A VHDL-AMS Simulation Methodology for Transient Supply Current Extraction

Richard Perdriau, Damien Lambert, Anne-Marie Trullemans, Mohamed Ramdani

A VHDL-AMS Simulation Methodology for Transient Supply Current Extraction

Richard Perdriau[*/**], Damien Lambert[*/***], Anne-Marie Trullemans[**],
Mohamed Ramdani[*] and Jean-Luc Levant[***]
[*]Ecole Supérieure d'Electronique de l'Ouest
4, rue Merlet-de-la-Boulaye - BP 926 - 49009 Angers Cedex 01 - France
[**]DICE - Laboratoire de Microélectronique - Université Catholique de Louvain
Place du Levant, 3 - 1348 Louvain-la-Neuve - Belgium
[***]ATMEL
La Chantrerie - Route de Gachet - 44000 Nantes - France
Corresponding author : Richard Perdriau
Tel. (33/0) 2 41 86 67 03 - E-mail : richard.perdriau@eseo.fr

Abstract

Transient supply current extraction plays a very important role in estimating performance level in the IC EMC field.
As far as complex circuits such as microcontrollers are concerned, transistor-level (SPICE-based) simulation leads to very long CPU times, mainly because of memory arrays which often represent more than 80 % of the transistors in a μC. A convenient way of cutting out simulation times is to describe memories at the behavioral level, while keeping the microcontroller core itself at the structural level.
In the first step, the dynamic supply current consumption of the microcontroller core alone is simulated by coupling purely digital (VITAL) VHDL models for Flash and RAM blocks to the transistor-level core. The next step consists in adding VHDL-AMS behavioral models of the consumption of the memory blocks themselves to the VITAL descriptions. This allows the designer to simulate the whole microcontroller without dramatically increasing simulation time.
This work is supported by the MESDIE project and complies with the ICEM proposal.

1 Introduction

When designing complex integrated circuits such as microcontrollers (μC), extreme care must be taken about their compliance to electromagnetic compatibility (EMC) rules; in fact, the later these rules are taken into account in the design phase, the more non-recurrent engineering (NRE) costs may increase, at first because of foundry costs. Therefore, a methodology allowing the designer to predict conducted emission levels before sending the design to the foundry would be of great interest.
The recent Integrated Circuit Electromagnetic Model (ICEM) proposal [5], represented in figure 5, uses an equivalent current generator to model internal activity. Consequently, extraction of dynamic supply current by the means of simulation displays an accurate image of this activity; even if only one given activity is needed in the proposal itself, being able to provide software-dependent currents opens new perspectives.
The first idea coming to mind in order to obtain instantaneous currents is to perform a transistor-level simulation on the complete netlist, using SPICE-like tools. However, many microcontrollers include on-chip SRAM, Flash EPROM and/or EEPROM. These memory blocks often contain many more transistors than the CPU core itself; hence, simulating millions of transistors at SPICE level would be tedious and time-consuming. Moreover, EMC studies do not require as much accuracy as the one provided by these simulations; indeed, results might be acceptable when the discrepancy between simulation results and measurements is as high as 20 %.
On the contrary, many authors have already demonstrated the advantages of behavioral simulation models over structural ones for complex systems, mainly in terms of simulation time. For that purpose, many high-level, mixed-signal languages have been developed in the last decade, including HDL-A, MAST and more recently VHDL-AMS and Verilog-AMS.
Nonetheless, as is well known, microcontroller core mod-

els are written in purely digital VHDL or Verilog in order to be synthesized; in addition, VITAL (VHDL Initiative Towards ASIC Libraries) behavioral models of memory blocks are often available. In fact, VITAL is a standard (IEEE 1076.4-1995, enhanced in 1999) which (among other features) allows memory read/write operations to be simulated in VHDL, including timings and path delays.
The use of a similar, compatible modeling language seems thus to match the requirements of a global simulation.
VHDL-AMS, described in [1], is a mixed-signal, multi-technology extension to VHDL, covering various domains including electronics, mechanics and optics. Moreover, purely digital VHDL models may be imported, compiled and simulated with no modification, and furthermore the available test models are written in VHDL. Nonetheless, the ICEM proposal allows the designer to specify the equivalent generator quoted above as a VHDL-AMS description. Therefore, it seemed that the VHDL-AMS language was best suited to our application.
Consequently, this paper introduces another technique, based on mixed-level (behavioral and structural) simulation, which may eventually reduce CPU times in a significant way, while preserving enough accuracy to meet the goals of an EMC-oriented study.

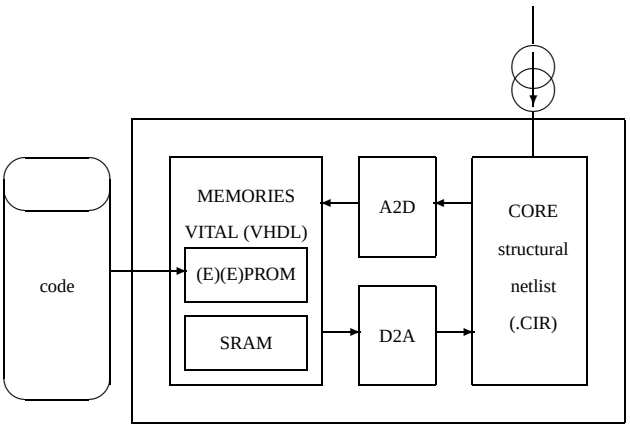

Figure 1. Simulation setup with VITAL memory models

2 Presentation

2.1 The methodology

From what has just been stated, we can suggest a 3-step process in order to model the equivalent supply current generator of a microcontroller.
At first, the equivalent supply current of the core can be extracted by using a transistor-level netlist of the core alone; this can be obtained after either the synthesis step (netlist with active devices only) or the RC extraction step (netlist with detailed parasitics depending on floorplanning and routing). This netlist can then be coupled with digital-only, VITAL VHDL models of the memory blocks by the means of analog-to-digital and digital-to-analog converters specified in VHDL-AMS; in addition, a given executable code can be taken into account by the (E)(E)PROM model, driving data pins according to its contents. This process is depicted on figure 1.
Obviously, this simulation does not include the current consumed by the memory blocks, but allows us to validate the method and the tool.
The second step consists in performing an analysis of the current consumed by the memory blocks themselves, and then trying to find a VHDL-AMS behavioral model of this current. Eventually, these models may be coupled with the core netlist in the same manner, thus giving the whole transient supply current (figure 2). Of course, previous VITAL models may be re-used as far as the digital behavioral part is concerned, since VHDL-AMS is upward-compatible.

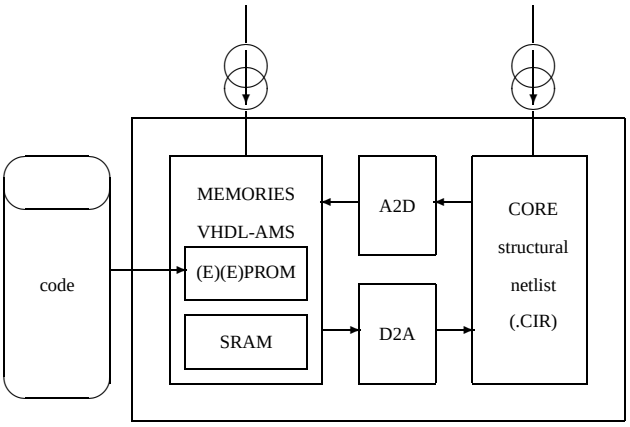

Figure 2. Simulation setup with VHDL-AMS memory models

Memory activity may not represent a substantial part of the current consumed by simple (8 or 16-bit) microcontrollers; however, more complex (32-bit) microprocessors always include cache memories, which are accessed at every clock cycle or so. Consequently, these may play more important a role in the total consumption of those high-performance devices.

The final study (and the most difficult one) might result in a global VHDL-AMS model of the current consumed by the core depending on its activity (instructions, input/output ports). Obviously, this should lead to far shorter simulation times, but at the expense of further theoretical research (figure 3).

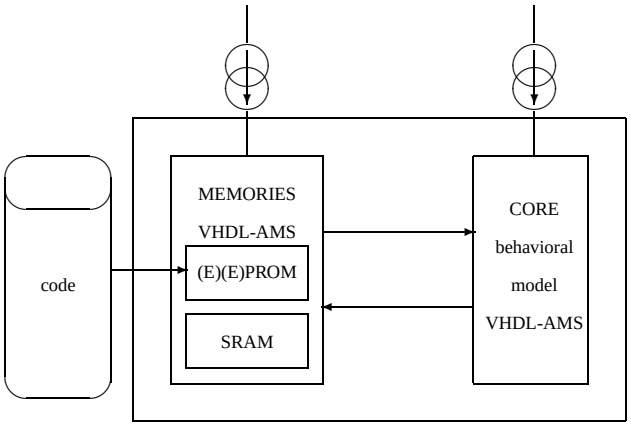

Figure 3. Simulation setup with full VHDL-AMS models

2.2 Example : current generator in VHDL-AMS

In order to better demonstrate the interest of the VHDL-AMS language, a closely related example will be examined. The ICEM model specifies an equivalent piece-wise linear (PWL) current generator for the chip itself; therefore, a generic VHDL-AMS behavioral model of this generator was written, which is introduced below. (I,t) values are obtained from two real vectors, the syntax of which complies with the SPICE/Eldo definition.

```
- PWL current generator
- Ti(0) = 0
LIBRARY DISCIPLINES;
USE DISCIPLINES.ELECTROMAGNETIC_SYSTEM.
  ALL;
ENTITY IbPWLGenerator IS
        GENERIC (Ti  : real_vector;
                 Ii  : real_vector);
        PORT (TERMINAL Vdd_n2, Vss_n2 :
          electrical);
END ENTITY IbPWLGenerator;

ARCHITECTURE a OF IbPWLGenerator IS
        QUANTITY Vb ACROSS Ib THROUGH
          Vss_n2 TO Vdd_n2;
        SIGNAL Istart : real :=
          Ii(Ii'low);
        SIGNAL Tstart : real := 0.0;
        SIGNAL Tend : real :=
          Ti(Ti'low+1);
        SIGNAL deltaI : real :=
          Ii(Ii'low+1)-Ii(Ii'low);
BEGIN
        ASSERT Ti(Ti'low) = 0.0
        REPORT "Error : Ti(Ti'low)
          should be equal to 0.0."
        SEVERITY ERROR;
        ASSERT Ii(Ii'high) = Ii(Ii'low)
        REPORT "Error : first and last
          I values should be the same."
        SEVERITY ERROR;
        PROCESS
        VARIABLE PeriodStart : real
          := 0.0;
        BEGIN
          LOOP
            FOR n IN Ti'low+1 TO Ti'high
              LOOP
                Istart <= Ii(n-1);
                deltaI <= Ii(n)-
                  Ii(n-1);
                Tstart <= PeriodStart
                  + Ti(n-1);
                Tend <= PeriodStart
                  + Ti(n);
                WAIT FOR Ti(n)-Ti(n-1);
              END LOOP;
              PeriodStart:=PeriodStart
                + Ti(Ti'high);
          END LOOP;
        END PROCESS;
        IF domain = quiescent_domain
        USE
          Ib == Istart;
        ELSE
          Ib == Istart + deltaI *
            (now-Tstart)/(Tend-Tstart);
        END USE;
        BREAK ON Tstart;
END ARCHITECTURE a;
```

This example shows the strong relationships between events defined within a process in the "digital" domain, and analog solution points (ASP) in the analog domain; moreover, it demonstrates VHDL-AMS's simplicity (the model is nothing but an implementation of a plain linear interpolation algorithm).

3 Validation and results

3.1 Simulation with VITAL models

The validation of the methodology proposed above was carried out on a 8051 microcontroller core from Atmel (DIVA project), including a 32KB code EEPROM and 2 data SRAMs (1KB and 256B); the core represents about 25000 equivalent NAND gates, both SRAMs represent about 18000, and finally, the EEPROM alone represents more than 150000 ...

The toolset used was ADVance-MS Mach from Mentor Graphics. ADVance-MS is a VHDL-AMS tool based on the Eldo simulation core, which is very accurate but very slow for 100000-transistor netlists; on the contrary, Mach is designed for million-transistor netlists and thus much faster (10 to 12 times), but at the expense of accuracy. ADVance-MS Mach is claimed to combine both cores in a unique environment.

Unfortunately, ADVance-MS's former release (1.5 at the time this article was initially written) did not allow coupling between VHDL-AMS models and the Mach core, because of non-functional A/D and D/A converters; consequently, an Eldo simulation had to be performed instead. It lasted about 80 hours (on a dual-processor Sun Blade 1000 workstation) for only 1 μs simulated time; thus, only the RESET phase could be studied. Much time was wasted with this release before receiving the latest one (1.6) in October 2002.

Finally, we managed to perform an ADVance-MS Mach simulation with two different sets of rise and fall times for the $XTAL1A$ clock signal (200 ps in red and 5 ns in green). The results are displayed in figure 4, in which $I_{R\_VCCCORE}$ represents the current flowing in the power supply line of the core, and I_{R_VCCBUF} the same for the I/O buffers.

Delay times depend on clock rise and fall times, as stated in Hirata's work [2]; consequently, the internal clock generator plays an important role in the correlation between simulation and measurements; for this reason, the ability to simulate the clock generator (with its added complexity) with the rest of the core should significantly increase the validity of the results. Moreover, this simulation covers only the RESET state, and general rules may not be inferred from such a short equivalent time. However, it can be seen that peak values of the currents generated on the core power lines are four times as high as the ones generated on the I/O buffer power lines (25 mA instead of 6 mA).

This method should be validated as soon as possible. For that purpose, the ICEM model may be used; a schematic of the power supply line model is presented in figure 5.

Previous research described in [3] proved the validity of such a model in "low" frequency bands (below 2 GHz). Since lumped elements were perfectly identified for our microcontroller, the predicted external current can be computed by simulating the core (represented by I_b) with these additional elements, and thus these results can be correlated with external current measurements.

However, the effects of internal parasitic elements (resulting from the place-and-route step) are not included at this time, and a method for DSPF (Detailed Standard Parasitic File) import in ADVance-MS Mach is currently being investigated (compatibility issues between formats used by CAD tools).

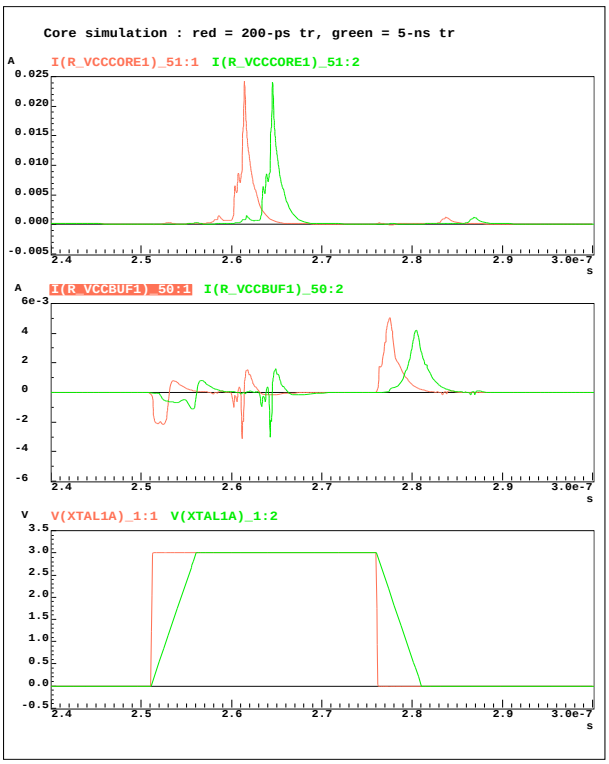

Figure 4. Simulation of the core in the RESET phase

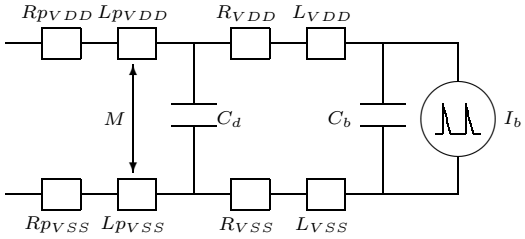

Figure 5. The ICEM model for supply lines

3.2 Simulation with transistor-level memories

In order to validate the mixed-signal simulation concept, the simpler memory block was chosen, i.e. the SRAM.
For that purpose, an interesting approach of power consumption issues is proposed by Saillé [4]. Even if only energy (and not transient signals) is addressed, the article clearly demonstrates the role of each building block (including address decoder, cells, sense amplifiers). The corresponding current waveforms should thus depend on successive addresses as well as on memory contents.
In order to state this, the transistor-level netlist was driven by a VHDL digital testbench browsing every decoded address, as shown in figure 6.

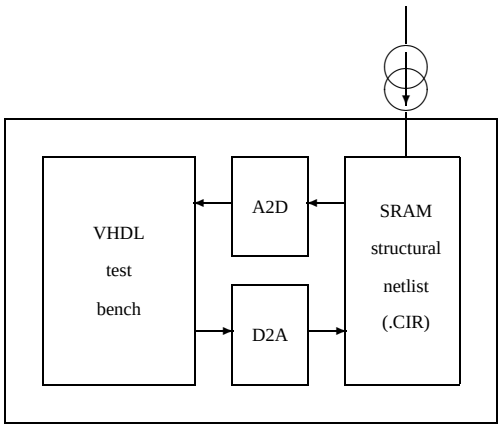

Figure 6. Setup for transistor-level SRAM simulation

For that purpose, two different sets of rise and fall times for incoming signals (200 ps in red and 5 ns in green) were chosen, in order to better demonstrate the influence of this parameter on the amplitude and timings of the generated current. The results are displayed in figure 7 for two given accesses (address $1FF after $1FE, address $200 after $1FF) differing by the number of switching bits (1 then 10) between successive addresses. ME represents the active-high memory enable (chip select) signal.
On these simulations, two different current pulses for each access can be clearly seen :

- a first pulse which may correspond to address decoding; as can be seen, its amplitude widely depends on the Hamming distance between successive addresses, which is an expected result; moreover, both its amplitude and rise time depend on the rise time of the address signals.

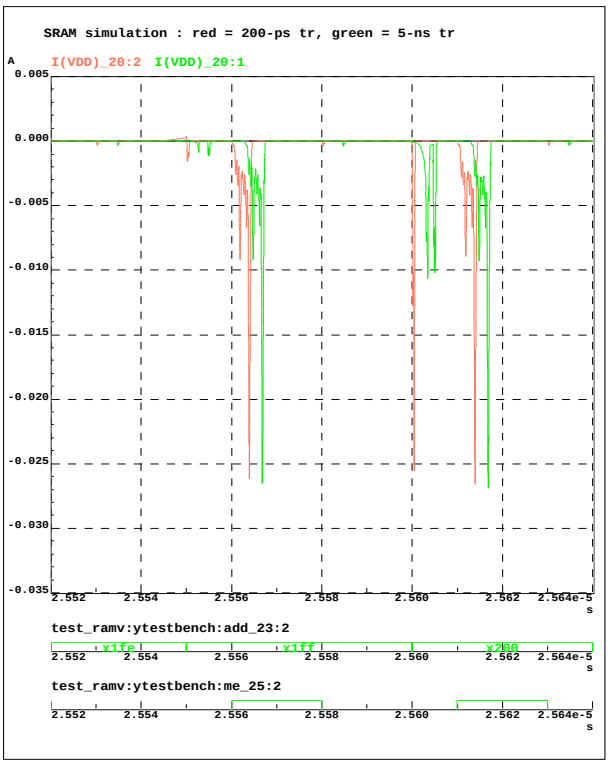

Figure 7. Simulation of SRAM accesses

- a second pulse which should correspond to the memory cells themselves; its shape depends neither on the address itself, and most remarkably, nor on rise/fall times of the incoming control signals; however, the delay between the transition of the MemoryEnable signal and the pulse itself depends on rise/fall times of this control signal, which is a general result already quoted in [6].

Moreover, peak values are nearly the same order of magnitude as the ones of the current generated by the core (i.e. 25 mA, see previous section), which shows that memory accesses should not be omitted in a global simulation of a microcontroller.
The influence of memory contents still remains to be studied. However, a few conclusions may be inferred from these results :

- the resulting SRAM VHDL-AMS model should be parameterized by rise and fall times of the incoming signals, or conversely include a test bench measuring these characteristics

- the addressing scheme plays a very important role in the the dynamic current frequency spectrum, and will be of course included in the model

As a general result, adjusting rise and fall times (by design or place-and-route) of address and control signals might help to desynchronize current pulses generated by the core and its associated memories, thus reducing the amplitude of some frequency components in the resulting spectrum and consequently improving EMC performance.

4 Conclusion

This paper suggests a methodology for extracting power supply currents in a microcontroller by using mixed-mode simulation. For this purpose, the use of the VHDL-AMS language, permitting high-level memory modeling, is proposed.
This methodology is about to be validated by measurements on the core, but this operation was delayed because of improper operation of the CAD tools used at that time. However, the results obtained on SRAM blocks show the feasibility of a dynamic current consumption macromodel.
Our immediate objectives, which should be met at the end of year 2002, are :

- the inclusion of RC parasitics in core simulations
- the validation of core simulations by the means of measurements coupled with the ICEM model
- the development of the behavioral model of the instantaneous current consumed by memory blocks

These results may provide guidelines for EMC-oriented microcontroller design.

References

[1] E. Christen, K. Bakalar, A.M. Dewey, and E. Moser. Analog and mixed-signal modeling using the VHDL-AMS language. In *36th Design Automation Conference*, June 1999.

[2] A. Hirata, H. Onodera, and K. Tamaru. Estimation of propagation delay considering short-circuit current for static CMOS gates. *IEEE Transactions on Circuits and Systems*, CAS-45(11):1194–1198, November 1998.

[3] J.L. Levant, M. Ramdani, and R. Perdriau. ICEM modeling of microcontroller current activity. In *Submitted to EMC'Compo 2002*, September 2002.

[4] D. Saillé, D. Chillet, and O. Sentieys. Modélisation de la consommation pour les mémoires SRAM. In *FTFC'01*, pages 27–33, 2001.

[5] IEC EMC Task Force. IEC62014-3 : Integrated circuit electromagnetic model. Draft technical report, IEC, November 2001. http://intrage.insa-tlse.fr/∼etienne/Emc/icemcdv.pdf.

[6] S.J.E. Wilton and N.P. Jouppi. An enhanced access and cycle time model for on-chip caches. Digital WRL research report 93/5, Digital Equipment Corporation, July 1994.