RAMSÉS - DEVELOPMENT OF AN 8-BIT DIDATIC PROCESSOR

G. M. Brigidi, L.M.Rech, R. L. Pitoni, R. M. Cornes, T. M. Araki, R. P. Ribas

Instituto de Informática – UFRGS UFRGS – Caixa Postal 15064
CEP 91501-970 – Porto Alegre – RS – Brasil
{tmaraki, rlpitoni, rmcornes, rpribas}@inf.ufrgs.br

ABSTRACT

The objective of this paper is to describe the development an 8-bit processor, the Ramsés, a hypothetical processor created for didactic purposes by Professors C. A. L. Lisbôa and R. F. Weber. The top-down project methodology, from architectural system to the electrical and design issues are considered in this work.

1. INTRODUCTION

The idea of developing a processor came in the course of Integrated Circuits Design II, a discipline that provides the chance for students to apply their knowledge in VLSI circuits on a project chosen by themselves. In our case, the chosen project was the development of Ramsés processor, which certainly includes most of Microelectronics issues.

The Ramsés is a hypothetical didactic processor created for Computer Architecture course and was first proposed on [1]. It appears as the next-step architecture for Neander processor, another well-known didactic processor that is used along introductory disciplines at Brazilian universities. The Ramsés was only described in software that emulated its behavior, but was based on real 8-bit processors.

In addition, is important to declare that the whole work involved only free tools. The logical circuit was developed using VHDL and it was simulated on a web version of Quartus II, by Altera, while the design involved tools such as Magic, a VLSI CAD software, used for the layout description, and Spice Opus , which was used for electrical design and also the entire system validation.

This work aims at MOSIS educational program, which offers free of charge fabrication of projects designed by under-graduate students.

In this paper, Section 2 describes the Ramsés architecture and instruction set. Section 3 explains our work from logical to physical implementation. Finally, conclusions and future projects are presented in Section 4.

2. THE RAMSÉS PROCESSOR

The Ramsés is a simple processor with the following characteristics: 8-bit data and address width; data codification in two's-complement; two registers for general use and one index register; and an 8-bit program counter. In addition, there are 4 different address modes: direct mode, in which the operand address follows the instruction code word; indirect mode, in which the address of the operand address follows the instruction code word; immediate mode, in which the operand follows the instruction code word; and index mode, in which the offset that follows the instruction code word is added to the index register to build the operand address. The memory addressing space is shared between data and program code.

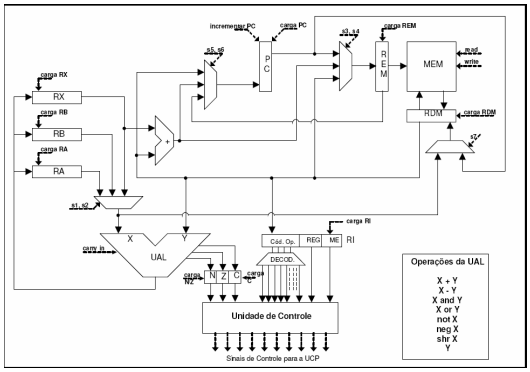

Figure 1: Ramsés hardware elements

Figure 1 describes the hardware elements of the Ramsés processor. In this figure is possible to view how registers, the arithmetic and logic unit, the control unit and memory are related, either by control signals or by bus connections. The memory unit was not in the scope of this project, and it was not designed, but a simulated memory was used for testing and debugging.

The Ramsés instruction set is described in Table I. In this table, ADDR represents the address, calculated by the address mode, and R indicates the register in use (A or B). Instructions that manipulate data operate in one of the three registers only.

The arithmetic and logic operations are computed on a dedicated entity, the Arithmetic and Logical Unit (ULA), which considers two 8-bit data inputs, eight operations and three flags: carry (C), zero (Z) and negative (N). Its output is connected to registers A, B and X, that stores the current data under manipulation. The program flow is controlled by the Program Counter, a register that can be reset on initialization process, incremented on sequential program flow, and loaded on branching operations. The memory interface is comprised by three buses - the address bus, the incoming data bus and the outgoing data bus - and read and write signals. The memory address register (REM) and the data memory

register (RDM) are registers responsible for accomplishing this interface.

The control of the internal signals is made by a finite state machine, implemented on the Processor Control Unit (UCP) and a Instruction Register (RI) that points to the current instruction under execution.

Table I: Ramsés instruction set

MNEMONIC	DESCRIPTION
STR R ADDR	MEM(ADDR) ← R
LDR R ADDR	R ← MEM(ADDR)
ADD R ADDR	R ← MEM(ADDR) + R
OR R ADDR	R ← MEM(ADDR) OR R
AND R ADDR	R ← MEM(ADDR) AND R
NOT	R ← NOT R
SUB R ADDR	R ← R – MEM(ADDR)
JMP ADDR	PC ← ADDR
JN ADDR	IF N = 1 THEN PC ← ADDR
JZ ADDR	IF Z = 1 THEN PC ← ADDR
JC ADDR	IF C = 1 THEN PC ← ADDR
JSR ADDR	MEM(ADDR) ← PC; PC ← ADDR + 1
NEG R	R ← - R
SHR R	R ← R/2

3. DESIGN FLOW

The project is divided in three parts: registers, control unit and arithmetic and logic unit. The registers and the program counter were based on implementations described in [2], and were chosen between the ones which could work at the processor estimated speed. The Arithmetic and Logic Unit (ULA) was built using the bit-ula approach. This means that the ULA circuit was first developed for one bit operands and then it was replicated eight times, creating the 8-bit ULA. The bit-ula is based on a circuit that calculates all operations at the same time, and a multiplexer, controlled by the UCP, that selects which operation is going to be in the output.

For the development of the Ramsés control unit, three implementation schemes were tested: combinational logic, read-only memory (ROM) and programmable logic array (PLA). A comparison between these schemes were made, taking in account that pass-transistor logic structures would be used. With this comparison, it was chosen to build the control unit in combinational logic, generating equations controlled by a finite state machine.

The Ramsés processor was implemented using a top-down methodology. At first, the behavioral VHDL description was created, to validate the logic idea of the system. Then, the processor was divided in different modules and these modules were developed into structural VHDL description.

The second phase of the project was the processor electrical design. Instead of building the circuit in standard CMOS structure, the logical family chosen was the PTL (pass-transistor logic) with the 0.35AMI library. This logical structure uses less PMOS transistors and spends less area in the circuit, although it has to be fed with dual-rail signals. Also, as source and drain terminals have variable capacitance levels, which could generate unexpected voltage gains, it was chosen only to connect variable signals in the gate terminal of the transistors. This construction leads to a better parasitic capacitance estimation, although needs larger area. The main electrical design techniques used here can be found in [2] and [3].

The third stage of the project was the development of the integrated circuit layout, which was based on a CMOS fabrication technology with minimal size for the transistor channel of 0.35µm. The first procedure was the floorplanning study, that demonstrates the best way to fit modules and their geometric format. Respecting these formats, the circuit layout was drawn using only metal 1 and 2 layers, and standard cell approach was used in the development of the control unit and registers, while the ULA was built using bit-ula technique. The resulting layout is shown on Figure 2, yet without PADS.

Figure 2: Ramsés processor layout

4. CONCLUSION AND FUTURE WORKS

This paper presented the complete design flow of an 8-bit simple didactic processor, considering logical, electrical and physical design. The processor chip presented here was fully designed by undergraduate students in a VLSI Project discipline and it took 5 months to be concluded. It is emphasized that the tools used to accomplish this project were all free software.

Currently, a test platform is being developed to test and debug the project. A revised layout will possibly be drawn in order to fabricate the chip.

5. REFERENCES

[1] Weber, R. F. *Fundamentos de Arquiteturas de Computadores.* Série Livros Didáticos do Instituto de Informática da UFRGS. Ed Sagra Luzzato, Porto Alegre, 2000.

[2] Weste, N., Eshraghian, K., *Principles of CMOS VLSI Design.* Addison-Wesley, 2nd edition, 1993.

[3] Rabaey, Jan, *Digital Integrated Circuits,* Prentice Hall, 1996.