Functional and cycle accurate models of PIC 16F84 microcontroller for IPs reuse based design

G. R. Laureano, L. Taglietti[*] , L. C. V. dos Santos[†]
Computer Science Department
Federal University of Santa Catarina
Florianópolis, SC, Brasil
{laureano,leonardo,santos}@inf.ufsc.br

ABSTRACT

The ascension of Systems-on-Chip (SoCs) is a result of millions of logic cells available in a single chip and the demand imposed by embedded systems industry for applications each time more challenging. The platform based design for SoCs is an answer for the high prices to engineering of common technologies and it is based in a reference architecture to IPs reuse. The IPs reuse works suitable only if the design space exploration is supported and the interface is designed to be reusable. The design space exploration asks for efficient models of each IP. This paper is about an efficient modeling IP in a two abstraction levels, one purely functional and another functional with cycle accurate.

1. INTRODUCTION

The evolution of integrated circuits, that is drove by Moore Law[1], result in a millions of logic cells in a single silicon device. This huge hardware available, combined with the growing demand for applications each time more challenging imposed by embedded systems industry result in a appearing of hardware (CPU, memory and peripherals) and software integrated systems called Systems-on-Chip (SoCs) [1].

To reduce the development costs and accommodate the time-to-market, the SoCs design leads to adopt a target platform [7], that is a reference platform built to allow the reuse of its components, called intellectual property blocks (IPs). This platform can be customized to a specific application, through of the remotion of unneeded IPs and inclusion of new IPs.

The process of platform customization asks an exploration of alternatives architectures resulting of differents selection of IPs sets. This exploration involve a verification of the system behavior in front of functional requisites, performance estimative in front of real time restrictions and power estimative in front of low power need. These asks for available efficient models for each IP.

This work talks about the efficient modeling of an IP, in two different abstraction levels: one purely functional and another functional with cycle accurate. The modeled IP corresponds to an microcontroller very used by industry, the PIC 16F84. Both models are formally describes and validated through set of benchmarks.

2. THE PIC 16F84 MODELING

The PIC is a microcontroller with architecture inspired by RISC, it is based in accumulator and adopt the Harvard machine. The data word has 8 bits while the instruction word has 14 bits. PIC has 35 instructions that are grouped in 3 different formats. All instructions has latency 0[2] while branch instructions has latency 1.

2.1 The architecture description language

To describe the PIC microcontroller, both at level purely functional and functional with cycle accurate, it was adopt an architecture description language (ADL) called ArchC. ArchC is licenced under the GNU Lesser General Public License (LGPL) and it generates simulators written in SystemC [8] automatically, and this is the main reason for the ArchC adoption in this project.

Even the simulation and co-verification are the beginning focus of ArchC, this ADL was design to offer support to tools like assembler generators [9] and other tools that are still over development. Other architectures like MIPS and i8051 are modeled too with ArchC language and they are available at ArchC official site [10].

An ArchC description is composed by two main parts. The first is the architecture description (AC_ARCH), where the organization of architecture components is described, such as pipeline structure and memory organization. The second part is the description of the instruction set architecture (AC_ISA), where the instruction characteristics such as names, formats, fields and sizes are defined. In this part also are mapped the mnemonics and operational codes.

After describe this two parts, the ArchC preprocessor can generate the framework of the simulator, that contains all methods signature that specify the behavior of instructions [10, 6].

2.2 The purely functional model

The purely functional modeling of a system does not worry about how the implementation is made. In this abstraction level only the correlation between inputs and outputs

[*]Co-Advisor.
[†]Advisor.
[1]The Moore Law says that each 18 months, the number of transistors in a integrated circuit (CI)) is multiplied by 2.
[2]Latency is used in this context, to express the number of cycles consumed between the end of instruction fetch and the beginning of next instruction fetch.

are important. For describe both purely functional and cycle accurate models of PIC 16F84 microcontroller we adopt ArchC [10].

The purely functional model was concluded and validated. After that it was certificated by ArchC support team, and it is available at ArchC public domain repository [10].

2.3 The functional model with cycle accurate

Besides the purely functional characteristics, the cycle accurate abstraction level also capture the delays of each operation execution. To build such model of PIC microcontroller was need a reverse engineering to deduce the pipeline and datapath since the user manual [4] is not much clean about that.

3. EXPERIMENTAL RESULTS

3.1 Experimental configuration

The experiments were realized in a PC computer with Intel® Pentium 4 processor, with 1.8 GHz of frequency and 256 MB of main memory. The operational system was Debian GNU/Linux, with kernel version 2.4.25-1-686. To generate the simulators, the ArchC ADL version 0.8.2 was used. The PIC compiler used was the PCW® (PIC C Compiler), version 3.182, with IDE version 3.41. The i8051 compiler used was Keil's i8051 C compiler.

3.2 Comparisons with i8051 model

This section shows a comparison between PIC model and i8051 model. The i8051 model is already certificate and is available at BrazilIP repository [2]. The quantitative results that will be shown was generated using the Dalton Project benchmarks [3], that was used to validate both models.

For each benchmarks, on table 1, there are comparisons between the two microcontrollers at number of executed instructions, time simulation and code size.

Benchmarks	Instruction Executed		Time Simulation		Code Size	
	PIC	8051	PIC	8051	PIC	8051
negcnt	150	312	2981	6221	60	33
int2bin	256	2659	5101	53161	72	55
cast	68	995	1341	19881	120	112
divmul	296	404	5901	8061	212	190
fib	358	1218	7141	24341	140	298
sort	2227	4245	44521	84881	220	534

Table 1: Data about comparisons

It is easy to note that the number of instruction executed is always less for PIC. This can be explain for the difficulty in generate code for CISC architecture of the i8051 or optimization deficiencies of i8051 compiler.

For all case tests, the time simulation of PIC 16F84 is less than i8051 times. For instance, the *sort* bench, executed almost 2 times more fast for PIC then i8051. For the *cast* bench, the PIC model executed almost 15 times more fast. The great times of i8051 can be explained by its more complex instruction set, that has many formats of instruction and size variable, while PIC instruction set has fixed size and only 4 different formats.

With a PIC cycle accurate model available another analyzes can be realized, the number of lost cycles because of control hazards. For 8 benchmarks of Dalton Project were found, at average, 17,59 % of lost cycles, 30,23 % to the worst case (*negcnt*) and 10,71 % to the best case (*sort*). This values could be improved with a branch prevision technic [5].

4. CONCLUSION AND PERSPECTIVES

The models built were shown robust in front of the number of experiments realized and they were efficient when compared with the already validate model of i8051.

The authors do not know other similar models available in public domain, what make this work a first contribution to the IPs reuse based design.

These models can still be used to tools automatically generation, such as simulators and assemblers that already are available, besides others under development.

With the cycle accurate model available, it is possible, for example, the validation of tools that need timing, such as code scheduler and code debug with cycle accurate. The model can still give informations for a estimative of power consume tool.

5. ACKNOWLEDGMENTS

We would like to acknowledge the ArchC support of the Computer Systems Laboratory at the Institute of Computing of the University of Campinas.

6. REFERENCES

[1] R. Bergamaschi. A to Z of SoCs. In *Tutorial apresentado na Escola de Microeletrônica da SBC Sul (EMICRO 2002)*, Florianópolis - Brazil, 2002.

[2] Brazil-IP Network. The Fênix Platform, 2004. http://www.brazilip.org.br/fenix.

[3] Dalton. Synthesizable VHDL model of 8051, 2001. http://www.cs.ucr.edu/ dalton/i8051/i8051syn/.

[4] Microchip, 1998. PIC 16F8X: 18-pin Flash/EEPROM 8 Bit Microcontrollers.

[5] D. A. Patterson and J. L. Hennessy. *Arquitetura de Computadores: Uma Abordagem Quantitativa*. Campus, 3rd edition, 2003.

[6] S. Rigo. *ArchC: Uma linguagem de descrição de Arquiteturas*. Instituto de Computação. Universidade Estadual de Campinas - Brasil, Julho 2004.

[7] A. Sangiovanni-Vincentelli and G. Martin. Platform-Based Design and Software Design Methodology for Embedded Systems. *IEEE Design & Test of Computers*, 18(6):23–33, November-December 2001.

[8] SystemC Homepage, 2003. http://www.systemc.org.

[9] L. Taglietti, J. O. C. Filho, D. Casarotto, O. J. V. Furtado, and L. C. V. dos Santos. Automatic ADL-based Assembler Generation for ASIP programming support. In *The 3rd International IEEE Northeast Workshop on Circuits and Systems*, Québec City, Canada, June 19-22 2005. http://www.newcas.org.

[10] The ArchC Architecture Description Language, 2003. http://www.archc.org.