Università degli Studi di Verona, Facoltà di Scienze MM. FF. NN.
Dipartimento di Informatica
EDALab: Embedded System Design Center

Ca' Vignal, Strada Le Grazie 15
37134 Verona, Italia
Tel. +39 045 8027069
Fax +39 045 8027068

Verona, 09/03/2007

FPGA Design

Nicola Bombieri

Ca' Vignal, Strada Le Grazie 15
37134 Verona, Italia
Tel. +39 045 8027069
Fax +39 045 8027068

1 Required Background

Students interested in learning FPGA design are required to know the fundamentals of VHDL language. Moreover, it is advisable that students are familiar with concepts of HDL device synthesis.

2 Goal

The goal of this lecture consists of guide students through the steps of FPGA design. The main steps are described by using one of the more important commercial tools for FPGA design.

Students will learn to:

- Analyze a real HW/SW board, by distinguishing the features of elements characterizing the Hardware.
- Synthesize RTL descriptions by setting the implementation details dependent on the board.

3 Introduction

After a system description has passed the verification phase by means of dynamic and/or static verification, it is ready to undergo the synthesis process. Figure 1 shows the whole design process and it underlines the exact point in which the synthesis task is set up.

The RTL device implementation is translated into a set of elementary blocks at gate level. Xilinx ISE will be presented as example of tool that aids designers in this task. It drives designers during the synthesis phase, starting from the RTL through the steps of architectural choices up to the board programming.

Ca' Vignal, Strada Le Grazie 15
37134 Verona, Italia
Tel. +39 045 8027069
Fax +39 045 8027068

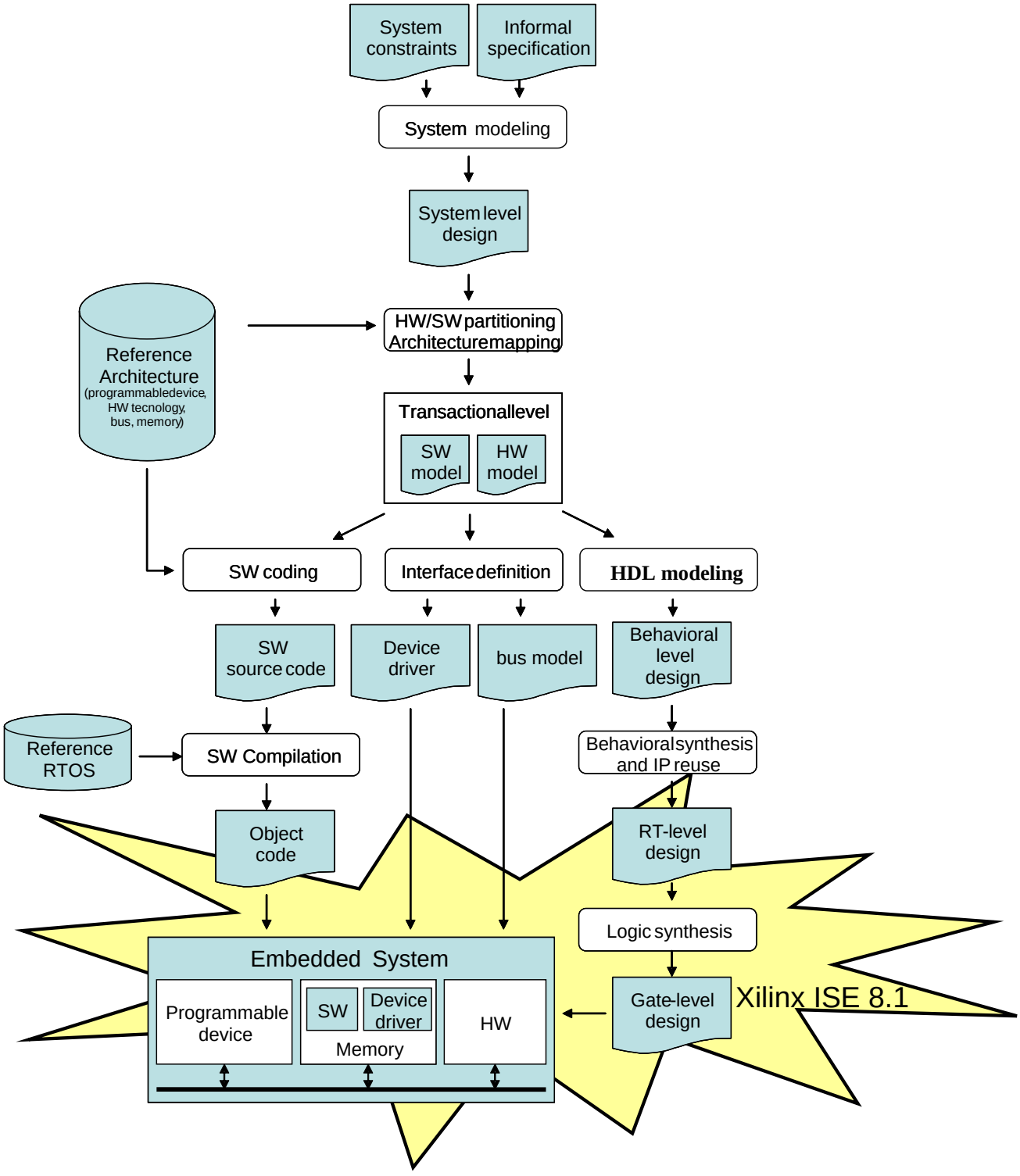

Figure 1: Embedded system design flow.

Ca' Vignal, Strada Le Grazie 15
37134 Verona, Italia
Tel. +39 045 8027069
Fax +39 045 8027068

4 Programmable Board

The programmable board Spartan3 by Xilinx is the case study. In the next session, its features and the tool for the programming task will be presented.

4.1 Xilinx Spartan3

Xilinx Spartan3 is a family of 90nm FPGA that fulfills industry demands for high-volume and I/O-optimized programmable logic solutions.

This technology allows designers to develop high volume applications for a variety of markets including broadband access, home networking, display/projection and digital television equipment. Spartan-3 devices offer the following:

Up to 5 million system gates (XC3S5000)

Up to 784 single-ended I/Os, and 344 differential I/O pairs

XCITE Digitally Controlled Impedance (DCI) technology

Designers can integrate embedded processing, digital signal processing (DSP), and connectivity capabilities into Spartan-3 devices. Xilinx supports each of these applications with customized tools, JTAG probes, IP cores, design services, and training.

Students will use the board XC3S200, a device belonging to the family of Spartan3 components. Figure 2 reports the board features.

Spartan-3	XC 3S200
System Gates	200K
Logic Cells	4,320
	12
Block RAM Bits	216K
Distributed RAM Bits	30K
DCMs	4
I/O Standards	24
Max Differential I/O Pairs†	76
Max Single Ended I/O†	173
FT256	173

Figure 2: XC3S200 board features.

1.

Ca' Vignal, Strada Le Grazie 15
37134 Verona, Italia
Tel. +39 045 8027069
Fax +39 045 8027068

4.2 Xilinx ISE 8.1

ISE 8.1 WebPACK is an industry's free, fully featured front-to-back FPGA design solution for both Windows and Linux. It is a solution for FPGA design offering HDL synthesis and simulation, implementation, device fitting, and JTAG programming. Now, ISE *WebPACK* 8.1i provides the tools and features along with the same easy-to-use design environment. Xilinx ISE allows convenient productivity by providing a design solution that is always up to date with error-free downloading and single file installation.

4.2.1 Creating a new project

A new project is crated by setting *New Project* from the *File* menu and choosing the HDL type. Device properties have to be set, by selecting:

- Family SPARTAN3
- Device XC3S200
- Package FT256

A new source file can be generated by clicking on *Next* button. Note that both source file name and source type (VHDL module) should be selected.

The last step consists of defining the Entity name, Architecture name and device ports. Port names should be defined as well as the direction (input or output) and the size. Note that a port with more than one bit is considered a Bus.

The tool shows, starting from the left :

- the source menu: it reports the source files composing the system;
- the processes menu: it reports every step of the synthesis process;
- the project status: it reports every detailed warning and message of any task.
- The VHDL code window: the designer can directly implement the functional behaviour, by exploiting the entity and architecture structures automatically generated by the tool.

4.2.2 Binding I/O ports

Designer can open the architecture and package views of the device by double clicking on *Assign Package Pins.* The I/O binding is performed by selecting the Package view and by dragging each VHDL entity port (reported on the left) on the point corresponding to the physical I/O port. The right mapping between device I/O objects and colour points is directly printed on the board.

The task is saved by selecting *Save* from *File* and setting *XST Default* as bus delimiter.

Ca' Vignal, Strada Le Grazie 15
37134 Verona, Italia
Tel. +39 045 8027069
Fax +39 045 8027068

4.2.3 Implementing the Top Module

The Top Module is created by clicking on the corresponding Button on the main menu (*Implement Top Module*). The synthesis process is now started and the report log is reported ad the follows:

```
=========================================================================
Started : "Synthesize".
=========================================================================
*                     HDL Compilation                   *
=========================================================================
Compiling vhdl file "C:/Xilinx/tesardo/example.vhd" in Library work.
Entity <example> compiled.
Entity <example> (Architecture <behavioral>) compiled.
=========================================================================
*                     HDL Analysis                      *
=========================================================================
Analyzing Entity <Example> (Architecture <behavioral>).
Entity <Example> analyzed. Unit <Example> generated.
=========================================================================
*                     HDL Synthesis                     *
=========================================================================
Synthesizing Unit <Example>.
    Related source file is "C:/Xilinx/tesardo/Example.vhd".
Unit <Example> synthesized.
=========================================================================
…
…
Device speed data version:  "PRODUCTION 1.37 2005-11-04".
Device Utilization Summary:
   Number of External IOBs           2 out of 173     1%
    Number of LOCed IOBs             2 out of 2       100%
Overall effort level (-ol):   Standard
Placer effort level (-pl):    High
Starting Placer
-------------------------------------------------------------------------


Generating Report ...


Number of warnings: 0
```

Ca' Vignal, Strada Le Grazie 15
37134 Verona, Italia
Tel. +39 045 8027069
Fax +39 045 8027068

Total time: 6 secs

Process "Generate Post-Place & Route Static Timing" **completed successfully**

4.2.4 Generating and downloading the bit-stream

Choose *Generate PROM, ACE or JTAG File* from *Generate Programming File* menu. Choose to configure the device using a boundary-scan (JTAG) by the automatic connection and select *Finish.* In this way the tool connects to the hardware device.

There are two ways to program the board:

1. *Permanently*, by downloading the bit-stream into the FPGA and saving it into the Flash memory. In this way the VHDL synthesized program remains into the board even if the power is switched off.
2. *Momentarily*, by downloading the bit-stream into the FPGA disregarding the Flash memory. In this way the board looses the program at any restart or reset step.

Choose to momentary program the board, by downloading the bit-stream file (example.bit) into the FPGA and bypassing the Flash.

Click on the FPGA block with the right key of the mouse and select *Program*. The process ends by clicking the OK button.

The board is programmed right now and its behaviour is ready to be verified.

Ca' Vignal, Strada Le Grazie 15
37134 Verona, Italia
Tel. +39 045 8027069
Fax +39 045 8027068

5 References

[1]http://www.xilinx.com/products/silicon_solutions/fpgas/spartan_series/spartan3_fpgas/overview.htm#s3table