



Using the Smart Card Interface on the LH7A400-10 Card Engine

Application Note 237

Jemiah Aitch, Mike Aanenson

Logic Product Development

Published: July 2004

Abstract

This application note briefly explains how to design a Smart Card interface. It includes CPLD bit control and address line 25 information. Applicable for LH7A400-10 Card Engine CPLD Revisions 3.7 and later.

This file contains source code, ideas, techniques, and information (the Information) which are Proprietary and Confidential Information of Logic Product Development, Inc. This information may not be used by or disclosed to any third party except under written license, and shall be subject to the limitations prescribed under license.

No warranties of any nature are extended by this document. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase or lease equipments. The only warranties made by Logic Product Development, if any, with respect to the products described in this document are set forth in such license or agreement. Logic Product Development cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

Logic Product Development may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering the subject matter in this document. Except as expressly provided in any written agreement from Logic Product Development, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

© Copyright 2002, Logic Product Development, Inc. All Rights Reserved.

REVISION HISTORY

REV	EDITOR	DESCRIPTION	APPROVAL	DATE
A	Jemiah Aitch, Mike Aanenson	Release	HAR	07/12/04

1 Smart Card Interface: Issues and Work Arounds

Applicable for LH7A400-10 Card Engine CPLD Revisions 3.7 and later.

1.1 Multiplexing the Smart Card IO Line and Bit 25 Issue

The LH7A400 Processor multiplexes a Smart Card IO line and bit 25 of the address bus. As a result, if the Smart Card is being used, no off-board memory mapped peripherals can be used simultaneously on the LH7A400-10 card engine.

Work Around: Setting the Multiplexed Smart Card IO Line and Bit 25

If you are using Logic's WinCE Smart Card Driver, you can skip this step. Otherwise, it is necessary to both properly control the Smart Card interface on the LH7A400 processor and control the card engine circuitry for the interface setup. The card engine circuitry may be controlled for proper Smart Card initialization by making the following bit manipulation per the LH7A400-10 IO Controller Specification:

- If using the Smart Card Interface, set the SCACT bit to enable the Smart Card and disable address line 25.
- If not using the Smart Card Interface, set the SCACT bit to disable the Smart Card and enable address line 25.

1.2 LH7A400-10 Silicon Issue

There is a known issue with the silicon on the A400's SCI module: the speed of the smartcard clock is limited to 5MHz, and the addition of a pull-up to the Smart Card clock net is required.

Work Around: Add a Pull-up to the Smart Card Clock Net

In order to use the Smart Card Interface, a pull-up must be added to the baseboard. According to Sharp, the value of the pull-up should be approximately 1K ohms for 1MHz and 680 ohms for 5MHz.

2 Using the Smart Card Interface

After a pull-up to the baseboard has been added, you can use a Logic's Smart Card Interface WinCE driver.

This driver can be located on our downloads site at: <http://www.logicpd.com>

2.1 The WindowsCE Driver: Implemented in the Windows Driver Model

The LH7A400-10 Smart Card driver for WindowsCE is implemented in the Windows Driver Model. The only functions that are exported, however, include:

- SMC_Init
- SMC_Deinit
- SMC_Open
- SMC_Close
- SMC_IOControl
 - *ReaderPower
 - *SetProtocol
 - *Transmit
 - *CardTracking
 - *VendorIoctl
- SMC_PowerUp
- SMC_PowerDown