MODELING OF ADC ARCHITECTURES IN HDL LANGUAGES

Marco Oliveira, Nuno Franca

Modeling Group, Chipidea Microelectronics, Inc.
Taguspark, Edifício Inovação IV, sala 733, 2780-920 Porto Salvo, Portugal
Phone (351) 21 422 01 70 Fax (351) 21 422 01 89
e-mail: marco.oliveira@chipidea.com

Abstract - *This paper describes the modeling of A/D converters in HDL languages such as Verilog [1] and VHDL[2]. It starts with an introduction about the importance of hardware modeling to support the flow of modern integrated circuit design, followed by the presentations of two case studies. The first case study is an A/D converter with successive approximations architecture [3][4][5]. The functional model was written in VHDL. The second case study is an A/D converter with pipelined architecture [3][4][6], whose functional model was written in Verilog.*

A/D – Analog to digital,
ADC – Analog to digital converter,
DAC – Digital to analog converter,
HDL – Hardware description languages.

1. INTRODUCTION

The design of a mixed-signal electronic system is a complex task. It is imperative to develop capable tools and methodologies to decrease the design cycle lifetime because of increasing competitive market forces. Developing functional and custom models of analog blocks improves design reuse, portability, decreases the likelihood of errors in the transistor level design and allows the system architecture to be rapidly tested to verify [7] interface and/or functional problems. Furthermore it is possible to carryout a mixed signal simulation of analog and digital blocks to qualify the overall functionality of complex mixed-signal system.

It is easier to solve a problem or correct an idea in a functional model of a system than in a transistor level phase. The simulation runs are shorter and problem solving becomes easier. With a library of analog circuit functional models one can test and see rapidly if a selected architecture solution is adequate. Then, the designer will determine the required specifications for the individual analog blocks to achieve the desired goals. At Chipidea, the design flows starts with a system level architecture analysis with Matlab/Simulink [8] followed by an HDL functional/behavioral model of the architecture.

This paper presents two case studies of ADC functional modeling recently implemented at Chipidea. In the first example, two levels of models were developed: one behavioral and one structural. The behavioral model contains a high-level functionality description whereas the structural model contains a behavioral model for each analog block used in the system. In the second example a structural model was developed.

2. CASE STUDY I – 10 BIT SUCCESSIVE APPROXIMATION ADC

For this first case study a functional model of a 10-bit Successive–Approximation ADC for application in Touch Screen Monitors is presented.

The functional model for this ADC was written in VHDL.

2.1 Structure

The top symbol for the Successive–Approximation ADC of 10 bits is presented in Fig. 1.
The reference voltage is configurable in differential and single ended topologies.

The input voltage is differential with the negative input always connected to the negative supply voltage.

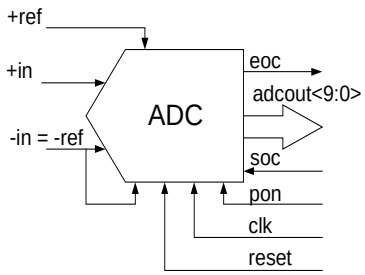

Fig.1 – Top Symbol of the ADC for touch screen applications.

The ADC has two voltage reference supplies and two input voltages. The negative input voltage is always connected to the negative power supply. The other inputs of the ADC are:

- clk – clock pin. Clock is at 480 Khz
- soc – bit indicating start of conversion;
- pon – power on bit;
- reset – reset bit, puts output in half scale (adcout = 1000000000)

The outputs of the ADC are:

- eoc – end of conversion;
- adcout – result of conversion.

ISBN 972-98115-5-5 © 2001 IT

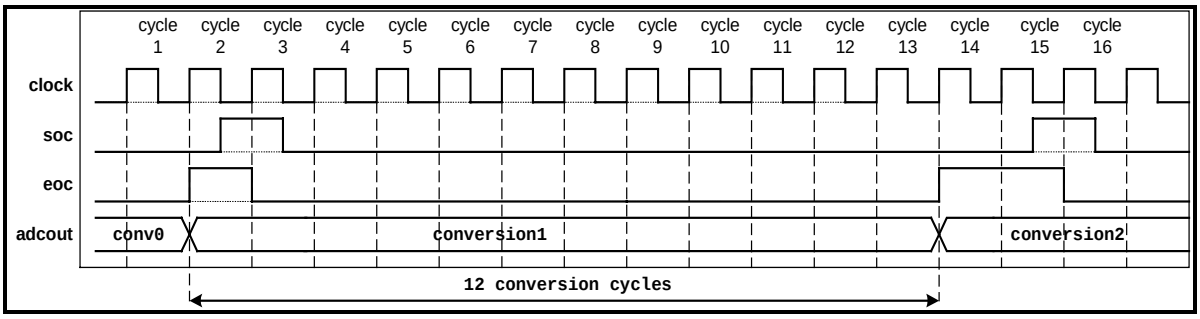

Fig. 2 – Desired timing diagram for the PenADC converter.

The timing diagram of a full conversion cycle of the ADC is shown in Fig. 2 when the conversion only starts with soc and eoc bit high.

The inside structural architecture of the ADC is described in Fig. 3. It is built, ideally, with one sample and hold, one voltage comparator, one SAR and one DAC that outputs the voltage for comparison.

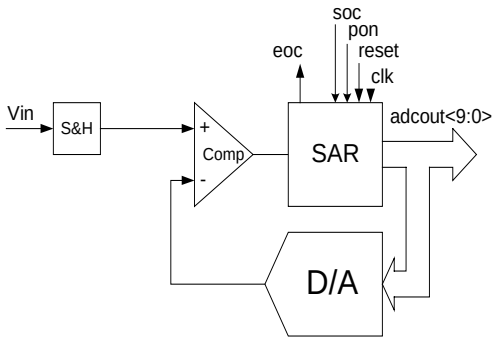

Fig. 3 – Internal Structure of the Successive Approximation ADC for Touch Screen.

The sample&hold takes samples of the input voltage when a conversion starts. This input voltage is compared with the output of the DAC for each clock cycle. The SAR and the result of the comparison determine the digital input of the D/A. After 12 clock cycles the eoc bit goes high waiting for a soc at high to appear and start all over again.

2.2 VHDL Model

There were two versions written, one behavioral and one structural.

The behavioral version models the functionality of the ADC from the inputs to the outputs. This first approach is used to check the behavior of the circuit and speed up simulation.

The structural model was built connecting all the models of the individual analog blocks (comparators, D/A, Sample and Hold) with the digital code of the SAR. This brings a more realistic approach to the structure of the circuit, thus allowing the validation of both analog and digital blocks simultaneously.

Below is the definition of the entity for the functional model:

```
entity touchScreen_adc is
  -- Parameters definition
  generic(
      nrbits : integer := 10
  );
  -- I/O definition
  port(
     refn: in real;  -- Positive power supply
      refp: in real;  -- Negative power supply
     clk: in std_ulogic;-- Clock Signal
     pon: in std_ulogic;-- Power on  Pin
     reset: in std_ulogic; -- Reset Pin
     soc: in std_ulogic; -- Start conversion signal
     inn: in real; -- Negative input
     inp: in real; -- Positive input
     adcout: out std_logic_vector(nrbits-1 downto 0); --Result of conversion
     eoc: out std_ulogic; -- End of conversion Bit
   );
      end penadc_adc;
```

2.3 Results

The simulation results are shown in Fig. 4. Both the

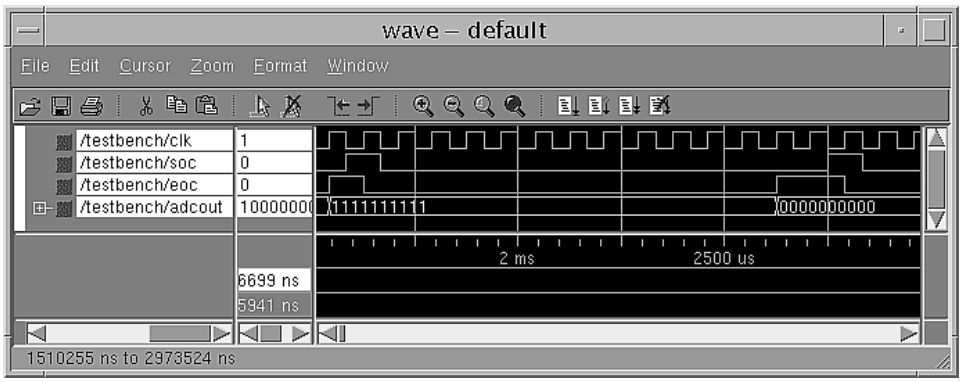

Fig. 4 – Simulation result for Successive Approximations converter.

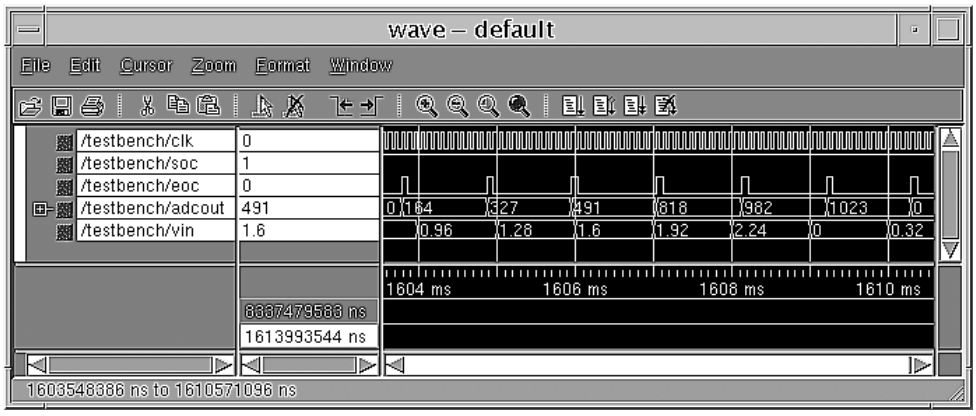

Fig. 5 – Simulation result for Successive Approximations converter for different input voltages.

behavioral and structural models exhibit the same output waves. In this test it can be seen that the eoc bit, when is high, waits for a soc bit high at a clock raising edge to start a new conversion. In Fig. 5 are represented several conversion cycles with the corresponding results and input voltage of the ADC.

3. CASE STUDY II – 10 BIT PIPELINE ADC

In this second case study a behavioral model of a fully differential pipelined ADC for applications requiring medium resolutions and high-speed conversion rates such as video and imaging is presented. This ADC has a 10-bit digital output code and employs digital error correction techniques.

The functional diagram of the ADC is shown in Fig. 6. The voltage references and inputs are differential. The ADC has the following I/O ports:

- vcm – input/output for common mode voltage decoupling and bypassing;

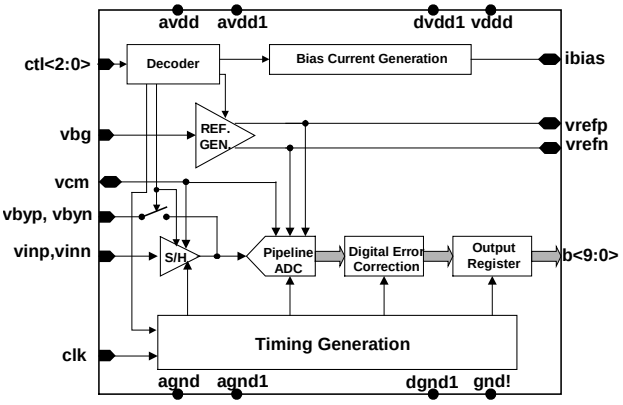

Fig. 6 – Functional diagram of the pipelined ADC

- vbyp, vbyn – input for bypass of S/H;
- vinp, vinn – input voltages;
- clk – clock input;
- ibias – bias current;
- vrefp, vrefn – reference voltage;
- b – output conversion code;
- clkout – clock output for data out synchronization;
- ctl – operating modes control.

This ADC as a structural architecture composed of a decoder, a current generator, a voltage reference generator, a sample&hold, a pipeline ADC, a digital error correction block, an output register and a timing generation block.
Besides several others, operating modes, the conversion mode 0 (normal conversion), of this ADC is selected by putting all ctl bits to low. In this mode the bias current is internally generated and the reference and common mode voltages are internally buffered.

The sampling rate of the ADC is defined by the frequency of the clk signal. The input voltage is sampled in the falling edge of clk. This input voltage is then processed by the pipeline stages.

The digital words (1.5 bits per stage) are synchronized by a chain of delay stages, overlapped and processed by the digital error correction logic to produce the 10-bit digital output code. The result is latched in the output register on the falling edge of clk with a latency of 5 clk periods the conversion timing diagram is shown in Fig. 7, where Tsd is the clock falling edge to sampling instant delay, Tod is the clock falling edge to data out delay and Ts is the sampling period.

3.1 Verilog Model

For this case a structural behavioral model was developed.

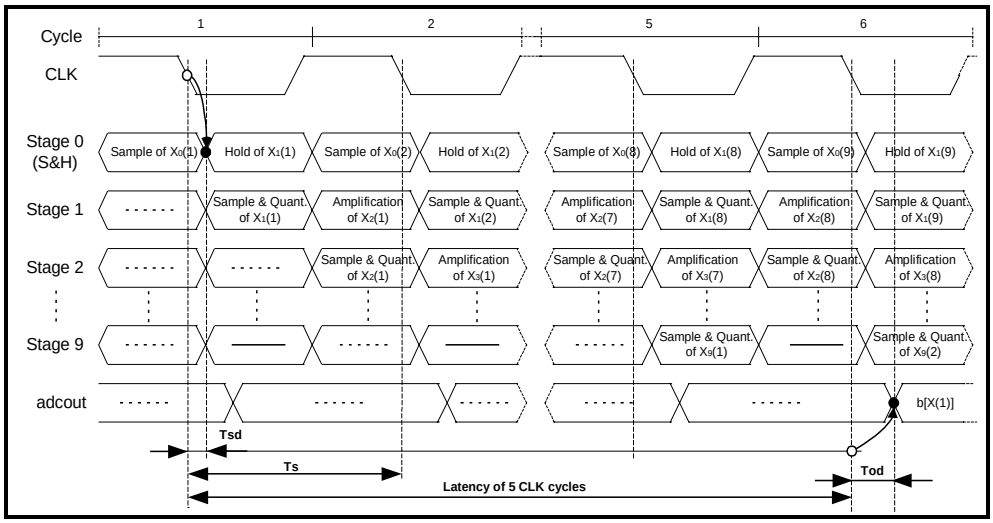

Fig. 7 – Desired timing diagram for the pipelining operation.

The structural model contains individual behavioral descriptions of the circuit components. The various functionalities in the real circuit where simplified and then proceeded to an architecture segmentation. This procedure isolates the parts that were more closely related and improves the circuit/model understanding, debugging and reusability.

The ADC model is built from the sub models of: ADC pipeline, reference voltage generator, digital correction logic, clock phases generator, a bias current generator and an output latch. The model was written in Verilog HDL language and this is the model header:

```
module ascad1032 (  b, clkout, ibias, vrefn, vrefp,
                    clk, ctl,vbg, vbyn, vbyp,
                    vcm, vinn, vinp );
```

3.2 Results

Fig. 8 shows the resulting timing diagrams from simulation. In this figure, the rl_vout signal is the sampled voltage from the sample and hold (see Fig. 7) and the stage signals are the output codes of the pipeline stages. Fig. 9 shows the output conversion code for different input voltages.

4. CONCLUSIONS

Modeling analog circuits in HDL languages is an important issue in modern integrated electronics. It improves the testing capability and verification of the circuit functionality as well as its reusability.

Having entire systems, and/or portions of it, modeled in HDL languages, decreases the project cycle time and thus reducing development cost and time to market.

Doing a prior modeling of the circuits behavior it helps detecting behavioral/functional bugs.

This paper described the importance of hardware modeling to support the flow of modern integrated circuit design with the presentation of two case studies. The first case study was an A/D converter with successive approximations architecture with functional model written in VHDL. The second case study was an A/D converter with

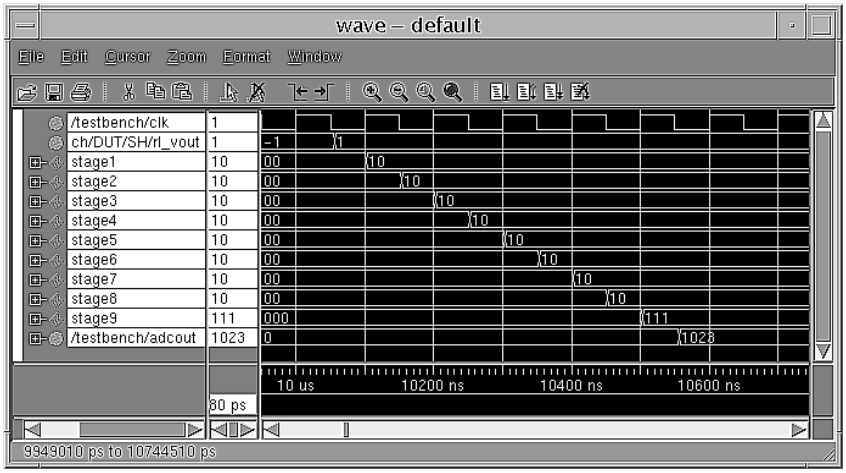

Fig. 8 – Simulation Result for pipeline AD converter for all stages.

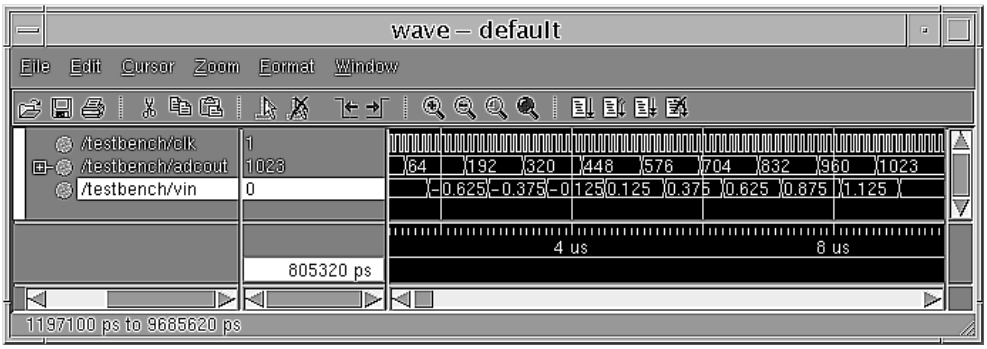

Fig. 5 – Simulation result for Successive Approximations converter for different input voltages.

pipelined architecture, whose functional model was written in Verilog.

ACKNOWLEDGMENTS

We would like to thank to our colleges Carlos Fachada, Luís Franco, Luís Laranjeira, Paulo Pereira as well as Profs. João Vital and Carlos Azeredo Leme for their support and useful information regarding the presented case studies.

REFERENCES

[1] James M. Lee, *Verilog QuickStart*, Kluwer Academic Publishers 1997

[2] Peter J. Ashenden, *The Designers Guide to VHDL*, Morgan Kaufmann 1996

[3] Rudy van de Plassche, *Integrated Analog to Digital and Digital to Analog Converters*, Kluwer Academic Publishers 1994

[4] David A. Johns, Ken Martin, *Analog Integrated Circuit Design*, John Wiley & Sons 1997

[5] Chipidea Microelectrónica, SA, *ADC for Touch-Screen Target*, Advanced Data Conversion Products 2001

[6] Chipidea Microelectrónica, SA, *AD1032AT57K*, Advanced Data Conversion Products 2000

[7] Janick Bergeron, *Writing Testbenches- Functional Verification of HDL Models*, Kluwer Academic Publishers 2000

[8] John O. Attia , *Electronics and Circuit Analysis using Matlab*, CRC press 2001