A 8051 µC FUNCTIONAL RTL DESCRIPTION USING SYSTEMC FOR PLATFORM BASED DESIGN

Diogo José Costa Alves, Tiago Sampaio Lins, Silvio Veloso Freire Neto, Edna Barros
(djca, tsl, svfn, ensb)@cin.ufpe.br

Centro de Informática, Universidade Federal de Pernambuco

ABSTRACT

A micro-controller is a highly integrated chip including a CPU, timers, watchdog, and peripherals. The 8051µC is a very complete microcontroller, developed by Intel, with a large amount of built-in control store, enhanced I/O ports, and the ability to access external memory.

The advances in integration technology has allow a exponential grow of the transistors numbers in a chip. The result is that components (like micro-controllers) that used to occupy entire chips now fit in a tiny corner of a chip, and entire systems that used to occupy boards now fit on one chip (system-on-a-chip). To cope with this scenario, platform-based design has been introduced in order to reduce the time to market. Also SystemC is specification mechanism for digital systems, witch can be used to create system level model, quickly simulate to validate and optimize the design.

The main objective of this work is the specification and simulation of a 8051 micro-controller in SystemC RTL, which is being used in a platform based design methodology.

1. INTRODUCTION

Design productivity is significantly behind what semiconductor technology can deliver: average yearly increase of 28% vs. 58%, respectively [2]. Time-to-market pressure is increasing continuously problem.

Platform-based design is the process of composing a new design by reusing existing components. In this respect, the driving forces for creating such complex chips using externally supplied core.

Motivated with the idea of building a functional description of the 8051 that would be useful in platform based design, we developed a SystemC synthesizable[3] model of the 8051 and a environment to visualize the simulation results. In next session we will describe the main features of our 8051 µC functional description CORE.

2. THE 8051 MICROPROCESSOR

The 8051 is an 8-bit microprocessor originally designed in the 1980's by Intel that has gained great popularity since its introduction. Its standard form includes several standard on-chip peripherals, including timers, counters, and UART's, plus 4kbytes of on-chip program memory and 128 bytes of data memory, making single-chip implementations possible. Its hundreds of derivatives, manufactured by several different companies (like Philips) include even more on-chip peripherals, such as analog-digital converters, pulse-width modulators, I2C bus interfaces, etc. Costing only a few dollars per IC, the 8051 is estimated to be used in a large percentage all embedded system products.

The 8051 memory architecture includes 128 bytes of data memory that are accessible directly by its instructions. A 16-byte segment of this 128 byte memory block is bit addressable by a subset of the 8051 instructions, namely the bit-instructions. External memory of up to 64 Kbytes can be accessed by a special "movx" instruction. Up to 4 Kbytes of program instructions can be stored in the internal memory of the 8051, or the 8051 can be configured to use up to 64 Kbytes of external program memory The majority of the 8051's instructions are executed within 12 clock cycles.

The 8051 processor description includes the following features: the design is fully synchronous; the SystemC description (RTL Level) is technology independent [2] and can be easily expandable by adapting/changing SystemC source code.

This version does not include serial ports, interrupts, timers and external memory since we would like to be concentrated in the 8051´s instruction set.

3. DESIGN METHODOLOGY

We have started by studying the micro-controller in a lower level, the design process has included specification, implementation, and verification phases.

So we started studying the microcontroller based in a tutorial founded on web[4] that refers to the standard 8051 version that we implemented. We consulted many books like [1],[5] and due the lack of a more detailed description of the implementation, we have implemented the processor according to the block diagram, which can be seen in Figure 1.

The instruction set of the 8051 microcontroller was completely implemented, excepting instructions for peripherals manipulation (external memory, interrupts, serial ports, etc).

The RAM memory was implemented using an array data structure, some especial register, PSW and accumulate, can be read and write by a particular input/output ports dedicated for them since more than 50% of instruction set can accesses these registers.

The control unit has been implemented as a Finite State Machine (FSM), this module manage the fetch, decode and execution of the instructions.

We have also implemented a bit manipulate unit to resolve the bit access and manipulation in instructions that address bit. This unit communicates with the control unit by a common signal, that is activated when the control decode a bit manipulation instruction. In this case, the bit manipulate unit executes the mentioned instruction and disables the flag, after that; so that the control come back to the principal control module.

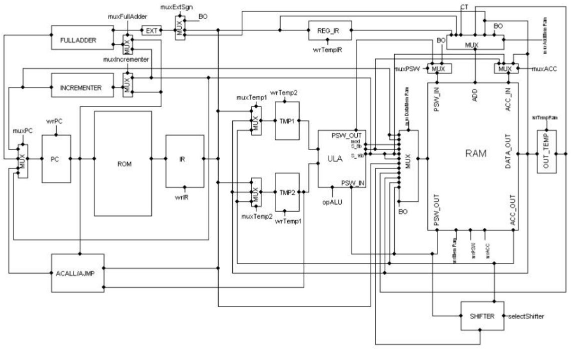


Figure 1 – Block Diagram

We have also developed a visualization tool in order to make easier the verification phase. It was too difficult know the contents of the RAM, and the values of the control signals without it. It was developed in JAVA. This visualization tool loads a trace file, which is generated during the execution of the 8051 SystemC description. The graphic interface of this tool can be view in Figure 2.

To develop some application we are using the Development Environment from Keil[6]. We can specify a program in 8051 assembly language or C language. The Keil Compiler is configured to build a HEX file after compiling process, which is loaded by 8051 and stored in internal ROM module when the simulation starts.

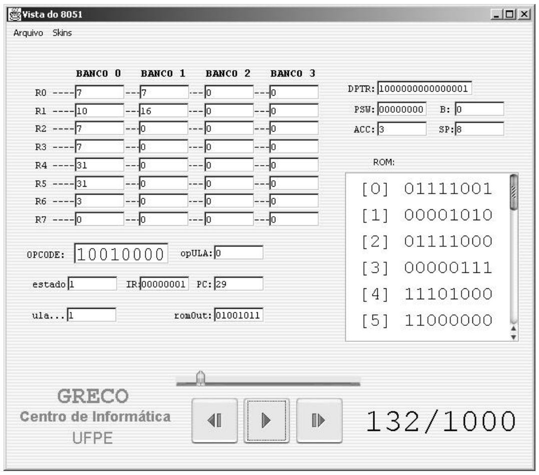
The verification process has been done in a hierarchical way: each module was validated individually and after that the whole description was simulated.

The functional verification of the 8051 SystemC description has been done by comparing the memory traces when executing a set of application in the proposed SystemC model and in the Keil simulator [6].

4. RELATED WORKS

Oregano Systems developed a processor core 8051[7] in VHDL that is configured by simple changing VHDL constants, serial interface, timers and interrupts, this IP Core is available for free in their website. We not yet implements serial interface neither timers nor interrupts but this work is been developing now and new versions this devices will be available.

The Dalton project focuses on the design of Intellectual Property (IP, or core) based embedded computing systems, i.e., systems-on-a-chip. Its present focus is on a "parameterized system-on-a-chip design" approach.



	BANCO 0	BANCO 1	BANCO 2	BANCO 3
R0	7	7	0	0
R1	10	16	0	0
R2	7	0	0	0
R3	7	0	0	0
R4	31	0	0	0
R5	31	0	0	0
R6	3	0	0	0
R7	0	0	0	0

DPTR: 1000000000000001
PSW: 00000000 B: 0
ACC: 3 SP: 8

ROM:
[0] 01111001
[1] 00001010
[2] 01111000
[3] 00000111
[4] 11101000
[5] 11000000

OPCODE: 10010000 opULA: 0
estado 1 IR: 00000001 PC: 29
ula... 1 romOut: 01001011

132/1000

Figure 2 – The Visualization Tool

They developed another 8051[8] VHDL implementation and benchmarks for tests in the same HEX format of ours implementation. The tool used for develop application for test in this project is the same that we used, the KeilSoftware[6].

5. CONCLUSIONS AND FUTURE WORKS

In this paper we presented an 8051µC functional description in SystemC. It includes the implementation of the complete instruction set and parallel ports.

The proposed 8051 SystemC model is being used in a Control Access System, a more complete system including the proposed 8051, an I2C bus, a card-reader, keyboard and display. The 8051 SystemC model has attained the demand of others modules and the result expected was reached.

Right now we are working on developing a complete description of 8051 microcontroller full compatible with the standard Intel 8051 with timers, interrupts and serial ports. The final objective of this work is describing an IP core of 8051 microcontroller that will be use on Fênix Platform in Brazil IP Project[9].

6. REFERENCES

[1] Predko, Myke Programming and Customizing the 8051 Microcontroller. McGraw-Hill, January, 1999.
[2] Küçükçakar, Kayhan, “Analysis of Emerging Core-based Design Lifecycle”, JOURNAL, PUBLISSHER.
[3] Synopsys SystemC Synthesis, “Describing Synthesizable RTL in SystemC”
[4] Site 8052.com, http://www.8052.com, in 10-jun-2003.
[5] Schultz, Thomas W.; Schultz, Tom, C and the 805: Hardware, Modular Programming & Multitasking – 2nd ed., Vol. I, Prentice Hall PTR, December, 1997.
[6] Keil 8051 Software, http://www.keil.com/demo/evaldl.asp?p=C51.
[7] Oregano Systems, 8051 Soft IP CORE, http://www.oregano.at/ip/8051.htm.
[8] Dalton Project, www.cs.ucr.edu/~dalton/8051/.
[9] Brazil IP Project, www.brazilip.org.