Design of a synthesizable processor didactic in FPGA

Fernanda P. Mota [1], Bruno Q. Leonardo[2], Vagner S. da Rosa [3]

Computational Science Center- Federal University of Rio Grande
Av. Itália km 8 – Campus Carreiros – 96.201-900– Rio Grande – RS – Brasil
[1]nandap.mota@gmail.com, [2]bruno_ql@hotmail.com, [3]vagner.rosa@furg.br

***Abstract* —This work presents the design and implementation of the microprocessor MAC-1, from Tanembaum's textbook. The goal of this work is to provide digital synthesis solution based on this microprocessor for didactic purposes. However, for the development of this project were necessary any modifications, such as changing to latch registers and the use of four clock cycles instead of using a single clock cycle and then divide it into subcycles as suggested in the book, was used four clock signals synchronously. Finally, we will present the preliminary results obtained in the simulation and implementation of MAC-1.**

***Keywords*— FPGA, MAC-1, Simulation.**

I. INTRODUCTION

The use of digital computers began to be applied to the '50s. Primarily in military operations – missile control and aircraft detection – they were later widely applied for non military purposes. At that time computers were built with vacuum tubes with very limited processing capability, programming and operation difficult.

The processing power of computers has increased exponentially because of the evolution and the emergence of transistors and integrated circuits constantly evolving. These advances have allowed computers were utilized in all applications, such as process control and complex mathematical calculations. Due to the reduction of price and physical size, computers started to be commercialized on a large scale for commercial or industrial applications.

In the 70s, the microprocessors were based on integrated circuits, having as precursor of the Intel 4004, since then there has been an intense search for new technologies for these components. The main reasons for the popularity of personal computers (those based on microprocessors) were: their use in various activities from academia [1] and its popularization in the household and ubiquity in modern embedded systems.

The microprocessor MAC-1, presented by Andrew Tanembaum [2] is based on microprogrammed architecture and used for teaching purposes. Microprogrammed architecture implements the control of the microprocessor simply through a microprogram ROM memory, rendering the understanding of control functions and operation of a microprocessor easier. Due to it is relatively simple to design and develop, this architectural style is useful for teaching applications. In this proposal, the microprocessor is implemented in programmable logic (VHDL-**VHSIC Hardware Description Language** and VHSIC stands for **Very High Speed Integrated Circuits**).

This paper is organized as follows: section 2 describes briefly the conceptual aspects of the microprocessor Mac-1, section 3describes any modifications to the architecture of MAC-1, section 4 shows the initial model is developed, section 5 reports the results that were generated by the model, and finally in section 6 have been concluding remarks and future work.

II. MAC-1: MICROPROCESSOR DESIGN

The MAC-1 microprocessor design (Fig. 1) has been divided in 4 main parts: Control Part that is responsible for the operations of them, Operative Part that is responsible for the operations of them, the RAM memory that is contains the program, i.e the search system, the information in main memory and also stores data and clock (Fig. 2) that distribute clock to all other parts.

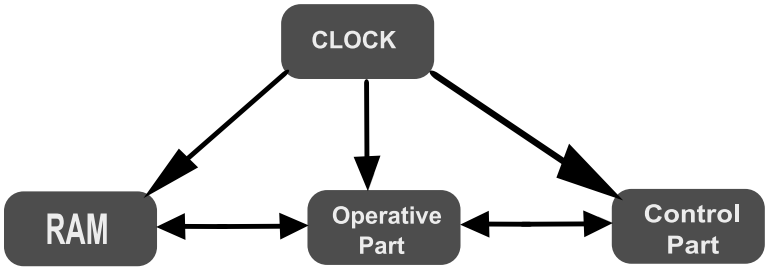

Fig. 1 Composition of MAC-1

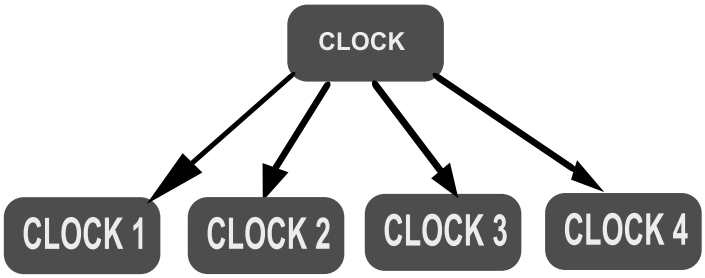

Fig. 2 Subcycle of clock

The operative part is composed of (Fig. 3):

- **Decoders:** The model consists of three decoders (one for bus A, bus B to another and finally one for the register C - the scratch register);
- **Registers:** This is the fastest storage component of the system, since it is inserted into the processor;
- **Latch A and B:** Ensure the correct shipping information coming from the scratch memory, are also controlled by the clock - second subcycle.
- **MUX1:** Selects if the registrar increment MPC will receive the data coming from the enhancer or address available on MIR.

- **MAR:** Have the address of data to be read. The signal is a unidirectional register to memory and provides the correct address of reading or writing, for that he respects the third subcycle clock.
- **MBR:** the data has to be read
- **Shifter:** Performs 3 operations (no shift, a shift to the left or right shift). However, this operation depends on the control bits of the word in the MIR.

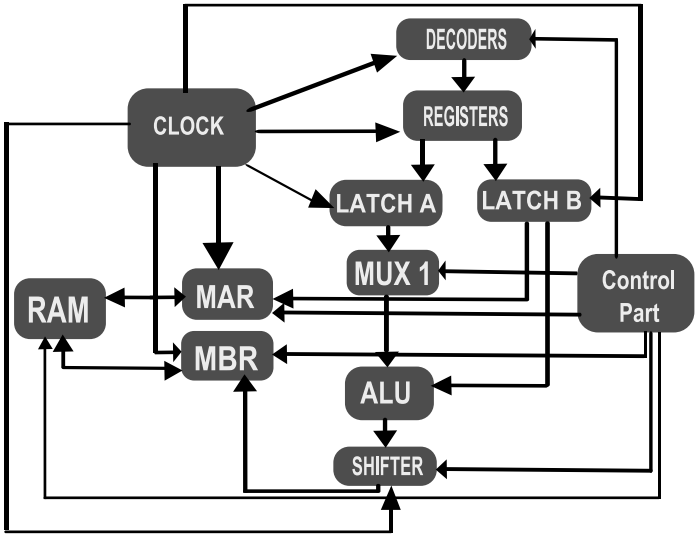

Fig. 3 Details of operation part

The control part is composed of (Fig. 4):

- **Icrementer:** This is only a block adder, taking the last address available on MPC and increasing unit.
- **MPC:** register those points toward the next microinstruction to be read from the control memory.
- **Memory Control:** Plays the role of storing instructions that enable the system to perform the tasks required by microprogrammer.
- **MUX2:** another multiplexer, but in this case it selects if the MPC register increments or receives a new value from the MIR control word.
- **MIR:** register controlled by the first clock subcycle, is responsible for ensuring safety, avoiding transients, the word out of control memory.

The Mac-1 has microinstructions, such as: PC, adds, jump, AC, and others, who are responsible for implementing functions like Lodd (loads DMA), STOD (Shop Direct) and JNEG (if negative changes). Thus, the microprocessor can perform multiplication, addition and division, demonstrating their efficiency and simple didactic.

The microcontroller implementation was done in VHDL language (VHDL-VHSIC Hardware Description Language VHSIC and stands for Integrated Circuits very high speed), which is the concurrent language, namely the commands involved in a given event happen simultaneously, unlike programming language software. The Fig. 1 shows the steps of a design using VHDL [3] and has been tested on board plate NEXYS 2 [4] with the Xilinx Spartan 3E FPGA XC3S500E [5].

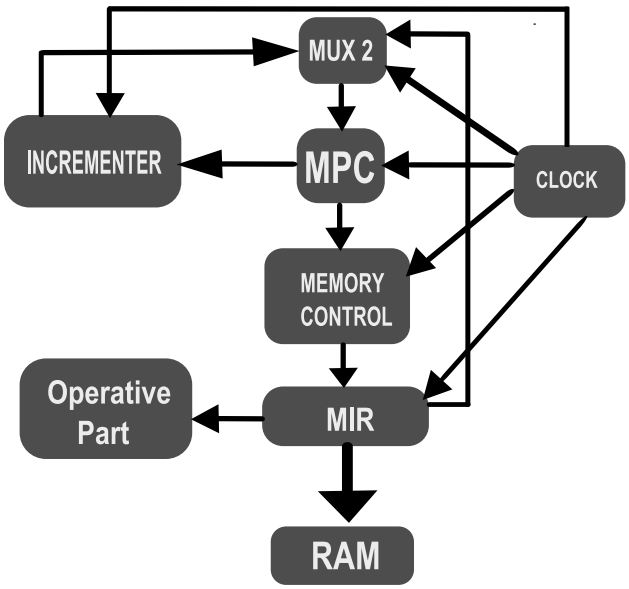

Fig. 4 Details of control part

To exchange information with the user's system was established to control the logic that provides for the use of the 7-segment displays (display is in hexadecimal) and 8 leds plate (displays results in binary form). Furthermore, to demonstrate the final results, we used the switches (5/1) that allow the user to select from a file of records concerning the registration or temporary memory to written information in RAM last. Moreover, to restart the system, simply use the 0 button [6] that the reset function is enabled.

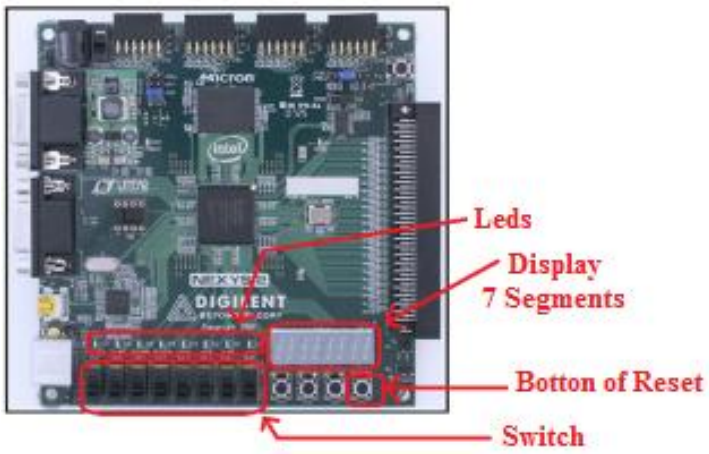

Fig. 5 Information board

In the Mac-1 is used the form of visual representation of the results since it is a didactic approach, and therefore it is necessary to ensure basic operation of the entire system on the platform, thus there is no need of external hardware and coupling any exchange of information with the computer.

III. MAC-1: MODIFICATIONS

Were necessary the following changes in the architecture of the book Tanembaum MAC-1:

- **Clock:** book description for subcycle clock is impractical, because it is not possible from one clock cycle to get 4 new subcycles. So we had to change it to 4 clock cycles that run synchronously.
- Latches: The latches were changed to 16-bit registers, for xilinx FPGA and not allow the construction of latches.

After these significant changes were possible to implement the architecture in VHDL so we had preliminary results both in implementation and in simulation.

IV. PRELIMINARY RESULTS

A. *Practical Results:*

The practical results were made Digilent Spartan 3E board FG-500 320, which has 2 USB ports, Flash Platform (configurating ROM) clocked at 50 MHz, 16MBytes Flash memory, 16MB SDRAM on, and its output: 8 leds , 2 pS2, data ports (8 bit color VGA, 2 RS232) and Expansion Connectors (PMOD x4 and Hi-Speed). However this work we used only the LEDs 8 and output the data, as can be seen with the video available at [6].

B. *Synthesis Results*

The simulated results were made ISE in the Xilinx Design Suite 12.0 (Fig. 6), as can be observed in Figure 2, showing the components that make up the architecture of MAC-1, there were simulated Flip Flops 385 and latches 16.

However, the architecture of MAC-1 had some problems during implementation, such problems are related to the latches, as it complicates the analysis of "timing" in synthesis tools, creating inconsistencies between the data read and written. Thus, these had to be replaced by registrars, which allowed the transmission of correct response, ensuring data consistency.

Another problem that was observed during the simulation of results was the use of a single cycle of cyclic clock for generating subcycles, because we can not generate timers as proposed in the book to delay a cycle and turn it into four subcycles, therefore, it was necessary to generate four clock cycles for implementing the architecture.

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Total Number Slice Registers	401	9,312	4%
Number used as FlipFlops	385		
Number used as latches	16		
Number of 4 input LUTs	2,387	9,312	25%
Number of occupied Slices	1,269	4,656	27%
Number of Slices containing only related logic	1,269	1,269	100%
Number of Slices containing unrelated logic	0	1,269	0%
Total Number of 4 input LUTs	2,436	9,312	26%
Number used as logic	1,331		
Number used as route-thru	49		
Number used for 32X1 RAMs	1,056		
Number of bonded IOBs	47	232	20%
IOB Flip Flops	11		
Number of RAMB 16s	1	20	5%
Number of BUFGMUXs	4	24	16%
Average Fanout of Non-Clock Nets	5.48		

Fig. 6 Results of the simulation architecture on Xilinx ISE Design Suite 12.0.

Finally, it was necessary to route the signal read / write to the MBR and then to RAM, as it was described in the architecture of Tanenbaum [2] there are issues of data consistency as when the signal is transmitted in response to MBR yet is not stable which generates a spread of incorrect response by the whole architecture.

V. CONCLUSION AND FUTURE WORK

Despite being a relatively simple architecture implementation requires a personal effort for its development and validation, while simple demonstrating that becomes a quite appropriate for a project discipline of digital systems or microprocessor.

In addition, future work will be implemented as improvements in control words to avoid repetition of commands that are generated in the current architecture, using a flag to indicate when the result of the MBR and MAR are stable. Finally, will be implemented using a CLP as microprocessor Mac-1.

REFERENCES

[1] Valente, J. A. O Computador na sociedade do conhecimento, OEANIED/UNICAMP, 1999.

[2] Tanenbaum, A. S. Organização Estruturada de Computadores, LTC, 1990.

[3] Conceitos Básicos. Available from: http://vhdl.com.br/site/aprenda-vhdl/ conceitos-basicos. Acessed at february 2013.

[4] Digilent. Available: www.digilentinc.com. Acessed at february 2013.

[5] Xilinx. Available: www.xilinx.com . Acessed at february 2013.

[6] Digilent Nexys2 Board Reference Manual. Available from: http://www. digilentinc. com /Data /Products/NEXYS2/Nexys2_rm.pdf, Acessed at february 2013.

[7] Pratical Results. Available: http://youtu.be/v5SnkSV79zE.

Antunes, Ana L. L. (2008). Algoritmos de Controle Distribuído em Sistemas. Universidade de Aveiro. Departamento de Electrónica, Telecomunicações e Informática Baseados em Microprocessadores, 2008.