Implementation of an Application Specific Microprocessor for PWL Computations using Synopsys.

A first pass successful development of a 150 K transistors micro architecture.

J. Agustín Rodríguez, Victor Jiménez, Omar Lifschitz, Pedro Julián, Osvaldo Agamennoni.

Instituto de Investigaciones en Ingeniería Eléctrica (UNS-CONICET)
Departamento de Ingeniería Eléctrica y de Computadoras
Universidad Nacional del Sur

September 2008

Introduction

Eighteen months ago, we started the analysis of a mathematical problem which was selected as the target application for the first fully automated ASIC development of the IIIE (Instituto de Investigaciones en Ingeniería Eléctrica). The PWL (Piece Wise Linear) Function Computation problem was chosen because this kind of function allows the representation of n-dimensional non-linear functions in a convenient way for computing systems, and non-linear functions are of relevance in control systems, one of the most active areas at the IIIE.

After defining and analyzing the computation algorithm, we concluded that a special microprocessor architecture was the most appropriate approach for this ASIC; the programming capability of this design included the flexibility required for PWL computations of 2 to 6 dimensional functions and the dedicated hardware structures provided the required performance. Also, the challenge of developing a microprocessor was an extra motivation for the design team.

The next step was the implementation of the design; it was done using a top down design flow where Synopsys' tools played the major role. After 12 months of design and implementation we sent the PWL uP's layout to MOSIS. Today, after the testing procedure, we have our PWL u-P working properly in a first pass success.

Note: the reader is referred to [1] and [2] for more details about PWL Computations and the implemented micro architecture.

Design Flow and Synopsys Tools

Working with Synopsys' tools was quite successful as well as challenging. The process that started with the installation of the software, continued with being a user from scratch and ended when we obtained what we wanted, was a valuable experience in electronic design automation.

The RTL level description was written in VHDL. Obviously, logic synthesis was done using Design Compiler. Due to the importance of this step we will dedicate the next section to describe this particular experience. For simulations of RTL and Design Compiler output's Gate Level description we opted for Mentor's Modelsim due to our long experience with it. We actually didn't know about VCS.

While simulating the Gate Level we detected some errors in the digital specification. Finding these errors was not an easy task; we had to simulate many times until we were able to identify what was failing. Of course, if we had known the capabilities that Formality provides we could have worked faster. But that didn't happened until the PWL u-P was ready for tape out, a little late but to be taken into account.

Since IC Compiler was not yet available for universities, physical synthesis was done using Cadence's tools. For our next development we will do the placement and routing with IC Compiler and we will certainly compare it against Encounter.

After P&R we used Prime Time (PT) to check the obtained blueprint layout. Using the DSPF extraction we verified that constraints were still met. Only one capacity constraint was violated but, after spice simulations of that part of the circuit, we concluded that it would not cause problems. As beginners we were not confident about Sign Off Static Timing Analysis (STA), so we did some fast spice simulations of the full extracted circuit. Again, our previous experience forced us to work with Cadence Ultrasim. For our next project we may probably avoid these 24/36 hour simulations because they only proved that STA was correct. PT also showed its potential at the testing stage. For example, max clock frequency were estimated for different Vdds and they were actually the frequencies where the IC started to fail; power predictions done with PT were quite similar to real values, specially the ratio between clock tree power and total power, predicted equal to 0.7 and measured equal to 0.7 too.

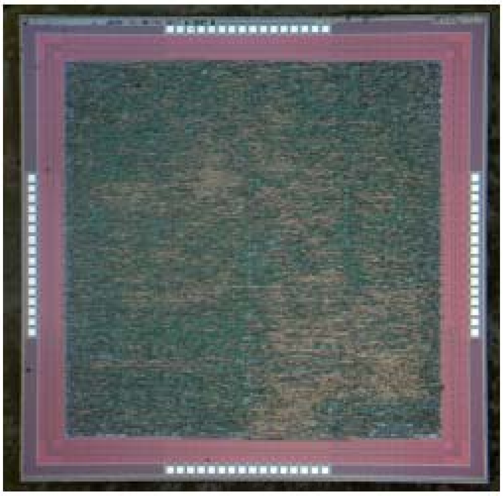
Design Compiler

Design Compiler proved to be a consolidated tool. It is usable and powerful, and we obtained good results quickly. The first interesting characteristic we noted is its ability to optimize combinational structures like the adder for different clock frequencies, jumping from the ripple adder to CLAA and more parallelized blocks as frequencies are higher. We used it to evaluate the performance and area of different implementations of digital blocks, i.e. the addressing logic for a 256 word internal RAM. As we went ahead with the constraints we found other relevant features like the possibility to synthesize with more than one clock.

The constraints evolved from a very simple setup, defining just one clock and input and output loads, to a more detailed one with three clocks, specifying false paths between clock domains, ideal networks (rst), and more precise signal characterizations (i.e. rising and falling times).

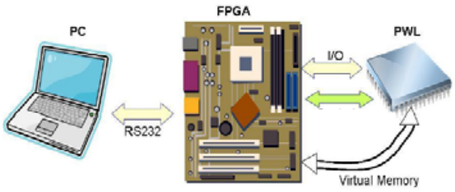
The PWL uP

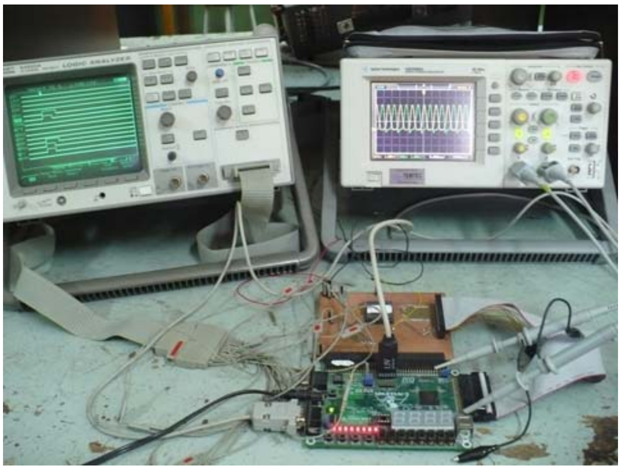
PWL uP’s die is 4×4 mm in the AMI 0.5 µm process; it was fabricated by the MOSIS service. This IC is composed of 150 K transistors and has been optimized for a 66 Mhz clock frequency.



Testing

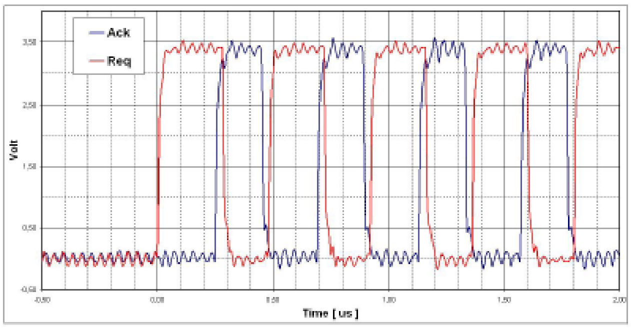
A testing environment was configured with a FPGA and a PC. The FPGA was used to emulate the external data RAM required for the system and to implement the I/O protocols required by the PWL uP. The PC was used as the user interface of the system.



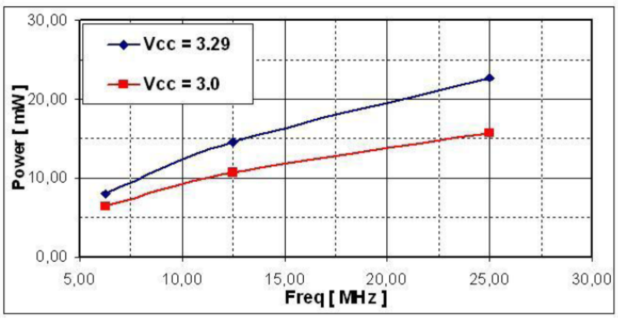


Many programs were loaded and executed in the PWLuP in order to test the different blocks and sequences of instructions; IO blocks, Register File, ALU, Memory interface were tested with different routines that i.e sorted a sequence of values from IO or added a sequence of values from Memory. DFT structures played a mayor role in partial results checking and verification.

The following figure shows the oscilloscope measurement of request and acknowledgment signals of one IO protocol.



The two main programs of this application specific processor were also executed. The simpler one reads a sequence of values from the IO and stores them in the data RAM. The other is the PWL function evaluation routine that computes the value of F(X) for a given X set by the user. Different cases were executed and results were consistent with mathematical simulations.
The circuit was tested at different frequencies reaching a maximum of 54 Mhz for Vcc = 3,3 V. Measurements for Vcc = 5 V are pending. Power was also measured at different frequencies and Vcc values.



Conclusion

The successful result of this project is consequence of hard working (we can not let the EDA tools take all the merits) and a mature software environment. EDA tools free up the designer to concentrate on system levels characteristics and application specific requirements, they reduce human errors very much and they enable really fast implementations. The time between the idea and the silicon is becoming shorter and the manageable complexity increases fast, however designers still need to design.

Acknowledgements

We would like to specially thank to our Guide in EDA design flow, Alfonso Chacon Rodriguez, for his support and patience. Thanks to Victor Grimblatt from Synopsys Inc. for his support and for inviting us to the “Escuela de Verano: Diseño de CI Digitales” January 2008, Santiago, Chile, where we learnt much of the EDA foundations. Thanks to Dave Wilder from Synopsys Inc. for his interesting lessons and tutorials, and for his key tips for Design Compiler.

References

[1] V. M. Jimenez, J. A. Rodriguez, P. M. Julian, O. Agamennoni, M. Di Federico, “Digital architecture for R6 PWL function computation”, *in Proc. Arg. School of Micro Nanoelectronics*, pp. 1-6, 2007.

[2] J. A. Rodriguez, V. M. Jimenez Fernandez, P. Julián, O. Agamennoni, O. Lifschitz, “VLSI Microprocessor Architecture for a Simplicial PWL Function Evaluation Core,” *Actas de la Escuela Argentina de Microelectrónica, Tecnología y Aplicaciones: Trabajos Regulares*, Vol. 1, No. 1, pp. 55-60, Sept. 2008, (editado en CD) ISBN 978-987-655-003-1.