# NEANDER - 8 BITS PROCESSOR DESIGN FLOW

*Renato E. B. Poli, Felipe R. Schneider, Daniel Barden, Diogo A. Fiorentin, Francisco M. Trindade, Fernando S. de Vasconcellos, Renato P. Ribas*

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

## ABSTRACT

In this work, it is presented the design flow of a didactic 8-bit processor, called Neander, from the system architecture conception to the chip layout drawing and testing. The fabrication of the chip was possible thanks to the MOSIS educational program and the partnership stabilished with UFRGS. The processor was conceived using the 0.5μm double-metal CMOS technology from AMI. The whole project was designed with free CAD tools available in the Internet. Project methodology, logic and electrical design as well as testing issues are considered on this work.

## 1. INTRODUCTION

Neander is a simple didactic processor that has been used for years on computer architecture introductory disciplines at Brazilian universities. It works with 8-bit data word and a set of nine instructions, which includes logic and arithmetic, conditional and unconditional branching and memory data transfer ones. It is designed for interfacing with a 256 bytes memory bank.

The design involved only free design tools, such as Magic [1], a VLSI layout system used for the layout conception, and Spice Opus® [2], used for electrical design and validation. The logical design was developed using VHDL and simulated on a free version of the Modelsim® simulator.

This work aims the MOSIS educational program, which offers the fabrication of projects designed by students under graduation free of charge. The circuit has already been fabricated and is under testing and debugging.

Section 2 describes briefly the logical design as well as the architecture of the processor. Section 3 talks about relevant points of the AMIS technology and makes other considerations about the electrical design. Section 4 describes some important testing issues. Finally, conclusions are presented.

## 2. LOGICAL DESIGN

Neander processor was first proposed in [3]. Basically, it is composed by simple entities which control the datapath and program flow. Neander has nine instructions shortly described in Table 1. The memory addressing is direct, so the data is always accessed through its memory position, which follows the operation code of the instruction, whenever needed. The memory addressing space is shared between data and program code as no distinction is made.

**Table 1:** Neander instructions set.

| MNEMONIC | DESCRIPTION |
|---|---|
| LDA | ACC ← MEMORY DATA |
| STA | MEMORY ← ACC |
| OR | ACC ← ACC OR MEMORY DATA |
| NOT | ACC ← NOT ACC |
| AND | ACC ← ACC AND MEMORY DATA |
| ADD | ACC ← ACC + MEMORY DATA |
| JN | IF ACC < 0 THEN PC ← MEMORY DATA |
| JZ | IF ACC = 0 THEN PC ← MEMORY DATA |
| JMP | PC ← MEMORY DATA |

The logical and arithmetic operations are computed on a dedicated entity, the Arithmetic and Logical Unit (ALU), which considers two 8-bit data inputs and five distinct operations. Its output is connected to the Accumulator Register, which stores the current data under manipulation. The program flow is controlled by the Program Counter that is basically a register that can be initialised, incremented, on sequential program flow, and loaded on branching operations. The memory interface is composed 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 the registers responsible for accomplishing this interface.

The control of the internal signals is made by a finite state machine, implemented on the Central Processing Unit (UCP) and a Instruction Register (RI) that points to the current instruction under execution. The temporisation of the UCP is done by a three bit counter, controlled by the system clock. A simplified block diagram of this approach is shown on Figure 1.

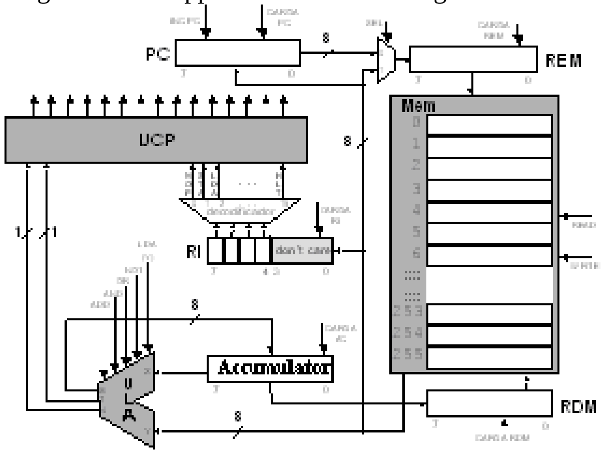

**Figure 1:** Neander block diagram.

## 3. PHYSICAL IMPLEMENTATION

The physical implementation of the project includes the electrical design and validation, and the layout drawing. The aimed technology process was the AMIS C5F/N [4], a CMOS technology with transistor with a minimal channel size of 0.5µm and that supports up to three metal layers. The chip area was limited to 2,5mm x 2,5mm by the educational program, but it was enough to the design proposed.

The electrical design considered most of the times the minimum transistor dimensions, except for some cells on which larger fanouts had to be considered, such as internal bus drivers. Simulations were made in order to estimate the cells driving capabilities given the influences of variable fanouts and parasite capacitances. Based on these simulations, a cell library was proposed containing cells of variable transistor dimensions, as needed. Then, the whole circuit was mapped to the proposed library and the layout was drawn. For a matter of illustration, Table 2 shows the delays related to some cells of the library. The main electrical design techniques used here can be found in [5] and [6].

**Table 2:** Delay simulation on cells of the library.

| | $T_{HL}$ (ns) | $T_{LH}$ (ns) | $T_D$ (ns) |
|---|---|---|---|
| NOR2 | 0.56 | 0.53 | 0.55 |
| NAND2 | 0.34 | 0.43 | 0.37 |
| XOR2 | 0.42 | 0.45 | 0.44 |
| MUX 2-1 | 0.42 | 0.30 | 0.40 |

The interface with external components is another point of interest. Due to the possibly large capacitance associated with the encapsulation pins and to problems related to them, robust pads were needed. The students chose to use pre-designed and tested unidirectional pads, developed by Tanner and available thorugh MOSIS WEB page [4].

The final chip layout is shown in Figure 2.

## 4. TESTING ISSUES

Given the difficulties imposed by an ASIC on monitoring internal signals and debugging the system, testing shows up as one of the most important issues of the project and should be considered from its beginning.

As shown on previous sections, the design is well modularised. In order to detect a fault, the modules should be isolated as much as possible in a way that they can be tested as a standalone block. Due to the area limitations imposed by the MOSIS program, the circuit could have up to forty pins, what means that, if one wants to monitor or control internal signals, pins have to be multiplexed.

On this matter, three test modes were included, each one with a distinct goal. The first one aimed on testing a single register, so that the flip-flops could be validated; the second test mode points to testing some ALU functions; and the third looked forward to testing the temporisation, which controls the whole circuit synchronising. The pertinent multiplexing structures were added to the circuit.

## 5. CONCLUSIONS AND FUTURE WORKS

This paper presented the complete design flow of an 8-bit simple educational processor. It considered the logical and physical design as well as the testing structures needed. The processor chip presented here was entirely designed by undergraduate students in a VLSI Project discipline and it took, approximately, 2 months to be concluded. It is emphasized that the tools used to accomplish this project were free software.

Currently, a test platform is being developed through field programmable devices, in order to test, debug and possibly draw a revised layout and fabricate a new chip.

## 6. REFERENCES

[1] Mayo, R. N. 1990 DECWRL/Livermore Magic Release. Western Research Laboratory. California USA, 1990

[2] Buermen, A. et al. Spice OPUS Homepage. http://www.fe.uni-lj.si/spice/welcome.html. CACD Group, 2002.

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

[4] MOSIS homepage. http://www.mosis.org.

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

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

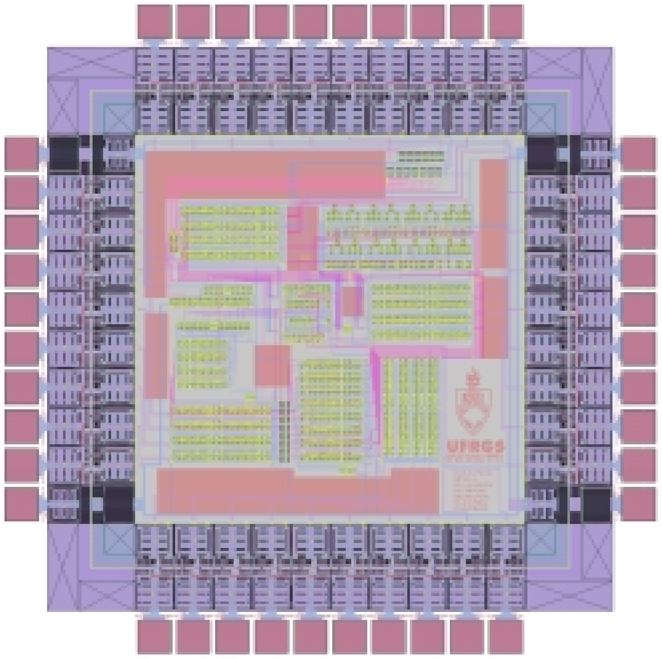

**Figure 2:** Chip layout.