Hardware Implementation of Protocols for High Speed Networks from a Formal Description Technique

Emerson Carli*, Ricardo Neimo Belem Lima*, Aloysio C. P. Pedroza*†
Luci Pirmez‡, Antônio C. de Mesquita§
Universidade Federal do Rio de Janeiro
COPPE/EE – Programa de Engenharia Elétrica
C. P. 68504 – CEP 21945-970 – Rio de Janeiro – RJ – Brasil

Abstract

High performance in communication subsystems is obtained with software optimization of high speed protocols and with their implementation in hardware. This work follows a methodology that efficiently translates specifications described in a Formal Description Technique (Estelle) into VHDL, allowing high level synthesis of protocols. We chose two current protocols, IPv6 and AAL5, which are able to support the requirements of novel applications and of the development of high speed networks.

Keywords: IPv6, AAL5, Hardware Implementation, Formal Specification.

1 Introduction

The advances in VLSI and optical transmission technologies are leading to the development of faster and more reliable communication networks. In this way, the performance bottleneck in communication systems becomes the amount of processing needed to run protocols in workstations and network servers and no more the transmission rate on the network.

These changes in the technology and the arrival of multimedia applications require modifications in the project of protocols. On one side, the protocols need to be simpler to allow high speed operations. On the other side, they should be more efficient to support multimedia traffic.

However, high performance in communication systems is not always possible only with software optimization. Sometimes it is necessary to implement these protocols in

*Grupo de Teleinformática e Automação (GTA): {emerson,belem,aloysio}@gta.ufrj.br
†Departamento de Eletrônica/EE
‡Núcleo de Computação Eletrônica (NCE): luci@nce.ufrj.br
§Laboratório de Projeto de Circuitos Integrados (LPC): mesquita@lpc.ufrj.br

hardware. In this case, it is more convenient to begin the project through specifications based on Formal Description Techniques (FDTs), such as Estelle, making easier to model the hierarchy and the communication among processes besides allowing the use of more familiar tools to the protocol designer.

Aiming the hardware implementation of communication protocols, a methodology that consists of the translation of specifications in Estelle to descriptions in VHDL (*Very high speed integrated circuit Hardware Description Language*) [1, 2, 3] is used. This step is necessary for the use of specific environments for hardware projects, such as the tools for high level synthesis. VHDL is a hardware description language used by most of the current synthesis tools.

This work uses this methodology for the implementation of two current protocols in hardware: the IPv6 protocol, a new proposed version for the IP protocol and the AAL5 protocol, that transfer messages according to the ATM framework [4].

This work is structured as follows. Section 2 presents a Estelle-VHDL methodology and the description style adopted. Section 3 the IPv6 and AAL5 protocols are presented. Section 4 presents an architecture defined for this work. The simulation and VHDL synthesis results are detailed in section 5. The final comments and conclusions are shown in section 6.

2 Estelle-VHDL translation and Description Styles

The starting point for the translation is a correctly compiled and simulated Estelle specification , obtained in a suitable protocol design environment, that is mapped into a VHDL description which is used as input to a circuit design environment.

2.1 The Estelle and VHDL Languages

Estelle is a Formal Description Technique developed by ISO for the specification of distributed systems. Estelle is used to remove the ambiguities found in the informal descriptions of communication protocols.

VHDL is a standard language developed by the Department of Defense (DoD) for the description and simulation of hardware structures in different levels and description domains, going from the behavioural to the structural. The behavioural description allows the specification of the functions of designs using familiar programming languages forms. In the structural domain, the structures of a design are described, that is, it is shown how they are decomposed into sub-designs, and how those sub-designs are interconnected.

2.2 The Translation Strategy

An Estelle specification is composed of a set of modules hierarchicaly organized and a VHDL description is composed of a set of entities described separately. Thus, each Estelle module can be translated into a VHDL entity, as shown in Figure 1.

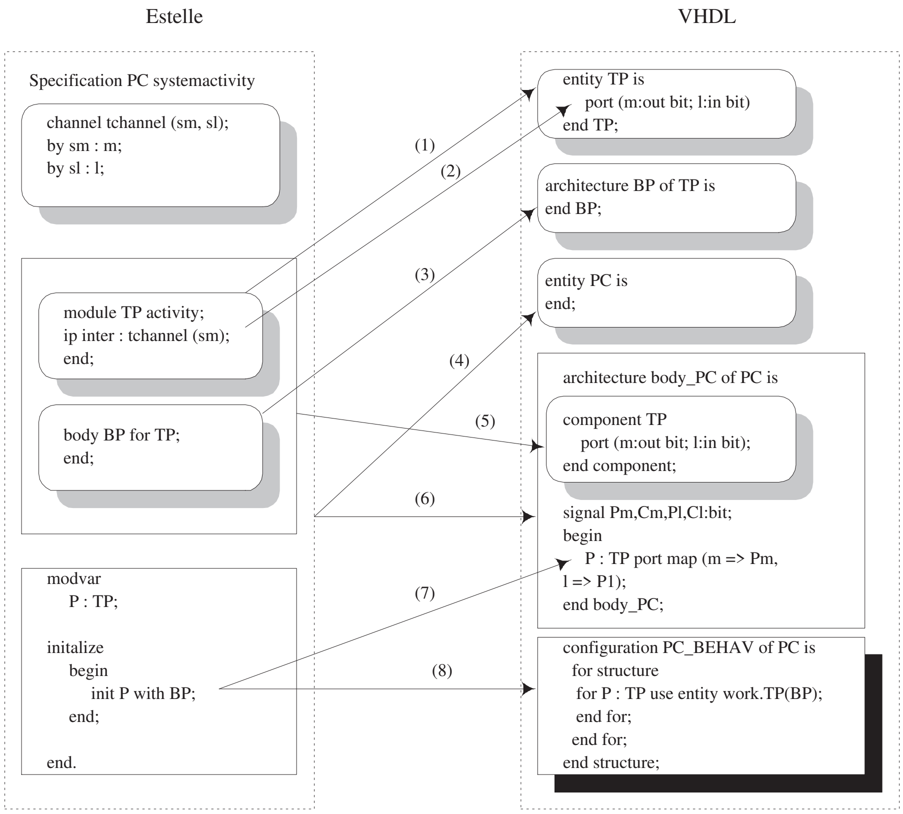

Figure 1: Estelle-VHDL Translation

The hierarchy among Estelle modules is mantained through a VHDL *component* declaration statement, as shown in Figure 1 (relation 5). Each VHDL component statement declares an entity and its description is placed in design libraries. The VHDL component instantiating construct is used to create VHDL entities intances (Estelle *init* function) whose ports are either connected to actual signals or to the entity ports (Estelle *connect* and *attach* functions), as shown in Figure 1 (relation 7). The binding of a VHDL entity to its instance (*component instantiation*) is achieved through a configuration declaration, as shown in Figure 1 (relation 8).

Each Estelle module is composed of two parts: a header (*module*) and one or more bodies (*body*). The external view is defined in the header and the internal behaviour is described in the body. Each VHDL description is also composed of two segments: an interface (*entity*) and a group of one or more bodies (*architecture*). In this way, the Estelle module-body pair corresponds to the VHDL entity-architecture pair, respectively, as shown in Figure 1 (relations 1, 3, 4 and 6).

Each interaction contained in the set of interactions of each interaction point of an Estelle module will correspond to a port (signal) of a VHDL entity, as shown in Figure 1 (relation 2). The signal type will be input (output) if the interaction point role is reception (transmission).

Since there are no VHDL equivalent mechanisms to the Estelle synchronous and asynchronous types of parallelism among module instances, an efficient Estelle- VHDL translation will require restrictions included in Estelle* [5]. These restrictions allows the representation of the parallelism among activity type module instances, in asynchronous form, within a system activity type specification.

The behaviour of a module is described by the initialization and the transition parts of the body. The execution of both parts is sequential. The translation of Estelle clauses into VHDL structures is trivial and the set of structures obtained in VHDL is included in a process statement. The Estelle clauses "**from** state", "**when** interaction-point.interation", "**provided** expression", "**to** state" and "**output** interaction-point.interaction(msg)" are directly converted into "**when** state => ...of case structure", "**wait until** interaction-condition", "**if** expression **then**..." , "**state** := state value" and "**interaction** <= msg" in VHDL. The Estelle instructions "**procedure** name (formal parameters queue)" and "**function** name (formal parameters queue): return" are directly converted into equivalent VHDL declarations.

A protocol can be described using different VHDL behavioural description styles. These styles provide semantically equivalent descriptions that differ only in the quality of the HLS result. The quality of this HLS output structural description is measured by the number of states and paths generated by the scheduling process performed by the HLS tool.

Different description styles are proposed and analysed in Luci et al. [6]. The style that gave the best results is described below and will be used in this work:

- the FSM protocol model corresponds to a VHDL process. The process is composed of two loops: an external loop that models the restart of the process and an internal loop that models the FSM of the protocol;
- the "**wait until...**" statement is combined with the inner loop where the control signals are checked;
- in order to avoid duplications in the code when specifying exceptions, the exceptions signals (reset, error situations) should be created and added to the sensitivity list of the wait statement. Thus, the exceptions signals are immediately checked after this wait statement;
- a state machine can be built using a "**case...**" or "**if...then...elsif...**" statement.

In both solutions, the decision to test initially the state or the input will depend on the processing. The first option will be used if the description is more dependent on states, otherwise, the second option will be chosen;

- after a "**case...**" or after an outer "**if...then...elsif...end if**" statement, the output control signals should be set to zero.

3 AAL5 and IPv6 Protocols

AAL5 and IPv6 belong to a new generation of protocols. They support high-speed networks and provide functionalities required by the new multimedia applications.

3.1 AAL5 Protocol

ATM (Asynchronous Transfer Mode) implements an universal network able to support a wide number of services. The adoption of ATM as a solution for high-speed networks implementation is due to the advantages that this technology introduces, such as flexibility, independence of the transmission rate and commutation of all services in only one structure.

Among the three layers contained in this technology - Physical, ATM and AAL (ATM Adaptation Layer) - the AAL layer adapts the whole current architectures of higher level to this new model. It makes the necessary adaptations to support the services required by the higher layers, using ATM technology.

There are several service classes supported by this technology and each one is provided of a different adaptation layer. This means that its purpose is to increase functionalities for each ATM adaptation layer, in a way to differ the service classes required by the higher layers.

Among those service classes and its respective adaptation layers, the AAL5 [7] was designed to be used in connection-oriented and connectionless services. The data transfer is made in ensured or not ensured operation, supporting error detection without recovery from lost packages. These options reduce the protocol processing and the transmission overhead, decreasing significantly the implementation complexity according to the needs of the new high-speed networks.

The AAL5 protocol is composed of two sublayers: SAR (Segmentation and Reassembly sublayer) and CPCS (Common Part Convergence Sublayer). The CPCS sublayer adds headers to the user data and assures integrity in transmission. The SAR sublayer is responsible for the fragmentation of the information, in order to be compatible with the ATM cells, and for the reassembly of the information enclosed into the incoming cells.

3.2 IPv6 Protocol

The current IP protocol (IPv4) became inadequate to support the new performance requirements and the Internet functionalities. In response, the research and engineering community have developed a "next generation" version of the Intenet protocol, officialy called IPv6 [8, 9]. In summary, the changes introduced by IPv6 can be grouped into three categories:

- Larger Addresses - The new address size is the most noticeable change. The IPv6 protocol quadruples the size of an IPv4 address from 32 bits to 128 bits.
- Flexible Header Format - IPv6 uses an entirely new datagram format. Unlike IPv4, which uses a fixed-format datagram header where all fields except options ocupy a fixed number of octets at a fixed offset, IPv6 uses a set of optional headers.
- Support for Multimedia Traffic - IPv4 provides minimal assistance in this area. IPv6 enables the labeling of packages belonging to a particular traffic flow for which the sender requests special handling. This aids in the support of specialized traffic such as real-time video.

The IPv6 protocol retains many of the features of IPv4, such as: (1) choose the "new hop" gateway or host for outgoing IPv6 datagrams and (2) fragment or reassemble the IPv6 datagrams.

As a first study, this work will approach the procedures related to the fragmentation and reassembly of IPv6 datagrams.

4 The Chosen Approach

The growth on the demand for information in the Internet directly affects the rate at which two computers can exchange data in a local area network. This is due to the access scheme of these networks that allocates bandwidth according to the global demand. As the number of users increases, the effective transmission rate decreases for each one of them.

ATM is seen as an emergent technology capable to overcome the current limits in the exchange of messages, once it allows the communication among several stations simultaneously. Besides, the high bandwidth availability causes great interest in its use.

Considering the future networks, the adopted architecture in this work is based directly on the IPv6 over ATM implementation strategy. This architecture can be seen in Figure 2, where the IPv6 and AAL5 protocols placement is shown. In the following section the synthesis results of the SAR sublayer will be presented.

It will be considered in this work that the virtual circuits used to transfer information are already established. Thus, the ATM signaling process as well as the existing

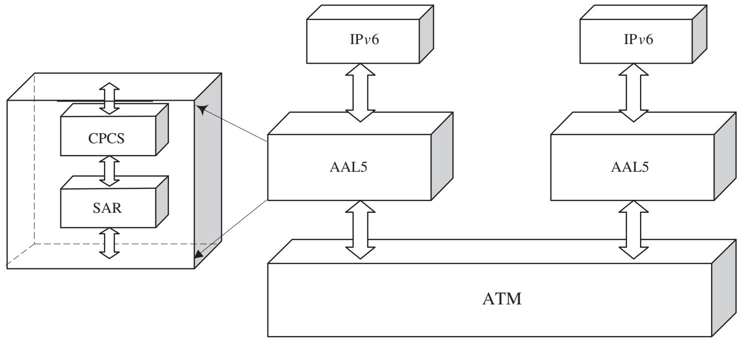

Figure 2: The chosen approach involving AAL5 and IPv6 protocols

communication among the user plane and the control plane will not be considered in this work.

5 Results

The Estelle specification of the AAL5 and IPv6 protocols were tested using a set of tools developed at GTA/COPPE/UFRJ. These tests included a simulation using a sequence of tests automatically generated and also a series of sequences manually generated. Thus, the correction of the specification of the protocols was verified.

After the translation of the AAL5 and IPv6 protocols into VHDL, simulations were made including exchanges of messages among users, showing that the protocol versions in VHDL corresponds to the Estelle specifications and that its translations were correctly made.

The AAL5 protocol contains about 750 lines of source code in Estelle while its description in VHDL has approximately 1100 lines. The IPv6 protocol contains around 1800 lines of source code in Estelle. That increase on the number of lines of source code in VHDL is due to the great redundance on the definition of the signals in the interfaces in VHDL.

In order to illustrate some important aspects of the synthesis of protocols, it was taken as example the SAR sublayer, responsible for the segmentation and reassembly of information.

Once the VHDL behavioural description of this sublayer was obtained, the DLS scheduling algorithm (*Dynamic Loop Scheduling*) was employed in order to use the logical synthesis tool Synopsys.

In addition, some experiences related to the storage of data in the SAR sublayer were made. These experiences are important due to the complexity of the generated hardware. The Table 1 below illustrates this fact.

As expected, this table shows that as larger the buffer for the storage of the information

Table 1: Size Buffer versus Synthesized Hardware

Buffer Size	Number of Synthesized Circuit Cells	Synthesized Circuit Total Area
Two buffers of 4 octets	1589	6,04 mm^2
Two buffers of 16 octets	2317	8,36 mm^2
Two buffers of 20 octets	2584	9,16 mm^2
Two buffers of 32 octets	3262	12,21 mm^2
One buffer of 62 octets and one buffer of 48 octets	4544	17,58 mm^2

is, larger and more complex the synthesized circuit will be. This fact should be taken into account by the protocol designer when specifying behaviourly a protocol, considering that a specification in a higher level of abstraction doesn't take it account the complexity of the hardware that will be generated.

6 Conclusions

This work has shown a methodology to translate Estelle specifications into VHDL descriptions. Two examples were developed, the AAL5 and the IPv6 protocols for high-speed networks. The correctness of the specifications was verified using simulation techniques. The experience has shown that the number of lines of source code in VHDL increases due to the redundancies on the definition of signals in the among VHDL entities interface.

The main motivation for using this translation is to facilitate the protocol designers task without experience in hardware design, that could use a hardware description language to obtain high performance subsystems. However, during a behavioural specification of a given protocol, it should be kept in mind that some structures result in more complex synthesized circuits.

References

[1] L. Pirmez, *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, 238 p., Rio de Janeiro, Brasil, 1996.

[2] L. Pirmez, A. Pedroza and A. C. Mesquita, "A methodology to developed integrated circuits from estelle specifications", in *38th Midwest Symposium on Circuits and Systems*, Brazil, 1995.

[3] L. Pirmez, A. Pedroza and A. C. Mesquita, "A methodology to the implementation of distributed system in hardware from a formal description", in *Fifteenth International Symposium on Protocol Specification, Testing and Verification*, Varsow, Poland, IFIP WG 6.1, 1995.

[4] E. Carli, R. Lima, L. Pirmez, A. Pedroza and A. C. Mesquita, "Implementação em hardware de protocolos para redes de alta velocidade a partir de uma descrição formal", in *II Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos*, Fortaleza, CE, Brasil, SFBSID'97, 1997.

[5] J. P. Courtiat, "Estelle*: A powerful dialect of estelle for OSI protocol description", in *International Workshop on Protocol Specification verification and Testing*, Toulouse-Moissac,France, 1988.

[6] L. Pirmez, A. Pedroza and A. C. Mesquita, "High level synthesis of protocols described by a formal description technique", in *IX IFIP International Conference on VLSI'97*, Gramado, RS, Brazil, 1997.

[7] International Telecomunication Union (ITU), ITU-T Recommendation I.363, *B-ISDN ATM Adaptation Layer (AAL) Specification*, Mar. 1993.

[8] S. Bradner and A. Mankin, *RFC1752: The Recommendation for the IP Next Generation Protocol.* Network Working Group, Jan. 1995.

[9] S. Deering and R. Hinden, *RFC1883: Internet Protocol, Version 6 (IPv6) Specification.* Network Working Group, Dec. 1995.