Biprocessor SoC in an FPGA for Teaching Purposes

Joaquín Olivares, Juan Gómez, José M. Palomares, and Miguel A. Montijano

University of Córdoba

olivares@uco.es

Abstract

Computer architecture science evolves continuously. This work describes a methodology to teach a system which is integrated by several processors in just one chip. The description of how to implement a biprocessor system within an FPGA is proposed. Thus, the student will simultaneously acquire advanced knowledge on microprocessors, focusing on the operational behavior and structure of a biprocessor system, which they implement in a real device. Finally, a software layer is provided to execute an application, which proves how the system works. The accomplishment of this practice has been carried out in the first four-month period of 2007 in the 4th year subject “Digital Electronic Systems” of the Engineering Degree in Automatics and Electronics at the University of Córdoba.

1. Introduction

The Digital Electronic Systems subject is structured in three thematic blocks. The first block is a review of programmable logic devices, digital systems (DS), and design of DS; this takes about 20% of the total. The second block is dedicated to train in advanced hardware programming, it takes about 60%. Finally, last block is dedicated to show the microcomputer as a complex digital system, taking the remaining 20% [1].

This work treats the third thematic block exposed. Traditionally, computer architecture has been taught with sw simulators [3][4][5].

We propose to provide students with a user-friendly tool for easy design and real implementation. Simple designs are exposed to the students at the beginning; finally showing them more complex designs as biprocessors, which is shown in this paper. In it, two MicroBlaze (MBlaze) processors are integrated in an FPGA; each of which has an independent memory system, and are intercommunicated with a unidirectional serial channel.

MBlaze is a 32-bit specific purpose processor developed by Xilinx in VHDL. It can be parameterized using XPS to obtain an “à-la-carte” processor. It is a RISC processor, structured as a Harvard architecture with separated data and instruction interfaces. [2]

2. Software tool

The Xilinx Platform Studio (XPS) is a Graphical IDE for developing and debugging hw/sw platforms for embedded applications. With it, the developers can detach themselves from the direct programming of the hardware. This environment simplifies the procedure to the users, allowing them to select, interconnect, and configure components in order to obtain the desired result.

In this work, a SoC implementation is proposed. In it, two MBlaze processors are integrated in a single Xilinx xc3s200 FPGA.

An embedded system performed with XPS can be summarized as a definition of a hw platform (HWP) and an sw platform (SWP), each defined separately.

2.1. Hardware Platform

The HWP contains the description of the system architecture, the memory map, and the configuration parameters. HWP can be defined as one or more processors connected to one or more peripherals through one or more buses. The definition of the activity follows this sequence:

- To add processors and peripherals.
- To connect them through buses.
- To determine the processor memory allocation.
- To define and connect internal and external ports.
- To customize the configuration parameters.

2.2. Software Platform

The SWP contains the description of drivers, component libraries, standard input/output devices, configuration parameters, interruption routines, and
other sw features. The sequence of activities which are needed to define the SWP is the following.

- To assign drivers to peripherals.
- To include EDK libraries.
- To select the operative system (OS).
- To define processor and drivers’ parameters.
- To assign interruption drivers.
- To establish OS and libraries’ parameters.

3. Biprocessor System Design

A computational system composed of two MBazes will be designed. Both MBazes will be interconnected using message-passing protocol. Each MBaze has its own non-shared memory for instructions and data.

In Fig. 1, a diagram with the structure of the design is shown. In it, the used buses and components have been detailed. It also includes how they are interconnected.

3.2. Specification of the SWP:

The first step is to select the drivers for the components, the SWP libraries (default in this project) and the OS (standalone) for each processor.

In the following step, the configurable parameters from processors and drivers must be edited. In this case, all fields are configured by default except the frequency of the processors, at 100 MHz.

In the final step, the OS libraries configuration parameters are edited and configured.

The proposed system requires the UART to be set as the I/O standard peripheral, which is implemented by assigning the UART to the MBaze master OPB.

3.3. Software Application:

To carry out the goals of this work, it is necessary to create two SW projects, one for each MBaze. With these, each MBaze will execute its own program.

The writer processor generates the data that will be sent to the reader processor through the FSL channel, using mutual exclusion primitives for message passing. The master OPB bus sends the data received by the reader processor to the UART.

Finally, the configuration file is loaded into the FPGA. The students execute the global system and verify that both processors work properly.

4. Conclusions

This activity was carried out last academic year.

XPS can be used as an excellent tool for the students to manipulate and implement complex architectures avoiding implementation details. Otherwise, they would spend a lot of time until they dominate concepts and techniques to develop them.

Students understand the functionality and the structure of the different components that altogether build up either a monoprocessor or a biprocessor. An improvement of 28% after examining the students in 2007 respect to the mean of the last five years was obtained.

Also a tutorial on how to implement a biprocessor system on a single FPGA chip has been exposed.

5. References