HADES-1: A Rapid Prototyping Environment based on Advanced FPGA's

M.A. Aguirre, J.N. Tombs, A. Torralba and L.G. Franquelo
Departamento de Ingeniería Electrónica. Escuela Superior de Ingenieros de Sevilla
Avda. Camino de los Descubrimientos s/n
{aguirre,jon,torralba,leopoldo}@gte.esi.us.es

Abstract

Rapid prototyping of large digital systems is becoming supported with the use of new advanced FPGA's. These FPGA's can give more information than functional simulation and emulation tasks, due to their inner inspection features.

This paper presents HADES-1, a new environment for rapid prototyping and hardware debugging. HADES-1 is based on one FPGA of the VIRTEX family, exploiting the advanced features of the SelectMap port and a fast link with the host PC.

1. Introduction

The improvements in the area of microelectronics lead to shorter design cycles and rapidly rising complexity of electronics designs [1]. Therefore, developers are forced to refine the system development process to avoid a productivity gap. Rapid prototyping is a technique to support the system development. The term *rapid prototyping* is defined as a type of prototyping in which emphasis is placed on developing prototypes early in the development process to permit early feedback and analysis.

The traditional development process can be represented in the V-model (fig 1). The V-model enhanced by different levels of rapid prototyping is called VP-model[2]. The idea of rapid prototyping is to skip time intensive steps of the development process, and reduce the time to get a first functional prototype of the entire system or a part of the system. This prototype allows the validation of functionality and early performance tests. The comparison between different design alternatives can be simplified and therefore the conceptual validation rises. Compared to system simulation, rapid prototyping offers a test in the real environment. Hence, real-time behaviour, interferences and interfaces with the real environment must also be taken into account.

Three different levels can be distinguished:

- *Concept oriented* rapid prototyping represents the fast conversion of an executable specification into a functional prototype. This prototype serves to clarify system goals and must support a widespread range of hardware interfaces. The concept is oriented at high level. The support environment is not specific and can be reused for other prototypes.
- *Architecture-oriented* rapid prototyping, where the final architecture of the system is fixed. Some components of the entire system are already developed or standard components are used, whilst other parts are still under fine development.
- *Implementation-oriented* rapid prototyping is highly specialized and precise, the prototypes don't permit a widespread applicability. Usually it is formed from several final release components surrounded by new components. This approach allows performance analysis.

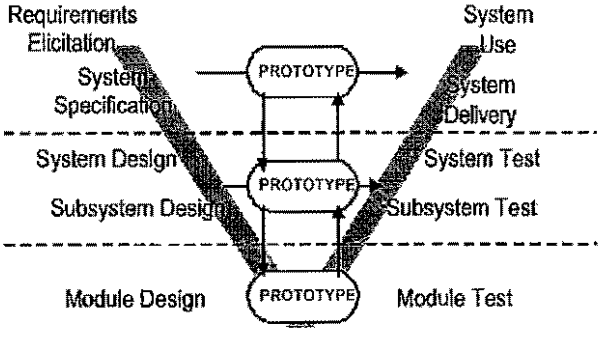

Fig 1. VP Diagram for Rapid Prototyping

This paper describes the solutions for a cost effective environment that supports prototyping in the first level and second level, named HADES-1 (HArdware DEbugging System). HADES-1 is a framework to exploit the most specialised characteristics of advanced FPGA's. It consists of a prototyping board which is presented as a digital microelectronic in-system emulator [3-4]. There are three main lines that have been extensively developed in HADES-1:

- For large digital system analysis, an advanced FPGA is used with an extensive use of its high flexibility, capacity and performance.

- For some primary performance analysis a system clock controller is introdueed.

now on referred to as the C-FPGA). This C-FPGA works as an interfaee between the host PC, the

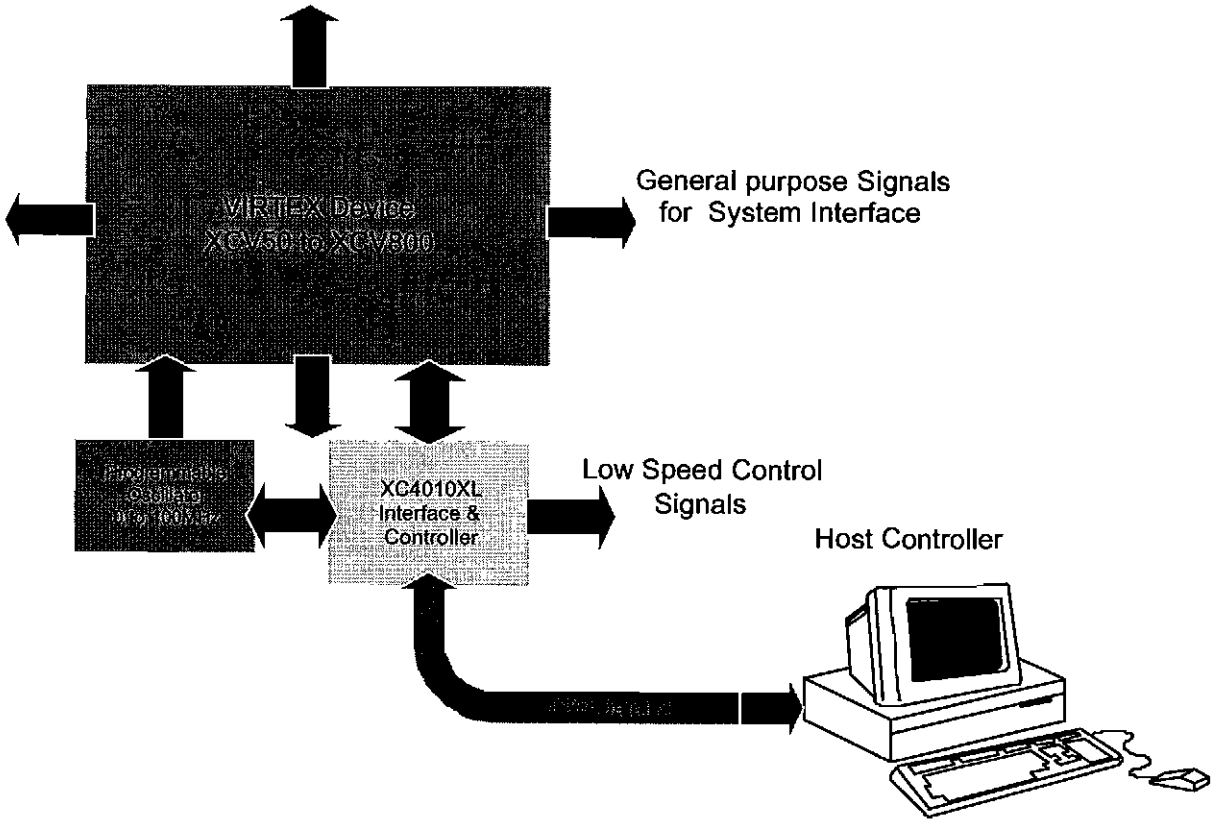

Figure 2. Overview of HADES-1 System

- For human analysis purpose a fast and seeure link with a host PC takes control of the internal features of the FPGA.

This paper is organised as follows. In Section 2 the system hardware is described, section 3 and 4 show the interface with the internal parts of the FPGA and the system clock controllcr and, section 5 the link with the host PC is shown and seetion 6 the softwarc is described.

2. Overview of HADES-1

HADES-1 consists of a system for emulating large digital circuits using an advanced FPGA. HADES-1 has bcen developed using a VIRTEX device from XILINX, using any mcmbcr of this family with a PQ240 footprint. A photo of HADES-1 is shown in figure 5. HADES has all the functional pins availablc for being use in the system, without any specialisation. In this way HADES can be adapted for the rcquiremcnts of any system. Figure 2 shows a scheme of HADES-1 and its most interesting features.

The core for control and analysis features is a second FPGA, onc XC4010XL also from XILINX, (from VIRTEX programming and debugging system, the programmable oscillator and some static Input/output lines. The C-FPGA solves the link between the host PC and board. This part will be described in detail in seetion 5. Internally, the C-FPGA is organised as a memory map deviee with 32 memory addresses. This C-FPGA runs with the built-in clock to make it absolutely independent of the on-board oscillator. C-FPGA has a fixed program that is downloaded during board start-up time. Howcver C-FPGA can bc bootcd from another program due to the availability of a flash memory.

As clock system is an essential part of a prototype we have introduced a programmable oscillator dcviec from Dallas, DS-1075. This programmable oscillator can be in-run-time programmed form thc host PC in runtime, without any cffect on the Virtex FPGA.

3. Interface with VIRTEX

The fastest and simplest way to interface with VIRTEX FPGA (from now on the S-FPGA) is using the call SelectMap method. SclectMap is a parallel port that can be read and written in an asynchronous process, this is the fastest configuration method. From the host PC cvery byte of the configuration file

is downloaded through this port and every sequence of the control signals is generated from an internal state machine. This state machine is also designed to be used for reading S-FPGA, allowing the configuration and the internal state of the S-FPGA to be transferred and analysed in the host PC.

Some internal checks are introduced in the downloading process to avoid potential damages inside the S-FPGA due to an almost impossible (note "almost") potential mismatch in the programming method. This requires the start-up process to be well controlled.

In the case of successful programming the host receives a flag on the start-up from the S-FPGA.

4. System Clock Controller

One of the most interesting features for a Rapid Prototyping system is a full controlled System Clock. We have selected the well-known programmable frequency clock generator from Dallas DS-1075. This oscillator can generate frequencies from a base frequency characteristic of every DS-1075 model. This oscillator can be programmed to frequencies given by f_{base}/N, with N=1,...,512, and f_{base} 60, 66, 80 and 100MHz. Other commercial prototyping boards also use this oscillator, but for its programming method it needs a power-off plus power-on sequence for the programmed process. In HADES-1 this process has been introduced automatically and controlled by the host PC through the C-FPGA. The System Clock can be stopped if needed, and from the PC the oscillator can be programmed, inspected and modified.

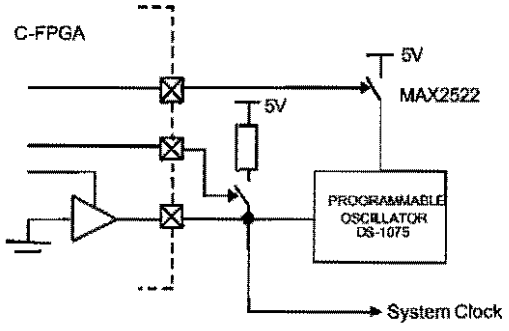

Figure 3.Programmable Oscillator Control Scheme

Due to its programming method, that shares the same wire for programming and for the system clock, the clock is connected to the C-FPGA to generate the programming sequence (Dallas 1-Wire property)[9]. The Power-off plus power-on programming sequence is only performed on DS-1075, as the C-FPGA controls a set of analog switches that allow the clock power, and clock stopping as well.

5. Link with the host PC

The system programming and analysis process can only be done using a fast and robust link. HADES-1 uses a well-known and common protocol called EPP[8]. EPP means Enhanced Parallel Port and is described in the IEEE-1284 Standard. EPP is a bi-directional and hardware controlled link that can achieve speeds between 500Kbytes/s to 2Mbytes/s that is more than enough for our purpose. A Virtex 800 S-FPGA needs 576 Kbytes for its configuration. Due to the control process the configuration would be performed in less than 2 seconds.

The C-FPGA has a state machine that produces a dialog between the PC and the board. It generates the necessary handshake signals and decouples addresses and data in the designed memory map. The state machine is depicted in figure 4, and it shows how it generates a single clock cycle read, write and address pulse and the wait handshake signal.

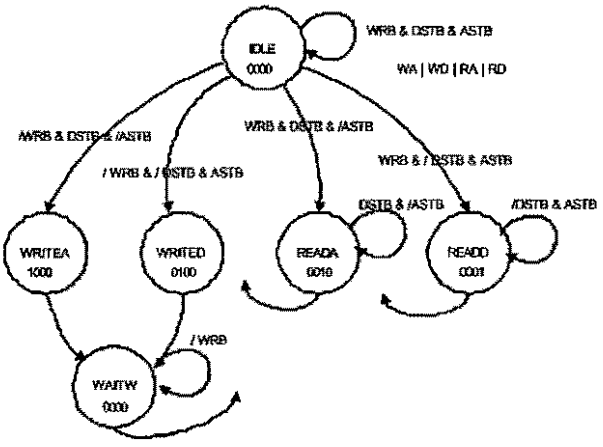

Figure 4. EPP Interface State Machine

6. HADES-1 Software.

The host PC implements the human-machine interface. The design is developed using the standard design flow using Xilinx tools. When the bitstream file is produced, HADES-Software can read this file and download it into the S-FPGA, during the configuration process. Also, the HADES-Software allows an easy control of the system clock frequency, and the configuration and readback processes from S-FPGA and partial reconfiguration of the device. The software is a Windows based tool and uses a low-level control of the parallel port.

Figure 6 shows the dialog window for the Oscillator and figure 7 shows the S-FPGA programming dialog window.

Other useful features that have been included in the software, such as inspection charts. These allow the observation of the current content of the internal flip-

flops of the S-FPGA after a *capture* command has been done.

The inspection process goes as follows:

1. During the standard design flow two files are generated: bitstream (*file.bit*) and eross reference (*file.ll*).
2. The bitstream generation have to keep the programming port active.
3. An special internal reading allows a downloading of the internal state from the FPGA. A new file (*file.CLB*) is created.
4. From the file.ll the exact place of the information inside the file.CLB and can be reconstructed.

The *file.ll* only gives the information for those nodes that has not been eollapsed by the synthesis process. If a particular node is desired to be explicit, adequate commands can be introduced for preserve hierarchy.

Figure 8 shows the window chart for the signal selection of an example design. Also shows the result of an inspection process uploaded from the board.

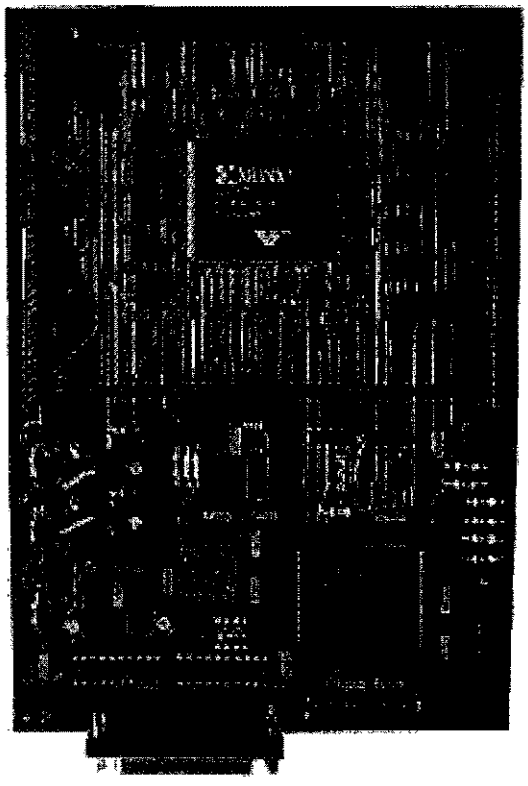

Figure 5. HADES-1 Board.

7. Conclusions and Future Work

In this paper we have presented a new board for large digital circuit emulation and rapid system prototyping. It can run as an in-system circuit emulator with many hardware debugging capabilities. It also has full control of the system clock. Currently, we've under development some new features for HADES-1:

1. Development of a dedicated port for the construction of a multi-board system, for parallel processing prototyping system.
2. Easy modification of the C-FPGA configuration.
3. HADES-2 will have one VIRTEX-E and one SPARTAN for the S-FPGA and the C-FPGA, respectively.
4. Run-Time partial reconfiguration using JBITS classes

The up to date information about this project is weekly maintained in the web site: http://woody.us.es/~aguirre/hades.html

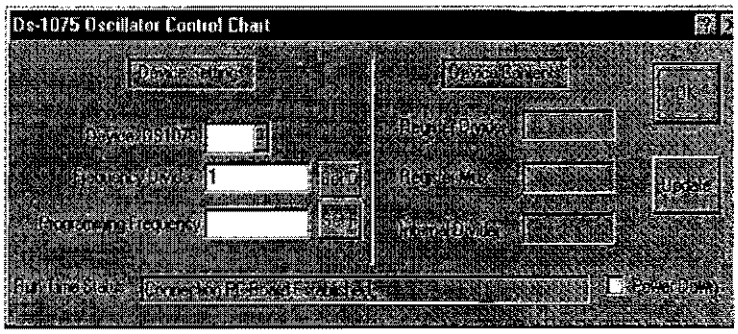

Figure 6. Programmable Oscillator Dialog Window

7. Bibliography

[1] Dollas, A. Kanopoulos, N. "Reducing Time to Market Through Rapind Prototyping". IEEEComputer, Feb. 1995.

[2]A.Burst, B.Spitzer, M.Wolff, K.D. Müller-Glasser. "On Code Generation of Rapid Prototyping Unsing CDFI". OOPSLA'98, Vancouver, CA

[3] Hutchings B., Nelson B. and Wirthlin M.J.. "Designing and Debugging Custom Computing Applications". IEEE Design & Test of Computers. Jan-March 2000. pp 20-28.

[4] Koch A. "A Comprehensive Prototyping-Platform for Hardware-Software Codesign". 11th IEEE Internation Workshop on Rapid System Prototyping. Paris Jun-2000. pp78-82.

[5] Xilinx. Co."The Programmable Logic Data Book". 2001.

[6] Xess Co. XSV Board V1.0 Manual. March 2000

[7] Insight. Virtex-E Development Kit. Aug-2000

[8] Peacock C. "Interfacing the enhanced parallel port". http://www.senet.com.au/~cpeacock. Dec-1997.

[9] DS-1075 Datasheet.

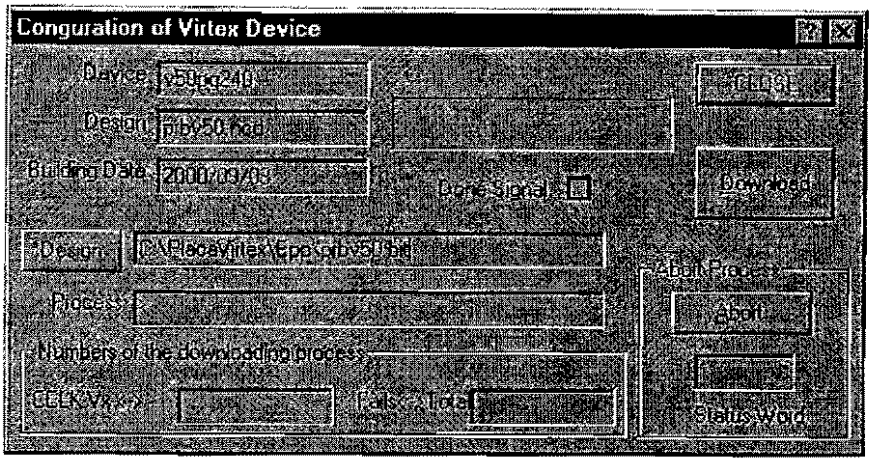

Figure 7. S-FPGA Programming Dialog Window

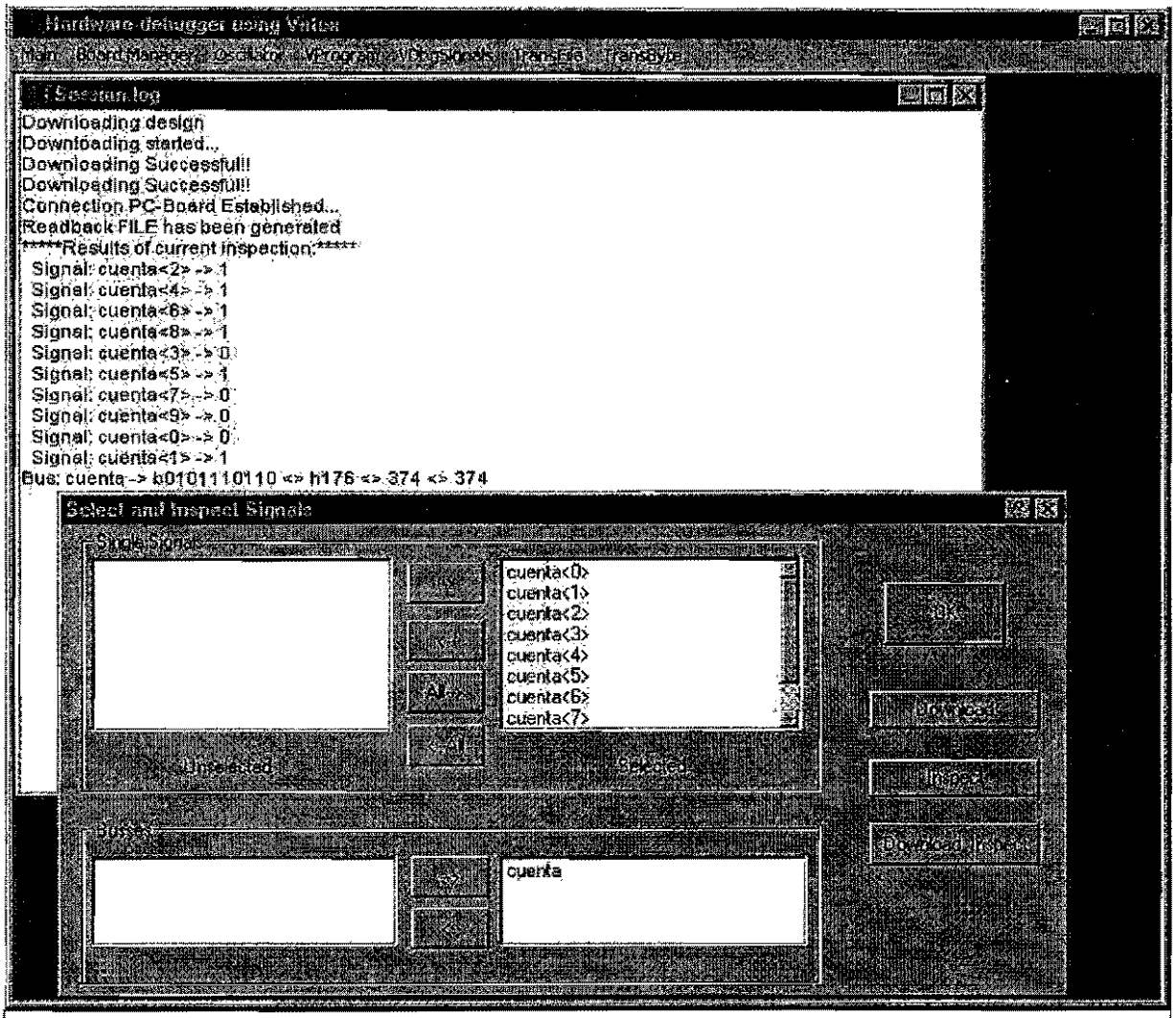

Figure 8 HADES-1 Analisys window