WTROPIC: a WWW-Based Macro-Cell Generator

João Leonardo Fragoso
UFRGS
Instituto de Informática
Av. Bento Gonçalves, 9500
Caixa Postal 15064
91501-971 – Porto Alegre
Brazil
fragoso@inf.ufrgs.br

Fernando Moraes
PUC-RS
Faculdade de Informática
Av. Ipiranga, 6681 - Prédio 30
90619-900 - Porto Alegre - Brazil
moraes@inf.pucrs.br

Ricardo Reis
UFRGS
Instituto de Informática
Av. Bento Gonçalves, 9500
P.O. Box 15064
91501-971 - Porto Alegre
Brazil
reis@inf.ufrgs.br

Abstract

This paper presents a www-based macro-cell generator tool integrated in a www-based framework. The Wtropic tool allows user connections through the internet and it provides a communication layer to execute the physical synthesis tool remotely from any machine with a web-browser. The Wtropic server processes the client request at the server-side. So, it is required a small machine at the client-side. The Wtropic may have many servers in order to take advantage of distributed resources that can execute the physical synthesis tool.

1. Introduction

During the last four years, some research effort was done to provide World Wide Web based design environments for integrated circuits [1] [2] [3] [4] [5]. There are different approaches, but the common motivation is the use of a widely known user interface - a web browser - and the possibility of remote access.

Some of the early research in this area deals mainly with remote execution of applications. In this case, the designer acts as a data provider and analyzer, with a low degree of interactivity. Later, with the arise of platform independent programming solutions such as Java programming language [6], some research efforts were done to distribute the data processing to both sides of the network: remote and local machine.

The CAVE Project [5] is one of the initiatives focusing on this approach. Its architecture is based on the distribution of design resources between client and server sides of the network. The design environment supports several servers and clients. On server side, the performed functionality includes the storage of the design tools, design data and the hypermedia structure of the environment. Also on the servers are executed remotely the non-interactive tools. On the client side the interactive tools are loaded from the server and executed. Some design data can be temporally stored. The non-interactive tool invocation and results analysis is also done from the client side.

The distribution of processing load among the servers is an interesting feature, since it is transparent to the user. So, the design environment may be designed to execute the heavier tasks on the computationally faster machines.

Otherwise, the client side may be a very simple machine, running only a web browser, which is the basic requirement for a WWW based environment.

The Wtropic is a macro-cell generator based on WWW. The Wtropic implements the interface and the necessary protocol of communication that allows the user to execute the layout synthesis tool remotely using the web browser. The Wtropic was implemented using the methodology proposed in CAVE Project [7] and it is integrated in the CAVE Framework.

In Section 2, the CAVE framework and its structure are described. In Section 3, the main features of physical synthesis using Wtropic tool are presented. The Wtropic and its implementation and integration in the CAVE framework are showed in Section 4. Finally, we present our conclusions and future work.

2. The CAVE framework

The Cave Project goal is to research issues for a WWW based design automation framework. The framework architecture is based a client-server model, so it distributes the framework resources between client and server sides of the network. To define the design automation tools distribution over the network, the tools are divided in two groups, regarding the level of interaction of the designer with each tool. Highly interactive tools are loaded from network and executed on the client side, while poorly or non-interactive tools can be execute remotely on the server, as shown on Figure 1.

0-7695-0843-X/00 $10.00 © 2000 IEEE

Based on this specification, a prototype of a web based integrated circuit design automation framework was developed. The CAVE architecture divides the framework in the following entities: framework server, hypermedia structure, graphic user interface and tools. So, each one of the prototypes must follow the CAVE standards.

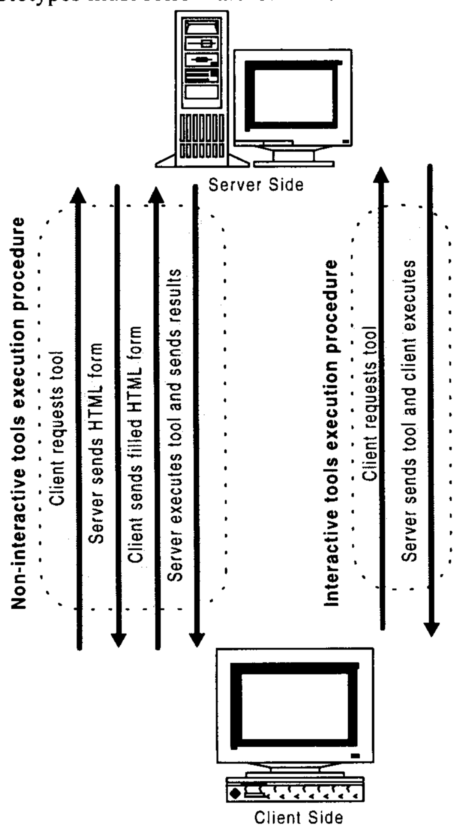

Figure 1. Cave Project client-server communication procedures

The framework server is based on the Java Web Server Model. Because of its extensibility using Java servlets, this product can be customized to supply the framework storage system, the HTTP service and the connection to external applications. As a web based environment, the building blocks of the framework are hyperdocuments. The hypermedia structure was designed targeting the ease of navigation across these documents, as show in Figure 2. The Project Home is the root of the framework hypermedia hierarchy. It is accessible from every document of the framework and from its hyperlinks the user can reach the root document of the main sections of the framework. The main sections are Project Manager, Tool Manager and the repository of the on-line documentation.

The project management tools, the user communications tools and the project tools are located in the Project Manager section. The facilities for tools integration and encapsulation, the framework management tools and the resources for tool developers are located in the Tool Manager section.

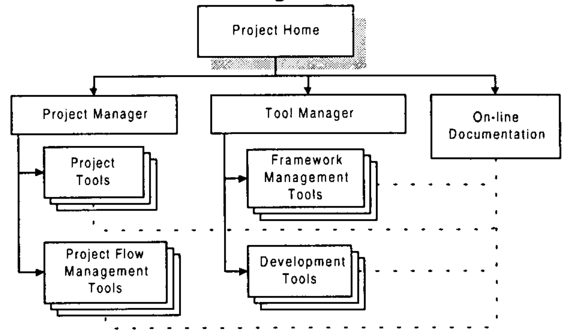

Figure 2. Cave Project hypermedia structure

The prototype graphic user interface follows the three-level standard proposed in [5]. The external level of the user interface is a web browser, which is the only software required on the client side of framework. The next level is defined by the hyperdocuments format. Most of the visual identity of the framework appears on this level. So, all the framework hyperdocuments must have the same appearance, defining a familiar user interface to ensure the simplicity of use and navigation.

The inner level is the tool graphic interface itself, so the visual aspect depends on the tool functionality, allowing limited customization.

2.1. Encapsulation of tools

Following the CAVE Project specification, tools with high level of interactivity run at the client side, so these tools must be either implemented or re-implemented using Java language and inserted in hyperdocuments. When the user request a tool with high interactivity from server using the hyperdocument URL, the hyperdocument and the tool are transmitted to client where the tool will be processed by the web browser. New client-server connections will open, if the tool needs configuration files, libraries or other files from the server.

However, tools with low level of interactivity run at the server side, remotely started by the designer. Once the designer uses a web browser to interact with the framework, the startup of the remote tools must be done via framework server, which is the framework entity in charge of responding the browser requests. The designer provides data and parameters via web browser, which one send them to server side. Finally, the server sends the results back to the web browser. It is necessary a communication layer between the framework server and the tools, in order to redirect the data and parameters received from the browser to the tool

3. Physical synthesis

The layout synthesis tool, Wtropic, is a *macro-cell generator*, i.e., a complete circuit is generated. The Wtropic is a new version for the TROPIC tool [8]. The initial SPICE description, obtained from a behavioral VHDL, is decomposed into leaf cells that will be assembled together by specific place and route tools, without constitute a separate library. Two instances of the same logic function can have different layouts, according to its environment.

The main features of the layout style:

- Linear-matrix layout style- each cell row is composed by two horizontal diffusion strips. The width of the transistors are orthogonal to the diffusion strips;
- Routing is implemented with 3 metal layers and stacked contacts, reducing the routing area;
- No layout compaction. This is the main feature of Wtropic, since it allows a very fast layout synthesis. Tools, like LAS [9], create an intermediate symbolic layout description, requiring layout compaction, consuming a lot of CPU time.
- Complete parasitic capacitance/resistance evaluation.
- Simple technology file to describe design rules (28 rules) and parasitic capacitances/resitances (26 rules).

At the cell level, we can enumerate the following features (Figure 3):

- connection between N and P plan directly in metal 1;
- minimum separation between N and P;
- gnd/vcc wires between transistors, in metal2;
- over-the-cell routing (OTC), connecting internal nodes of SCCGs (static CMOS complex gates) and nets belonging to only one channel;

and the Wtropic improvements at the cell level:

- optimized jogs are automatically inserted in the polysilicon wires, to reduce diffusion area (capacitance reduction);
- just the input/output pins are aligned. Jogs in polysilicon are inserted to connect the gate to the routing region of the circuit. This allows minimum space between almost all transistors, because the transistor does not need alignment.

As the TROPIC tool, the layout generator has only 2 input files: the SPICE netlist and the design rules file. Its outputs are the layout (CIF format) and the parasitic capacitance/resistance estimation (flat spice netlist). Figure 4 shows some transistor densities for Wtropic for 2 different processes.

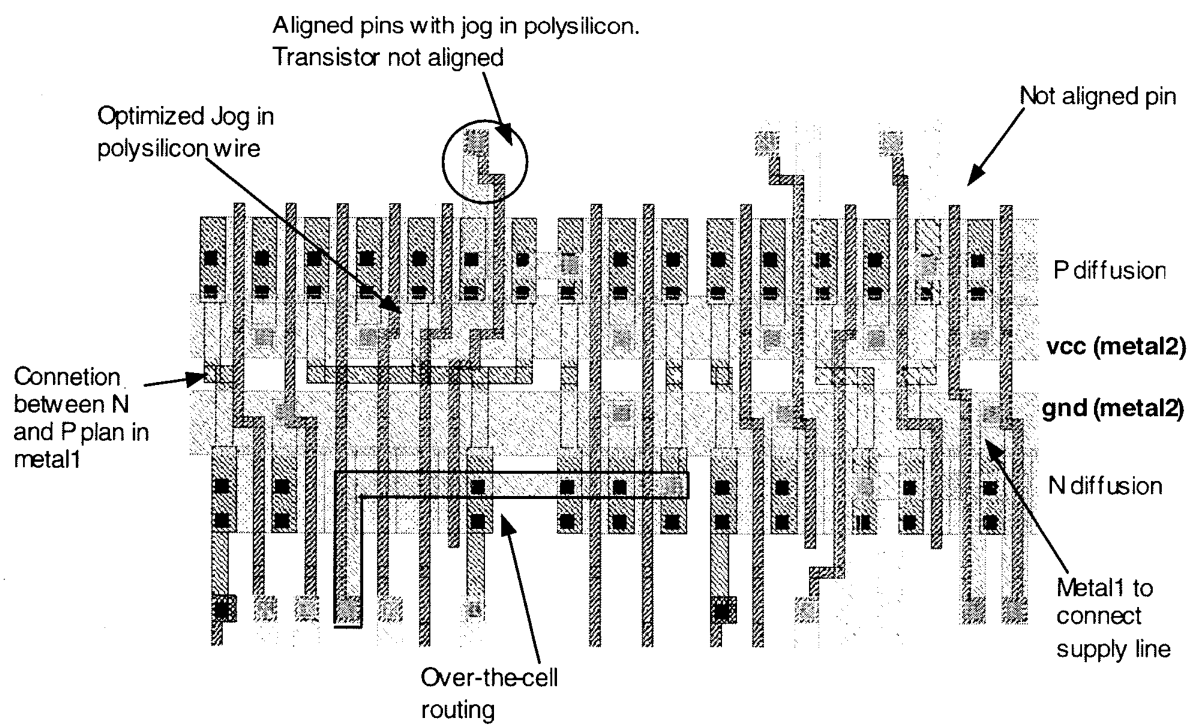

Figure 3. Layout example (row level)

For a 0,25 μm process, an average transistor density of 60000 transistors/mm^2 was obtained. Refinements in the routing algorithm can improve this density up to 20% [8]. The CPU time, to generate the layout and estimate parasitic capacitances, for the largest example (15000 transistors), was 227,67 seconds (in an Ultra-10 Sparc).

As CPU time for layout generation is no more a bottleneck, iterations can be made to obtain an optimized circuit. The logic synthesis tool can execute initial iterations to get accurate information on routing length (parasitic capacitances) and area. This can guide the technology mapping and also indicate where buffers must be inserted, since the real load in each node is calculated during the layout synthesis.

There are three parasitic capacitance components: line-to-ground, line-to-line and crossover capacitances. The line-to-ground and crossover capacitances are calculated by the traditional formulation based in the area and perimeter of the interconnections. However, the line-to-line capacitance is a function of the distance between two connections and the thickness of the layer. We use the empiric formulation described in [10].

It is used a simple and accurate 2½D methodology described in [11] to compute coupling capacitances. For each connection in the layer *i*, we analyze the immediate neighbor in the same layer, all crossunders in the layer i-1 and all crossovers in the layer i+1, treating the layers i±2 like ground planes.

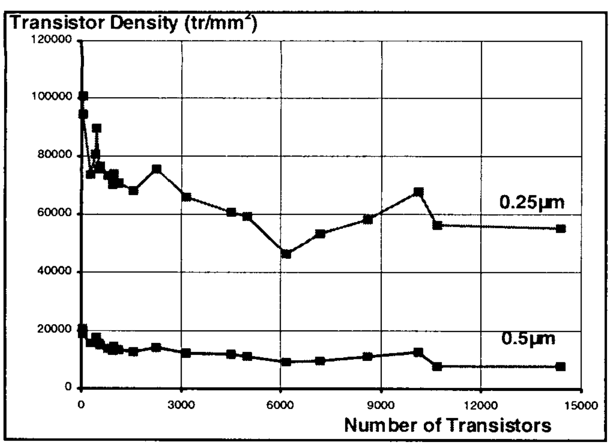

Figure 4. Transistor densities for Wtropic

The components of the diffusion capacitance, the area and perimeter of drain/source areas are also computed by the layout generator.

The circuit delay obtained through electrical simulation using the evaluated capacitances was compared to extracted capacitances. Preliminary results show an average difference of only 5% between estimation and extraction.

The resistances are calculated using the number of squares in each wire, plus the number of contacts and vias. This simple model is been improved. If resistances are also evaluated, the user cans chose between 3 models (with one stage): L, T and Pi.

The Table 1 shows the reduction of circuit width (diffusion strips) for Wtropic when it is compared for the TROPIC tool original version.

Table 1. Width reduction for Wtropic tool

Circuit	Xtors	Tropic Width	WTROPIC Width	% Reduction
Adder	28	20.45	18.6	9.05%
Adder2	40	16.8	14.8	11.90%
Alu	260	74.45	66.05	11.28%
Alugate	432	90	80	11.11%
Rip	448	88.7	79.15	10.77%
Cla	528	111.45	101.65	8.79%
Mult6	972	135.7	120.65	11.09%
c1355	2244	200.45	175	12.70%
mult2	4512	284.7	252.75	11.22%
c5315	10656	636.45	571	10.28%
c7552	14376	764	679.1	11.11%
			Average	**10.84%**

4. The Wtropic tool

The WTropic tool is a web based interface for the TROPIC layout synthesis tool. This interface is integrated in CAVE project and it follows the project specifications, so the WTropic implements all necessary functionality to allow the user to execute remotely the TROPIC.

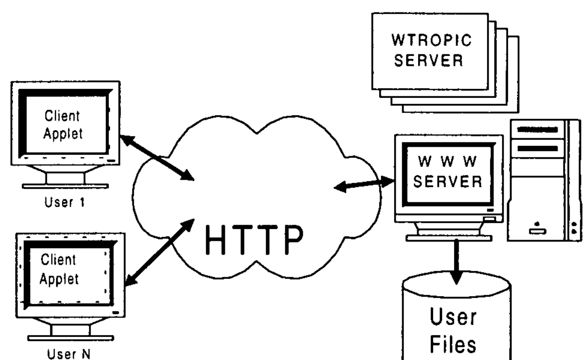

Figure 5. WTropic client-server structure

The TROPIC tool is a non-interactive tool, therefore a communication layer was implemented. On the server side, a server socket was developed to answer the client requests. A socket is a stream network connection. Sockets are the method used for communication on the Internet. When URLs are used, a socket provides a short-term or connectionless communication interface between the client and server. This results in the memoryless protocol associated with HTML and cgi-bin programs. However, there is a second kind of socket interface that stays active for the time that the connection client-server is active. This connection-based socket is what Java can provide. In essence, a private client-server interface is set up.

The client side was implemented using the Java applets-based interface. This interface provides a more efficient interface than HTML forms and CGI, because the data to be sent to the server can be validated at the client-side. Also, the client applet can help the user to construct his configurations file for the TROPIC tool, manage his data files on server side and the applet will guide the user throughout the physical synthesis. Figure 5 shows the Wtropic client-server structure.

Because multiple users may run the applet simultaneously, the server side of the program must provide multiple connections of the same type. This is done using the C language fork. The fork call creates a new version of the program each time a connection is established. Since a single socket can support more than one user, this is perfectly acceptable. In addition, the Tropic tool was developed using C language that has done easy its encapsulation in a C/C++ server socket.

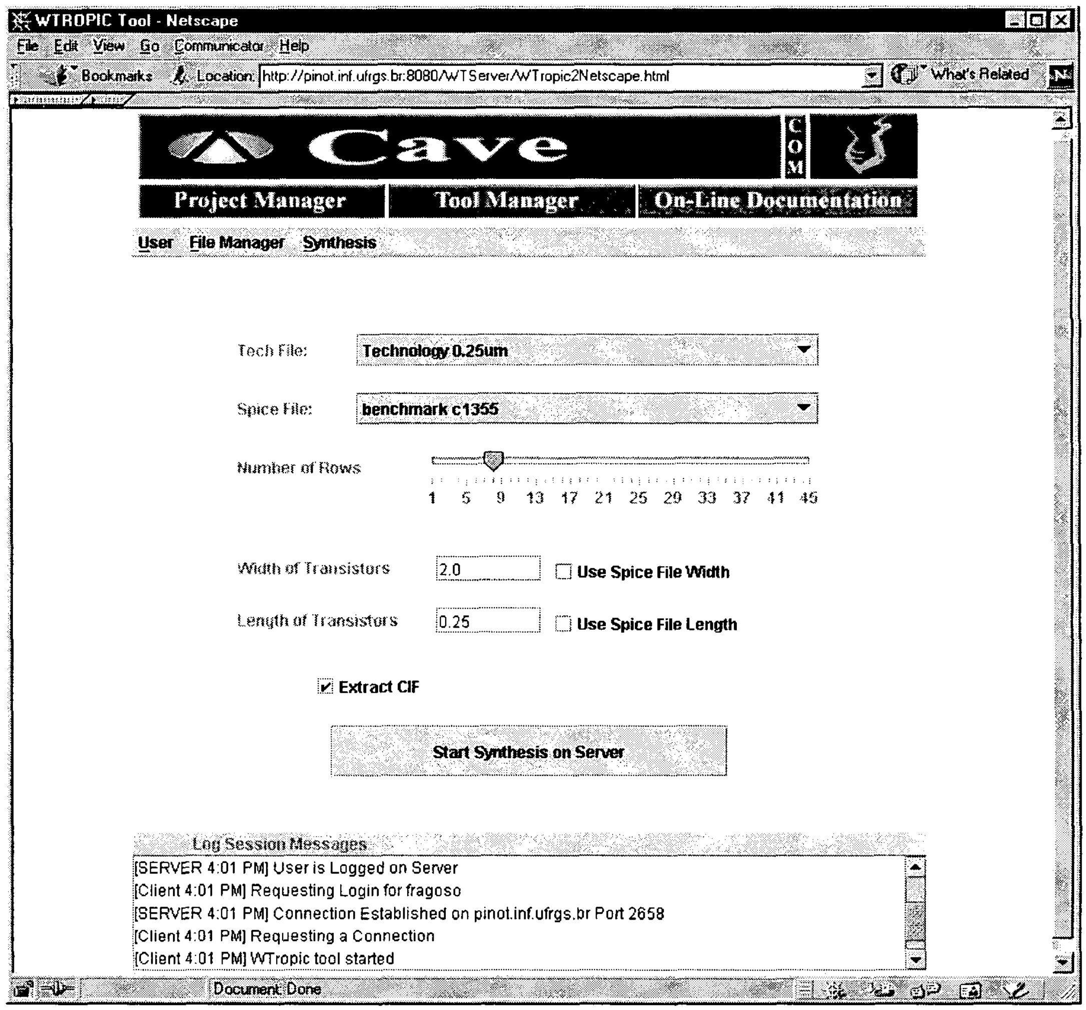

Figure 6. WTropic Interface

The socket connection allows the WTropic server to send back to client the results of the physical synthesis continuously. This feature allows the user to inspect the WTropic execution and the synthesis results remotely, before receiving the physical layout.
The Wtropic tool is integrated in the CAVE framework through hyperdocuments. The hyperdocuments contain the client-side applets that will be used to connect to Wtropic server. The Wtropic server can be installed in any host reachable on the Internet. More than one Wtropic server host can be provided to distribute the processing of client requests. The Figure 6 shows the interface of Wtropic and the three-level framework standard proposed in [5].

5. Conclusions

The web-based macro-cell generator, Wtropic tool, was presented. The Wtropic tool is a web-based layout synthesis tool that is a integrated environment for fast on-the-fly implementation of macro-cells.

The main goal of this work is to show that is possible to execute remotely a physical synthesis tool, integrated in a www-based framework, through the internet. The user can perform his synthesis from any simple machine with a web-browser. The Wtropic tool creates a user layer communication and a web-based interface to the TROPIC tool. The Wtropic tool was integrated in CAVE framework following all the CAVE project specifications.

The circuits generated with Wtropic are in average 10% smaller than the ones generated with Tropic.

6. References

[1] L. BENINI, A. BOGLIOLO, G. de MICHELI. *Distributed EDA tool integration: the PPP paradigm.* In: ICCD Proceedings, 1996. pp.448-453.

[2] A. R. NEWTON, et. Al. *WELD Project – Web-Based Electronic Design.* U.C. Berkley, 1996. Available via WWW at http://www-cad.eecs.berkley.edu/Respep/Research/weld/.

[3] M. SILVA, R. KATZ. *The Case for Design Using the World Wide Web.* In: ACM/IEEE DESIGN AUTOMATION CONFERENCE, DAC, 32., 1995. Proceedings... Los Alamitos: IEEE Computer Society, 1995.

[4] W. WESTFELDT, F. TOTH, N. JACOBSON, S. LEWIS. *Silicon Xpresso: Designing with the Web.* XCELL, Issue 30, 1998, pp.16-17.

[5] Leandro INDRUSIAK. *Ambiente de Apoio ao Projeto de Circuitos Integrados baseado no World Wide Web.* Masther Thesis. Porto Alegre: CPGCC UFRGS, 1998.

[6] J. GOSLING, B. JOY, G. STEELE. *The Java Language Specification.* SUN Microsystems. July 1996. Available via WWW at http://www.sun.com/doc/language_specification.html

[7] Leandro INDRUSIAK, Ricardo REIS. *Project Management and Design Methodology Support for the Cave Project: A Hyperdocument-Centric Approach.* In: SBCCI, 1999, pp 188-191.

[8] Fernando MORAES, Michel ROBERT, Daniel AUVERGNE. *A Virtual CMOS Library Approach for Fast Layout Synthesis.* In: VLSI, 1999, pp 415-426. Lisbon - Portugal.

[9] CADENCE. *"Virtuoso layout synthesizer - LAS - user guide".* CADENCE™ Version 4.2, October 1991

[10] Jue-H. CHERN, Jean HUANG, Lawrence ARLEDGE, Ping-C. LI, Ping YANG. *Multilevel Metal Capacitances Models for CAD Design Synthesis Systems.* IEEE Electron Devices Letters, v.13, n.1, p.32-34, Feb. 1992.

[11] Jason CONG, Andrew B. KAHNG, David NOICE; Nagesh SHIRALI, Steve H. YEN. Analysis and *Justification of a Simple, Practical 2 1/2D Capacitance Extraction Methodology.* UCLA Computer Science Technical Report 970013, 1996.