Demonstration of Tools

Paulo Flores

ESPRIT III Project # 6043 "QuickChips"

Task 1–5.4.3 - Deliverable 16

INESC

Instituto de Engenharia de Sistemas e Computadores

May 1994

1 Introduction

This deliverable is a demonstration of the selected CAD tools for the QuickChips design environment. These tools were integrated in the selected framework (Design Framework II), providing the alpha version of the CAD system, as reported in deliverable 5.

The set of selected tools comprises the *Synopsys VHDL System*, for high-level VHDL simulation and synthesis, *Synopsys Test Compiler*, for automatic test pattern generation, *Synopsys VSS* and *Cadence Verilog*, for logic simulation, *Cadence Gate Ensemble*, for place and routing, and *Cadence Diva*, for physical design verification.

The necessity to support a low cost front-end design system directly targeted at SMEs (Small and Medium Enterprises), providing a set of libraries and simulation models for design system based on personal computers, has also been recognized. The OrCAD design environment, which has a very large installed base, has been chosen to provide the PC (Personal Computer) based front-end.

The table 1 summarizes the design tasks and the selected tools required to implement the QuickChips design methodology. An overview of each tool is presented in the following sections of this report.

Design Task	Workstation Environment		PC Environment	
	Design Tool	Vendor	Design Tool	Vendor
Design Entry	DFII - Composer	Cadence	Schematic Tools	OrCAD
Behavioral Simulation	VHDL System Simulator	Synopsys	Simulation Tool	OrCAD
Technology Mapping	Design Compiler	Synopsys		
Logic Simulation	Verilog-XL	Cadence		
Place & Route	Gate Ensemble	Cadence		
Physical Verification	Diva	Cadence		
Post Layout Simulation	Verilog-XL	Cadence		
Test Generation	Test Compiler	Synopsys		

Table 1: CAD tools used in the QuickChips design environment

2 PC Based Environment

The PC based design environment is a low-cost design solution that has been regarded, taking into consideration the needs of small and medium enterprises.

In the PC based front-end the design is specified using traditional schematic entry tools, but a technology independent library is used. The supported design entry tool in the PC based design environment is the **OrCAD Schematic Design Tool**. This tool is an affordable option for a SME and is extensively used by Portuguese SMEs. The IDPS core library, developed for this environment, acts as a technology independent library allowing the creation of technology independent descriptions. These descriptions are simulated in the the **OrCAD Simulation Design Tool**, using a simplified delay model to verify the behavior of the schematics.

Using the EDIF format, which has been selected as one of the standard interface formats to the QuickChips design environment [Flores 94], the design is exported to the workstation based environment and mapped to the target technology using logic synthesis.

3 Workstation Environment

The main design flow of the QuickChips design environment is based on high performance CAD tools, running on workstations. The selected set of tools for this environment performs both, front-end and back-end design tasks.

Composer

In the workstation design environment the schematic editor integrated in the **Cadence** framework - **Composer** was selected for the design entry task. In this environment the circuit specifications are captured in a mixed-level description that includes VHDL RTL and gate level blocks.

The Composer schematic editor allows hierarchal design entry, where the defined blocks can be either a representation of another schematic, or specified using a textual description, in VHDL or Verilog. The selection of a tool integrated in the framework, for

design entry, simplifies the access to the design database, through the use of the DFII extension language - *Skill*.

VHDL System Simulator

The VHDL simulation of the design is accomplished using the **Synopsys VHDL System Simulator**. This simulation of the design is performed to check the compliance of the VHDL description with the original design specifications.

The VHDL System Simulator is fully compliant with the IEEE-1076 VHDL standard and it is integrated in the Synopsys synthesis environment. The simulation results can be checked on the schematic representation of the circuit or using the waveform display. This simulator has also a graphical debug environment with capabilities such as, step by step execution, signal monitoring and source code breakpoints, that simplifies VHDL bug detection and correction.

Design Compiler

The logic synthesis and technology mapping tasks are performed by the **Synopsys Design Compiler** synthesis tool. This tool has the capability to accept hierarchal circuit descriptions either in a hardware description language, such as VHDL or Verilog, or in EDIF format; a mixed description of the previous formats is also supported. Although, the VHDL language is not fully supported (as described in [Flores 93]), Synopsys claims that the Design Compiler supports the largest VHDL subset available on commercial synthesis tools.

The Design Compiler is an interactive synthesis tool that optimizes a design based on specified constraints imposed by the designer. The built-in static timing analyzer lets the designer impose complex timing constraints and analyze the temporal behavior of the circuit, for example, to evaluate the critical path delay. The synthesized circuit can be exported from the tool in VHDL, Verilog or EDIF.

Verilog-XL

The logic simulation task is performed using the **Cadence Verilog-XL** simulator. This task intents to check the fulfillment of circuit specifications considering technology specific delays.

The Verilog-XL simulator accepts Verilog descriptions ranging from "behavioral level" to "switch level", but in this task only the logical level will be used. The main advantages of this Verilog simulator are its performance in the simulation of large circuits. Verilog libraries are also supported and certified by all the major silicon foundries.

The capability to read a SDF (Standard Delay Format) file, makes this simulator also suitable for the Post Layout Simulation task. In this task the circuit is simulated with more accurate time delays extracted from the layout.

Gate Ensemble

The tool used for place and route tasks is **Gate Ensemble** from **Cadence**. This tool reads an EDIF netlist and generates the circuit physical layout for a specified channeless gate array.

The Gate Ensemble engine can route a circuit using up to seven metal layers in sea-of-gates technology. It supports over-the-channel routing, and handles off grid pins and routing obstructions. Placement regions and soft groups can be specified, providing a more efficient starting point for the placer. Besides being a completely automatic tool, it also supports manually editing for small changes. The final circuit layout can be written in GDSII standard format.

Diva

The physical verification task, that checks for layout errors and verifies the circuit implementation, is performed by **Diva** from **Cadence**. Diva is a set of verification tools that can be used interactively to locate and correct design errors.

The Diva design rule checker verifies the layout geometry to find deviations from technology design constraint. It works with any technology and handles all layout methodologies (from full custom to automated layout). Diva extractors operate in flat or hierarchical mode to produce, from the layout, the circuit netlist and the device interconnection delays. The Diva layout versus schematic comparator, checks the netlist extracted from the layout with the schematic netlist. The mismatching nets, devices, and parameters can be view interactively using error probing and correspondence cross-probing. The Diva electrical rules checker, checks netlists connectivity. It highlights electrical problems such as floating interconnect and devices, and abnormal connections in the design.

Test Compiler

The insertion of scan structures in the circuit and the generation of test vectors, to support device testing after fabrication, are done by **Synopsys Test Compiler**. This "tool" is integrated in the Design Complier enabling circuit optimization, for speed or area, with test logic in place.

The Test Compiler is a test synthesis tool that automates design-for-testability (DFT) and provides automatic test pattern generation (ATPG). This tool supports three common internal scan methodologies, multiplexed flip-flop, LSSD flip-flop or LSSD latch, and one standard boundary scan methodology, the IEEE 1149.1 JTAG standard. A built-in rule checker ensures that a design meets the needs of the chosen methodology and reports any violations that might reduce the fault coverage. The automatic generated test vectors can be written in TSSI standard format for interface with the test equipment.

4 Conclusions

In this report the CAD tools used in the INESC design environment were presented. This set of tools was chosen according to the QuickChips design methodology.

In order to support a low-cost design solution targeted at SMEs, the selected set of tools provide not only a high performance design environment, based on workstations, but also a design environment based on PCs.

All the tools that comprise the workstation design environment are integrated in the DFII framework which simplifies its invocation and design data transfer. However, the capability of the selected tools to use standards formats, to import and export design data, provide the necessary foundations to extend the QuickChips design environment to other tools.

References

[Flores 93] Paulo Flores. Set of Synyhesis Tools. Technical report, Project ESPRIT 6043, November 1993.

[Flores 94] Paulo Flores. Selected Languages and Formats. Technical report, Project ESPRIT 6043, May 1994.