

# Post layout timing simulation with accurate modelling of interconnections using a VHDL-simulator

Bernhard Wunder, Gunther Lehmann, Klaus D. Müller-Glaser

Institute for Information Processing Techniques (ITIV)

University of Karlsruhe

D-76128 Karlsruhe, GERMANY

## Abstract

*This paper describes a new modelling concept for the timing simulation of digital electronic systems with special respect to the influence of interconnections.*

*A new signal representation allows the modelling of waveform dependencies, accurate models for interconnections will catch the increasing influence of that passive part on the overall performance of electronic systems, using information extracted from layout data.*

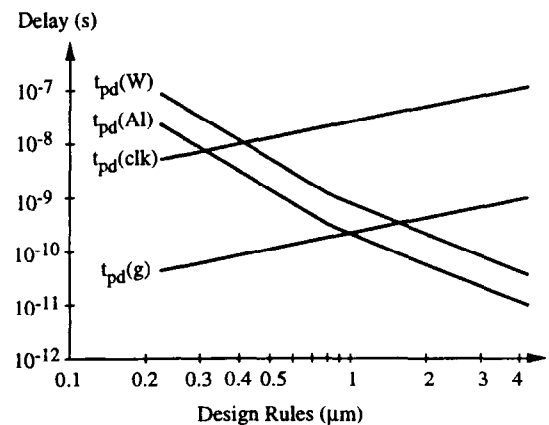
*The models are implemented in VHDL, some modules in the programming language C, which can be linked to a conventional VHDL-simulator. VHDL-netlists, e.g. generated by synthesis tools, are automatically modified to use the new signal representation and models.*

## 1 Introduction

Progress in the field of semiconductor technology leads to a permanently decreasing feature size and design rules. This evolution is accompanied by also decreasing propagation delays of single logic cells. Simultaneously, the size of the chips, that can be manufactured with a tolerable yield, increases. All in all, the complexity, i.e. the number of cells or transistors in a single circuit, grows in a considerable amount.

On the other hand, global interconnections be-

tween functional blocks get longer and cause an extra high contribution to the propagation delay on the corresponding logic paths. The resulting disproportion between the propagation delays caused by the cells and by global interconnections is shown in figure 1, [1].



**Figure 1:** Propagation delays of single logic cells ( $t_{pd}(g)$ ) and clock drivers ( $t_{pd}(clk)$ ) in contrast to global interconnections of the length  $0.5cm$  ( $t_{pd}(AI, W)$ ) as a function of the design rule

These effects can be proved by the scaling theory. Many parameters for smaller feature sizes can be derived using the scaling factor  $S$ , which represents the shrinking of the minimum feature size, and  $S_C$  denoting the increasing chip dimensions.

Table 1 shows the results of the scaling theory for gate and interconnection delay [2].

Scaling of MOS transistors	
Dimensions ( $W, L, t_{gox}$ )	$1/S$
Area per device ( $A = WL$ )	$1/S^2$
Voltages ( $V_{dd}, V_{th}$ )	$1/S$
Intrinsic gate delay ( $t_{pd}(g)$ )	$1/S$
Scaling of local interconnections	
Cross dimensions ( $w, h, w_{sp}, t_{ox}$ )	$1/S$
RC constant per unit length ( $R'C'$ )	$S^2$
Interconnection length ( $l_i$ )	$1/S$
Interconnection delay ( $t_{pd}(i) \sim R'C'l_i^2$ )	1
Scaling of global interconnections	
Die size ( $L$ )	$S_c$
Interconnection length ( $l_g$ )	$S_c$
Interconnection delay ( $t_{pd}(i) \sim R'C'l_g^2$ )	$S^2 S_c^2$

Table 1: Scaling theory for gate and interconnection delay

Notice that the  $RC$ -constant ( $= R'C'l^2$ ) as a first estimation for interconnect delay  $t_{pd}$  is constant for local interconnections while it grows by the factor  $S^2 S_c^2$  for global interconnections.

Since propagation delay and other effects caused by long interconnections (reflexions, coupling, etc.) played 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 influence on the dynamic behaviour and on the efficiency of today's electronic systems, even on chip. The performance is limited by the effects of interconnections in many cases.

Besides that, the shape of a signal transition also has a major influence on the dynamic behaviour (propagation delay, output rise time, ...) of logic cells and interconnections. These “analog aspects” in the field of digital circuits will play a more and more important role in the future.

The discussed trends and effects are not taken into consideration by conventional logic and timing models for digital cells with the necessary accuracy. Modelling the interconnections with lumped elements (resistors, inductors and capacitors) and

the logic cells by their equivalent transistor circuits and using a circuit simulator for timing verification leads to the required accuracy, but also to a huge amount of computation time. This method can be used for smaller circuits or subcircuits, but is unacceptable for large ones or complete systems.

Therefore, new methods, models and tools must be developed to meet the advanced requirements for an efficient design flow that allows the design of complex systems with high performance.

## 2 Concept for accurate modelling of interconnections

To catch the dependency of the shape of signal transitions, a new signal representation and appropriate timing models have to be created.

In contrast to traditional multi valued logic systems, the signal representation used here holds information about the shape of the signal transition in form of a piecewise linear function. Using this signal representation, models for interconnections can reflect the waveform dependency of the propagation delay.

To parametrize the models for interconnections, information about the structure (length, dimension and physical layer of each section or branch) and the used materials is needed. This information can be extracted from the layout data for each relevant interconnection net and stored in a suitable format, described in chapter 4.

Furthermore, appropriate models for logic cells currently under development, which also use this signal representation, will lead to a common environment for an accurate timing simulation of digital electronic systems.

Another step towards a combined simulation of logic cells and interconnections is the conversion of the gate level netlists (e.g. derived from logic synthesis). Usually interconnections in these netlists are treated as a single, dimensionless node where each connected gate is mapped to this single node. Using the accurate models for interconnections with different behaviour from each input port to each output port means to split these single nodes and to integrate individual instances of the interconnection models into the netlist. These instances will be mapped to different nodes as con-

nection to the associated cells (figure 2).

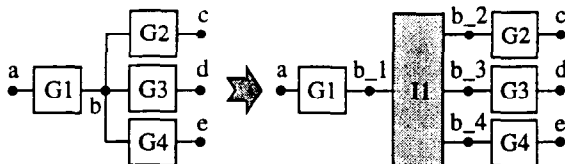


Figure 2: Netlist conversion integrating instances for the interconnection models

### 3 Signal representation

Analyzing the behaviour of CMOS-cells and interconnections proves that not only the signal shape at the output ports is depending on the input signal shape, but also important parameters like the propagation delay. Conventional signal types such as the VHDL-types *bit* or *std\_ulogic* do not contain information about the signal shape, so that the fundamental dependencies could not be modelled using these types. Transitions are interpreted with zero rise or fall time in the shape of a step function.

Some timing simulation tools meanwhile use information about the signal rise or fall time, representing the signal transition as a ramp. Some effects could be reproduced by these models, others such as overshoots caused by reflexions or strongly nonlinear waveforms, could not.

Because of this reason, signals are provided here with more information: Each signal transition is described as a sum of ramps, resulting in a piecewise linear (PWL) shape.

Figure 3 shows these three fundamental types of signal representation for a low-high transition ( $0V \rightarrow V_{dd}$ ).

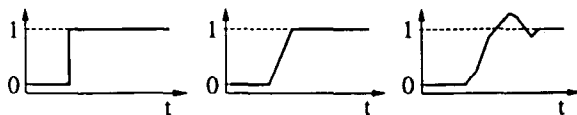


Figure 3: Step, ramp and PWL-representation of signal transitions

The PWL-representation is realized in the VHDL-types *pwl\_signal* and *pwl\_signal\_vector* in an

appropriate package, shown below. Each of the ramps consists of a time-voltage-pair,  $(\Delta t, \Delta V)$ . The voltage  $\Delta V$  is an integral multiplier of  $1mV$ , time  $\Delta t$  is an integral multiplier of the base unit *ps*. This implementation provides the necessary accuracy and results in less numerical effort inside the algorithms.

PACKAGE pwl IS

```
CONSTANT max_ramp_number : natural := 20;
SUBTYPE int_constrained IS integer
    RANGE 0 TO max_ramp_number;
```

```
TYPE ramp_rep IS RECORD
    ramp_height : integer;    -- in mV
    ramp_time   : natural;    -- in ps
END RECORD;
TYPE ramp_array IS ARRAY (int_constrained)
    OF ramp_rep;
```

```
TYPE pwl_signal IS RECORD
    ramp_number : int_constrained;
    ramp        : ramp_array;
END RECORD;
TYPE pwl_signal_vector IS ARRAY
    (natural RANGE <>) OF pwl_signal;
```

END pwl;

The precision of the implemented VHDL-type for this piecewise linear signal representation can be controlled by several parameters for the corresponding models and conversion functions. Signals of the type *bit* or “analogous” functions (which are also piecewise linear functions with very small timesteps) can be interpreted as special cases of the proposed VHDL-type.

Several conversion routines will provide high flexibility for using the new signal types together with conventional VHDL-types *bit* or *std\_ulogic*.

### 4 Description of interconnections

The mentioned geometrical information about structure, dimension and physical layer of the used material for each section is extracted from the layout and stored in a special netlist format for each interconnection net, while the technological data about the used materials are kept in a separate file, referenced by the *TECHNOLOGY* statement inside the interconnection net description:

```

BEGIN

TECHNOLOGY cmosln

BEGIN_NET net21

.LUMP_R      r1 1500          in  n1;
.INT_CMOS_MET2  n1 7E-4    9E-7 5E-7 n1  n2;
.CONT_MET1_MET2 ch1          n2  n3;

# branch 1

.INT_CMOS_MET1  n2 1.2E-3 9E-7 5E-7 n3  n11;
.CONT_MET1_POLY ch2          n11 n12;
.INT_CMOS_POLY  n3 2E-4    8E-7 4E-7 n12 n13;
.LUMP_C         c1 2E-13          n13 n14;

# branch 2

.INT_CMOS_MET2  n4 2.3E-3 9E-7 5E-7 n3  n21;
.CONT_MET1_MET2 ch3          n21 n22;

# branch 2.1

.INT_CMOS_MET1  n5 3.3E-3 9E-7 5E-7 n22 n23;
.LUMP_C         c2 1E-13          n23 n24;

# branch 2.2

.INT_CMOS_MET1  n6 3E-4    9E-7 5E-7 n22 n25;
.CONT_MET1_MET2 ch4          n25 n26;
.INT_CMOS_MET2  n7 1.1E-3 9E-7 5E-7 n26 n27;
.CONT_MET1_MET2 ch5          n27 n28;

...

END_NET

BEGIN_NET net24

...

END

```

Each instance line in this netlist format comprises information about one interconnection section (INT...), one contact hole (CONT...) or one lumped element (LUMP...). This information consists of the element type, a reference name, the dimensions and the connecting node names.

The netlist will be parsed by the interconnection models and stored in a C-data structure. All further operations are performed using this data structure.

## 5 Model for interconnections

The behaviour of interconnections is nearly linear, so that frequency domain methods can be used in an algorithm to compute it. Therefore, the transfer functions for each possible path from the input port to each output port of a complicated interconnection net are to be computed using fourpole theory.

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

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

with

$$Z = \sqrt{\frac{R' + j\omega L'}{G' + j\omega C'}} \quad (2)$$

$$\gamma = \sqrt{(R' + j\omega L')(G' + j\omega C')} \quad (3)$$

The parameters  $R'$ ,  $L'$ ,  $G'$  and  $C'$  represent the distributed parasitics for resistance, inductance, conductance and capacitance respectively and can be calculated using geometrical information and technological data [3], [4].

The implemented algorithm neglects the conductance  $G$  because of the very good insulation properties of the dielectric at the given frequency range and uses an approximation for the hyperbolic terms in equation 1.

In the same manner cascade matrices are built for contact holes ( $\mathbf{A}^{(CH)}$ ) and lumped elements ( $\mathbf{A}^{(R)}$ ,  $\mathbf{A}^{(C)}$  and  $\mathbf{A}^{(L)}$ ) as representatives for output resistance and input capacitance of logic cells or bond wires:

$$\mathbf{A}^{(CH)} = \begin{bmatrix} 1 + \frac{sC_{CH}R_{CH}}{2} & R_K + \frac{sC_{CH}R_{CH}^2}{4} \\ sC_{CH} & 1 + \frac{sC_{CH}R_{CH}}{2} \end{bmatrix} \quad (4)$$

$$\mathbf{A}^{(R)} = \begin{bmatrix} 1 & R \\ 0 & 1 \end{bmatrix} \quad (5)$$

$$\mathbf{A}^{(C)} = \begin{bmatrix} 1 & 0 \\ sC & 1 \end{bmatrix} \quad (6)$$

$$\mathbf{A}^{(L)} = \begin{bmatrix} 1 & sL \\ 0 & 1 \end{bmatrix} \quad (7)$$

In the direct path of the considered input-output pair, the cascade matrices can simply be multiplied, while branches ( $B$ ) require a special handling. Based on the final cascade matrix of the branch ( $\mathbf{A}^{(B)}$ ) the corresponding input admittance  $Y^{(B)}$  is computed using the following equation ( $C_{L,B}$  represents the load capacitance at the end of the branch):

$$Y^{(B)} = \frac{A_{22}^{(B)} + A_{21}^{(B)} / (s \cdot C_{L,B})}{A_{12}^{(B)} + A_{11}^{(B)} / (s \cdot C_{L,B})} \quad (8)$$

In a second step this admittance is interpreted as a shunt conductance and transformed into a corresponding cascade matrix:

$$\mathbf{A}^{(Y^{(B)})} = \begin{bmatrix} 1 & 0 \\ Y^{(B)} & 1 \end{bmatrix} \quad (9)$$

All branches are recursively transformed this way. Finally there are only series connected cascade matrices for each path in an interconnection net from input port  $i$  to output port  $j$ , so that the cascade matrix can be built for this path using the following formula:

$$\mathbf{A}_{(i,j)} = \prod_n \mathbf{A}^{(n)} \quad (10)$$

The transfer function  $H_{(i,j)}(s)$  from input port  $i$  to output port  $j$  considering the load capacitance  $C_{L,j}$  can then be written as:

$$H_{(i,j)}(s) = \frac{V_j(s)}{V_i(s)} = \frac{1}{A_{11(i,j)}(s) + sC_{L,j} \cdot A_{12(i,j)}(s)} \quad (11)$$

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

$$\begin{array}{ccc} v_{j,k}(t) & & v_{i,k}(t) \\ \downarrow & & \downarrow \\ V_{j,k}(s) & = & H_{(i,j)}(s) \cdot V_{i,k}(s) \end{array} \quad (12)$$

Since  $v_{i,k}(t)$  is a simple linear function, the output voltage  $v_{j,k}(t)$  can be derived analytically using the poles of the transfer function  $H_{(i,j)}(s)$ . As our models only take into consideration linear, quadratic and cubic terms of the frequency parameter  $s$ , there are only two possible situations:

1) The transfer function has three real poles ( $\alpha$ ,  $\beta$  and  $\gamma$ ). Each ramp  $k$  then produces the following proportion  $v_{j,k}(t)$  to the output signal  $v_j(t)$ :

$$v_{j,k}(t) = \frac{\Delta V_k}{\Delta t_k} \cdot \left[ a(t - t_k) \cdot \tilde{f}(t - t_k) - a(t - t_k - \Delta t_k) \cdot \tilde{f}(t - t_k - \Delta t_k) \right] \quad (13)$$

with

$$\begin{array}{ll} \Delta V_k & : \text{Voltage of the } k\text{-th ramp} \\ \Delta t_k & : \text{Time of the } k\text{-th ramp} \\ t_k & : \text{Start time of the } k\text{-th ramp} \\ a(t) & = 0 \quad \text{for } t < 0 \\ & = 1 \quad \text{for } t \geq 0 \\ \tilde{f}(t) & = r_1 + r_2 t + q_1 e^{\alpha t} + \\ & \quad q_2 e^{\beta t} + q_3 e^{\gamma t} \end{array} \quad (14)$$

2) The transfer function has one real pole ( $\alpha$ ) and one complex pole pair ( $\sigma \pm j\omega$ ). In this case the function  $\tilde{f}(t)$  has the following form:

$$\tilde{f}(t) = r_1 + r_2 t + q_1 e^{\alpha t} + 2e^{\sigma t} \cdot [x \cos(\omega t) + y \sin(\omega t)] \quad (15)$$

The sum of all these portions  $v_{j,k}(t)$  represents the output signal  $v_j(t)$ , which is finally transformed back to a piecewise linear representation, scheduled for the output node  $j$ . Optimal values for the parameters controlling this transformation can automatically be derived from input transition time and the properties of the transfer function itself.

## 6 Implementation

We chose the hardware description language VHDL ([5], [6], [7], [8]) as base for the implementation of the proposed models because of the following advantages:

- Using VHDL we are not restricted to a single simulator and/or vendor, because VHDL is a vendor-, tool- and platform-independent language. Especially there are also a lot of VHDL-simulators and debuggers on different platforms, integrated in the most design environments.
- VHDL is a worldwide established and accepted standard; it is used by a lot of design engineers around the world.
- VHDL offers a wide range of modelling capabilities including new signal types, attributes and functions.

Two aspects caused us to implement some modules in the programming language *C* rather than in VHDL:

- The efforts to create a standardized VHDL-package for mathematical functions (IEEE-package `MATH_REAL`) are still ongoing and the problem of precise types isn't solved sufficiently yet. The current version of the IEEE-package does not provide double precision types.
- The used algorithms are based on nontrivial mathematical operations such as matrix multiplications. Investigations on implementations in *C* and VHDL using standard mathematical packages (`math.h` respectively `MATH_REAL`) proved the advantages of the *C*-realization regarding simulation performance.

The *C*-modules are linked to a conventional VHDL-Simulator using a corresponding, but tool specific *C*-language interface.

We chose the widespread simulator *vss* (SYNOPTIS) and its *C*-language interface *CLI* [9]. Other simulators offer similar interfaces, so that there is no general restriction to that concept.

*CLI* uses four different routines for elaboration, evaluation, error handling and closing functionality. The elaboration routine is only activated once during the initialization phase of the VHDL-simulation. It can be used to calculate the transfer function (equations 1 to 11) for the different paths in the interconnection net. The calculation of the output signals via Laplace transformation (equations 12 to 15) is done inside the evaluation routine called by each input event.

Some problems arise because the analog waveform interpretation can't be displayed clearly enough by conventional digital waveform display tools. Therefore we use an analog display tool provided by a circuit simulator. Nodes to be examined are provided with a special module to save the analog waveforms in an appropriate format.

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. An example for a structural VHDL-model is listed in chapter 7.

## 7 Results

Investigations on a complex, global interconnection net prove the usability of the model for post layout timing simulation with VHDL as well as for clock skew calculations. The used net depicted in figure 4 connects several separated functional blocks on an integrated circuit.

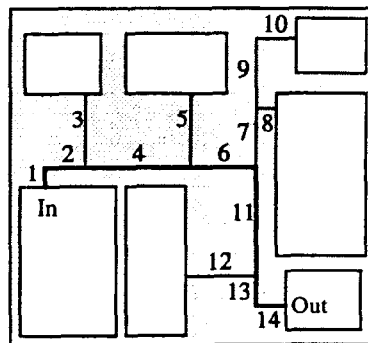


Figure 4: Example of an interconnection net with several output ports

After generating the corresponding netlist description for this structure, the simulation can be

started. Using a single event at node *In* in form of a *1ns*-ramp, the results of a VHDL-simulation for node *Out* are shown in figure 5 compared to the results of a corresponding circuit simulation using *SPICE3* and its lossy transmission line model *LTRA*.

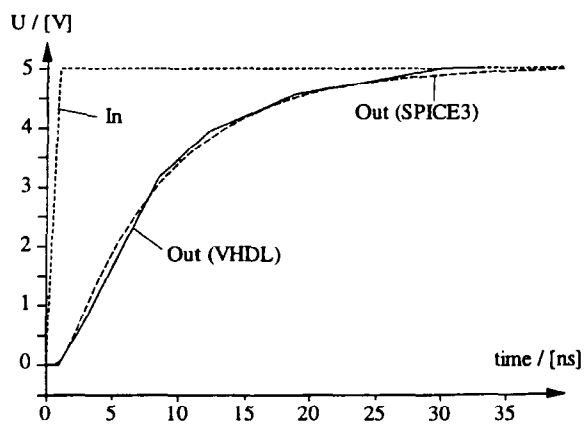


Figure 5: Simulation results for node *Out* (on-chip parameters)

Using typical off-chip (PCB-) parameters instead of on-chip parameters yields in simulation results shown in figure 6. Here, overshoots caused by reflections, prove the model capability for this kind of interconnections, too.

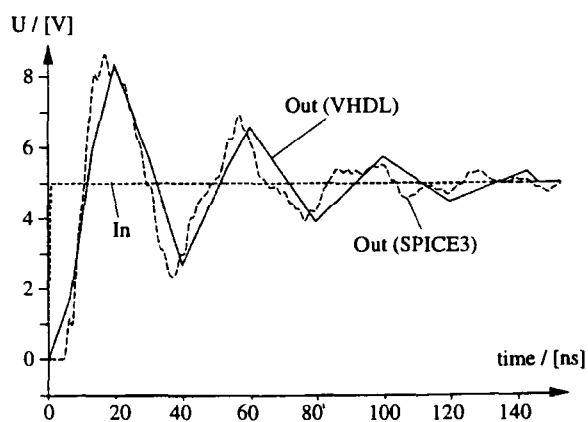


Figure 6: Simulation results for node *Out* (off-chip parameters)

In both cases, the VHDL-simulation using the new models was 100 (on-chip parameters) to 125 times

(off-chip parameters) faster than the equivalent circuit-simulation.

To validate the usefulness of the model for clock skew examinations, the waveforms were calculated for several output nodes, shown in figure 7. Here the signals at the output nodes of the branches 3 and 8 and node *Out* are compared for the same input signal.

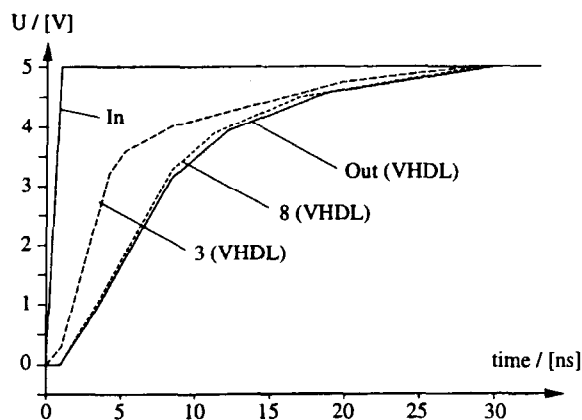


Figure 7: Skew for various output nodes (on-chip parameters)

As a second example, a critical path (figure 8) inside a larger circuit has been simulated together with conventional models for the rest of the circuit.

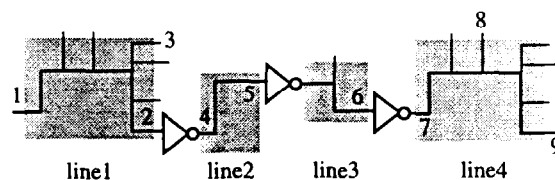


Figure 8: Critical path

The VHDL-netlist for this example is shown below and demonstrates the use of the new models (including special plot routines) in VHDL. Different kinds of models have to be used for *line1* to *line4* according to the different numbers and types of ports.

```

USE work.pwl.ALL;
USE work.pwl_components.ALL;

ENTITY crit_path IS
  PORT ( ... );
END crit_path;

ARCHITECTURE structural OF crit_path IS
  SIGNAL a,e,d,f,g,k : pwl_signal;
  SIGNAL b : pwl_signal_vector(2 DOWNT0 0);
  SIGNAL c : bit_vector(2 DOWNT0 0);
  SIGNAL h : bit_vector(0 DOWNT0 0);
  SIGNAL l : bit_vector(5 DOWNT0 0);
BEGIN

  line1: p_pv_bv
    GENERIC MAP ( ... ) PORT MAP (a,b,c);
  inverter1: p_inv
    GENERIC MAP (15 ns) PORT MAP (b(2),e);
  line2: p_p
    GENERIC MAP ( ... ) PORT MAP (e,d);
  inverter2: inv
    GENERIC MAP (15 ns) PORT MAP (d,f);
  line3: p_p_bv
    GENERIC MAP ( ... ) PORT MAP (f,g,h);
  inverter3: inv
    GENERIC MAP (15 ns) PORT MAP (g,k);
  line4: p_bv
    GENERIC MAP ( ... ) PORT MAP (k,l);

  plot1: p_plt
    GENERIC MAP ("(1)") PORT MAP (a);
  plot2: p_plt
    GENERIC MAP ("(2)") PORT MAP (b(2));
  ...

END structural;

```

The path, consisting of four interconnection nets and three simple inverter models, is stimulated by a single ramp. Some of the nodes (1, 2, 4, 5, 6 and 7) use the new signal representation, while others (3, 8 and 9) have the type *bit*. The results of the simulation are plotted in figure 9, where *bit*-signals are drawn as waveforms between 0 and 1V.

## 8 Conclusion

We have discussed an approach for accurate modelling the influences of interconnections on the timing behaviour of digital circuits. To catch the dependencies of the shape of the signal transi-

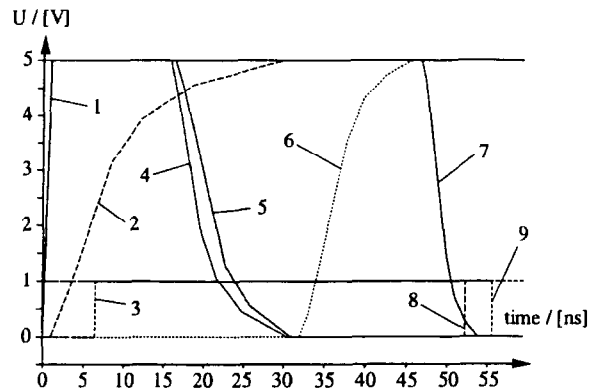


Figure 9: Simulation results for a critical path (on-chip parameters)

tions, a new piecewise linear signal representation is used. The proposed concept has proved its usability for post layout timing simulation using a conventional VHDL-simulator.

The advantages of this concept are:

- The model for interconnection represents the layout structure including branches, their loads and the various interconnection layers with their individual electrical properties. The model is based on distributed parasitics: no lumped circuit equivalents are used to model lossy transmission line sections.
- The results of the model do not only include the propagation delay, but even the waveform at each of the output nodes of the interconnection as a function of the input waveform and the interconnection itself.
- The concept can be adapted to be used by other VHDL-simulators or on other hardware platforms. It can be used for other levels of interconnections such as multi chip modules (MCMs) or printed circuit boards (PCBs) and is not restricted to integrated circuits.
- There are several ways of using the model for interconnection:
  - Modelling complete circuits (together with appropriate PWL-models for the logic cells)

- Modelling only critical paths (together with appropriate PWL-models and conventional models for the logic cells)
- Modelling single interconnections (e.g. for investigations on clock skew)
- Pre-layout usage (with estimated interconnection data based on the floorplan)
- Post-layout usage (with exact interconnection data extracted from the layout)

This concept can be integrated into a typical VHDL-design environment using VHDL-tools for simulation and synthesis. It will provide a method for fast and accurate investigations on the post layout timing behaviour of digital electronic systems.

## 9 Future Work

Future work will concentrate on the realization of an environment for a *mixed mode simulation* together with conventionally modelled components. This will include:

- Development of corresponding and accurate models for logic cells being capable of handling the PWL signal representation,
- Automatic netlist conversion based on the synthesized VHDL-netlist and layout data integrating instances for the interconnection models and using accurate models for the logic cells,
- Extraction of the necessary information about the structure and used materials for the relevant interconnection nets from the layout.

Furthermore we will modify the only tool specific element, the *CLI*-specific interface description, so that the models can be used by other VHDL-simulators, too.

## Acknowledgements

The authors would like to express their gratitude to Peter Schlarb, Ralf Gräbner, Gerd Lehmann, Christian Beck and Andreas Pitsch for their help during development and implementation of the proposed methods and models.

## References

- [1] Marc Rocchi: *High Speed Digital IC Technologies*, Artech House, Norwood, 1990
- [2] H. B. Bakoglu: *Circuits, Interconnections and Packaging for VLSI Circuits*, Addison-Wesley, Reading, MA, 1990
- [3] E. T. Lewis: *An Analysis of Interconnect Line Capacitance and Coupling for VLSI Circuits*, IEEE Journal of Solid State Circuits, Vol. 27 (8), 1984
- [4] Yuan, Lin, Chiang: *Properties of Interconnection on Silicon, Sapphire and Semi-Insulating Gallium Arsenide Substrates*, IEEE Transactions on Electronic Devices, Vol. 29 (4), 1982
- [5] The Institute of Electrical and Electronics Engineers: *IEEE Standard VHDL Language Reference Manual (ANSI/IEEE 1076-1993)*, New York, 1994
- [6] J. Berge, S. Maginot, A. Fonkoua, J. Rouillard: *VHDL Designer's Reference*, Kluwer, Dordrecht, 1992
- [7] D. L. Perry: *VHDL*, McGraw Hill, New York, 1991
- [8] G. Lehmann, B. Wunder, M. Selz: *Schaltungsdesign mit VHDL*, Franzis, Poing, 1994
- [9] Synopsys, Inc.: *C-Language Interface Reference, Version 3.1*, Mountain View, CA, 1994