Interfacing to IO Devices via the Static Memory Controller on LH7xxxx Card Engines

Application Note 303

Bruce Rovner

Logic Product Development

Published: August 2005

Abstract

This document provides software examples for interfacing to IO devices via the Static Memory Controller on LH7xxxx card engines.

REVISION HISTORY

REV	EDITOR	DESCRIPTION	APPROVAL	DATE
A	Bruce Rovner	Initial Release	HAR	8/23/05

1 Introduction

Logic Product Development's *LH7xxxx IO Controller Specification* documents were written before we were aware of Sharp's application note for its LH7xxxx series processors: "Interfacing the Static Memory Controller with I/O Devices.**"** This document can be located online at: http://www.sharpsma.com/part.php?PartID=177.

Due to the information contained in Sharp's application note, Logic added a barrier function to device accesses in production-ready software offerings for the Sharp processors (e.g. LogicLoader, LoCE Windows CE BSP, etc…). Barrier functions are applied between each access to devices located in memory regions controlled by Sharp's static memory controller.

1.1 Barrier Function Information

The barrier function is an important feature because many asynchronous devices require that the chip select line (/CS) toggle between each access when reading and/or writing to the device. These devices include all CPLD registers, the compact flash interface, the flash devices (when using the CFI to check status and program them), the SMSC 91C111 Ethernet chip, all devices accessed via the SPI interface in the CPLD, and any external devices on the static memory controller bus. This is done in order to make sure that the chip select line (/CS) toggles.

For further information about the IO devices on your product, please refer to the corresponding Schematic, Hardware, and IO Controller Specification documents. The most recent revisions of these documents can be located online at: https://www.logicpd.com/auth/login.php.

2 Barrier Function Examples

The following software examples demonstrate how to communicate to IO devices via the Static Memory Controller of the System on Chip (SOC) for LH7xxxx card engines.

2.1 CPLD registers:

An example of a barrier function used with CPLD registers.

```
void barrier_cpld(void)
{
// flash is guaranteed not to be cached, and to be in
// a different region from the CPLD, and will force the LH7XXXX SMC
// to toggle the chip select lines.

volatile u_int tmp = *(volatile u_int *) MEMAP_BASE_FLASH;
tmp = 1;
}
```

2.2 Flash CFI interface:

An example of a barrier function used with flash CFI devices.

```
void barrier_flash_cfi(void)
{
// The CPLD is guaranteed not to be cached, and to be in
// a different region from flash, and will force the LH7XXXX SMC
// to toggle the chip select lines.

volatile u_int tmp = *(volatile u_int *) CPLD_ CODE_REVISION_REG;
tmp = 1;
}
```

2.3 SPI register in the CPLD:

An example of a segment of code using a barrier function that communicates with the SPI register in the CPLD.

```
⋮
barrier_cpld();  //force /CS toggle

// load SPI data into shift register
REG16(dev->data_reg) = src[i];

barrier_cpld();  //force /CS toggle

// write out the command
REG16(dev->control_reg) = (CPLD_CE_SPI_CNTL_LOAD_REG | dev->chip_select);

barrier_cpld(); //force /CS toggle
⋮
```

3 Summary

Logic added barrier functions appropriate to each device as a result of the information published in Sharp's application note: "Interfacing the Static Memory Controller with I/O Devices."

Please refer to the Sharp website for their Application Note and background information. In addition, refer to the corresponding Schematic, Hardware, and IO Controller Specifications for further information about the IO devices on your product.