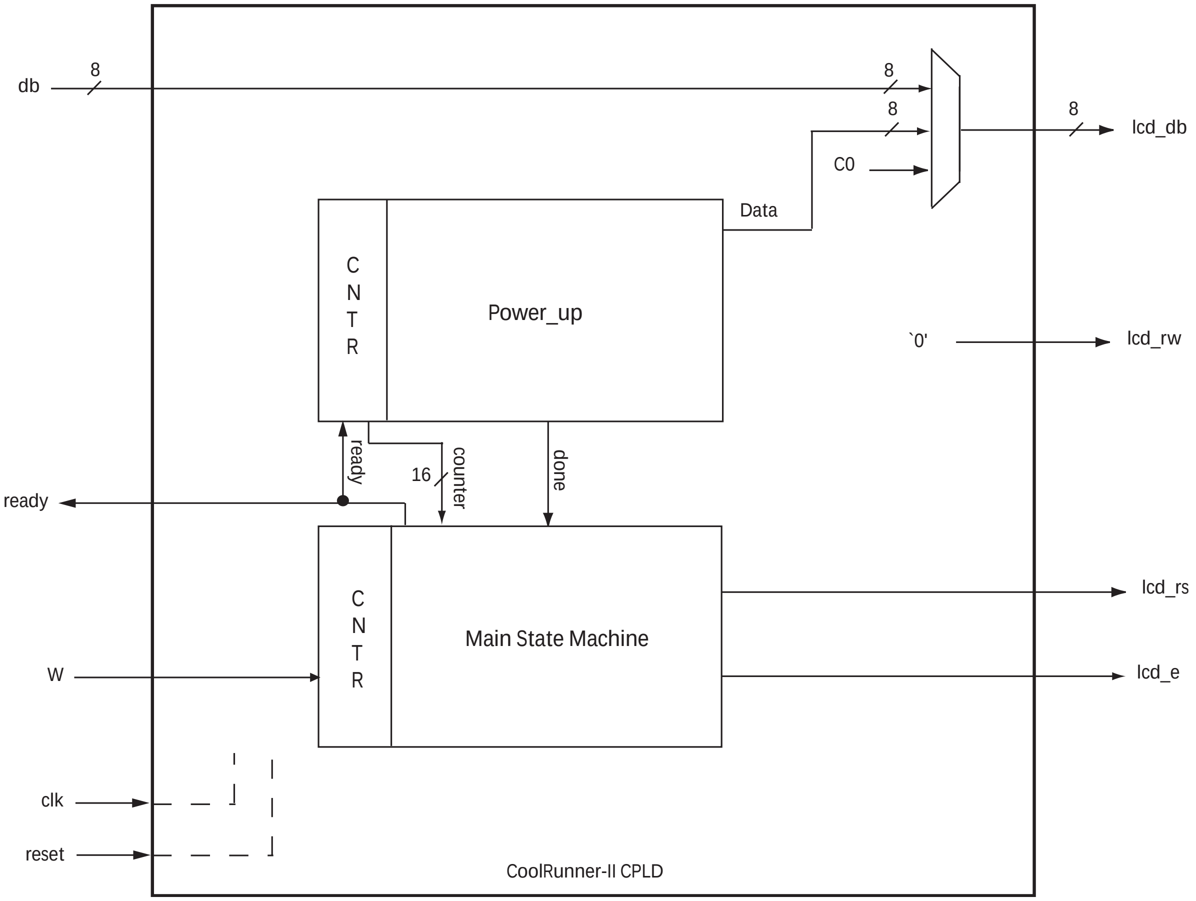

Figure 2: **LCD Controller Block Diagram**

The LCD Controller implementation is comprised of two state machines - a Power_Up State Machine, and an LCD Controller Main State Machine. These two blocks are discussed in the following sections.

Table 1 defines the primary inputs/outputs of this LCD Controller design.

Table 1: **Signal Descriptions**

Signal Name	Port Specification	Description
Reset	input	Reset Signal (Active High)
DB[7:0]	input	8-bit Data bus.
W	input	Write strobe (Active High)
Ready	output	Ready Signal - Asserted by CPLD to signal when more data can be sent (Active High)
Clk	input	Clock input (This design assumes 1.8MHz clock)
LCD_RS	output	LCD Register Select
LCD_RW	output	LCD Read/Write
LCD_E	output	LCD Enable Strobe
LCD_DB[7:0]	output	8-bit LCD Data Bus

Power-Up State Machine

LCDs require an initialization procedure to be executed each time the module is turned on or reset. The Power-Up State Machine is designed to automatically take care of this procedure by sending a sequence of hex codes from the CPLD to the LCD module. Upon completion, the LCD cursor will be enabled, the display will be cleared and the LCD module will be set for auto-increment mode. (See Figure 3)

During this sequence, the CPLD waits for 15 ms to allow the LCD to power up, then sends 0x38, 0x06, 0x0E and 0x01. This initialization sequence is automatic, and will also occur upon every power-up or reset. All timing requirements will be met, assuming a 1.8 MHz external clock frequency.

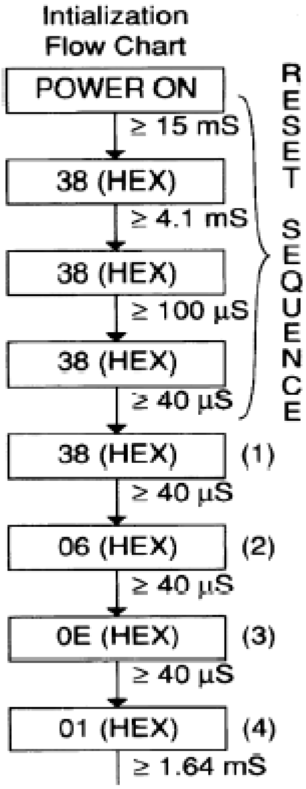

Figure 3: **LCD Module Initialization Sequence**

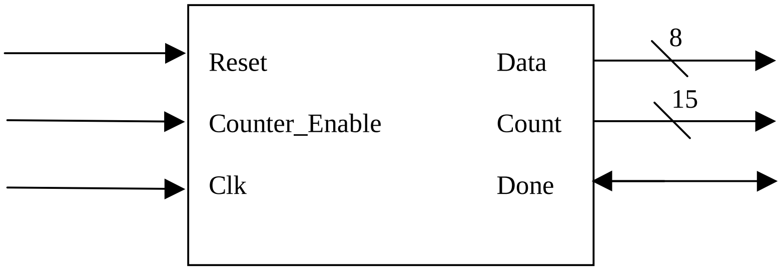

Figure 4: **Power_Up Module Block Diagram**

Figure 4 shows a block diagram of the Power_Up module. This Power_Up module utilizes a terminal counter to insure that data is held for the appropriate amount of time, as specified in Figure 3. As shown, the largest wait time required by the LCD is 15 ms (during the Power_ON sequence). Thus, to simplify the design, data is held for 16 ms in every subsequent block. Since this design assumes a 560 ns clock period (1.8 MHz), a terminal count value of 30,000 is used. Alter this terminal count if a different clock frequency will be used.

An internal 'Done' signal is output by this Power_Up module to let the main LCD State Machine know when the Power_Up sequence has completed.

LCD Controller Main State Machine

After the LCD has been initialized, control is released to the LCD Controller Main State Machine. The CPLD will drive the ’Ready’ line high in order to signify that the power up state machine has completed and that it is ready to accept data to be written to the CPLD. Hence, a CPU, or some other upstream device, sends a data byte corresponding to a specific LCD character to the CPLD along with a Write strobe. The CPLD will latch the data byte on the falling edge of the Write strobe, drive the ’Ready’ signal low, and drive the appropriate LCD signals in order to make the character appear. Input Data will be ignored by the CPLD until the ’Ready’ signal is high again.

Figure 5 shows the complete sequence of events, starting from the power up initialization process. As can be seen, the CPLD first sends the initialization codes 0x00, 0x38, 0x06, 0x01 followed by 0x80. Immediately after, the ’Ready’ line is asserted, and the CPU begins sending data on ’DB’ on every edge of the ’W’ line. The CPLD latches the data on every rising edge of W, and drives the ’LCD_DB’, ’LCD_RW’, ’LCD_RS’ and ’LCD_E’ pins accordingly in order to make the character appear. The CPU continuously monitors the ’Ready’ line and sends data only when Ready is high.

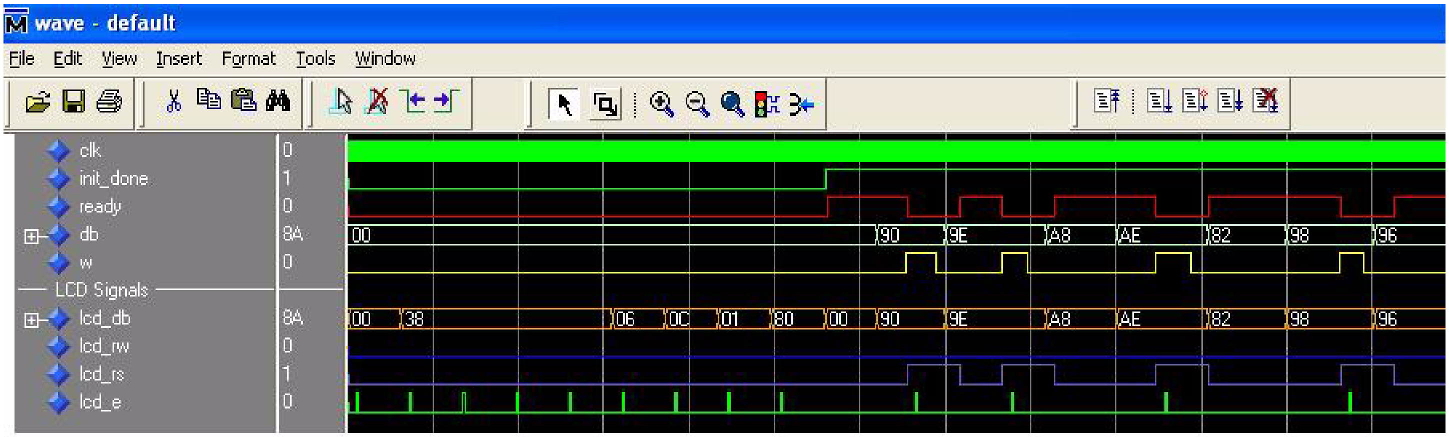

Figure 5: **Signal Sequence of Events**

Table 2: **CPLD Resource Use Summary**

Resource Summary

This design uses a total of only 40 macorcells, allowing it to fit into an XC2C64A device.

```
*************************  Mapped Resource Summary  ********************
Macrocells     Product Terms    Function Block   Registers     Pins
Used/Tot       Used/Tot         Inps Used/Tot    Used/Tot      Used/Tot
40/64(62%)    64/224(29%)    61/160(38%)    26/64(41%)   23/33(70%)
```

Additional Information

CoolRunner-II Data Sheets and Application Notes

Conclusion

The LCD Interface presented in this application note is simple and straightforward. Additionally, CoolRunner-II devices are ideal candidates for driving LCDs due to their low cost, low power and ease of use.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
08/22/05	1.0	Initial Xilinx release.