

Accurate VHDL Delay and Power Characterization of CMOS Logic Cells

N. Dumitru, R. Nouta

Department of Electrical Engineering

Delft University of Technology

Mekelweg 4, 2628 CD Delft, the Netherlands

Email: dumitru@cas.et.tudelft.nl, reinder@cas.et.tudelft.nl

Abstract

This paper presents a model for characterizing delay and power for CMOS logic cells that accounts for input slope and output capacitance loading. A method for deriving the model parameters and VHDL modeling for simple logic gates is presented. The model makes feasible delay and power estimation at VHDL simulation speed, the errors of the model prediction are less than 5% of Spice results.

1 Introduction

Analog circuit simulators suffer from severe memory and execution time constraints and are hence unsuitable for VLSI circuits. Logic and timing simulators are much faster, but their accuracy depends upon the accuracy of the model used for simulation. Usually, delay and power models used by logic simulators assume signal slew rate as being constant, and model only the load influence for delay and power dissipation of the cell. In order to improve the simulation accuracy, the model should include also the dependency upon the signal slope together with cell output load.

In this paper we present a CMOS cell delay model which accounts for both input slope and output load in deriving the delay and power values. Our characterization is based on piece-wise linear fitting of the model parameters to a number of Spice3 simulations for both delay and power model. The model parameters for each cell are derived from circuit simulations of the netlist based on cell's layout using lateral interconnect capacitance. Level-3 transistor modeling is used. We have determined the delay and power parameters for our Sea-of-Gates (a gate-

isolation image in a 1.6 μ m CMOS process with two layers of metal) library cells, such as: buffer, inverter, nand2, nor2, exclusive-or, edge triggered D flip-flop, edge triggered D flip-flop with synchronous reset.

This paper is structured as follows: Section 2 introduces the delay model. Section 3 presents the cell dissipation parameters. Section 4 presents the VHDL implementation of the model. Simulation results and conclusions are presented in Section 5. An example of a VHDL code which describes a nand2 gate using our delay and power model is presented in the Appendix.

2 The delay model

Our characterization is based on the observation that CMOS logic gates exhibit two different modes of switching, one for relatively fast input slope and other for relatively slow input slope. The two modes of operation – named *FAST* mode and *SLOW* mode – depend upon the relative values of the input slope and output load. In order to introduce the two switching regions, we take as an example the switching of the CMOS inverter and compare the output switching in both cases.

The inverter is said to switch in a fast mode when the input transition ends before the output starts to move. This happens when the input signal is fast and the load of the inverter is sufficiently low. In this case, the output capacitance is charged (discharged) by a constant current-source of a value given by the saturation current of one transistor, the current contribution of the second transistor can be neglected.

If the inverter is heavily loaded and/or the input is a slow-moving signal, the discharge time of the output capacitance is small compared to the input transition-time, and the inverter is switching in a slow mode. The current contribution from the second transistor cannot be neglected totally and the current which charge (discharge) the load capacitance is less than the maximum drive current possible. For infinitely slow inputs, the output transition follows the DC characteristic of the inverter [1].

Our modeling approach reflects the existence of the two modes of operation. The most important features are: i) the presence of fast and a slow regions separated by critical slope (CS) line. ii) the critical input slope is a linear function on the output load.

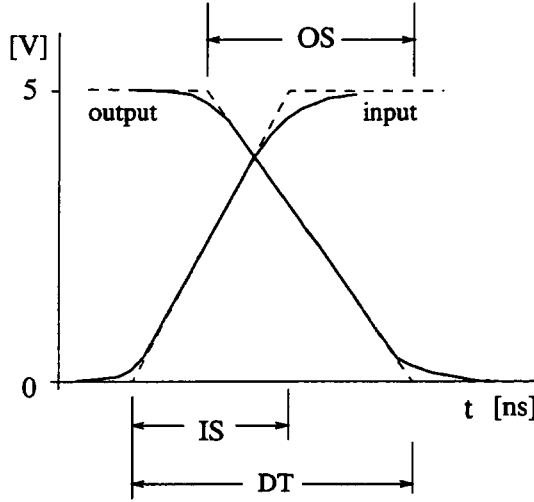


Figure 1: Definition of IS, OS and DT.

The delay model is developed in terms of the geometrical quantities IS, OS, and DT presented in Figure 1. IS (input slope) and OS (output slope) represent the linear approximation of the input and output transition-time expressed in ns. Conventionally, the slope is calculated from the time difference between the 90% and 10% points on the output curve. We found that the conventional way of expressing slope is not always suitable for our modeling approach, and therefore we have used for slope approximation the quasi-linear region of the output waveform which is around the logic threshold voltage. The logic threshold voltage of a gate is defined as input voltage which determines the output to switch, and for the inverter used in this example is about 2.0 V.

Figure 2 plots Spice predictions of OS versus IS for the inverter output for several loads equally spaced. Due to the fast and slow regions separation, the output slope can be described by two linear functions, one for the fast region and the other for the slow region. The two functions are equal on the fast-slow boundary determined by CS line.

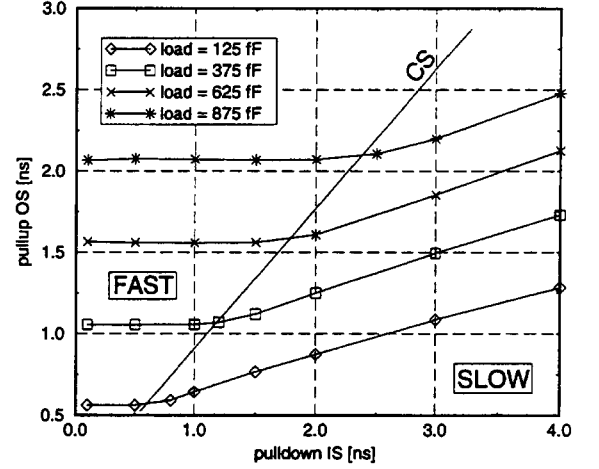


Figure 2: Spice prediction of inverter OS vs. IS (pullup transition).

Our model to estimate the OS versus IS variation is given by

$$OS = \begin{cases} a + b \cdot C_L & \text{when } IS \leq IS_{cr} \text{ (FAST)} \\ c + d \cdot C_L + m \cdot IS & \text{when } IS > IS_{cr} \text{ (SLOW)} \end{cases} \quad (1)$$

where IS_{cr} is derived from the CS line and represents the critical slope value for a fixed output load C_L . IS_{cr} is a linear function on the output load of the inverter.

$$IS_{cr} = \frac{a-c}{m} + \frac{b-d}{m} C_L \quad (2)$$

The a , b , c , d , and m values from Equation 1 are the characterization parameters for OS. The intuitive interpretation of these parameters is as follows:

a - is OS of the unloaded gate when the input is the step signal.

b, d - are the parameters which model OS variation with load

m - is the parameter which models OS versus IS variation for slow switching region

c - is a fictitious parameter for the slow region which play the same role as the a parameter for the fast region.

The gate-delay versus input slope plots also present fast and slow switching regions separated by a CS line. However, when the gate-delay is plotted against IS, the slow region plots of the gate delay can hardly be approximated by a linear function. Therefore, another geometrical quantity represented in Figure 1, namely *delay time* (DT), is selected for a piece-wise linear approximation. Figure 3 shows Spice results of DT versus IS plots of the CMOS inverter for several loads. Again a fast region – slow region partition can be made in the same way as for the OS.

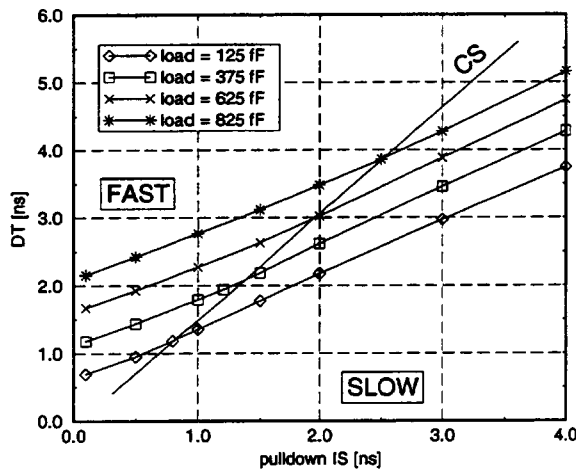


Figure 3: Spice predictions of the inverter DT vs. IS (pullup transition).

The piece-wise linear approximation for DT is given by

$$DT = \begin{cases} a + b \cdot C_L + m_1 \cdot IS & \text{when } IS \leq IS_{cr} \text{ (FAST)} \\ c + d \cdot C_L + m_2 \cdot IS & \text{when } IS > IS_{cr} \text{ (SLOW)} \end{cases} \quad (3)$$

where IS_{cr} is

$$IS_{cr} = \frac{a-c}{m_2-m_1} + \frac{b-d}{m_2-m_1} C_L \quad (4)$$

The parameters m_1 and m_2 model the DT vs. IS variation for the fast and slow region respectively.

From Figure 1, the 50% gate-delay (denoted as Δ) can be expressed as

$$\Delta = DT - \frac{IS}{2} - \frac{OS}{2} \quad (5)$$

Delays other than 50% gate-delay can be derived as well using IS, OS, and DT values. For example, the 40% gate-delay would be $\Delta_{40\%} = DT - 0.4(IS + OS)$.

In order to determine the characterization parameters, a number of Spice simulations with varying input slope and output load are performed. There are 11 parameters for every input-output pair per input transition (5 params. for OS and 6 params. for DT characterization), which makes 22 parameters in total for the inverter cell characterization. It is important to notice that the minimal number of Spice simulations required by this method is 6 (3 for the fast mode, 3 for the slow mode). However, only 6 simulations are not enough to generate a model which can predict delays with less than 5% error. We have used about 20 Spice simulations for every input/output pair of the library cell.

For simplifying the task of determining the parameter values, we have created a tool which runs Spice in a batch mode and captures automatically DT, OS and power parameters. For the current value of the input slope, the input capacitance of the cell under simulation is captured and displayed. This feature is useful for VHDL modeling, as the model uses the output capacitive loading to estimate the cell-delay and power dissipation.

2.1 Temporal proximity of transitions on different inputs

The problem of the dependence of the gate-delay on the input slope should consider also the case when two or more inputs of the gate switch simultaneously. The single-input-change model – commonly used for its simplicity – can lead to significant errors in the gate-delay, especially in situations where signal delay times have large spread. In such cases, a single-input-change model can seriously underestimate the gate-delay. The dependence of the gate-delay on the overlap of the transitions on different inputs is introduced by another example: a CMOS nand2 gate. Figure 4 presents the Spice results for pullup delay

versus IS of a nand2 gate. The input transitions are skewed by δ , referred to as the *input data-skew*, defined as the interval elapsed from the moment the first and the second transition reaches 50% of their voltage swing. Figure 4 suggests that gate-delay of a nand2 gate depends almost linearly on the input data-skew factor δ .

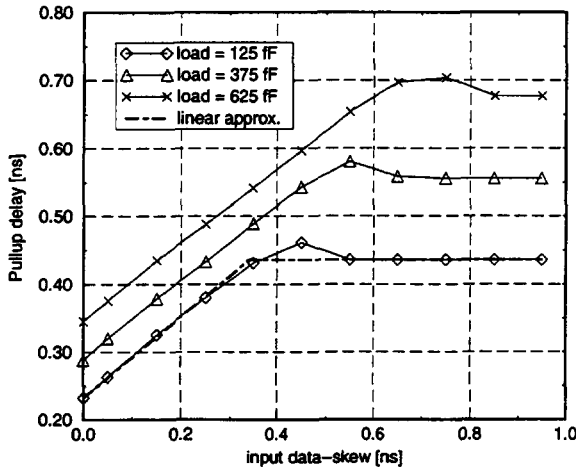


Figure 4: Spice results of delay vs. input data-skew for nand2 gate. (IS = 0.2 ns)

Our model to estimate the gate-delay vs. input data-skew for a nand2 gate is given by

$$Delay = \begin{cases} k \cdot \Delta_1 + (1-k)\Delta_2 & \text{when } \delta \leq K \cdot \Delta_1 \\ \Delta_1 & \text{when } \delta > K \cdot \Delta_1 \end{cases} \quad (6)$$

where

Δ_1 - nand2 gate delay when only one input changes

Δ_2 - nand2 gate delay when both inputs change simultaneously

δ - input data-skew ($0 \leq \delta \leq \Delta_1$ for skewed inputs)

$$k = \delta / \Delta_1$$

K - a constant which depends on the logic gate ($K=0.85$ for Sea-of-Gates nand2 gate).

3 The power model

Power is dissipated in a CMOS cell whenever the output switches from one voltage level to another. Usually, power estimation is based on a toggle frequency formula $P = C_L \cdot V \cdot f$ where C_L = capacitance being charged, V = voltage swing, f = toggle frequency. Such approach cannot take into account the slope of the signal input or power dissipated due to the short-circuit current.

We present a modeling approach for power which accounts for input slope and output loading as well as for the short-circuit current to estimate the power dissipation of a cell. The model is similar to the delay model presented in the previous section and relies on the fact that the behavior of the energy dissipated by CMOS cells exhibit also *FAST* and *SLOW* operation modes. Both static and dynamic power dissipation of the cell are modeled in our approach. The energy dissipated for one output transition is estimated as

$$E_n = \int_0^t V_{dd} \cdot i(t) dt. \quad (7)$$

Figure 5 presents energy plots for the CMOS inverter when the output makes a low-to-high transition for several loads equally spaced.

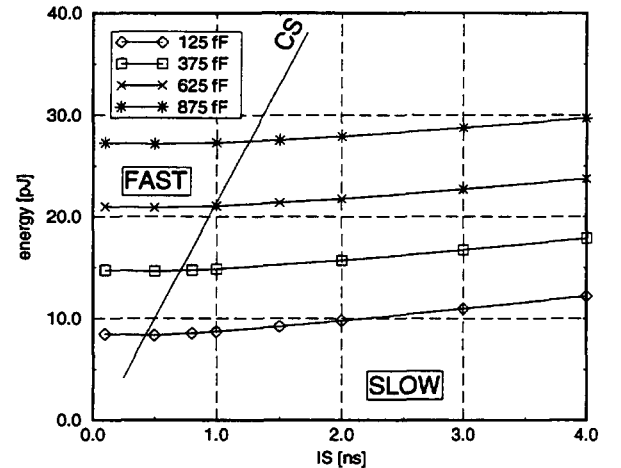


Figure 5: Energy vs. IS for the CMOS inverter (low-to-high transition of the output).

For fast input slopes overshooting/undershooting of the signal at the output, known as the feedforward effect, may occur. For the inverter example, when

the input node makes a low-to-high transition, the output voltage is temporarily driven up above V_{dd} before the NFET pulls the output node down. When this is the case, the output "pumps" current into the power supply, and the dissipated energy is negative.

4 Modeling in VHDL

For interconnected cells, the output signal of a cell serves as input signal for the subsequent cells. In order to have the slope propagated along with the signal, a record type would be the perfect choice to describe signals in VHDL. We have defined `s_logic` record type as a compound (*value*, *slope*), where *value* is `std_logic` and *slope* is a real number representing OS value. The cell load is passed through a generic statement.

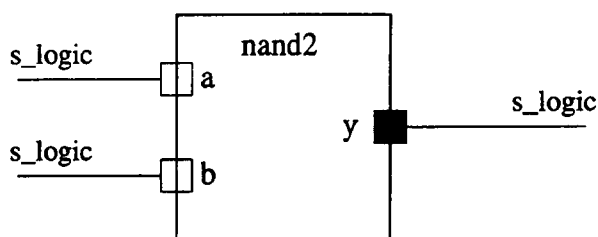


Figure 6: Interface aspect of nand2 gate.

The `s_logic` type and its derivative `s_logic_vector` type are declared in `s_logic_package`. The `s_logic_package` contains also the characterization parameters and function definitions for delay and power modeling.

Delay modeling uses *f_slope*, *f_dtime*, *f_delay* functions to estimate the OS, DT, and gate-delay value respectively.

For power modeling two global signals `added_energy` and `imed`, and two associate functions and subtypes are declared. The function `sw_energy` evaluates the energy dissipated by the cell each time the output makes a '0'→'1' or a '1'→'0' transition. Signal `added_energy` gives the total energy – expressed in pico-Joules [pJ] – which is dissipated during the simulation. The signal `added_energy` is resolved by the function `resolve_energy` which adds up the energies dissipated by each individual cell. The use of more global energy signals makes possible to estimate dissipation in different design blocks using configuration decla-

rations which bind each component instantiation to a different energy signal.

The signal I_{med} estimates the short-circuit current (I_{sc}) which appears during the transition of the output signal. The value I_{med} is derived as

$$I_{med} = E_n / (V_{dd} \cdot OS) \quad (8)$$

The I_{sc} current which has a triangular shape, is truncated to a rectangle of height I_{med} and width OS. This approximation matches relatively well the peak value of the real current I_{sc} , but gives big errors in case of the partial overlap of the rectangles. A solution to this problem is to use 3 rectangles of different heights to obtain a better match for I_{sc} .

In order to use the delay and power models one should i) make visible the package `s_logic_package`, and ii) use `s_logic` or `s_logic_vector` signals, as in the following example:

```
library ieee;
use ieee.std_logic_1164.all;
use work.s_logic_package.all;
entity chain_inv is
  port (a: in s_logic;
        . . .
```

We have generated VHDL descriptions which use the input slope and output load in delay and power estimation for several CMOS cell, such as inverter, nand2, nor2, exclusive-or, edge triggered D flip-flop, edge triggered D flip-flop with synchronous reset. An example of the VHDL code for the nand2 gate which includes delay and power modeling is presented in the Appendix.

5 Results and conclusions

VHDL simulation results were compared against Spice3 results for several test circuits. All VHDL models have been simulated with the Leapfrog 2.1 simulator (from Cadence) and the VSS V5.400 simulator (from Vantage).

Table 1 presents the error predictions of the VHDL models compared to Spice results for several test circuits. The VHDL descriptions of nand2 and nor2 include the simultaneous transition model described in Section 2.1.







<i>test circuit</i>	Delay / oscillation period				Error	
	VHDL		Spice3		Δ [%]	
						
chain of 20 inverters	6.31 ns	6.36 ns	6.50 ns	6.53 ns	2.9 %	2.6 %
chain of 20 nand2	8.77 ns	8.67 ns	8.78 ns	8.75 ns	0.1 %	0.9 %
chain of 20 nor2	11.06 ns	10.88 ns	10.98 ns	10.93 ns	0.7 %	0.4 %
ring osc. of 11 inverters	7.33 ns		7.15 ns		4.9%	
ring osc. of 11 nand2	10.03 ns		9.99 ns		0.3%	
ring osc. of 11 nor2	12.31 ns		12.39 ns		0.6%	

Table 1: Delay simulation results

<i>test circuit</i>	VHDL sim.	estimated from Spice	Error
chain of 20 inv.	157.22 pJ	165.38 pJ	4.9 %
chain of 20 nand2 (one input)	189.62 pJ	195.89 pJ	3.2%
chain of 20 nand2 (inputs together)	236.72 pJ	239.70 pJ	1.2 %
chain of 20 nor2 (one input)	210.67 pJ	218.05 pJ	3.3 %

Table 2: Power simulation results

Results of our dissipation model were found to be qualitatively and quantitatively similar to the delay model. Table 2 presents the error prediction for the energy dissipated by several test circuits. For the nand2 chain two simulations are considered: one for the case when the gate inputs are connected together (the nand2 gate works as inverter), and the other for the case when only one input is driven, the other input being connected to '1' logic.

Table 3 presents the amount of CPU time needed to simulate for 5 μ s a chain of 1000 nand2 gates on a HP 735/9000 machine. We have used for this simulators available at TU Delft: a VHDL simulator, Spice3, and SLS timing simulator. The acronym SLS stands for *Switch-Level-Simulator*, and the simulator can be used for simulating the logical and timing behavior of digital MOS circuits. In the simulator transistors are modeled by grounded capacitors and a switched resistor [2]. Three delay models have been used to model in VHDL the behavior of the nand2 gate: a simple delay model (y <= a nand b after 1 ns;), single-input-change model, and two-inputs-change model. Our VHDL simulator was VSS release V5.400. The evaluation time and initialization times have been subtracted from the total CPU time reported by VSS. When only delay models are used, the simulation time increases, but re-

mains negligible compared to Spice simulation time. When both delay and power models are used, the increase in CPU time becomes quite substantial. However, the total time is less than 1% of the CPU time required by Spice.

A similar method for estimating delay was presented by Misheloff in [3]. Misheloff's characterization uses 12 parameters to characterize an inverter, and requires only 4 Spice simulations to be done for a cell. In his approach Misheloff uses also the DT quantity. He considers that DT is a totally linear function on load and input slope. We found that a piece-wise approximation for DT is a more accurate approach. In the assumption that our model estimates DT as a totally linear function, our inverter characterization would have 20 parameters instead of 22 currently used.

A method for characterizing delay and power dissipation of a logic gate, and VHDL implementation of the model is presented. The characterization method allows accurate delay and power dissipation to be carried out at VHDL simulation speed, and can predict the gate-delay and dissipated power over a range of input signal slopes and output loading. Results obtained from VHDL simulations have been found to be typically within 5% of Spice.

chain of 1000 nand2	VHDL			Spice3	SLS timing
	simple delay	single-input	two-inputs		
CPU time (delay only)	0.15 sec	1.17 sec	2.28 sec	55÷150 min*	5.1 sec
CPU time (delay & power)	-	23 sec	24.3 sec	-	5.7 sec

Table 3: CPU time for 5 μ s simulation of a chain of 1000 nand2 gates (55÷150 min* - estimated time).

References

- 1). Shoji Masakazu, "CMOS Digital Circuit Technology", Prentice Hall, 1988.
- 2). A.C. de Graff, A.J. van Genderen, "SLS: switch-level simulator user's manual", Delft University of Technology, 1993.
- 3). M.N. Misheloff, "Improved Modeling and Characterization System for Logic Simulation", Proc. of the IEEE Asic Intl. Conference, 1992.
- 4). Navabi Zainalabedin, "Analysis and Modeling of Digital Systems", McGraw-Hill, 1993.

Appendix

VHDL model of nand2 gate (delay & power)

```

library ieee;
use ieee.std_logic_1164.all;
use work.s_logic_package.all;
entity nand2 is
  generic (load: real := 128.0);           -- gate load in fF
  port (a,b: in s_logic;
        y: out s_logic );
end nand2;
--
architecture pr_nand2 of nand2 is
begin
  a1: process (a.value, b.value)
    variable buf_val, old_val           : std_logic ;
    variable out1_delay, out2_delay, buf_delay : time := 0 ns;
    variable out1_slope, out2_slope, buf_slope : real := 0.0 ;
    variable out1_dtime, out2_dtime          : real := 0.0 ;
    variable in_slope, med_sl              : real := 0.0 ;
    variable k, pow1, pow2, buf_en_tr, buf_en : real := 0.0 ;

```

continued on the next page

continued from the previous page

```
begin
buf_val    := a.value nand b.value;
-- select input slope (IS)
if (a.value'last_event < b.value'last_event) then in_slope := a.slope;
  else in_slope := b.slope;
end if;
  -- single_input_change model
  out1_dtime := f_dtime (data_na1, load, in_slope, buf_val);
  out1_slope := f_slope (data_na1, load, in_slope, buf_val);
  out1_delay := f_delay (out1_dtime, in_slope, out1_slope);
  pow1       := sw_energy (nand2_pw, load, in_slope, buf_val);
  -- two_input_change model
  med_sl := a.slope/2.0 + b.slope/2.0 ;
  out2_dtime := f_dtime (data_na2, load, med_sl, buf_val);
  out2_slope := f_slope (data_na2, load, med_sl, buf_val);
  out2_delay := f_delay (out2_dtime, med_sl, out2_slope);
  pow2       := sw_energy (nand22_pw, load, med_sl, buf_val);
-- select switching model
if abs(a.value'last_event-b.value'last_event) <= 0.85*out1_delay
  and a.value=b.value then
  -- select two_input_change model
  k := real(abs(a.value'last_event - b.value'last_event)/out2_delay);
  buf_slope := k*out1_slope + (1.0 - k)*out2_slope;
  buf_delay := k*out1_delay + (1.0 - k)*out2_delay;
  buf_en_tr := k*pow1      + (1.0 - k)*pow2 ;
else
  -- select single_input_change model
  buf_slope := out1_slope;
  buf_delay := out1_delay;
  buf_en_tr := pow1;
end if;
  y <= (buf_val, buf_slope) after buf_delay;
-- power
if ((to_X01(buf_val) = '1' and to_X01(old_val) = '0') = TRUE) or
  ((to_X01(buf_val) = '0' and to_X01(old_val) = '1') = TRUE) then
  buf_en := buf_en + buf_en_tr;
  added_energy <= buf_en;
  imed <= buf_en_tr/(5.0*buf_slope), 0.0 after buf_slope*1 ns;
end if;
  old_val := buf_val;
end process a1;
end pr_nand2;
```