# VAMP: A VHDL Based Concept for Accurate Modeling and Post Layout Timing Simulation of Electronic Systems

Bernhard Wunder, Gunther Lehmann, Klaus D. Müller-Glaser
Institut für Technik der Informationsverarbeitung, Universität Karlsruhe, D-76128 Karlsruhe, Germany

## Abstract

*This paper presents a new concept for accurate modeling and timing simulation of electronic systems integrated in a typical VHDL design environment, taking into account the requirements of deep submicron technology.*

*Contrary to conventional concepts, autonomous models for gates and interconnections are used. A piece-wise-linear signal representation allows to model waveform dependent effects. Furthermore, the gate models catch pattern dependencies, the models of interconnections take into account post layout information with ramified structure, different layers and even contact holes.*

## 1 Introduction

During the design of electronic systems, each design phase is accompanied by a verification phase. As an increasing portion of errors on integrated circuits can be traced back to timing problems, the timing behaviour plays a very important role in the IC verification process.

Technological progress leads to smaller feature size resulting in decreasing propagation delay and higher output signal slew rate of single logic gates. Higher clock frequencies can be realized. Besides that, the shape of a signal transition and the input pattern (often summarized as *input situation*) have a major influence on the dynamic behaviour (propagation delay, output rise time, etc.) of logic gates. These *analog aspects* in the field of digital circuits will play a more important role in the future [1].

On the other hand the chip size increases, so that global interconnections cause a growing contribution to the propagation delay of the corresponding paths. Since propagation delay and other effects (reflexions, coupling, etc.) caused by long interconnections have been playing an important role between PCBs for a long time, they were treated as second order effects on PCBs and even could be neglected for the small on-chip dimensions. But the mentioned tendencies have a major impact on the dynamic behaviour and on the efficiency of todays electronic systems, even on chip (see also: scaling theory [2]). The performance is limited by the effects of interconnections in many cases. This disproportion between the delay of gates and global interconnections is shown in figure 1 [3].

As the mentioned pattern and waveform dependencies can hardly be covered by static timing analysis, we focus on (dynamic) timing simulation in this work.

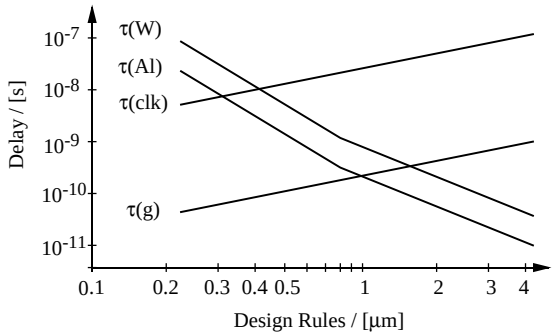


**Figure 1:** Propagation delay of single gates ($\tau(g)$) and clock drivers ($\tau(clk)$) in contrast to global interconnections ($\tau(Al, W), l = 5mm$) as a function of the design rules

There are four fundamental types of models and algorithms for timing simulation, demonstrating the existing trade-off between simulation accuracy and simulation expense:

**Boolean algorithms** for digital gates use two valued (boolean) signals or multi-valued logic systems. The function and the timing behaviour are calculated separately. The function (output signal value) is evaluated by boolean equations, whereas the propagation delay of the output signal is calculated using simple formulas or lookup-tables and interpolation methods. The needed parameters for it are calculated by extensive presimulation runs (characterization) [4].

**Switch-level algorithms** calculate function and timing by evaluating the node voltages represented by discrete levels using charge or discharge equations. These algorithms operate directly on the transistor circuit structure. Each transistor is represented by a bidirectional resistive switch, nodes are represented by capacitances [5]. Sometimes interconnections are taken into account by equivalent lumped RC elements [6], [7]. Although these models can take into consideration feedback loops and pass transistors, there are still deviations from the results of circuit simulations that cannot be tolerated for the design of submicron systems.

**Analytical differential equation solvers** use macromodeling techniques and nonlinear transistor equations benefitting from the unidirectional signal flow in MOS circuits [8], [9]. Mostly, the complex nonlinear differential equations resulting for gate behaviour are reduced by mapping all gates into equivalent primitive blocks (e.g. inverters) and/or using linearized equations. Doing this, pattern dependencies are neglected.

Timing simulation using the most simple models for boolean algorithms can be performed with so-called boolean gate-

33rd Design Automation Conference ®

DAC 96 - 06/96 Las Vegas, NV, USA 

level simulators. This method is state of the art in digital design, often enhanced with backannotation methods using the established Standard Delay Format (SDF) [10]. On the other hand, the extensive modeling capabilities of hardware description languages like VHDL [11] - [13] and corresponding simulators make it possible to perform timing simulation not only by using boolean but even by using switch-level models and simple analytical models. Beneath the implementation of the discussed algorithms with conventional digital simulators, some of the proposed more advanced models need specialized tools for timing simulation [14], [15].

As conventional signal types for boolean algorithms such as the VHDL types *bit* or *std_logic* only represent the new signal value and do not contain information about the signal shape, waveform dependent effects cannot be reflected. Some algorithms therefore use information about the signal rise or fall time, representing the signal transition as a ramp [1], more advanced models use a two-segment approximation [16]. But even with this type, overshoots caused by reflexions, strongly nonlinear waveforms, or non completely switching transitions cannot be modeled. Signal representations by mathematical functions or piece-wise-linear approximations have to be applied [17].

Taking a closer look on the discussed models and algorithms proves that in nearly every case simplifications (heuristic formulas, linearization, inverter reduction) are made to be capable of the nonlinear transistor behaviour with all the parasitic effects.

None of the models mentioned above offers a solution for an accurate integration of the interconnection behaviour into timing simulation which also includes effects of inductive parasitics. At the moment, only one modeling technique masters this problem with the necessary accuracy:

**Numerical differential equation solvers** represent the logic gates by their equivalent transistor circuits and interconnections with lumped elements (resistors, inductors, capacitors) or special lossy transmission line models. This results in Kirchhoff networks that have to be solved with numerical algorithms [18]. This is done by circuit simulators like SPICE [19].

Using a circuit simulator for timing simulation leads to the required accuracy, but also to very time consuming simulation runs. This method can only be used for smaller circuits or subcircuits, but is unacceptable for complete systems.

Since all discussed models and algorithms have deficiencies regarding the accurate representation of submicron effects or simulation performance, new methods, models and tools must be developed to meet the advanced requirements for an efficient design flow that supports the design of complex systems with high performance. The proposed VAMP concept will close that gap in the trade-off range of accurancy and simulation expense. Timing simulation using this concept will increase simulation speed by a factor of one hundred. Simultaneously, the simulation leads to deviations less than 5% compared with circuit simulation results.

## 2 Concept and implementation basis

Figure 2 shows the VAMP concept (VAMP = **V**HDL based **a**ccurate **m**odeling for **p**ost layout timing simulation). This new concept processes gate-level netlists and layout data.

VAMP is an event-driven approach based on a new signal representation and separated, autonomous models of the various logic gates and interconnections. This separation stands

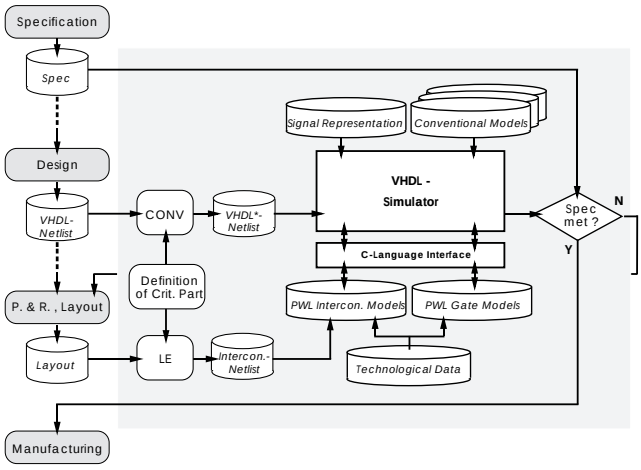


**Figure 2:** The VAMP concept

in contrast to other event-driven modeling concepts, where interconnect delay is integrated into the gate models either at the output stage [4] or at the input stage [10].

**Signal representation:** To catch the dependencies on the shape of signal transitions, a new signal representation is used which holds more information of the signal shape than traditional multi-valued logic systems. This signal representation is defined as a VHDL type in an appropriate package (section 3) [20].

**Models of interconnections:** Interconnections will no longer be treated as a single, dimensionless node like by conventional simulators, but as an independent component with several input and output ports and different behaviour at each output port. Resistive, capacitive and inductive parasitics are taken into account (section 4) [20].

**Models of logic gates:** Appropriate models of the logic gates, capable of handling the new signal representation, also take into consideration the waveform and pattern dependencies as well as the parasitic load caused by driven interconnections and gates (section 5).

**Layout extraction (LE):** To calculate the necessary parameters for the models of interconnections, information about the structure (length and dimension of each section and branch) and the used materials (interconnection layer) is needed. This information is extracted from layout and stored in a netlist format for each interconnection by the extraction module.

**Netlist conversion (CONV):** Usually, interconnections in conventional VHDL netlists (e.g. resulting from logic synthesis) are treated as a single, dimensionless node each connected gate is mapped to. Using the accurate models of interconnections with different behaviour from each input port to each output port means to split these nodes and integrate individual instances of the interconnection models into the netlist (figure 3). Furthermore, conventional gate instances have to be replaced by new, accurate PWL models. All these steps are done by the conversion module.

It is possible to investigate the complete system using the advanced models and signal representation, but one can also investigate only critical paths or parts of the system. Other

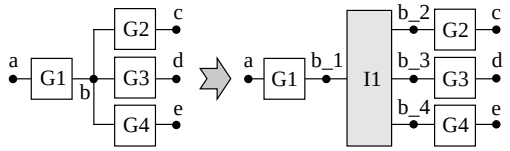


**Figure 3:** Netlist conversion integrating an interconnection model instance for node $b$

parts are modeled in a conventional way using signal types like *std_logic*. Information about the system part to be modeled with the new methodology is necessary for netlist conversion. Proper conversion modules to connect the different signal types (PWL $\leftrightarrow$ *std_logic*) have to be used in this case.

As an additional feature, the power dissipation can be calculated as a function of time (section 6).

The simulation results can finally be compared with the specification in order to decide, whether the layout meets the given timing and power requirements.

This concept is implemented using the standardized hardware description language VHDL. Therefore, we are not restricted to a single simulator and/or vendor. Furthermore, VHDL offers a wide range of modeling capabilities including new signal types, attributes, and functions [11] - [13]. Aspects of accuracy[1] and simulation performance[2] caused us to implement some modules in the programming language $C$ rather than in VHDL. The $C$ modules are linked to the VHDL simulator using a corresponding, but tool-specific $C$ language interface. Once the $C$ functions are integrated into the interface and linked to the simulator, the models can be used in structural VHDL models like any other VHDL component. At the moment we provide interface routines for *vss* (SYNOPSYS) and *Leapfrog* (CADENCE). As most of the VHDL simulators offer a $C$ language interface, the concept is not restricted to these specific simulators.

## 3 Signal representation

As discussed, the signal shape has a major impact on the timing behaviour of gates and interconnections. Modeling these effects means to calculate and propagate the waveform of each signal, i.e. the shape of each individual event.

As piece-wise-linear (PWL) approaches are used in the field of circuit simulation [22], we use it for our event driven approach, too. The VAMP concept describes each signal transition as a sum of ramps, resulting in a PWL shape. The PWL representation is realized as a VHDL type (`pwl_signal`) in an appropriate package, shown below. The voltage component `ramp_height` of each ramp is an integral multiplier of $1mV$, the time component `ramp_time` is an integral multiplier of the base unit $ps$. This memory saving implementation provides the necessary accuracy. Boolean signals or even analog signal waveforms (which are also piece-wise-linear functions with very small timesteps) can be interpreted as special cases of the proposed PWL type. Therefore, simple conversion modules are provided to use the new signal types together with conventional VHDL types *bit* or *std_logic* in a mixed-mode simulation.

[1] The current version ot the IEEE mathematical package does not provide double precision types.

[2] Investigations on comparable $C$ and VHDL implementations using the mathematical packages `math.h` respectively `MATH_REAL` indicated broad performance advantages of the $C$ realization.

```
PACKAGE pwl IS

   SUBTYPE ramp_range IS integer RANGE 0 TO 20;

   TYPE ramp IS RECORD
      ramp_height : integer;      -- in mV
      ramp_time   : natural;      -- in ps
   END RECORD;
   TYPE ramp_array is ARRAY (ramp_range) OF ramp;

   TYPE pwl_signal IS RECORD
      ramp_number : ramp_range;
      ramp_data   : ramp_array;
   END RECORD;
   TYPE pwl_signal_vector is ARRAY
            (natural RANGE <>) OF pwl_signal;

END pwl;
```

The model equations for the output voltage $V$ are sampled for discrete values of time $t_n$ and the resulting sampled functions $V_n$ have to be transformed back into the PWL form by approximation using least squares method. This process is controlled by the user defined parameters maximum number of ramps and maximum voltage error.

## 4 Models of interconnections

Since the behaviour of interconnections is nearly linear, frequency domain methods can be used to compute it. The transfer function of every path from the input port to every output port of an interconnection net is computed using fourpole theory.

Each section of the interconnection net is treated as a homogenous twin wire ($TW$) with distributed parasitics. The exact solution of this problem can be written in the form of the following cascade matrix:

$$\mathbf{A}^{(TW)} = \begin{bmatrix} \cosh(\gamma \cdot l) & Z \cdot \sinh(\gamma \cdot l) \\ \frac{1}{Z} \cdot \sinh(\gamma \cdot l) & \cosh(\gamma \cdot l) \end{bmatrix} \qquad (1)$$

Because of the very good insulation properties of the dielectric at the relevant frequency range, the conductance $G$ can be neglected. $Z$ and $\gamma$ can then be described as:

$$Z = \sqrt{\frac{R' + sL'}{sC'}} \quad , \quad \gamma = \sqrt{(R' + sL')(sC')} \qquad (2)$$

Parameters $R'$, $L'$ and $C'$ represent the distributed parasitics for resistance, inductance and capacitance [23], [24], $s$ stands for the complex frequency $j\omega$. Furthermore, the implemented algorithm uses an approximation for the hyperbolic functions sinh and cosh.

Cascade matrices of contact holes ($\mathbf{A}^{(CH)}$) and lumped elements ($\mathbf{A}^{(R)}$, $\mathbf{A}^{(C)}$ and $\mathbf{A}^{(L)}$) as representatives for resistance, gate input capacitance or bond wires can be built in the same manner.

Branches ($B$) of the regarded input-output path terminated by the capacitance $C_{l,B}$ have to be modeled by their input admittance $Y_B$, which is integrated into the direct path as a shunt conductance with the corresponding cascade matrix $\mathbf{A}^{(Y_B)}$:

$$\mathbf{A}^{(Y_B)} = \begin{bmatrix} 1 & 0 \\ Y_B & 1 \end{bmatrix} \qquad (3)$$

$$\text{with:} \quad Y_B = \frac{A_{22}^{(B)} + A_{21}^{(B)}/(s \cdot C_{l,B})}{A_{12}^{(B)} + A_{11}^{(B)}/(s \cdot C_{l,B})}$$

In this way, the interconnection path from input $i$ to output $j$ is reduced to a simple series connected line of twin wires, lumped elements or shunt conductances. For this structure, the final cascade matrix can be built by matrix multiplication:

$$\mathbf{A}_{(i,j)} = \prod_n \mathbf{A}^{(n)} \tag{4}$$

The corresponding transfer function $H_{(i,j)}(s)$ can be directly derived from $\mathbf{A}_{(i,j)}$ taking into account the terminating capacitance $C_{l,j}$:

$$H_{(i,j)}(s) = \frac{1}{A11_{(i,j)}(s) + sC_{l,j} \cdot A12_{(i,j)}(s)} \tag{5}$$

As the input signal $v_i(t)$ is represented as a sum of ramps $v_{i,k}(t)$, each of these ramps contributes the portion $v_{j,k}(t)$ to the output signal $v_j(t)$. These portions $v_{j,k}(t)$ are calculated using the rules of Laplace transformation:

$$\begin{array}{ccccc} v_{j,k}(t) & & & & v_{i,k}(t) \\ \circ & & & & \circ \\ | & & & & | \\ \bullet & & & & \bullet \\ V_{j,k}(s) & = & H_{(i,j)}(s) & \cdot & V_{i,k}(s) \end{array} \tag{6}$$

The continuous output signal $v_j(t)$ equals the sum of all portions $v_{j,k}(t)$. It is finally approximated by a piece-wise-linear form again.

## 5 Models of logic gates

As discussed, waveform dependent effects play an important role in the timing behaviour of submicron devices. Therefore, we use a continuous signal representation in piece-wise-linear form. In the MOS gate primitive by Y.-H. Shih and S. M. Kang [15], [17] we found an adequate model to represent the desired behaviour. The circuit primitive is shown in figure 4. The point of interest is the output node $V$ with its

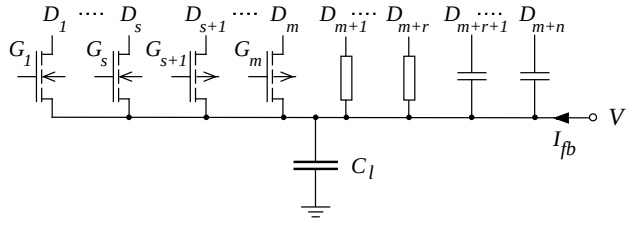


**Figure 4:** Circuit primitive by Shih/Kang

capacitance $C_l$, which is charged by the various substitute components in this model. Transistors $1 \ldots m$ represent parallel connected NMOS or PMOS transistors whereas series connected transistors are modeled as one equivalent transistor. Each transistor generates a contribution to the charging current of $C_l$, which depends on the piece-wise-linear gate and drain voltages $G_1 \ldots G_m$ and $D_1 \ldots D_m$. The transistor currents are computed using the level 0 equations [19]. Additional parasitics $m+1 \ldots m+n$ and even an external feedback current $I_{fb}$ also contribute to the (de)charging current of $C_l$. The resulting differential equation for node $V$ has the form of a so-called quadratic Riccati differential equation:

$$\frac{dV}{dt} = K \cdot V^2 + P(t) \cdot V + Q(t) \quad \text{with:} \quad V(t_0) = V_0 \tag{7}$$

Assuming linear gate and drain voltages and a special form of $I_{fb}$, the time variant parameters have a linear ($P(t) = p_1 \cdot t + p_0$) or quadratic form ($Q(t) = q_2 \cdot t^2 + q_1 \cdot t + q_0$). In this case, equation (7) can be solved analytically for suitable time intervals. These intervals (linear form of all input signals $D_i, G_i$ and constant transistor state) have to be ascertained depending on the input waveforms and the transistor states.

Although this model is one of the most accurate used for timing simulation, we found some aspects of inaccuracy that have to be avoided in an implementation for submicron timing simulation.

**Pattern dependency:** The circuit primitive, as proposed in [17], can only reflect pattern dependencies caused by parallel connected resistors (e.g. PMOS-transistors in CMOS NAND-cells). In spite of the minor effects caused by series connected transistors, we enhanced the model to be capable of them using a pattern and slope dependent calculation of the load capacitance.

**Transistor equations:** The Shih/Kang circuit primitive is only capable of the simplified level 0 transistor equations. Therefore, the output conductance $\lambda$ (level 1 transistor equations) has been integrated by an operating point dependent linearization to maintain the necessary form of the coefficients in equation (7).

**Influence of non capacitive loads:** The node capacitance $C_l$ in figure 4 includes the output capacitance as well as the capacitance of driven gates and interconnections. Resistive properties of interconnections are neglected. By calculating a proper feedback current $I_{fb}$, we are now able to consider the non pure capacitive impedance of the load.

The analytical solution of equation (7) is evaluated for discrete timepoints $t_n$. The resulting sampled signal $V_n$ is then approximated by a piece-wise-linear form and propagated to the subsequent interconnection models.

## 6 Calculation of power dissipation

Neglecting the leakage current, CMOS circuits do not drive any transverse current in the case of constant input voltages of valid logic levels. Power dissipation is only produced by transverse current and charging current during switching processes. Our gate models are activated for each input event and calculate the output waveform just for the time intervals the relevant supply current contributions are non zero. As all currents of transistors or parasitic elements are computed during the evaluation of an event anyway, we can use this information to calculate the current taken from the power supply. We just have to sum up all the relevant contributions, all currents of PMOS transistors. As we want to keep the computational effort small, we just integrate the supplying current and form a mean value for the time, the output node switches. Finally it has to be multiplied with the supply voltage:

$$P_{event} = \frac{V_{dd}}{t_{rise}} \int_{t_{event}}^{t_{event}+t_{rise}} \sum_r I_{PMOS,r}(t) \quad dt \tag{8}$$

This value is propagated to a summing module. The resulting power dissipation of blocks or the whole circuit as a function of time can be displayed like any other signal voltage. The results can be taken to find hot spots inside the design or to dimension the power supply. An example is presented in the next section.

## 7 Results

Results of the model of interconnections, even of complex interconnection structures with inductive parasitics, can be found in [20]. In this work we focus on two simple application circuits that prove the interoperability of both model types (interconnections and gates). They also demonstrate the usability of the VAMP concept for timing simulation.

The first example is a simple ring oscillator. We connected six inverters plus one NAND2 primitive to control the oscillation. With different modeling techniques we obtained the oscillation frequency $f$ and the CPU time for a $20ns$ simulation performed on a SPARC 10/30 workstation. For gate-level simulation we used SYNOPSYS' *vhdlsim* and a conventional *std_logic* gate library, the circuit simulation was done using METASOFT's *HSPICE*.

To investigate the layout dependencies, we simulated the oscillator with different lengths of interconnection, from no interconnection up to $3mm$ metal interconnection lines with an interconnection length of $21mm$ in total. The results of both extremes are presented in table 1.

| Simulation level | gate | VAMP | circuit |
|---|---|---|---|
| $f/[MHz]$ (no int.) | 254.1 | 360.4 | 377.2 |
| $f/[MHz]$ ($3mm$ int.) | 78.4 | 113.7 | 119.1 |
| CPU time/$[sec]$ (no int.) | $< 0.2$ | $< 0.8$ | 17.5 |
| CPU time/$[sec]$ ($3mm$ int.) | $< 0.2$ | 2.4 | 75.1 |

**Table 1:** Simulation results of the inverter ring

Notice that the frequency obtained by gate-level simulation is much lower than by circuit simulation or our models. This results from the very pessimistic (worst case) timing parameters used in a gate-level simulation. Even though the VAMP simulation needs a significant smaller CPU time than an equivalent circuit simulation, the deviation is small (less than 5%).

As a more complicated, but still comprehensible example, we chose a one bit full adder, built by circuit primitives. The structure is shown in figure 5. This circuit is stimulated by certain signal waveforms at the inputs $a$, $b$ and $c\_in$ (carry in), shown in the upper part of figure 6. Again, two simulation runs were performed. The lower part of figure 6 shows the results of the simulation without interconnections at node $s$ (sum bit) and $c\_out$ (carry bit), where the VAMP results (dashed lines) are plotted together with the results of an equivalent circuit simulation (solid lines). A difference can hardly be seen. Please consider the sum bit in the interval between 40 and $65ns$. The different paths from input signals $a$ and $c\_in$ to the output $s$ lead to a small positive spike at $t \approx 47ns$ and a small negative spike at $t \approx 57ns$.

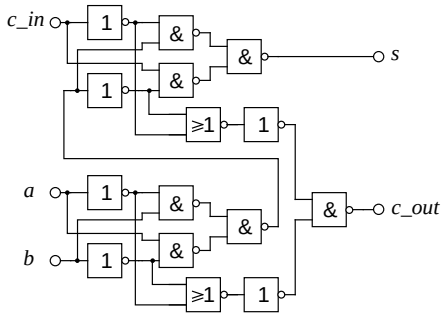


**Figure 5:** Structure of a one bit full adder

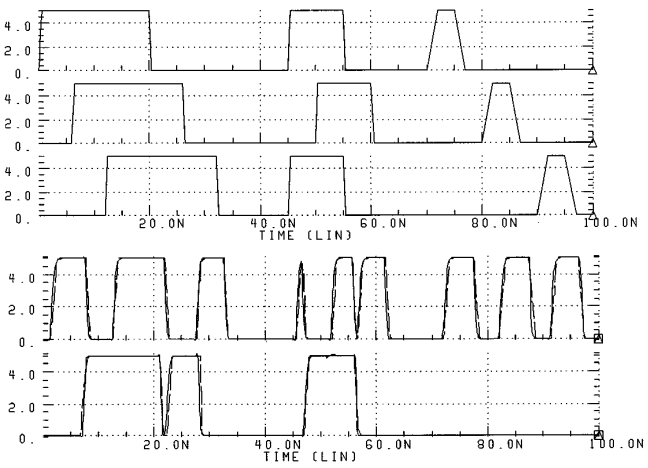


**Figure 6:** Stimuli for the full adder ($a$, $b$ and $c\_in$) and simulation results without interconnections (sum bit ($s$) and carry bit ($c\_out$))

If we integrate interconnection models into simulation, we get the results shown in figure 7. The sum bit shows a totally different behaviour in this simulation. Additional delays in the logic paths broaden the spikes mentioned above, so that we cannot speak of spikes any longer. This example proves that even in small systems additional interconnection delay does not only affect the timing behavior, but even the logic behavior when integrated in critical paths.

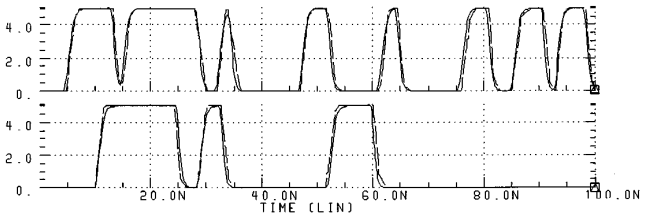


**Figure 7:** Results of the full adder with $3mm$ interconnections (sum bit ($s$) and carry bit ($c\_out$))

Figure 8 finally demonstrates the calculation of the power dissipation. For the time interval $44ns \ldots 65ns$ the power dissipation of the VAMP models is plotted by a dashed line, the solid line is the result of an equivalent circuit simulation.

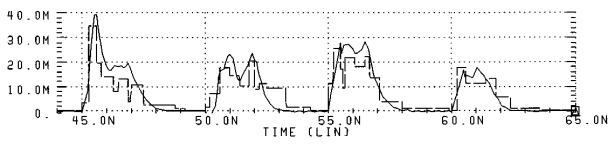


**Figure 8:** Power dissipation of the full adder (zoom)

The number of ramps per signal affects the simulation expense in a nearly linear manner. We found an optimum of

accuracy and simulation expense in the case of well damped interconnections (on chip parameters) at a ramp number of 3...5. Off chip interconnections with overshoots require a higher number of ramps per transition. The simulation expense of the VAMP models grows linear with the number of gates, while circuit simulation expense typically grows by $n^{1.2}$ up to $n^2$ [19]. Further advantages result from the latency in less active circuits. Circuit simulators compute voltages anyway whereas our event driven approach only activates the models during input transitions.

## 8 Conclusion

We demonstrated that an accurate modeling of interconnections as well as pattern and waveform dependencies is essential for simulating the timing behaviour of submicron digital circuits. The proposed VAMP concept has demonstrated its usability for this purpose with remarkable improvements regarding accuracy compared to conventional methods and simulation expense compared to circuit simulation ($\approx 1\%$). In our concept, the modeled behaviour of gates depends on the input signal shapes, the input pattern and the layout dependent load. In addition, the power dissipation is calculated, too. Models of interconnections also take into account the distributed parasitics and the interconnection structure, even with branches. By using conventional VHDL simulation tools and some additional layout extraction and conversion modules, the concept has been integrated into a typical VHDL based design flow.

The concept provides a variety of operation modes:

- Standalone use of the models for the logic gates (e.g. inside smaller functional blocks with very short interconnections) as well as standalone use of the models for interconnections (e.g. to calculate the clock skew).
- Use of the models without layout data, based on estimated interconnection lengths as result of the floorplan as well as post layout usage with exact data.
- Modeling complete systems as well as only critical system parts (*mixed-mode simulation* with conventionally modeled system parts or even analog components).

Although the simulation results are promising, VAMP is no accelerated circuit simulation that can be used for verification on transistor level. VAMP rather represents the missing type of timing models and simulation in the trade-off range between simulation accuracy and expense that suffices the needs of cell based submicron circuit design.

## 9 Future Work

We implemented models of interconnections and basic gates (Inverter, NAND, NOR, latches and flip flops). Future work will therefore concentrate on the realization of parametrizable models for complete ASIC libraries.

The final goal of the VAMP concept is a *push button solution* for a *mixed-mode simulation* of critical paths/parts together with conventionally modeled components. Automatically generated information about critical paths/parts have to be taken as a basis for layout extraction and netlist conversion.

If integrated in floorplanning and layout tools, the simulation concept could be very helpful to reduce the radius of the timing verification loop. Therefore, we will investigate on the possibilities of such an integration into existing tools.

## References


[1] G. Lehmann, P. Nagel, K. D. Müller-Glaser: *An Approach to Detailed Modeling of Digital CMOS Gates for Logic Simulation using VHDL*, International Journal of Electronics and Communication, Vol. 49, No. 2, 1995

[2] H. B. Bakoglu: *Circuits, Interconnections and Packaging for VLSI Circuits*, Addison-Wesley, Reading, MA, 1990

[3] Marc Rocchi: *High Speed Digital IC Technologies*, Artech House, Norwood, 1990

[4] E. Hörbst: *Logic Design and Simulation* Advances in CAD for VLSI, Vol. 2, North Holland, Amsterdam, 1986

[5] J. P. Hayes: *Switch Level Modeling*, IEEE Design and Test, August 1987

[6] J. Rubinstein, P. Penfield, M. A. Horowitz: *Signal Delays in RC Tree Networks*, IEEE Transactions on Computer Aided Design, Vol. 2, No. 3, 1983

[7] J. L. Wyatt, Q. Yu: *Signal Delay in RC Meshes*, Proceedings of the IEEE International Conference on Computer Aided Design, Santa Clara, CA, 1983

[8] T. Sakurai, A. R. Newton: *Alpha-power Law MOSFET Model and its Application to CMOS Inverter Delay and Other Formulas*, IEEE Journal on Solid State Circuits, SC-25, pp. 584-594, 1990

[9] A. Nabavi-Lishi, N. C. Rumin: *Inverter Models of CMOS Gates for Supply Current and Delay Evaluation*, IEEE Transactions on Computer Aided Design, Vol.13 , No. 10, 1994

[10] IEEE 1076.4 Working Group: *VITAL V3.0 Specification*, FTP-address: ftp.vhdl.org, 1995

[11] The Institute of Electrical and Electronics Engineers: *IEEE Standard VHDL Language Reference Manual (ANSI/IEEE 1076-1993)*, New York, NY, 1994

[12] D. L. Perry: *VHDL*, McGraw Hill, New York, 1991

[13] G. Lehmann, B. Wunder, M. Selz: *Schaltungsdesign mit VHDL*, Franzis, Poing, 1994

[14] R. E. Bryant: *A Survey of Switch Level Algorithms*, IEEE Design and Test, August 1987

[15] Y.-H. Shih, Y. Leblebici, S.-M. Kang: *ILLIADS: A Fast Timing and Reliability Simulator for Digital MOS Circuits*, IEEE Transactions on Computer Aided Design, Vol. 12 , Nr. 9, 1993

[16] A.-C. Deng, Y.-C. Shiau: *Generic Linear RC Delay Modeling for Digital CMOS Circuits*, IEEE Transactions on Computer Aided Design, Vol. 9, No. 14, 1990

[17] Y.-H. Shih, S.-M. Kang: *Analytic Transient Solution of General MOS Circuit Primitives*, IEEE Transactions on Computer Aided Design, Vol. 11 , Nr. 6, 1992

[18] A. E. Ruehli: *Circuit Analysis, Simulation and Design*, Advances in CAD for VLSI, Vol. 3, North Holland, Amsterdam, 1986

[19] L. W. Nagel: *SPICE2: A Computer Program to Simulate Semiconductor Circuits*, Electronic Research Laboratory Report #ERL-M520, University of California, Berkeley, CA, 1975

[20] B. Wunder, G. Lehmann, K. D. Müller-Glaser: *Post layout timing simulation with accurate modeling of interconnections using a VHDL-simulator*, VHDL International Users Forum Fall 1995, Newton, MA, 1995

[21] Synopsys, Inc.: *C-Language Interface Reference, Version 3.3*, Mountain View, CA, 1995

[22] H. W. Buurman: *From Circuit to Signal: Development of a Piecewise Linear Simulator*, PhD-thesis, University of Eindhoven, 1993

[23] E. T. Lewis: *An Analysis of Interconnect Line Capacitance and Coupling for VLSI Circuits*, IEEE Journal of Solid State Circuits, Vol. 27 (8), 1984

[24] Yuan, Lin, Chiang: *Properties of Interconnection on Silicon, Sapphire and Semi-Insulating Gallium Arsenide Substrates*, IEEE Transactions on Electronic Devices, Vol. 29 (4), 1982