Open FPGA-Based Development Platform for Fuzzy Systems with Applications to Communications

Federico Montesino, Ángel Barriga, Diego R. Lopez, Santiago Sánchez-Solano

Abstract

Soft computing techniques are gaining momentum as tools for network traffic modeling, analysis and control. Efficient hardware implementations of these techniques that can achieve real-time operation in high-speed communications equipment is however an open problem. This paper describes a platform for the development of fuzzy systems with applications to communications systems, namely network traffic analysis and control. An FPGA development board with PCI interface is employed to support an open platform that comprises open CAD tools as well as IP cores. For the development process, we set up a methodology and a CAD tools chain that cover from initial specification in a high-level language to implementation on FPGA devices. PCI compatible fuzzy inference modules are implemented as SoPC based on the open WISHBONE interconnection architecture. We outline results from the design and implementation of fuzzy analyzers and regulators for network traffic. These systems are shown to satisfy operational and architectural requirements of current and future high-performance routing equipment.

1 Introduction

Soft computing techniques, and fuzzy systems in particular, are gaining momentum as tools for network traffic modeling, analysis and control. Fuzzy systems find applications in a number of areas such as traffic regulation in routers [1], support for differentiated services within the DiffServ architecture policy and quality of service evaluation real-time traffic measurement, analysis and monitoring, power saving for wireless networks, as well as end-to-end traffic control [2] and end-to-end control for wireless networks [3].

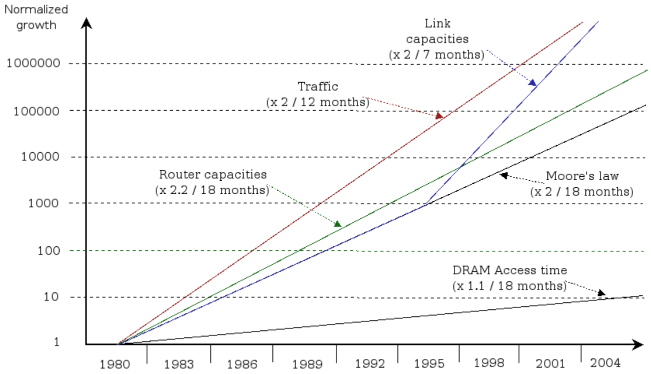

Figure 1: High-end routers technological trends [4, 5, 6].

Technological trends in Internet core routers and high-end communications hardware in general (see figure 1) lead to hard constraints specially regarding packet processing rates. Within this context, two main constraints arise: scalability (processing units must be able to process up to millions of packets per second), and flexibility and reconfigurability of implementations (required because of the fast increasing diversity of protocols and technologies involved).

Both academia and major vendors are currently pushing for distributed and modular router designs, where routers are composed of modules that can be mapped onto different processing elements and communicate through well-defined open interfaces over an internal network [4]. Hardware for high-end communications systems has been traditionally developed in a custom and unstructured manner. Nevertheless, reconfigurable architectures are employed in practice by most vendors and design methodologies for easing the development process are sought.

Federico Montesino Pouzols, Angel Barriga Barros and Santiago Sánchez-Solano are with the Microelectronics Institute of Seville (IMSE-CNM), CSIC/University of Seville, Avda. Reina Mercedes s/n. Edif. CICA. E-41012 Seville, Spain (phone: +34 955 056 666; fax: +34 955 056 686; email: {fedemp,barriga,santiago}@imse.cnm.es).

Diego R. Lopez is with RedIRIS, the Spanish National Research and Education Network, Edificio Bronce, Pza. Manuel Gómez Moreno s/n. Planta 2. E-28020 Madrid, Spain (phone: +34 912 127 625; fax: +34 915 568 864; email: diego.lopez@rediris.es)

This work has been supported in part by projects TEC2005-04359/MIC from the Spanish Ministry of Education and Science and TIC2006-635 from the Andalusian regional Government.

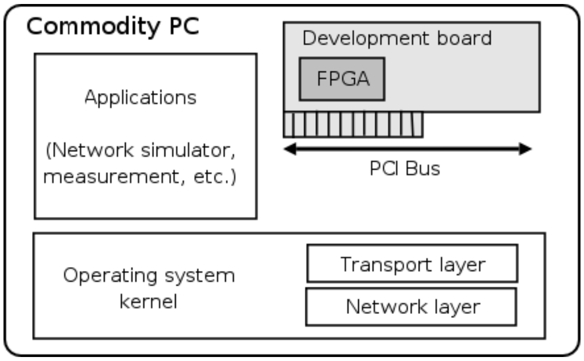

Figure 2: Prototyping platform scheme

Although fuzzy systems are usually computationally intensive, its performance can be boosted by means of hardware implementations based on optimized architectures. The hardware implementation of fuzzy systems is a well established field of microelectronics [7], thus making them a feasible solution for processing massive traffic volumes in real-time. We present an open FPGA-based platform for the development of fuzzy components for communications systems that has been succesffully employed to develop intelligent traffic analyzers and regulators that can achive real-time operation within current high-performance Internet routers. The platform has been developed with a twofold objective in mind:

- Making it possible the efficient implementation of a number of fuzzy systems proposed throughout the last years.
- Fostering the research on fuzzy logic based solutions to Internet traffic analysis and control.

2 Prototyping platform

Throughout more than a decade, strategies and methodologies for prototyping fuzzy logic based controllers have been developed. To date, most work on this topic has been focused on industrial applications [8]. Considering the specific requirements as well as the high cost and complexity of high performance routers deployed on the Internet today, we have developed a flexible prototyping platform for Internet traffic analysis and control.

Additional requirements considered are seamless integration into current router architectures [6], flexibility, and performance scalability up to higher requirements in current and foreseeable network technologies. The platform provides a complete set of tools and an environment for easing the development of prototypes of fuzzy systems for communications as well as performing its validation.

Our prototype development architecture, outlined in figure 2, is based on a commodity PC equipped with an FPGA development board with PCI interface, thus making a flexible and cheap solution with no specific hardware requirements yet able to emulate the behavior of complex and expensive routing equipment.

When implementing fuzzy systems, two main function blocks are distinguished: those directly related to fuzzy inference and those that can be classified as auxiliary functions (such as initialization, timing, pre/post-processing, etc.) [8]. For the implementation of prototypes of fuzzy systems for analysis and control of Internet traffic the following model is used:

- As fuzzy inference modules (FIM) are the potential system bottlenecks, they are implemented on FPGA devices and described by means of VHDL according to an specific implementation architecture [7] tailored for efficient and fast fuzzy inference. The methodology and tools employed for the development of the FIM is described in the next section.
- In the basic configuration of the platform, all auxiliary functions are implemented as software. Software can run on the PC operating system as well as on optional components implemented on the FPGA of the development board.

A flexible and open architecture for implementing fuzzy systems on the FPGA has been defined. Within this architecture, depicted in figure 3, FIM modules are integrated as subsystems of a potentially complex and reconfigurable fuzzy logic based digital system.

Interconnection between the host PC and the fuzzy inference module (FIM) is done through a standard PCI bus. The internal bus of the fuzzy digital system conforms to the WISHBONE [9] public domain standard. WISHBONE is a SoC interconnection architecture for portable IP cores, that connects a variable number of components.

While the WISHBONE-PCI bridge and WISHBONE controller (using the Conbus IP core) are implemented in Verilog as provided by OpenCores, the FIM module is implemented in VHDL. All top level interfaces are designed to be WISHBONE compatible. Both the WISHBONE bus controller and the PCI-WISHBONE Bridge IP cores have been developed under free distribution licenses by the OpenCores [10] organization as well as other entities.

WISHBONE Systems can easily interface with other SoC bus standards, such as the On-Chip Peripheral Bus (OPB) license). In its basic configuration, the system comprises three cores: the WISHBONE controller, the PCI-WISHBONE

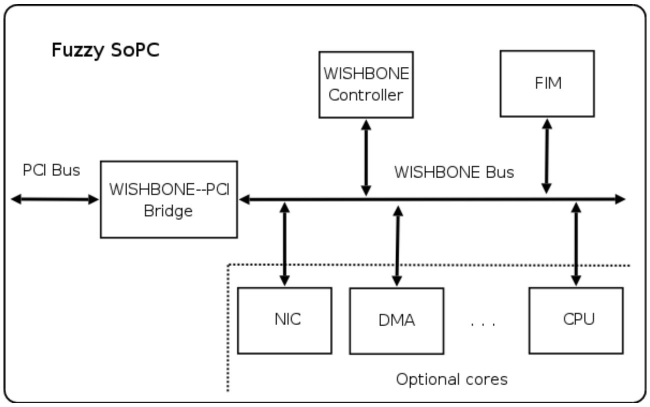

Figure 3: Fuzzy SoPC as a PCI device

bridge (as a master WISHBONE device), and a fuzzy logic control module (FLC) (as a slave WISHBONE device). The WISHBONE controller IP module employed supports up to 8 master and 8 slave devices.

This way, software tasks can be defined using common programming languages and can be run on the generic purpose processing units of the PC as well as on specific processing units implemented on the FPGA. For instance, a fuzzy logic based traffic analysis application can be implemented incorporating an OpenRisc Core for which the GNU/Linux operating system is available. The PCI interface of the prototypes eases integration with routing architectures by major vendors [6, 11].

Within routing architectures currently deployed in the Internet [5, 6], fuzzy traffic analyzers and controllers could be seamlessly integrated as processing engines whether at the network processing unit (NPU) and/or the output/input cards depending on the quality of service architecture implemented on the router. In addition, when the development board employed includes a network interface card, as is the case for our AvNet board, a whole fuzzy logic based traffic analysis application can be implemented as a standalone SoPC on the FPGA.

As we will describe in section 4, the configuration presented so far allows for the implementation of fuzzy processing units that can be generally applied to network traffic analysis and control. Additional IP cores (such as network interface card control, DMA devices and CPUs) can be incorporated in order to develop extended fuzzy processing units or complementing them with extensions for other soft computing techniques.

Within the prototyping platform presented, a fully automated design flow has been employed. The fuzzy systems design flow, described in the next section, covers from an initial high level fuzzy system description to an FPGA implementation of FIMs by means of the tools included in the Xfuzzy development environment as well as tools in the Xilinx ISE environment.

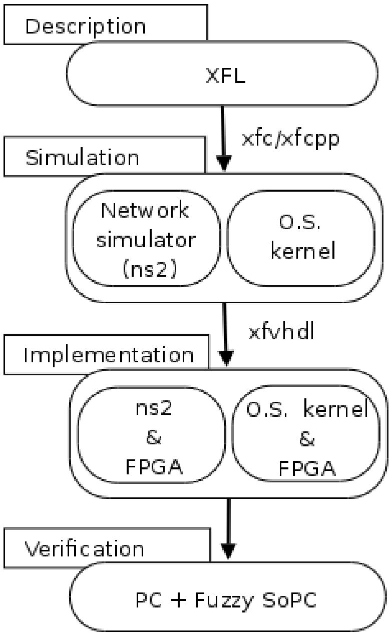

Figure 4: Design Flow of Fuzzy Systems for Communications

3 Development methodology and design flow

As a result of more than 10 years of research experience on the digital implementation of fuzzy systems, the fuzzy group at IMSE has developed methodologies and CAD tools that fulfill the design flow of fuzzy systems. We leverage on the Xfuzzy [12] CAD suite of tools and a methodology [8] for the development of fuzzy controllers to define a methodology and design flow tailored for the development of fuzzy systems applied to Internet traffic analysis and control.

The design flow we have defined and applied in order to develop fuzzy inference modules is depicted in figure 4. The design flow covers the whole development process, from initial specification to final implementation whether as software or hardware.

The first stage (description) is performed using a high level fuzzy systems specification language, XFL [13], which is turned into C and VHDL code (among other implementation options) by means of the tools included in the Xfuzzy development environment. The tool chain comprises:

- The xfc and xfcpp tools (included in Xfuzzy), which turns an XFL specification into C and C++ code that can be employed in both user and kernel space.

- The xfvhdl tool (also part of Xfuzzy), which turns an XFL specification into synthetizable VHDL code generated for a specific efficient parallel architecture for the implementation of fuzzy systems [12]. xfvhdl applies an active-rule driven architecture for fuzzy inference, using simplified defuzzification methods and parallelization in order to provide high inference rates. The output of xfvhdl can be feeded to a number of synthesis tools, such as those from Xilinx and Synopsys. As we wil show in section 4, this architecture can provide efficient implementations of fuzzy systems even with a high number of variables, linguistic terms and rules.

- ns-2 [14], an open network simulator widely spread within the Internet research community.

- Operating system kernel (currently Linux and FreeBSD).

The development stages after specification have been tailored for Internet traffic controller development as follows.

- For both simulation and implementation, we have defined as general hardware-software partition the implementation on hardware of the FIM module and its interfacing logic whereas all other tasks are implemented as software running on a PC.

- Two options are considered for simulation: network simulation in user space (ns-2) and network emulation in kernel space.

  - ns-2 has been used for simulation in user space. ns-2 is an object oriented discrete event driven simulator with support for a vast variety of transport protocols, queueing systems, routing schemes and access media, thus enabling us to evaluate the performance of traffic controllers under complex and realistic simulated scenarios. Fuzzy controllers are integrated into ns-2 as components implemented in C.
  - Simulation in kernel space allows for analyzing emulated scenarios where a router is emulated by a prototyping PC. This is accomplished by replacing queue control functionality in the operating system network layer with functionality provided by software implementations of FIMs.

- Similarly, implementation in real and simulated scenarios is also possible by using the prototyping PC whether as a simulation platform or as a router. To this end, kernel drivers have been developed to make it possible to access the fuzzy controller in the FPGA development card from networking modules in the operating system kernel as well as user space.

  - By replacing queue control functionality in the operating system network layer with functionality provided by the FIM in the development board, a full prototype implementation can be validated in real scenarios, where the prototyping PC acts as a router. Drivers have been developed for FreeBSD 6.x and Linux 2.6.x kernels.
  - For implementations in simulated scenarios, queue control functionality in ns-2 is replaced with functionality provided by the FIM in the development board.

Implementation of novel hardware components and experimental deployment on high-end equipment poses major practical problems. Deployment on high-end (around and above 1 million euro cost per unit) routing equipment requires the adoption of a new technology by vendors of routing hardware (a market with high inertia), which is a long term objective of our research. Nonetheless, by means of our prototype architecture, verification can be performed the same way as simulation through emulated scenarios.

By following a well defined development methodology, we provide a much more efficient and formal approach that those currently used for the development of Internet routers from major vendors [6, 5].

4 Application to Internet traffic analysis and control

This section provides results (in terms of inference rate, occupation and power consumption) of a set of example applications of the described platform to the area of Internet traffic analysis and control.

A number of research results have been reported on the application of fuzzy systems to the general area of network traffic regulation [1]. Additionally, recent results have been obtained on the development of a fuzzy queueing theory as an extension to classical queueing theory [1], which is the basis of many traffic processing mechanisms in the current Internet.

The prototyping platform described in previous sections has been employed in order to develop Internet traffic analyzers and regulators. A feasibility study has shown that implementations of fuzzy inference systems on FPGA devices can satisfy operational requirements of current and fu-

System	Inputs	Linguistic terms	Rules
RxBufferSize	2	5,5,5	25
AQMBestEffort	2	7,7,7	37
DSSelect	2	5,5,2	14
AQMDSAF	2	3,3,4	7
XAIMD	2	5,4,4	5
RTperf	4	5,5,5,5,5	27

Table 1: Complexity of fuzzy systems.

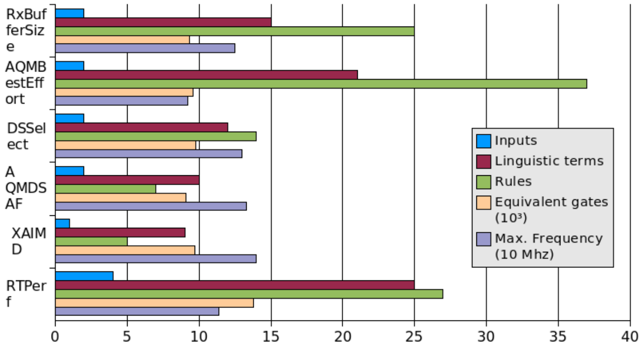

Figure 5: Implementation results summary

ture high performance routing hardware in terms of both inference speed and resource consumption.

A summary of implementation results is presented in what follows. A more detailed report can be found in [15]. We outline the results of the microelectronic implementation of a set of fuzzy systems prototypes on FPGAs. The focus is on the implementation results for FIM modules as they are the key component with higher operational requirements. All the prototypes have been implemented on a Xilinx Spartan-3 FPGA, xc3s1500- fg456-5 device (1.5 millions of equivalent gates) included in the development board employed, an AvNet ADS-XLX-SP3-EVL1500.

The tool xfvhdl was used to generate VHDL descriptions from XFL specifications as described in section 2. xfvhdl provides several FIM implementation options. In particular, we set ROM based storage for both the rule base and membership functions.

A performance evaluation of the FIM architecture in terms of inference speed, area and power consumption was conducted. Synthesis as well as place and routing were performed by means of the tools included in the Xilinx ISE environment, namely `xst` and `par`. ISE 8.1i, `xst I.24` and `par I.24` were employed for the reported prototypes.

We will focus on the fuzzy systems listed in table 1 which implement intelligent Internet traffic analysis and regulation systems [16, 2, 15]. Figure 5 shows a summary of implementation results for the above listed systems for a precicion of 8 bits for inputs, outputs and membership function, which was found to satisfy overall precision requirements. These systems perform dynamic adjustment of reception buffers [15], active queue management in best-effort and differentiated services schemes [16], and analysis of the network performance from a real-time application viewpoint [2]. The systems for active queue management have been shown to clearly outperform schemes deployed in commercial systems.

In current router architectures, traffic analysis and regulation subsystems are integrated into output interface cards together with the virtual output queue processing logic. In general, these subsystems can be thought of as queue schedulers that run, at most, at frequencies around the maximum per interface packet processing speed.

As for inference rate, prototypes implemented on a Xilinx Spartan-3 FPGA could achieve around 100 MFLIPS. Routers from the Cisco 12000 and CRS series as wel as Juniper M & T series process up to 25 Mpackets/s [17] per interface output queue. Thus, even our prototype implementation using medium cost FPGAs can provide the required inference speed in current high performance routers.

Power consumption, between 13 and $335mW$ for the FPGA circuit, turns out to be negligible for current high performance routers as it is two to three orders of magnitude below the overall consumption of an output interface card. Also, processing units of current routers have a power consumption of the same orders of magnitude and above.

A number of hardware implementations of fuzzy systems for tasks belonging to the physical and link layers of communications systems (such as signal filtering) have been reported in the literature. Some of them are based on FPGAs. However, we are not aware of proposals of FPGA based implementations of fuzzy systems applied to network traffic control and network layer tasks in general.

The most closely related work we are aware of [18] reports an inference rate of 3.3 MFLIPS for a 60Mhz clock, which would not fulfill current requirements. We note though that there are major differences between our proposal and the aforementioned work. In the latter case, the target application is traffic control for ATM networks. Additionally, it is based on a substantially different architecture (using the concept of fuzzy processor) and the controller is implemented as ASIC.

Furthermore, our solution provides a development methodology and a tool chain that fulfill an important gap in current custom, unscalable and inefficient design schemes [5]. The complexity introduced into a routing system is negligible as compared to the complexity increment that is taking place at present and will happen in foreseeable high performance routers.

In fact, an FPGA approach to the implementation of router components is in line with the current trend towards FPGA based development router design of major vendors [6, 11]. In particular, providing a PCI compliant interface eases integration of fuzzy inference modules as processing units within current network processing architectures.

5 Conclusions

We have described a platform that eases the development of fuzzy systems and its implementation as SoPC on FPGAs. The platform integrates both open tools and open IP cores. The Xfuzzy environment automates development from initial high-level fuzzy specifications to synthetizable VHDL.

Fuzzy inference modules are integrated into a SoPC architecture made of open IP cores that is suitable for prototyping fuzzy systems applied to communications among many other possible areas. Those SoPC developed using the outlined architecture can be integrated in current router architectures as processing units.

Successful results of the application of the platform to Internet traffic analysis and control fuzzy systems were also outlined. These fuzzy systems were validated and shown to satisfy operational requirements of current and future high performance routing hardware in terms of both inference speed and resource consumption. In addition, the prototypes have been designed for easy integration with routing architectures currently deployed in the Internet.

The open prototyping platform presented paves the way for further development of efficient intelligent traffic controllers but also foster the development of fuzzy systems for a number of areas where intelligent analysis systems are sought, such as packet and flow identification, classification and filtering, among many others.

References

[1] R. Zhang, R. Phillis, and V. Kouigoglou, *Fuzzy Systems for Queuing Control*. Surrey, UK: Springer-Verlag, Jan. 2004.

[2] F. Montesino, D. R. Lopez, A. Barriga, and S. Sánchez-Solano, "Fuzzy End-to-End Rate Control for Internet Transport Protocols," in *15th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'06)*, Vancouver, Canada, July 2006, pp. 1347–1354.

[3] R. de Oliveira and T. Braun, "A Delay-based Approach Using Fuzzy Logic to Improve TCP Error Detection in Ad Hoc Networks," in *IEEE Wireless Communications and Networking Conference*, Atlanta, USA, Mar. 2004.

[4] M. Hidell, "Decentralized Modular Router Architectures," Ph.D. dissertation, KTH-Royal Institute of Technology, Sept. 2006.

[5] M. Hidell, P. Sjödin, and O. Hagsand, "Control and Forwarding Plane Interaction in Distributed Routers," Laboratory for Communication Networks, Department of Signals, Sensors, and Systems. KTH Royal Institute of Technology, Stockholm, Sweden, Tech. Rep. TRITA-S3-LCN-0501, Mar. 2005.

[6] A. K. Kloth, *Advanced Router Architectures*. CRC Press, Nov. 2005, ISBN: 0849335507.

[7] I. Baturone, A. Barriga, S. Sanchez-Solano, C. J. Jimenez, and D. R. Lopez, *Microelectronic Design of Fuzzy Logic-Based Systems*. CRC Press., 2000, ISBN: 0-8493-0091-6.

[8] A. Cabrera, S. Sánchez-Solano, P. Brox, A. Barriga, and R. Senhadji, "Hardware/Software Codesign of Configurable Fuzzy Control Systems," *Applied Soft Computing*, vol. 4, no. 3, pp. 271–285, Dec. 2004.

[9] R. Herveille *et al.*, "WISHBONE System-on-Chip (SoC) Interconnection Arhitecture for Portable IP Cores." OpenCores Organization, Tech. Rep. Revision B.3, Sept. 2002.

[10] OpenCores Organization, "OpenCores.Org: Free Open Source IP Cores and Chip Design," http://www.opencores.org, Mar. 2007.

[11] W. J. Goralski, *Juniper and Cisco Routing. Policy and Protocols for Multivendor IP Networks*. Indianapolis, Indiana: Wiley Publishing Inc., 2002, ISBN: 0-471-21592-9.

[12] F. Moreno-Velo, I. Baturone, S. Sánchez-Solano, and A. Barriga, "Rapid Design of Fuzzy Systems With Xfuzzy," in *12th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'03)*, St. Louis, MO, USA, May 2003, pp. 342–347.

[13] F. Moreno-Velo, S. Sánchez-Solano, A. Barriga, I. Baturone, and D. López, "XFL3: a New Fuzzy System Specification Language," in *5th WSEAS/IEEE Multiconference on Circuits, Systems, Communications and Computers (CSCC'01)*, Rethymon, July 2001, pp. 361–366.

[14] Information Sciences Institute. University of Southern California, Viterbi School of Engineering, "The Network Simulator – ns-2," http://www.isi.edu/nsnam/ns/, Apr. 2006.

[15] F. Montesino, A. Barriga, D. R. Lopez, and S. Sánchez-Solano, "FPGA Based Implementation of Fuzzy Controllers for Internet Traffic," in *XII IBERCHIP Workshop*, San José, Costa Rica, Mar. 2006.

[16] F. Montesino, D. R. Lopez, A. Barriga, and S. Sánchez-Solano, "Intelligent Scheduling of Aggregate Traffic in Internet Routers by Means of Fuzzy Systems," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, Paris, France, July 2006.

[17] Cisco Systems, Inc, "Cisco series 12000 router performance evaluation," Cisco Systems, Tech. Rep., 2005.

[18] G. Ascia, V. Catania, G. Ficili, S. Palazzo, and D. Panno, "A VLSI Fuzzy Expert System for Real-Time Traffic Control in ATM Networks," *IEEE Transactions in Fuzzy Systems*, vol. 5, no. 1, feb 1997.