High Level Synthesis of IPv6 and AAL5 Protocols

R.N.B. Lima, E. Carli, A.C.P. Pedroza, L. Pirmez and A.C. de Mesquita

Universidade Federal do Rio de Janeiro
COPPE - Departamento de Engenharia Elétrica
FAX: +55 21 2906626
e-mail: {belem,emerson,aloysio}@gta.ufrj.br

Abstract

In this paper the hardware implementation of two protocols, the IPv6 and the AAL5, is proposed. The starting point of this implementation is a VHDL behavioral description of these protocols, which is translated by high-level synthesis into a structural description suitable to a VLSI integrated circuit implementation.

1. Introduction

The advances in optical transmission technologies are leading to the development of faster and more reliable communication networks. In these networks, the performance bottleneck becomes the amount of data processing needed to run protocols in workstations and network servers and not on the transmission capacity of the physical media.

Multimedia applications and services generate additional demand for data processing that require modifications in the design approach of protocols. In this case, the protocols need to be simple to allow high-speed operations and should be more efficient, as far as data processing is concerned, in order to support multimedia traffic. These performance requirements may not always be possible to obtain by traditional protocol design and software optimisation. This suggests alternative protocol design approaches that assume the hardware implementation of the protocol controllers.

In this work, a Design-for-Hardware implementation approach to the design of protocols is proposed, based on the use of VHDL (*Very High-speed Integrated Circuits Hardware Description Language*) [1, 2] and High Level Synthesis (HLS) techniques for digital systems design automation [3].

The starting point of the design process is a VHDL system level behavioral description [4] of the protocol similar to the protocol functional specification that is mapped into a structural description through HLS techniques. Of special concern in the HLS step is the use of a sheduling algorithm well suited to the synthesis of control flow dominated behavioral compilers [5]. In this case, the Dynamic Loop Scheduling (DLS) [6, 7, 8, 9] algorithm is employed. The resulting structural description is then used as input of the logical synthesis tool Synopsys [10] and finally the Opus [11] tool generates the circuit layout. In this work two protocols, the IPv6 and the AAL5 [12] are synthesized as a means to evaluate the complexity of the resulting hardware.

The contents of the paper is outlined in the following. In section 2 the main characteristics of the IPv6 and AAL5 protocols are briefly discussed. In section 3 the proposed approach of design-for-hardware implementation of protocols is presented and the main steps involved are discussed. In section 4, the obtained results are examined for the effectiveness of the hardware implementation proposed in terms of area of the synthesized circuits. Finally, conclusions are drawn on section 5.

2. The AAL5 and IPv6 Protocols

A. The AAL5 Protocol

The adoption of ATM (*Asynchronous Transfer Mode*) 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 a single structure. Among the three layers contained in this technology - Physical, ATM and AAL (ATM Adaptation Layer) - the AAL layer is the interface between all current architectures of higher level to the ATM model. It makes the necessary adaptations to support the services required by the higher layers when using ATM technology. There are several service classes supported by the ATM technology, each one is provided with its own adaptation layer. Among these service classes and their respective adaptation layers, the AAL5 [13] layer was devised to used in connection-oriented and connectionless services. In this case, the data transfer is done in ensured or not ensured operation, supporting error detection without recovery of 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 of the 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.

B. The IPv6 Protocol

The IPv6 protocol [14, 15, 16] was introduced as the "new generation" version of the current Internet Protocol (IPv4), which is unable to support the new performance requirements and Internet functionalities. The main differences between the IPv6 and the IPv4 protocols are:

- 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 as, for example, the support of specialized traffic such as real-time video.

The IPv6 protocol retains many of the features of IPv4. For example, the IPv6 protocol is still a connectionless protocol. In this work, some basic functions of the IPv6 protocol, like the procedures related to the fragmentation and reassembly of IPv6 datagrams, will be implemented in hardware.

3. The Proposed Protocol Design Methodology

The steps involved in the proposed methodology of design-to-hardware implementation of protocols are summarized in the diagram of figure 1. The procedure involves the use of a set of standard tools for VLSI circuit design which, in the figure is called IC-CAD. This includes the Synopsys logic synthesizer [10] and Opus layout editor [11].

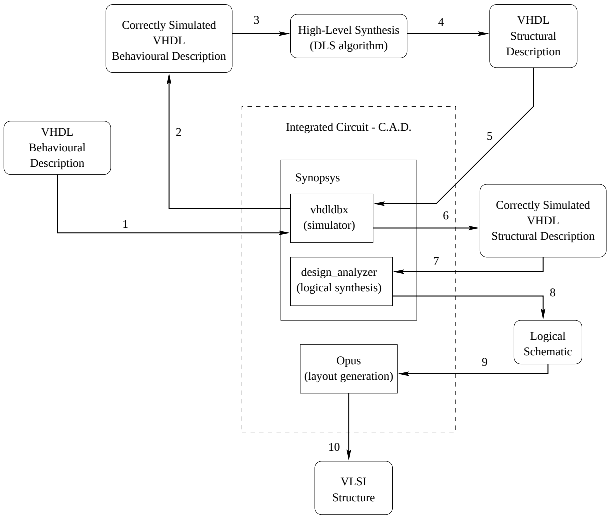

Fig. 1: The proposed design methodology.

Synopsys can be used in simulation of VHDL behavioral and structural descriptions and in logical synthesis of structural descriptions. Opus is applied to the layout generation of the VLSI structures, after a logical level description of a correctly simulated behavioral description of the protocol (1,2) is obtained. The High-Level Synthesis step is implemented by the DLS scheduling algorithm (3,4). It maps the input behavioral description into a finite state machine (FSM) description that can be considered as the input structural description of Synopsys. Prior to logical synthesis (7,8), a simulation step (5,6) is performed in order to verify the correctness of the obtained structural descriptions. The resulting logic description is then submitted as input of the Opus tool for layout generation (9,10).

A. The VHDL Behavioral Description

The basic component of an VHDL description is the "entity/architecture" pair [1]. The VHDL "entity" describes, through a set of input/output ports, the external interface of a given system. Ports are the signals that enable the exchange of information between VHDL entities. The "architecture" describes the entity's function either in the behavioral or the structural domains. In the case of protocols, for an efficient information exchange between VHDL entities it is necessary some kind of a synchronization among them in such a way that, an entity cannot send data to another one until it is enabled to receive the data. This synchronization is stablished by control signals included in both entities.

B. High Level Synthesis

The HLS process consists mainly in a scheduling step. To this purpose the DLS algorithm is employed. The input to DLS is a VHDL behavioral description of the protocol. The main steps of the DLS algorithm are:

- Control-Flow Graph (CFG) extraction
- Generation of paths
- Generation of states and transitions

The internal representation of the system used by the DLS is a control-flow graph. It provides a convenient way to represent inherent properties of control-flow dominated circuits that are described by nested loops and synchronization constructs such as *wait* statements. In the CFG the nodes represent the operations of the original description and the edges indicate the flow of signals. In this CFG the *wait* nodes are of basic concern since it indicates to the algorithm the beginning of a new path. In fact a *wait* statement always means that a change of state will take place. A path $C = (n_1, n_2, ..., n_i)[n_s]$ is defined as a set of nodes, starting at n_1 and having as successor a new path started at n_s. Starting at the first node of the CFG, new paths are generated as the algorithm proceeds. The generation of a given path is stopped and the generation of new path begins when one of the following two conditions is met:

- A *wait* statement is found (Condition C_1)
- A new node that is data dependent on one of the nodes already in the path is attained (Condition C_2)

The first path always starts with the first node in the CFG. At the generation of sates and transitions phase, each generated path will constitute a transition of a finite state machine (FSM). Appended transitions will be generated due to the implicit loop in all *wait* statements. All paths starting with the same node are placed together in the same state. The result of DLS is, therefore, a transition table that can be handled by the logical synthesis tool.

4. Results

Following the procedure outlined in the previous section, a VHDL behavioral description of the AAL5 and IPv6 protocols was iniatilly obtained and simulated using the Synopsys tool. This included the message's exchanges between entities. Once the functional specifications were verified, these descriptions were submitted to the DLS scheduling algorithm. In this case, the structural description obtained consisted of a single finite state machine of each protocol. The corresponding state-transition tables were generated by DLS. The IPv6 behavioral description produced a transition table with 13 states. For the AAL5 protocol a transition table with 29 states was obtained. The structural description of both protocols was simulated with the same data exchange sequence of the behavioral descriptions, cheking the correct usage of the algorithm.

In order to illustrate some relevant aspects of the protocol synthesis, some experiments have been performed relative to data storage at the IPv6 protocol and at the SAR sublayer in the AAL5 protocol. The tests performed are time consuming due to the hardware complexity generated. A first test performed over the logic descriptions generated by Synopsys, consisted in associating different length buffers to the data storage. Figure 2 illustrates some values taken from the SAR sublayer of the AAL5 protocol. In the figure, starting with a buffer's length of 4 octets a circuit with a estimated area of 6,04 mm^2 was obtained. A buffer with 32 octets in length produces an area estimate of 12,21 mm^2. These values were obtained using the ES2, 0,7μm technology standard cells library [17]. In the case of the IPv6 protocol the obtained values are depicted in figure 3. Figure 4 presents a small part of the logical schematic generated by Synopsys.

5. Conclusions

The advances in 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. However, high performance in communication systems is not always possible only with software optimization. The solution proposed in this work was the hardware implementation of AAL5 and IPv6 protocols in order to overcome the data processing bottleneck imposed by them. To obtain a first evaluation of the implementation cost in terms of layout area as a function of the data staorage buffer length, some tests were performed and the results discussed.

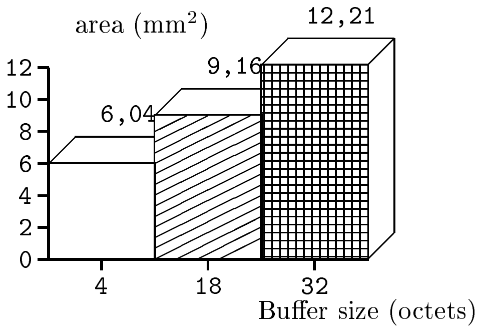

Fig. 2: Buffer Size × Synthetized Hardware Area.

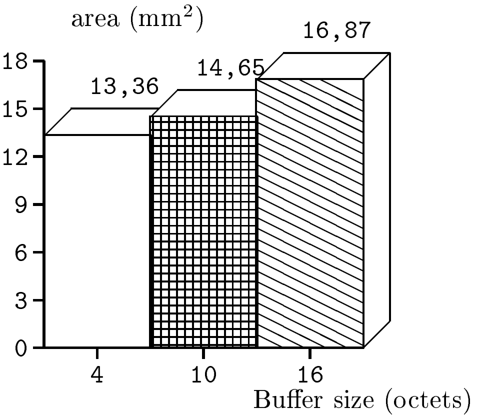

Fig. 3: Buffer Size × Synthetized Hardware Area.

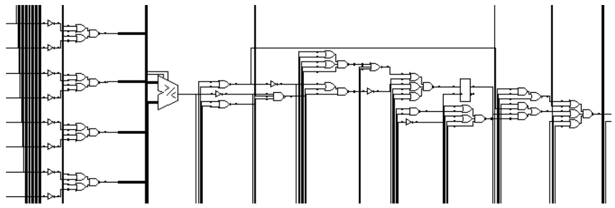

Fig. 4: Small part of the IPv6 logical Schematic.

References

[1] P. J. Ashenden, *The VHDL Cookbook.* Computer Science Dept. of University of Adelaide, 1990.

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

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

[4] L. Pirmez, M. Rahmouni, P. Kission, A. Pedroza, A. C. Mesquita and A. A. Jerraya, "Analysis of different protocol description styles in VHDL to high-level synthesis", in *EURO-DAC'96*, Geneve, 1996.

[5] Walker, R., Camposano, R., *A Survey of High-Level Synthesis Systems.* University of California, Irvine: Kluver Academic Publishers, 1991.

[6] Rahmouni, M., O'Brien, K., "A Loop-Based Scheduling Algorithm for Hardware Description Languages", *Journal of Parallel Processing Letters*, vol. 4, no. 3, no. 3, pp. 351–364, 1994.

[7] O'Brien, K., Rahmouni, M., Jerraya, A. A., "DLS: A Scheduling Algorithm for High-Level Synthesis in VHDL", in *EDAC'93*, Paris, Franca, 1993.

[8] Camposano, R., "Path-Based Scheduling for Synthesis", *IEEE Transactions on CAD*, vol. 10, pp. 85–93, Jan 1991.

[9] Camposano, R., Bergamaschi, R. A., "Synthesis Using Path-Based Scheduling: Algoritms And Exercises", in *27th DAC*, Orlando, pp. 450–455, Jun 1990.

[10] Synopsys, Inc., *Synopsys Online Documentation, v1997.01*, 1997.

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

[12] Carli, E., Lima, R. N. B. et al., "Implementação em Hardware de Protocolos para Redes de Alta Velocidade a partir de uma Descrição Formal", in *II Simpósio Franco-Brasileiro de Sistemas Informáticos Distribuídos, SFBSID'97*, Fortaleza, CE, pp. 211–220, Nov 1997.

[13] ITU-T Recommendation I.363, *B-ISDN ATM adaptation layer (AAL) specification*, 1993.

[14] Bradner, S., Mankin, A., *The Recommendation for the IP Next Generation Protocol.* RFC 1752, Harvard University, ISI, Jan 1995.

[15] Deering, S., Hinden, R., *Internet Protocol Version 6 (IPv6) Specification.* RFC 1883, Xerox PARC, Ipsilon Networks, Dec 1995.

[16] Stallings, W., "IPv6: The New Internet Protocol", *IEEE Communications Magazine*, Jul 1996.

[17] ES2 - European Silicon Structures, Silicon Processing Department, *ECPD 07 - Dual Layer Metal 0.7 Micron Design Rules*, Oct 1995.