Control System Design Using PBD Techniques

Emanoel Xavier, Cristiano Araújo and Edna Barros

Centro de Informática, UFPE

Caixa Postal 7851 - Cidade Universitária

50732-970 Recife – PE Brazil

embx,cca2,ensb@cin.ufpe.br

ABSTRACT

In this work is presented the modeling of a platform based on the 8051 microcontroller and a high performance serial bus. The primary objective of the work is to provide enough knowledge to support the automatic building of such platforms and its performance analysis. In order to evaluate the process of building a platform it has been chosen a medium complexity system to be implemented. This system was built and simulated then, it was possible to validate its functionality and to get the simulation times of its bus communication.

1. INTRODUCTION

With the increase in the complexity of integrated systems, the use of platforms has become more and more common among the designers [1][2]. These platforms allow the reuse of hardware and software components, as a consequence the development time of complex system is drastically reduced.

Most of the platforms used in commercial and academic systems are based in a small set of processor architectures and interconnection structures, reducing the design space available for the designers. Some commercial tools like CoWare allow some flexibility but the cost of the tools is prohibitive for academic use. The automation of the process of building processor based platforms needs a deep knowledge of how to build them. Building the platform is not enough, it is also necessary to obtain performance results that can drive the designer decision regarding the platform.

In this work is implemented a medium platform in order to evaluate the complexity and challenges in the task of building platforms automatically. The platform can be easily extended both in the number of master and slave devices. It is composed of a microcontroller connected to a serial bus that allows connections to other devices that have OCP-IP protocol [4].

The rest of the paper is structured as follows. Section 2 presents a description of the platform structure. Section 3 describes the microcontroller interface. In section 4 are described the bus interfaces. In section 5 the bus used is described. An access control system is shown as a case study in the section 6. Finally, some conclusions are given in section 7.

2. THE BASE PLATFORM

The design of the platform has been divided in two parts. The first is a base platform. It consists of a high performance serial bus based on the I^2C communication protocol [7], OCP-IP compliant interfaces and a 8051[8] microcontroller core. Figure 1 depicts the base platform. It shows the 8051, the serial bus and the OCP- Serial bus interfaces.

One pre-requisite of the platform is that all master and slave devices connected to the bus have OCP-IP compliant interfaces. The OCP-Serial interfaces allow the easy connection of devices, which can be masters or slaves, to the bus. In this way, the 8051 is an example of a master device to this platform. It is used to run the control software of the system. Microcontroller, interfaces and bus are constructed in SystemC RTL [5]. The platform was built in SystemC 2.0[6].

In a second phase the base platform has been extended to implement an access control system. A total of four slave devices have been added, together with software support in the microcontroller. The access control system is described in more detail in section 0.

3. 8051 MICROCONTROLLER

In order to integrate the 8051 microcontroller into the platform it was necessary to implement an OCP-IP master interface on it. It was used an 8051 core developed in synthesizable RTL SystemC by the Brazil IP[3] project team. The OCP interface was constructed as an external device to microcontroller. It uses one of the external interruptions available in the 8051. Figure 2 depicts the interface hardware. It is possible notice in this figure that the OCP-IP interface was built as a external module. In this way, it can be accessed like a memory.

Beyond the interface hardware it was also necessary to write a driver to use it and also an interrupt routine. The driver allows the programmer to read/write words using the OCP interface. The interrupt routine is used to indicate the end of the OCP protocol.

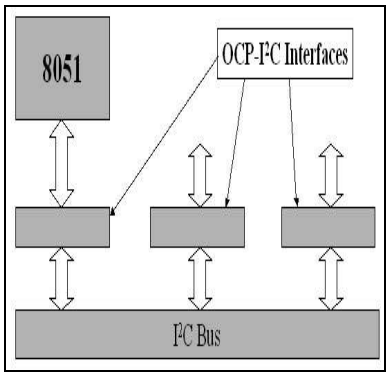

Figure 1: Platform

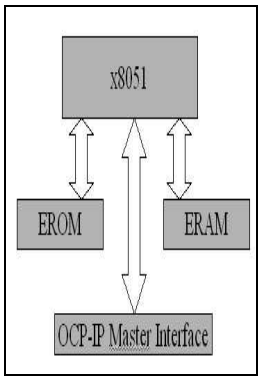

Figure 2: Interface

4. SERIAL BUS WITH OCP-IP INTERFACES

The serial bus used in the platform works according to I^2C protocol. The main difference between it and a I^2C implementation resides on the clock frequency used by them. While the highest frequency of the I^2C bus is 3.4 Mbits/s, the implementation used here is uses the system clock. This allows the bus to perform transfers at much higher speed.

The bus is composed by two wires: sda and scl. The sda transfers the data, while the scl transfers synchronism signal used during the transferences. A detailed description of the I^2C protocol can be found in [7].

The master/slave OCP interfaces implement the basic signals of the OCP protocol. These signals are: MCmd, MData, MAddress, SResp, SData, SCmdAccept. A detailed description of the used OCP signals can be obtained in [4].

The basic signals permit read, write and read exclusive write operations. A second role played by the interfaces is that they do the conversion between the OCP and serial protocols. The great advantage of such scheme is that the serial bus can be latter replaced by a parallel one without having to make changes on the rest of the system.

5. CASE STUDY – ACCES CONTROL SYSTEM (ACS)

The base platform has been extended to implement an Access Control System (ACS) that controls the access to restricted areas. The purpose of the system is to permit the access to the restricted area only to persons identified by a customized card and password..

The ACS platform is composed of the base platform and peripheral modules: the card reader, the keypad, the display and the device that lock/unlock a door. All this peripherals have OCP interface and are slaves. They are constructed in SystemC, but not SystemC RTL.

To simulate this system it was written a software for test that performs write and read operations in all the peripheral devices. This software works based on the following algorithm: (1) a message is sent (WR display) to the display; (2) a card code is read (RD card), (3) if this code is correct other message is sent and (4) the password is read (RD password), if this password is correct (5) an unlock command is sent (WR lock).

In Table **1** are shown the simulation time intervals for the communication operations on the bus performed by the ACS software. This simulation was done using a clock period of 30 ps.

Function	#opers.	Exec. time
WR display	1	64.800 ns
RD passwd	8	600.510 ns
RD card	4	414.750 ns
WR lock/unlock	1	34.506 ns

Table 1: Simulation results

6. CONCLUSIONS

This paper has presented a platform that will help the building of platforms like itself and the analyzing of their performances. This will be possible because this platform model allows the accurate measurement of the bus communication times.

The way the processor and peripherals are connected to the bus through OCP-IP interfaces was described. Besides that, a case study was implemented using the platform and its communication time intervals were obtained.

A next step to this work is to parameterize the bus used here. This is needed because this bus will help the building of other platforms.

7. ACKNOWLEDGMENTS

This work is being supported by the CNPq under project number 52.1869/96-0 and Synopsys University Program.

8. REFERENCES

[1] A, Snagiovanni Vincentelli; Carloni, Luca; Bernardinis, Fernando De; Sgroi, Marco. “Benefits and Challenges for Platform Based-Design”. June 2004.

[2] A, Snagiovanni Vincentelli. “Defining Platform-based Design”. EE-Design, February 2002.

[3] Brazil IP. available at www.brazilip.org.br , 2004 jun 16.

[4] OCP-IP Association. Open Core Protocol Specification, Release 2.0, rev 0.95, January 2002.

[5] Synopsys. *Describing Synthesizable RTL in SystemC*, version 1.0, May 2001.

[6] SystemC Community. available at www.systemc.org, 2004 jun 16.

[7] The I^2C bus specification version 2.1, January 2003.

[8] 80C51 8-bit microcontroller family datasheet; 2002 jan 24; Philips Semiconductors; http://www.semiconductors.philips.com/pip/P87C58X2FN.html.