Logic and High Level Synthesis for Communication Protocols

Ricardo N. B. Lima[1,2] belem@gta.ufrj.br
Emerson Carli[1,2] emerson@gta.ufrj.br
Aloysio C. P. Pedroza[1,2] aloysio@gta.ufrj.br
Luci Pirmez[2] luci@nce.ufrj.br
Antônio C. de Mesquita[2] mesquita@lpc.ufrj.br

[1] Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro
[2] COPPE/EE - Programa de Engenharia Elétrica/Departamento de Eletrônica
Caixa Postal 68504 - CEP 21945-970 - Rio de Janeiro - RJ - Brasil
FAX: + 55 21 2906626

Abstract

This work presents the results obtained using a methodology that allows the hardware implementation and the rapid prototyping of communication protocols through Logic and High Level Synthesis. The implementation results of different description styles of protocols and a comparison among the protocol implementation using the High Level Synthesis technique with standard cells library and the implementation using Programmable Logic Device (PLD) are presented. The results include area analysis and clock frequency evaluation of synthesized hardware.

1 Introduction

This work presents the results obtained using a methodology that allows the hardware implementation and the rapid prototyping of communication protocols through Logic and High Level Synthesis. This methodology is based on the use of Estelle [6] and VHDL (*Very High-speed Integrated Circuits Hardware Description Language*) [5] languages, High Level Synthesis (HLS) techniques for digital systems design automation [4] involving two design environments, one for protocols design and another for the integrated circuit implementation.

Comparisons using different VHDL behavioral descriptions are shown in order to evaluate the complexity of the resulting hardware. Besides, a comparison among the protocol implementation using the HLS technique with standard cells library and the implementation using Programmable Logic Devices (PLD) is presented.

The paper is organized as follows. Section 2 presents the HLS concepts. Section 3 contains the main steps involved in the methodology for the hardware implementation of protocols. Different description styles for protocols in VHDL efficient for HLS are presented in section 4. In section 5, the obtained results are examined for the effectiveness of the hardware implementation proposed in terms of the synthesized circuits area and clock frequency. Final remarks are drawn on section 6.

2 High Level Synthesis

A complete description of a VLSI (*Very Large Scale Integrated*) circuit involves three domains of representation: the *behavioral*, the *structural* and the *physical* domain. Each domain can be hierarchically divided in several abstraction levels. HLS corresponds to a behavioral-to-structural mapping in a level higher than RTL (*Register Transfer Logic*). The behavioral synthesis tool used through out this work is the Synopsys Behavioral Compiler (SBC) [10]. It maps a high-level circuit design specification into a RTL equivalent level and then synthesizes this lower-level specification by the logical synthesis tool Synopsys Design Compiler (SDC) [11].

Communication protocols are control flow dominated machines. This simplifies the synthesis procedures for this kind of system. The starting point of the HLS procedure is a behavioral VHDL specification in the system level. The target architecture of HLS is a general computing model that contains data path, memory, and control elements.

This architecture is synthesized automatically by the SBC and SDC tools through scheduling, allocation, register sharing, memory and control inferencing. Designers do not need to specify the exact architecture of a design, and they can explore many implementations to find the optimal one.

3 The Adopted Design Methodology

The steps involved in the methodology of design for hardware implementation of protocols [3, 7] are summarized in the diagram of Figure 1. The procedure uses initially a suitable design environment of protocols in Estelle, the CAD Estelle. An Estelle specification correctly simulated (steps 1,2) is converted into a VHDL description using the Estelle-to-VHDL Translation Strategy (step 3) developed by Pirmez in [8].

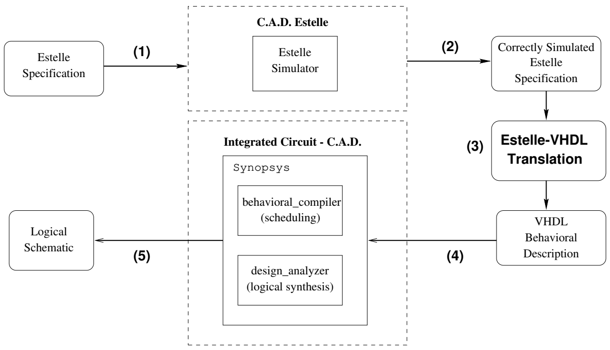

Figure 1: The design methodology.

This VHDL description will be used as input (step 4) of a HLS tool, that includes the SBC tool, SDC tool and Opus [2] layout editor. The SBC tool maps a high-level specification into a RTL or an equivalent level description which is synthesized by the logical synthesis tool (SDC) (step 5). The Synopsys tool perform also VHDL behavioral and structural simulations. Opus is applied to the layout generation of the VLSI structures [7].

4 Description Styles for Protocols in VHDL

A protocol can be described by many different VHDL behavioral descriptions. These descriptions are semantically equivalent and produce the same simulation results. However, they can produce results in the HLS procedure widely different. The quality of synthesis process is measured by the number of control states generated by the scheduling process performed by the SBC tool, which is responsible by the total area of the resulting integrated circuit. It will be shown that the quality of the HLS output description will depend on the type and order of statements used in the HLS input description, which characterizes a “description style” [9].

To illustrate this fact, consider the ISO’s ABRACADABRA reference protocol modelled by state machines that interact through the exchange of messages. This model will be used to illustrate the influence of the several description styles possible in VHDL on the HLS results. Table 1 shows the main characteristics of six different VHDL description styles used for implementing the state machine associate to the ABRACADABRA protocol. These different description styles are introduced in SBC and SDC tools. The circuits synthesized in each case and the complexity of the resulting hardware will be analyzed in the section 5.

5 Results

In this section, the effect of the different VHDL behavioral description styles in the synthesized hardware is evaluated and a comparison among the protocol implementation using the HLS technique with the standard cells library and the implementation using the PLD technique is shown.

Style	Main Characteristics
1st	1) "*case state*" clause 2) "*wait until...*" statement to each branch of "*case*" clause 3) output signals are reseted inside the "*case*"
2nd	equal to First style, but "*wait until...*" is placed outside the "*case*" clause
3rd	1) "*if..then..elsif..end if*" construct 2) test variable is *state* variable 3) "*wait until...*" is placed outside the "*if*" 4) output signals are reseted inside the "*if*"
4th	equal to Third style, but test variable is *input signal*
5th	1) "*case state*" clause 2) "*wait until...*" is placed outside the "*case*" 3) output signals are reseted immediatly after the "*case*"
6th	equal to Fifth style, but "*if..then..elsif..end if*" instead "*wait until...*"

Table 1: Main characteristics of VHDL description styles.

5.1 Description Styles

As discussed, VHDL allows the description of a given behavior in many different ways. In order to improve the performance of the synthesized hardware, it is necessary to adopt a VHDL description style that accounts for the structure of the scheduling algorithm used by the synthesis tool.

Styles	Control States	Area (mm^2)	Clock Frequency
1st	14	1,33	125 MHz
2nd	9	1,10	149 MHz
3rd	9	1,13	130 MHz
4th	9	1,37	130 MHz
5th	9	1,27	154 MHz
6th	4	0,66	154 MHz

Table 2: Effect of differences in VHDL descriptions on the synthesis results.

To illustrate this fact, a comparison among the synthesis results generated by the SBC tool for six different description styles is shown on Table 2. This comparison is done in terms of the number of states corresponding to the state machines generated by the synthesis algorithm for each description style, synthesized circuit area and clock frequency. The clock frequency evaluation is made from the critical path analysis found in the synthesized circuit. From this table some important conclusions are drawn:

1. use the "*wait until...control signals*" statement outside the "*case state*" clause. The circuit area decreases and clock frequency increases;

2. use the "*case*" statement instead "*if..then..elsif..end if*" structure. The circuit area decreases and clock frequency increases;

3. reseting the different output control signals inside the "*case*" clause the circuit area decreases. While reseting the different output control signals immediatly after the "*case*" the clock frequency increases;

4. employ the "*if..then..elsif..end if*" with the *state* variable as the outer signal instead the *input* variable. The circuit area decreases and the output controls are smaller;

5. use the "*if..control signals*" structure instead "*wait until...control signals*" constructs. The circuit area decreases.

In terms of circuit area and clock frequency the sixth description style presents the best results. In Figure 2 the logical schematic of the opening/closing connection process of the ABRACADABRA protocol generated by the HLS tool for the sixth description style is shown. The synthesized circuit has a 0,66 mm^2 estimated area, obtained using the AMS, 0,8μm technology standard cells library.

5.2 Comparison of Implementation Techniques

A comparison was performed using the opening and closing connection phase of the ABRACADABRA protocol. This specification was implemented using AMS, 0,8μm technology standard cells library and the ALTERA

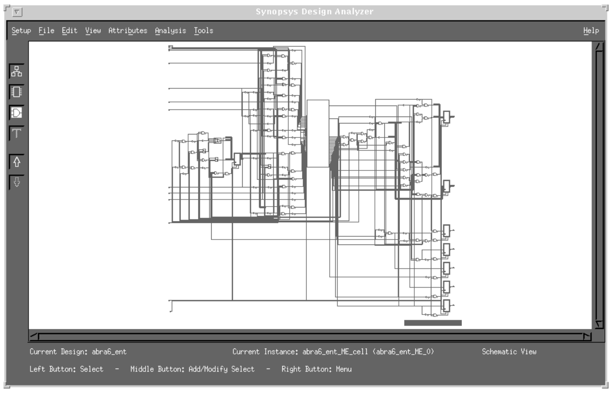

Figure 2: Logical Schematic of part of ABRACADABRA protocol.

Max+Plus II (*Multiple Array Matrix Programmable Logic User System*) design compiler [1].

This example shows a clear advantage of the standard cells implementation by report to PLD implementation. The clock frequency obtained in each case was 154 MHz for the standard cells implementation and 33 MHz for the PLD implementation. Moreover the area of the standard cells implementation was considerably smaller than the PLD implementation. Note that once chosen, the ALTERA component has a fixed area independent of the system being implemented. On the other hand, it must be stressed that PLD implementations are immediate, that is the system can be re-programmed without a new fabrication cycle, as is the case of standard cells, being an advantageous tool for rapid prototyping.

6 Final Remarks

The results obtained from a methodology that allows the hardware implementation and rapid prototyping of communication protocols through Logic and High Level Synthesis were presented. It was shown that the circuit resulting from the HLS procedure depends on the VHDL description style used to specify the system to the HLS tool. This was evaluated from the number of control states generated by the schedule step in the HLS process, that are directly coupled with the area and clock frequency of synthesized circuit. An example using the opening/closing connection phase of the ABRACADABRA protocol was discussed. Depending on adopted description style, a 49% reduction of the synthesized circuit area and a 23% increase of the clock frequency were noticed. The logical schematic of this protocol process, considering the best description style, was presented.

It was also shown that the standard cells implementation is 4.5 times faster than a PLD implementation. Nevertheless, PLDs are important tools for rapid prototyping of communication systems.

References

[1] Altera, Corp. *Data Book and Max + PlusII Getting Started*, 1997.

[2] Cadence Design Systems, Inc. *Cadence Online Library*, 1994.

[3] Carli, E. Projeto de Protocolos visando a Implementação em Hardware: aplicação ao Protocolo IPv6 no Ambito da Abordagem IP over ATM. Master's thesis, Universidade Federal do Rio de Janeiro, PEE/COPPE/UFRJ, Rio de Janeiro, Brasil, 1998.

[4] Gajski, Daniel D. et al. *High-Level Synthesis - Introduction to Chip and System Design*. Kluver Academic Publishers, University of California, Irvine, 1992.

[5] IEEE Std 1076-1987. *IEEE Standard VHDL Language Reference Manual*, Mar. 1988.

[6] ISO/TC97/SC21/WG16-1. Estelle: a Formal Description Technique based on a Extended State Transiction Model. *DP9074*, 1987.

[7] Lima, R.N.B. Projeto de Protocolos visando a Implementação em Hardware: Aplicação ao Caso do Protocolo AAL5 para Redes ATM. Master's thesis, Universidade Federal do Rio de Janeiro, PEE/COPPE/UFRJ, Rio de Janeiro, Brasil, 1998.

[8] Pirmez, L. *Uma Metodologia para a Implementação de Sistemas Distribuídos em Hardware a partir de uma Descrição Formal*. PhD thesis, Universidade Federal do Rio de Janeiro, PEE/COPPE/UFRJ, Rio de Janeiro, Brasil, 1996.

[9] Pirmez, L. and Rahmouni, M. and Kission, P. and Pedroza, A. and Mesquita, A. C. and Jerraya, A. A. Analysis of Different Protocol Description Styles in VHDL to High-Level Synthesis. In *EURODAC'96*, Geneve, 1996.

[10] Synopsys, Inc. *Behavioral Compiler User Guide Online Documentation, v1998.02*, 1998.

[11] Synopsys, Inc. *Synopsys Online Documentation, v1998.02*, 1998.