Interfacing the fido1100 Microcontroller to LCD Display Modules

An Innovasic Semiconductor Application Note
fido1100 Application Note 160

Version 1.1

December 2006

Table of Contents

Abstract

This application note address how the **fido1100** can be used to interface to an LCD display.

Introduction

LCD modules can be added to electronic applications to provide valuable human-interface information to the user. Usually, small systems rely on a handful of LEDs to provide status that can only be interpreted by a trained eye. LCD modules, by comparison, are only limited by the software programmer's patience to provide accurate, unambiguous information to the user.

Here are some suggestions for the type of information that could be displayed:

- Automatic test procedure (ATP) and diagnostic status
- Exception information
- User-specific text
- Exceeded analog value limits on the A-D converter

Available modules provide many possibilities, from simple 7-segment numeric displays to pixel matrix alpha-numeric displays to full color graphics displays. Fortunately, LCD modules have controllers that allow relatively easy interfacing to a host microcontroller. For the purpose of this application note, we will discuss how to drive a pixel matrix alpha-numeric display controller, specifically the Samsung S6A0069.

Approach

Interfacing Details

While it is tempting to attach the MPU interface of the S6A0069 to the main external bus and treat it like a peripheral, a quick inspection shows the interfaces are incompatible. The main external bus has the topology of a PowerPC bus, while the S6A0069 has a Motorola 68XX bus topology. Also, the 26A0069 bus interface is very slow when compared to the more modern devices that will be attached to the main external bus, potentially reducing the bus bandwidth and processor performance. Fortunately, the **fido1100** UIC allows the user to mimic a 68XX type of bus cycle.

Electrical Characteristics

The S6A0069 can operate at from 3.0 to 5.0 VDC. While the **fido1100** I/O operates at LVTTL levels and is 5.0 VDC tolerant, the S60069 require CMOS I/O levels which are dependent on the supply voltage. Therefore, without level translation, the **fido1100** can only interface to an S6A0069 device operating at 3.3 VDC.

Signaling

The S6A0069 has a CPU interface that is reminiscent of the Motorola 68XX bus. This 11 pin bus easily fits within the 18 available pins of a single UIC. With the UIC operating on a 15-20 ns bus cycle, and the S6A0069 bus working on the order of 500 ns between signal transitions, the **fido1100** UIC can easily generate sequenced outputs to the S6A0069 and oversample the inputs from the S6A0069 to make branch decisions. The table below details the bus interface:

S6A0069 Signal	***Suggested fido1100 Signal***	***Comment***
DB6-DB0	uic0_0 to uic0_6	Bidirectional data bus for message data and display instructions
DB7	uic0_7	Data input during writes, “busy” flag during reads
E	uic0_8	Read/write enable signal
R/W	uic0_9	Read/write command (must bracket E strobe)
RS	uic0_10	Data/Instructions command – effectively selects between two different registers

Software Considerations

With the hardware connectivity addressed, let's turn to the software side. There are two items required in software to allow useful communication between the S6A0069 and the **fido1100**:

- UIC firmware
- **fido1100** CPU software

A firmware program for the UIC must be written to properly handle the character data and display instructions between the UIC and the **fido1100** CPU, and the signals going between the selected UIC and the S6A0069. The program will recognize that data and/or instructions are available in the UIC FIFOs, then initiate the proper bus sequence to transfer them to the S6A0069. It appears that each bus cycle sequence contains about 5 transitions of data and control signals that can be precisely spaced out in time using the UIC loop counter. The only other subroutine required will simply wait for the S6A0069 busy flag (DB7) to fall before initiating the next bus cycle. This short list of simple routines will operate out of a single thread in the UIC.

Finally, software for the **fido1100** must be written to act as a driver for the LCD controller, deciding what messages should be sent when, then properly queuing the data/instructions through the PMU to the selected UIC.

Conclusions

The flexibility of the **fido1100** and its UIC blocks will permit easy construction of the relatively simple interface to the Samsung S6A0069 LCD controller. Since the UIC can be set up to handle batch writes to the display, there will be no waiting for the processor. This is a distinct advantage over conventional bus interfaces. Thus, any **fido1100**-based system requiring human interface feedback should be able to utilize commercial, character-based LCD modules. The same principles can also be extended to graphics LCD displays utilizing the power of the UICs to provide the control.

THANK YOU

Thank you for taking the time to review this application note. We hope you have found the information included in this application useful and east to understand. Please feel free to contact the Innovasic Support Team any time with questions or comments.

Innovasic Support Team
3737 Princeton NE
Suite 130
Albuquerque, NM, 87107

(505) 883-5263

support@innovasic.com
http://www.innovasic.com