

Didactic Architectures and Simulator for Network Processor Learning

Henrique Cota de Freitas¹, Carlos Augusto P. S. Martins²

*Postgraduate Program in Electrical Engineering
Pontifical Catholic University of Minas Gerais, Brazil*
cota@pucminas.br¹, capsm@pucminas.br²
<http://www.inf.pucminas.br/projetos/pad-r/r2np.html>

Abstract

In our university, we are developing a project about Reconfigurable Network Processors (RNP). There are four important results: Reconfigurable CISC Network Processor (RCNP) architecture, Reconfigurable RISC Network Processor (R2NP) architecture, Network Processor Simulator (NPSIM), and a performance analytical model for the ISA (Instruction Set Architecture). The architectures and the simulator are not commercial products, but conceptual models. This paper shows the main functionality of those four results and their applicability on the Network Processor learning. As our Network Processor architectures and simulator are simpler than commercial products, their conceptual models can aid students to learn network processors concepts, as a first step to understand other complex architectures.

Keywords: Reconfigurable Network Processors, Didactic Architectures and Simulator, Learning Process.

1. Introduction

Until the 1990's, network equipments used the traditional general-purpose processor (GPP) to process many types of packets and management services. The main advantage was flexibility. The software was capable to define many functions and applicability for the GPP's, but the performance was very harmed.

Another solution to solve the performance problem was the use of dedicated hardware. Using Application Specific Integrated Circuit (ASIC), the processing speed increases enough, but the flexibility was harmed.

So, there are two very important features: flexibility and performance. Nowadays, the Internet [28] is the main type of network, and the Quality of Service (QoS) is very important. For this reason, a best approach between those two features is necessary in network equipments. For example, a router is concentration point or a bottleneck in a network. The flexibility to process any kind of packets and the performance

(processing time and throughput) are essential features that the Network Processor has to be capable to implement.

Network processors [19,20] appeared during the 1990's to replace some GPP's and ASIC's in network equipments. These processors were developed using architecture models like ASIP (Application Specific Processor) and SoC (System-on-Chip) [29] adding to RISC (Reduced Instruction Set Computing) [24] technique for a better computing performance. These processors have a dedicated ISA (Instruction Set Architecture) model for network operations. Thus, the instruction set and the architecture of Network Processors are specific to execute typical operations in a data communication network [28].

In Post-graduation Program in Electrical Engineering, we have a project called RNP (Reconfigurable Network Processor). The research goal is the development of a conceptual model of network processor using SoC and Reconfigurable Computing [17] techniques to improve computing performance and flexibility. The partial results are: Reconfigurable CISC Network Processor (RCNP) architecture [10], Reconfigurable RISC Network Processor (R2NP) architecture [11], Network Processor Simulator (NPSIM) [12,13] and the performance analytical model for the ISA (between RCNP and R2NP). During this paper, these four results will be presented.

Our main **objective** in this paper is to present didactic models of Network Processor architectures and a simulator to aid students to learn simple Network Processor architecture concepts. This is a first step to understand some details and complex features.

We searched for documents about words like learning and didactic models. However, nothing related with Network Processors was discovered. Thus, our **motivation** is present a simple way to learn the main features of Network Processors using didactic architecture models and a simulation tool.

2. Network Processors Overview

In this section we will describe the state-of-the-art of Network Processors. The main architecture features, the main functionalities, some related researches and companies.

The main logic blocks (figure 1) of a Network Processor are:

- ✓ Multiple RISC processors, co-processors or programmable ASIC's;
- ✓ Dedicated hardware for network operations;
- ✓ High speed of memory interface;
- ✓ High speed of I/O interface;
- ✓ General-purpose processors interface.

Each Network Processor has a typical architecture and uses some or all blocks showed. A Network Processor can use one RISC processor and co-processors like the packet processors, or only multiple RISC processors. If the SoC technique is used, possibly dedicated hardwares like switching fabric and memory can be in the architecture. However, GPP interface like PCI and I/O interface always appear. It's an important detail, because a Network Processor needs to communicate to other processors (to help in system management) and the network (the main of functionality).

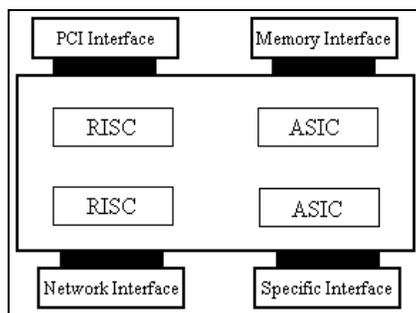


Figure 1 – Architecture Reference

The main functions of a Network Processor are:

- ✓ To analyze and classify the contents of head fields of a packet;
- ✓ To find in tables association rules related to head fields;
- ✓ To solve the destination path or QoS requirements;
- ✓ If necessary, to modify the packet (type of service or Diffserv, for example).

Nowadays, the active networks [6] are very important for QoS requirements, and equipments like active routers [27] appeared to improve performance and quality for Internet. The Network Processors have dedicated functionalities that provide flexibility and performance. For this reason, it has a large application in many network equipments.

During the developing process of Network Processors some companies joined among them we remark: Lucent / Agere [1], Motorola / C-Port [5] and Sitera / Vitesse [25]. Below, the main Network Processors and the companies are:

- ✓ IXP 1200 – Intel Corporation [16];
- ✓ NP4GS3 – IBM Corporation [15];
- ✓ C-5 Family – Motorola / C-Port [5];
- ✓ ASI/RSP/FPP – Lucent / Agere [1];
- ✓ IQ2000 Family – Sitera / Vitesse [25];
- ✓ AnyFlow 5400/5500 – MMC Networks [23];
- ✓ NP-1 – EZChip Technologies [8];
- ✓ NetVortex – Lexra Inc. [18];
- ✓ CS2000 – Chameleon Systems [3].

The Chameleon Systems was the first company to produce a Network Processor using the Reconfigurable Computing technique. This is an important characteristic. Reconfigurability is a technique that can be used in the NP architecture to improve flexibility.

Some researches about Reconfigurable Network Processors developed in universities are:

- ✓ “Reconfigurable Network Processors Based on Field Programmable System Level Integrated Circuits”, University of Patras, Greece [22];
- ✓ “Design and Analysis of a Layer Seven Network Processor Accelerator Using Reconfigurable Logic”, University of California, Los Angeles [9];
- ✓ “Design and Analysis of a Dynamically Reconfigurable Network Processor”, University of Florida [14].

After section 1 and 2, it is possible to say that flexibility and performance are two important features during network processing. Thus, we conclude that some concepts are very important, and so students must know them before study a commercial Network Processors:

- ✓ CISC and RISC models;
- ✓ The concepts of ASIC's and ASIP's;
- ✓ The concepts of SoC's;
- ✓ The concepts of Reconfigurable Computing;
- ✓ The main logic blocks of Network Processor architecture;
- ✓ The main functions of Network Processor.

The next sections will present the didactic models of RNP project and the results that can aid students to understand the functioning of Network Processors, based on the features above.

3. Didactic Architecture Models

This section presents two architecture models: CISC model (RCNP) [10] and RISC model (R2NP) [11]. Both architectures were developed and simulated using

Reconfigurable Computing [17] and SoC [29] concepts and techniques to increase flexibility and performance.

Reconfigurable Computing: Input Ports and Crossbar has more flexibility in time execution. Buffer sizes and topologies can be created dynamically.

System-on-Chip: Functional blocks, that are found externally, as memory, I/O ports and switching fabric, are internally in the same chip. Like hierarchical memory, the proximity between functional blocks reduces processing time and increases performance.

The use of didactic architectures (RCNP and R2NP) is presented in section 5. Subsection 3.1 and 3.2 presents only technical features. We will implement both architectures using VHDL (VHSIC Hardware Description Language) [21] and FPGA (Field Programmable Gate Array) [21] in the future.

3.1 RCNP (Reconfigurable CISC Network Processor) Architecture

The basic features of RCNP [10] architecture are the following (figure 2):

- ✓ Eight input ports;
 - Temporary buffers (one static buffer for each port);
 - Permanent buffers (dynamic buffers, reconfigurable size buffers);
- ✓ Eight output ports;
- ✓ Reconfigurable Crossbar;
- ✓ Direct Access Memory (DMA);
- ✓ Eight general-purpose registers (8 bits);
- ✓ Data bus (8 bits) and address bus (24 bits);
- ✓ Maximum of size memory (16Mbytes)

The RCNP architecture was developed as a System-on-Chip. Memory, I/O ports and crossbar are placed internally.

The Reconfigurable Computing appears in Permanent Buffers and Crossbar. Thus, in execution time the size of the buffers and topologies (defined by crossbar) modifies dynamically.

The main features of instruction set are:

- ✓ General-purpose instruction set
 - Arithmetic instructions (Ex.: ADD and SUB);
 - Logic instructions (Ex.: AND and OR);
 - Memory access instructions (Ex.: LOD and STO);
 - Branch instructions (Ex.: JMP and JNZ);
- ✓ Dedicated network instructions
 - Input port reading (Ex.: ENT);
 - Output port writing (Ex.: SAI);
 - Crossbar control (Ex.: SEC);
 - Status register control (Ex.: SRS and LRS);

The RCNP architecture was not designed with pipeline technique. For this reason, all instructions are executed sequentially. In section 7, the performance analytical model for the ISA shows the impact of the architecture without pipeline.

Like all CISC projects, other instructions (different of load and store) access memory. The general-purpose instruction set of RCNP is not optimized. There are 256 instructions that can be found in the project homepage (<http://www.inf.pucminas.br/projetos/pad-r/>). The simulation tool (NPSIM) also has the instructions described in figure 9.

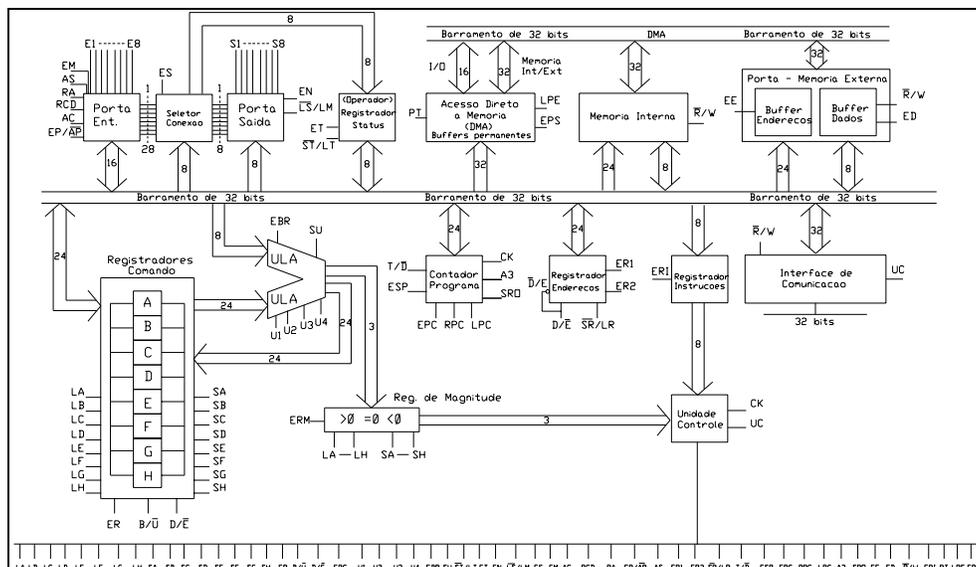


Figure 2 – RCNP Architecture

3.2 R2NP (Reconfigurable RISC Network Processor) Architecture

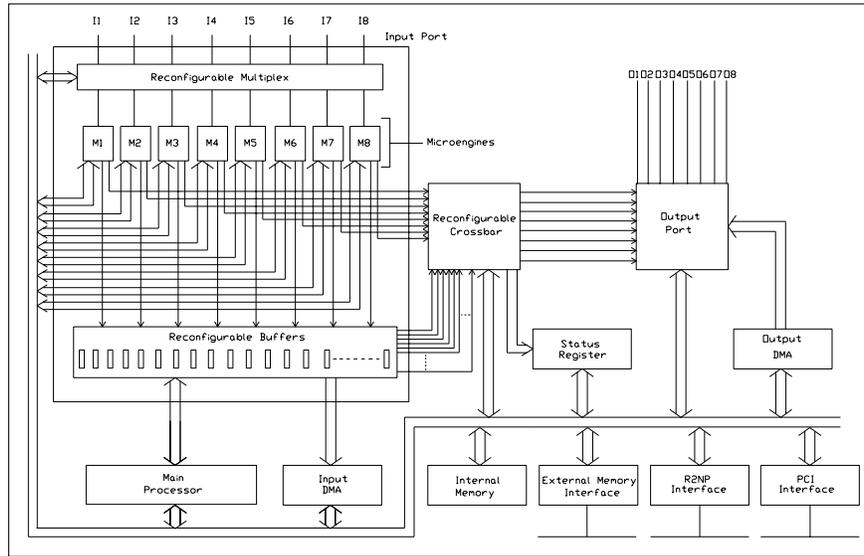


Figure 3 – R2NP Architecture

The evolution of RCNP is the R2NP [11]. This architecture uses the RISC model, pipeline and other reconfigurable blocks. The figure 3 shows the R2NP architecture.

The basic features of RCNP architecture are the follows:

- ✓ Eight input ports;
 - Reconfigurable Multiplex;
 - Programmable Microengines (one microengine for each port);
 - Permanent buffers (dynamic buffers, reconfigurable size buffers);
- ✓ Reconfigurable Crossbar;
- ✓ Eight output ports;
- ✓ Internal memory;
- ✓ Main RISC processor with data cache and instruction cache;
- ✓ Direct Access Memory (DMA), dedicated hardware;
- ✓ 256 registers (64 bits);
- ✓ Data bus (32 bits) and address bus (32 bits);
- ✓ Maximum of size memory (16Gbytes)

In R2NP project we add two important network blocks: Reconfigurable Multiplex and Microengines.

Microengines: Are responsible for the first analysis on the packet head. In this case the packet can be forward to output ports with no intermediary copies to buffers or memory.

Reconfigurable Multiplex: If you lost one microengine or need to use it in other function, the multiplex connects two or more ports to one microengine.

The instruction set of R2NP is more optimized than RCNP. Based on the RISC model, the instruction format is fixed and only load and store instructions access the memory. We present in table 1 the instruction set of R2NP.

Table 1 – Instruction Set of R2NP

General-purpose			
<i>ADD A,B,C</i>	<i>MOV A,B</i>	<i>SPUSH A</i>	<i>CONV</i>
<i>SUB A,B,C</i>	<i>INC A</i>	<i>LPOP A</i>	Network
<i>MUL A,B,C</i>	<i>DEC A</i>	<i>JMP A</i>	<i>FCX A,B,C</i>
<i>DIV A,B,C</i>	<i>LOD A,End32</i>	<i>JZ A</i>	<i>LOB A</i>
<i>AND A,B,C</i>	<i>LDA A,End16</i>	<i>JMZ A</i>	<i>BRC A</i>
<i>OU A,B,C</i>	<i>LOX A,B</i>	<i>JMI A</i>	<i>SAI A,B</i>
<i>XOR A,B,C</i>	<i>LDI A,Imed16</i>	<i>JNZ A</i>	<i>LRS A</i>
<i>NEG A</i>	<i>STO End32,A</i>	<i>JNI A</i>	<i>SRS A</i>
<i>ROD A</i>	<i>STR End16,A</i>	<i>CALL A</i>	<i>SEC A,B</i>
<i>ROE A</i>	<i>STX A,B</i>	<i>RET</i>	<i>ENT A,B,C</i>

There are two kinds of store and load instructions: with internal and with external memory access.

The internal memory is smaller and the instructions number 16 (Ex.: *LDI A,Imed16*) use sixteen bits to access the 64kbytes memory. The instructions number 32 (Ex.: *LOD A,End32*), access only the external memory. If the instruction has 32 bits of address, two fetch cycles will be necessary.

The network instructions (table 2) are very similar with the RCNP network set. However, by optimization, some differences appear in the instructions format.

The pipeline of R2NP is showed in figure 4. This is very similar with the conceptual model of pipeline [24], but the difference is the Buffer stage (together memory). One instruction that access buffer does not access memory.

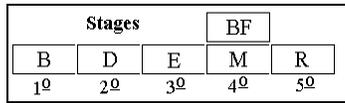


Figure 4 – Pipeline Stages

The stages mean:

1. Fetch of instruction (B);
2. Decoding of instruction. Reading of register bases (D);
3. Execution of instruction (E);
4. Reading or writing in memory of reconfigurable buffers (M/BF);
5. Results. Writing in register bases (R);

The performance analytical model for the ISA, in section 7, will show how the pipeline project increases performance.

4. Didactic Network Processor Simulator

This tool [12,13] was constructed with C++, and the main interfaces are capable to aid and guide the student in the learning of Network Processor theory and functioning. The simulator has six interfaces and it simulates main logic blocks as memory, registers, buffers, crossbar switch, DMA, I/O ports and others that are responsible to store, process, receive and transmit data. The student can modify and visualize the status and movement of the data inside and between logic blocks. There are two edition boxes, a program assembler and an editor of network packets.

This tool simulates the RCNP architecture and is available to download (<http://www.inf.pucminas.br/projetos/pad-r/r2np.html>) in two languages (idioms): Portuguese and English. It makes functional tests in all logic blocks of the processor. Through this tool, it is possible to write and execute many algorithms (assembly programs) and visualize the execution and the results in objects like: registers, stacks and arrays represented in components of C++ Builder 5.0 (used to construct and compile the simulator).

The user interface has one main module and six other modules:

- Memory, Registers and Fast Access Buttons (compose the main module – figure 5)
- Assembler (assembly program window – figure 5)
- Permanent Buffers (reconfigurable buffers) – (*)
- Temporary Buffers (eight buffers for each input) – (*)

- Input Packets (network packet edition box – figure 6)
- Internal Crossbar (commutation array) – (*)
- DMA Registers (Direct Memory Access) – (*)

(*) These modules are not shown in this paper. In figure 5, the number 1 shows where other modules can be found. For more details see the references [12,13].

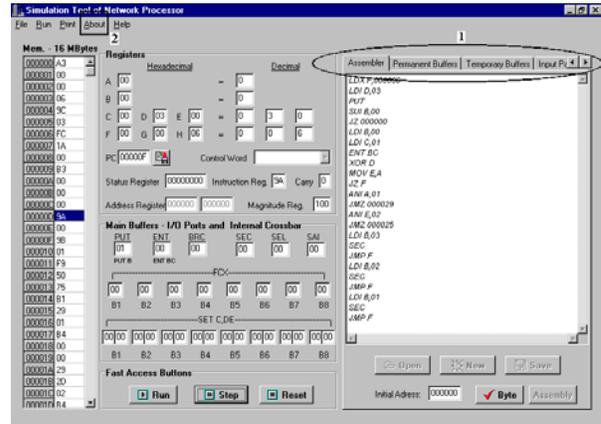


Figure 5 – Main Module (Assembler)

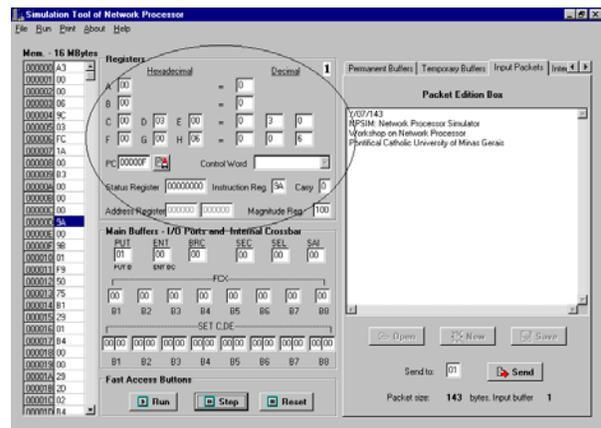


Figure 6 – Input Packets

In this section, we only describe the edition modules like the assembler and input packets. The figure 10 and 11, show the part of the Temporary Buffers and Crossbar modules, and its application as a way to learn Network Processor concepts.

With this simulator, it is possible to write and simulate routing algorithms (section 7), study the functioning of CISC network processor (RCNP) and understand the functioning of the Network Processors, described in section 5.

Other information about this simulator can be found in the papers: Simulation Tool of Network Processor for Learning Activities [12] and NPSIM: *Simulador de*

Processador de Rede (NPSIM: Network Processor Simulator) [13].

5. Using RNP Project to Learn NP Concepts and Functioning

This section will show how RNP project (didactic architectures and simulator) can help the students to learn network processor functioning and concepts.

The figures 7 and 8 show interfaces of NPSIM, which RCNP architecture and technical information. These interfaces aid students to understand the NPSIM modeling of RCNP proposal architecture. Basic features and important blocks as buffers and crossbar are described. Thus, concepts can be read and functional blocks visualized before the software execution. These interfaces can be found in the “About” option (figure 5, number 2).

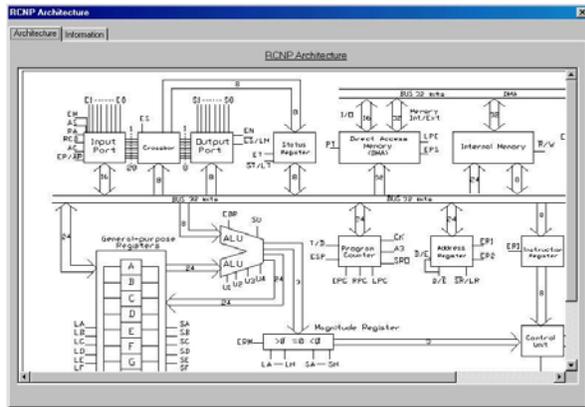


Figure 7 – RCNP Architecture

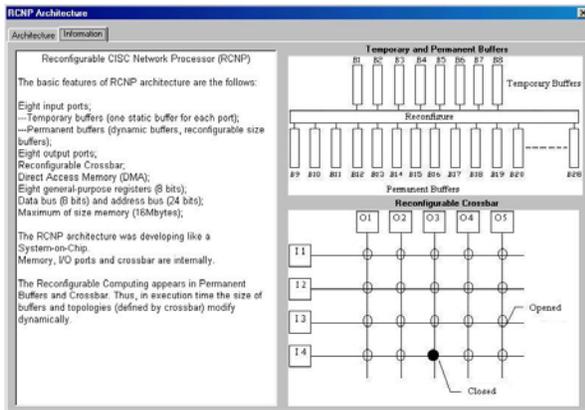


Figure 8 – Technical Information

The RCNP architecture has important blocks described in the architecture reference (section 2). Through the diagram architecture of RCNP we can visualize these functional blocks that represent important concepts of Network Processor, as flexibility

(reconfiguration) [17] and performance (ASIC’s). We present in table 2 these concepts:

Table 2 – Concepts of RCNP blocks

Functional Blocks	RCNP Concepts
Input Ports	Reconfigurable ASIC (Buffers)
Output Ports	ASIC
Internal Crossbar	Reconfigurable ASIC
Internal Memory and External Memory Interface	ASIC
Communication Interface	ASIC
Typical and dedicated CISC processor blocks	ASIP

The instruction set of RCNP can be visualized through the NPSIM interface. The figure 9 shows the instructions. This interface also can be found in the “About” option (figure 5, number 2).

The figure 10 shows interface modules that represent Temporary Buffers (inside Input Ports) and Internal Crossbar together Output Ports. A network packet sent by the Packet Interface (figure 6), arrives in Temporary Buffers and the behavior can be analyzed by students through control registers (figure 6, number 1). The behavior of routing algorithms and the destination of packets are showed through buffers, registers, crossbar, inputs and outputs. The buffers receive and allocate packets from inputs, the registers aid visualize the manipulation and data behavior, and the crossbar (figure 10) shows the way from input to output.

Instruction	Op Code						
HLT	0/00	ORI_H_Imed	64/40	MOV_E_E	128/80	RET	192/00
ADD_A	1/01	NEO_A	65/41	MOV_F_G	129/81	PUSH_A	193/01
ADD_B	2/02	NEO_B	66/42	MOV_F_H	130/82	PUSH_B	194/02
ADD_C	3/03	NEO_C	67/43	MOV_G_A	131/83	PUSH_C	195/03
ADD_D	4/04	NEO_D	68/44	MOV_G_B	132/84	PUSH_D	196/04
ADD_E	5/05	NEO_E	69/45	MOV_G_C	133/85	PUSH_E	197/05
ADD_F	6/06	NEO_F	70/46	MOV_G_D	134/86	PUSH_F	198/06
ADD_G	7/07	NEO_G	71/47	MOV_G_E	135/87	PUSH_G	199/07
ADD_H	8/08	NEO_H	72/48	MOV_G_F	136/88	PUSH_H	200/08
ADI_A_Imed	9/09	ROE_A	73/49	MOV_G_H	137/89	PUSH_T	201/09
ADI_B_Imed	10/0A	ROE_B	74/4A	MOV_H_A	138/8A	POP_A	202/0A
ADI_C_Imed	11/0B	ROE_C	75/4B	MOV_H_B	139/8B	POP_B	203/0B
ADI_D_Imed	12/0C	ROE_D	76/4C	MOV_H_C	140/8C	POP_C	204/0C
ADI_E_Imed	13/0D	XOR_A	77/4D	MOV_H_D	141/8D	POP_D	205/0D
ADI_F_Imed	14/0E	XOR_B	78/4E	MOV_H_E	142/8E	POP_E	206/0E
ADI_G_Imed	15/0F	XOR_C	79/4F	MOV_H_F	143/8F	POP_F	207/0F
ADI_H_Imed	16/10	XOR_D	80/50	MOV_H_G	144/90	POP_G	208/10
SUB_A	17/11	XOR_E	81/51	LOD_A_End	145/91	POP_H	209/11
SUB_B	18/12	XOR_F	82/52	LOD_B_End	146/92	POP_T	210/12
SUB_C	19/13	XOR_G	83/53	LOD_C_End	147/93	INR_A	211/13
SUB_D	20/14	XOR_H	84/54	LOD_D_End	148/94	INR_B	212/14
SUB_E	21/15	INOC	85/55	LOE_End	149/95	INR_C	213/15
SUB_F	22/16	INOC	86/56	LOD_F_End	150/96	INR_D	214/16

Figure 9 – Instruction Set of RCNP

These interfaces aid students to understand basic functions of Network Processors. Algorithms that execute functions described in section 2, as analysis and modification of contents, search for association rules, destination resolution, and QoS requirements can be found in NPSIM.

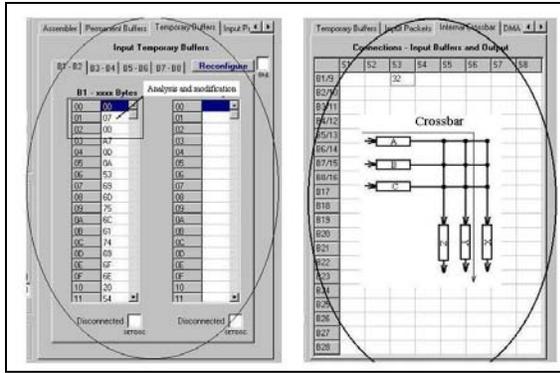


Figure 10 – Interface Modules

Although, the RCNP has many features of a Network Processor, one main characteristic does not exist, the RISC technique. RISC processors have better performance than CISC processors. Instruction format and pipeline are very important to increase processing speed and reduce processing time.

R2NP is the evolution of RCNP project. In this project, the goal is add RISC concepts and optimize the architecture and instruction set. The main difference is the pipeline and the instruction format. In section 7, the relation will be described.

The R2NP project is robuster than RCNP, and we present in table 3 some concepts related with Network Processors.

Table 3 – Concepts of R2NP blocks

Functional Blocks	R2NP Features
Input Ports	Reconfigurable ASIC (Buffers)
	Reconfigurable ASIC (Microengines)
	Reconfigurable ASIC (Multiplex)
Output Ports	ASIC
Internal Crossbar	Reconfigurable ASIC
Internal Memory and External Memory Interface	ASIC
Direct Access Memory	ASIC
Communication Interfaces	ASIC
Typical and dedicated RISC processor blocks	ASIP

The figure 11 shows the project evolution, based in hierarchical memory [24].

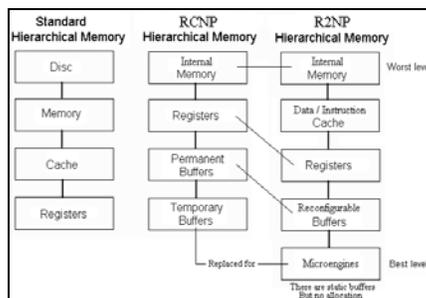


Figure 11 – Hierarchical Memory

The R2NP has data and instruction cache and RCNP has not. The Temporary Buffers were replaced by Microengines. In RCNP, packets could be store in Temporary Buffers, but in R2NP the microengines that also have static buffers, analyze and decide the destination of a packet, with no packet allocation. There are three routes: through crossbar and output ports, in the reconfigurable buffers or memory. In R2NP project, the main processor analyzes packets in buffers and memory.

6. Commercial Architectures of Network Processors

Some commercial architectures of Network Processors are presented and related with the reference architecture, also described in RNP project. The main features are numbered and appear in each figure. It's important to say that each commercial example represents details, features or concepts presented in our proposal, proving the real capability of RNP project as a didactic environment to learn Network Processors.

NetVortex Architecture [18] (figure 12): Each NetVortex is composed of many packet processors. It uses multi-threading in hardware and has the same instruction set of MIPS. Some features related to reference are:

1. Encryption Coprocessor: This architecture uses coprocessors for specific applications;
2. Crossbar Switch: Also uses dedicated hardware (ASIC) to increase performance;
3. Packet Processor: Specific processors for packet processing.

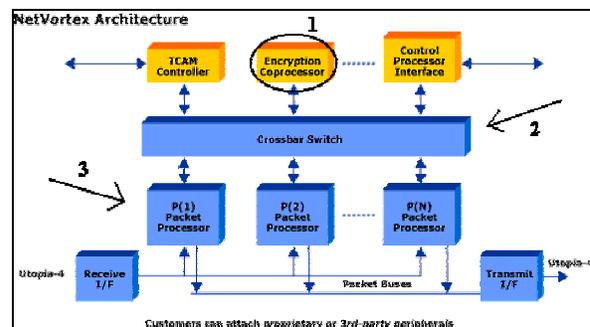


Figure 12 – NetVortex Architecture

IQ2000 Architecture [25] (figure 13): The IQ2000 Network Processor has four scale processors inside the chip. It has native support for MIPS, PowerPC and others RISC processors. There are specific coprocessors and hardware support for Quality of Service (QoS). Some features related to reference are:

1. Multiples CPU's: The IQ2000 uses parallel processing based in multiples processors;

2. QoS Engine: Dedicated hardware (ASIC) to increase performance.

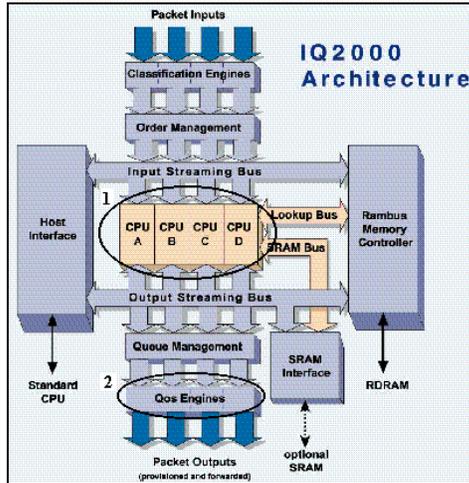


Figure 13 – IQ2000 Architecture

IXP1200 [16] (figure 14): The IXP1200 is composed of seven RISC processors. The first processor (StrongARM) is responsible for managing the network and for complex processing. The other six processors (the microengines) are responsible for processing and routing packets. Some features related to reference:

1. StrongARM Processor: The main processor responsible for complex processing.
2. PCI Unit: Dedicated communication hardware (ASIC).
3. Multiples microengines: For parallel processing of network packets.

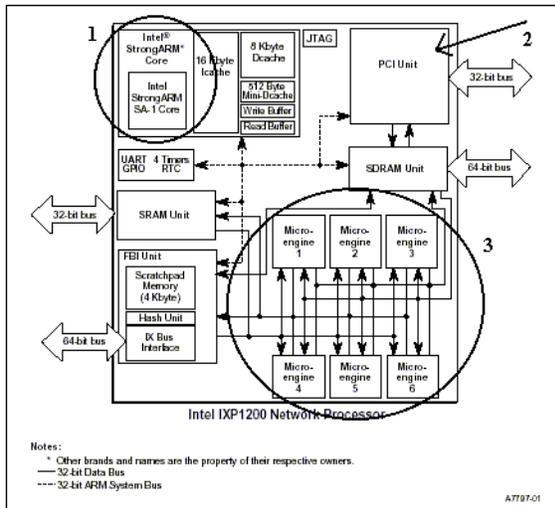


Figure 14 – IXP1200 Architecture

Reconfigurable Fabric of CS2000 [3] (figure 15): This figure shows the reconfigurable block of CS 2000. This block is divided in four slices with three blocks.

Each block has Datapath Units, Local Memories, Multipliers and Control Logic Unit.

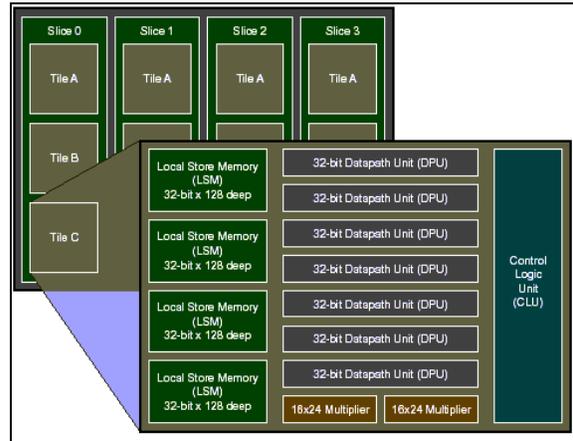


Figure 15 – Reconfigurable Fabric of CS2000

This section presented some features of RNP project, how to learn Network Processors using it and four commercial Network Processors. During all the description we related the features of RNP project and commercial NP to the architecture reference. The next section will present the experimental results from simulations and analytical model that contribute to validate the evolution of RNP project.

7. Experimental Results using NPSIM

The simulation results and the performance analytical model for the ISA were based in three topologies. For each topology was written three algorithms. These simulations were very important to validate the concept of RCNP. Using the results, an analytical model was proposed to verify the performance between ISA of RCNP and ISA of R2NP. The topologies (figure 16) are the follow:

Hypercube topology: Where each vertex is a simulated network processor. The routing algorithm is based in different bit resulted by XOR operation between source address and destination address.

Unidirectional ring topology: Constructed using the internal crossbar. The unidirectional ring program shows how internal crossbar can construct a topology.

Balanced Tree: Where each vertex is a simulated network processor. In this case the routing program uses the standard address by each vertex. The addresses grow from left to right.

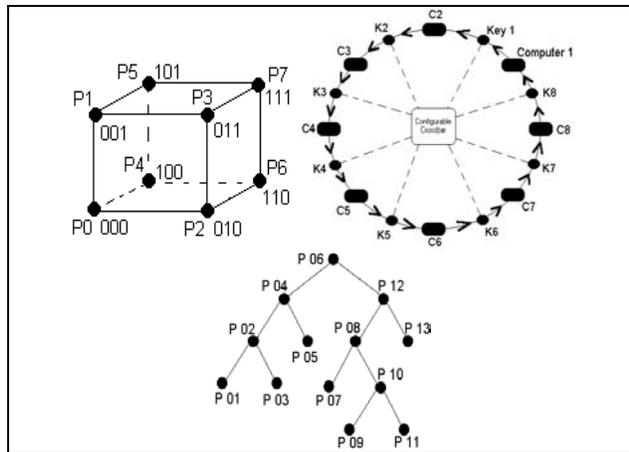


Figure 16 – Simulated Topologies

The metrics defined to analyze performance for ISA are:

- Cf → Clock Frequency (Hz)
- Tp → Time of Processor
- Ncpc → Number of cycles of program clock
- Cpi → Cycles per instructions
- Npi → Number of program instructions
- Pf → Performance factor

We can related through these equations:

$$Cpi = Ncpc / Npi$$

$$Tp = Npi * Cpi / Cf$$

It's important to say that the RCNP model does not use pipeline, the instructions are executed sequentially. The RCNP and R2NP does not exist physically, for this reason, the clock frequency is 500Mhz for definition.

For the Unidirectional Ring topology, the results are the follows:

RCNP proposal
 Npi = 45, Ncpc = 191, Cpi = 191 / 45 = 4,244
 Tp = 191 / 500 10E6 = 0,382 μs

R2NP proposal
 Npi = 38, Ncpc = 43, Cpi = 43 / 38 = 1,131
 Tp = 43 / 500 10E6 = 0,086 μs
 Pf = 0,382 / 0,086 = 4,44
 The R2NP is 4,44 faster than RCNP for this simulation.

For the Hypercube topology, the results are the follows:

RCNP proposal
 Npi = 17, Ncpc = 73, Cpi = 73 / 17 = 4,294
 Tp = 73 / 500 10E6 = 0,146 μs

R2NP proposal
 Npi = 17, Ncpc = 21, Cpi = 21 / 17 = 1,235
 Tp = 21 / 500 10E6 = 0,042 μs
 Pf = 0,146 / 0,042 = 3,47
 The R2NP is 3,47 faster than RCNP for this simulation.

For the Balanced Tree topology, the results are the follows:

RCNP proposal
 Npi = 12, Ncpc = 56, Cpi = 56 / 12 = 4,666
 Tp = 4,666 / 500 10E6 = 0,112 μs

R2NP proposal
 Npi = 15, Ncpc = 19, Cpi = 19 / 15 = 1,266
 Tp = 19 / 500 10E6 = 0,038 μs
 Pf = 0,112 / 0,038 = 2,94
 The R2NP is 2,94 faster than RCNP for this simulation.

Through this analytical model, the students can understand how RISC processor can be faster than CISC processors, using project techniques as pipeline, for example.

8. Conclusions

Nowadays, there is a great need of high performance in the data communication network [2,28]. The study of various equipments [4,27] and their functions, influenced in the project, and development of dedicated processors, that can supply the need of performance and quality. Thus, the Network Processors were initially developed to contribute with the increase of speed and quality of service in the communication systems.

In this paper we presented a project called Reconfigurable Network Processor (RNP) that has a main goal: to aid students to learn and know initial basic concepts of Network Processors, as a first step to understand commercial products.

RCNP [10] and R2NP [11] architectures, and NPSIM [12,13] simulator were described with many options to understand the reference architecture and the basic network processors functioning. In section 5, the architectures and NPSIM were shown for the student to compare the learning possibilities. The same features in reference architecture appear in RNP project. Using these didactic proposals it is possible to learn the basic concepts. Four commercial architectures were presented and related with the reference to show the use of didactic models before the studying of commercial Network Processors.

Didactic models and simulator were looked for, but they were not found anywhere. However, one correlate paper was presented in NP1, "A Methodology and Simulator for the Study of Network Processors" [7]. But they have different goals. That paper describes a model of the Cisco Toaster architecture and show simulated performance results of a Diffserv implementation. It describes a commercial product and simulates performance. Our goals in this paper are present a didactic model to introduce the main concepts of Network Processors before the study of complex architectures. A paper or research with didactic features for NP's, were not found.

The main presented results of our research, are the RCNP architecture, R2NP architecture, NPSIM

simulator and experimental results. Those results validated our goals and showed how conceptual models can aid students to understand complex architectures of Network Processors.

Thus, our main contribution, in this paper, is present didactic architectures and simulator for beginning process of Network Processors learning.

Our future works are: To simulate R2NP using Rconf_KMT (Reconfigurable Simulation Tool) [26] and VHDL (VHSIC Hardware Description Language) [21], prototype using FPGA (Field Programmable Gate Array) [21], simulate it in a real network system [2,28], and to develop didactic environment to learn Network Processors.

9. References

- [1] Agere System, Fast Pattern Processor (FPP) Product Brief, April 2001, <http://www.agere.com>
- [2] Buya, R., High Performance Cluster Computing, Volume 1, Prentice Hall, 1999
- [3] Chameleon Systems, "CS2000 Reconfigurable Communications Processor", Family Product Brief, 2000
- [4] Cisco Systems White Paper, "The Evolution of high-end Router Architectures-Basic Scalability and Performance Considerations for Evaluating Large-Scale Router Designs", 2001, <http://www.cisco.com>
- [5] C-Port, C5e Network Processor Product Brief, January 2002, <http://www.motorola.com>
- [6] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, G. J. Minden, "A Survey of Active Network Research", IEEE Communications Magazine, Volume 35, Nº 1, pp.80-86, 1997
- [7] D. Suryanarayanan, G. T. Byrd and J. Marshall, "A Methodology and Simulator for the Study of Network Processors", Workshop on Network Processor (NP1 at HPCA 8), Cambridge Massachusetts, February 2-6, 2002
- [8] EZChip Network Processors, <http://www.ezchip.com>
- [9] G. Memik, S. O. Memik, W. H. Mangione-Smith, "Design and Analysis of a Layer Seven Network Processor Accelerator Using Reconfigurable Logic", The 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines FCCM'02, Napa, California, 21-24 April, 2002
- [10] H. C. Freitas, C. A. P. S. Martins, "Processador de Rede com Suporte a Multi-protocolo e Topologias Dinâmicas", II Workshop de Sistemas Computacionais de Alto Desempenho, WSCAD'2001, Pirenópolis - GO, Brasil, pp.31-38 (in portuguese)
- [11] H. C. Freitas, C. A. P. S. Martins, "R2NP: Processador de Rede RISC Reconfigurável", III Workshop de Sistemas Computacionais de Alto Desempenho, WSCAD'2002, Vitória, ES, Brasil, pp. 60-67 (in portuguese)
- [12] H. C. Freitas, C. A. P. S. Martins, "Simulation Tool of Network Processor for Learning Activities". Frontiers in Education Conference (FIE 2002), Boston, MA, USA, November 2002, Session S2F, pp.1-6
- [13] H. C. Freitas, C. A. P. S. Martins, "NPSIM: Simulador de Processador de Rede". XXVIII Latin-American Conference on Informatics, CLEI'2002, Montevideo, Uruguay, November 2002 (in portuguese)
- [14] I. A. Troxel, A. D. George, S. Oral, "Design and Analysis of a Dynamically Reconfigurable Network Processor", IEEE Conference on Local Computer Networks (LCN), Tampa, Florida, November 6-8, 2002
- [15] IBM PowerNP NP4GS3 Databook, <http://www.ibm.com>
- [16] Intel, "IXP 1200 - Network Processor", Datasheet, May 2000, <http://www.intel.com>
- [17] K. Compton, S. Hauck, "Reconfigurable Computing: A Survey of Systems and Software", ACM Computing Surveys, Vol. 34, No. 2, June 2002, pp. 171-210
- [18] Lexra, NetVortex Network Communications System Multiprocessor NPU, <http://www.lexra.com>
- [19] Lucent Technologies, Building for Next Generation Network Processors, September 1999
- [20] Lucent Technologies, The Challenge for Next Generation Network Processors, September 10, 1999
- [21] M. Glesner, A. Kirschbaum, "State-of-the-Art in Rapid Prototyping", XI Brazilian Symposium on Integrated Circuit Design, SBCCI'98, Búzios, Rio de Janeiro, 1998, pp.60-65
- [22] M. Iliopoulos, T. Antonakopoulos, "Reconfigurable Network Processors Based on Field Programmable System Level Integrated Circuits", Field-Programmable Logic and Applications, The Roadmap to Reconfigurable Computing, 10th International Workshop, FPL 2000, Villach, Austria, August 27-30, 2000, pp. 39-47
- [23] MMC Networks, "EPIF-105, EPIF-200, GPIF-207, XPIF-300, Packet Processors", <http://www.mmcnet.com>
- [24] Patterson, D. A., J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann Publisher, 1997
- [25] Sitera IQ2000, Network Processor Product Brief, <http://www.sitera.com>
- [26] T. H. Medeiros, C. A. P. S. Martins, "Reconf_KMT, Uma Ferramenta Reconfigurável para a Simulação de Microprocessadores", III Workshop de Sistemas Computacionais de Alto Desempenho, WSCAD'2002, Vitória, ES, Brasil, pp. 32-38 (in portuguese)
- [27] T. Wolf and J. Turner, "Design Issues for High Performance Active Routers", International Zurich Seminar on Broadband Communications, Zurich, Switzerland, 2000, pp. 199-205
- [28] Tanenbaum, A. S., Computer Networks, Prentice-Hall, 1996
- [29] W. D. Mensch. Jr. and D. A. Silage, "System-on-chip Design Methodology in Engineering Education", International Conference on Engineering Education, ICEE2000 (IEEE/CS), Taipei, Taiwan, August 2000, pp. 224-228