TSIM Manual

Charles Fabian

March 24, 1995

1 Introduction

TSIM is a event-based, logic-transition simulator based on GSIM. [?] It is designed specifically for use in estimating dynamic power dissipation of a logic circuit. TSIM takes a circuit description and a sequence of input events and simulates the circuit at a logic gate level, recording the transition of each logic gate as it occurs. TSIM will also detect hazards caused by multiple input lines of a logic gate changing simultaneously. TSIM also allows specification of the amount of power dissipated per transition of each gate.

1.1 Simulation Engine

A simulation is composed of a circuit and a sequence of input events which occur at discrete time steps. The circuit consists of a network of basic logic gates; feedback is allowed in the circuit, but each gate's input can be driven by the output of at most gate. *(Add caution about using delays to avoid infinite loops)* The input event sequence specifies logic values (`0`, `1`, or `X`) for the primary inputs for each time-step of the simulation.

If, for a particular time-step, no value is specified for primary input, the value of that primary input signal is left unchange from its previous value. At the start of the simulation, all primary inputs and all gate outputs are initialized with the value X, which represents the unknown state.

At each time-step, a new output value is computed for every gate with an at least one input event. (TSIM calls changes in the logic value of a line *events*.) If the newly computed output value differs from its previous value, the event is recorded in the simulation's history of the activity for that gate. The time recorded for the event is the current time-step plus the appropriate rise/fall time of the gate. Also at that recorded time, the event will be forwarded to each of the gate's descendants, propagating the input change through the circuit. If the newly computed value does not differ from the previous value of the gate's output, no change event is recorded in the gate's output history and no event will be forwarded to the gate's descendents.

The simulator continues until no events are pending for the current time-step; each time-step is processed in this same manner.

TSIM does not perform any levelization or depth analysis preprocessing on the circuit. Therefore it is possible that a gate's value would be re-computed multiple times during the same time-step.[1] Rather than record multiple changes during the same time-step for such a gate, TSIM will revise the gate's output history to record just one event, representing the gate's final value for that time-step. (See the hazard discussion below.) This approach differs from that in GSIM, which will record

[1] Even with levelization, feed-back loops would force the simulator to re-compute the values of gates.

multiple transitions per gate per time-step. TSIM's approach has the advantage that a gate's history will be independent of the simulation order, including the order that gates and primary inputs are specified in the input file.

1.2 Hazard Determination

In the case where a gate's output value does not differ from its value at the end of the previous time-step, it is still possible that an output transient may occur. TSIM detects transients from two potential causes: simultaneous changes in a gate's input; and differing signal rise and fall times for a gate.

Transients can potentially occur when more than one input to a gate changes simultaneously. This may be illustrated by an 2-input XOR gate with one input changing from 1 to 0 and the other input changing from 0 to 1. Although the begining value and lasting value of the XOR's output are both the same (1), the possibility exists that sometime while the inputs are changing that the output may momentarily change to a 0. In TSIM, if the two input changes occur on the same time-step a DOWN_SPIKE will occur on the output. The DOWN_SPIKE has the same logical value as a 1 value, but has different implications for power consumption calculations (see the discussion on power computation below). An UP_SPIKE could occur when both inputs to the XOR have the 1→0 transition.

The rules for transients caused by simultaneous input changes are a little more complicated for the AND, OR, NAND and NOR gates. Each of these gates can only produce one type of transient, either the UP_SPIKE or the DOWN_SPIKE. For example, the AND gate can only produce the UP_SPIKE; it is produced only when the previous and lasting value of the AND's output are both 0 and each input to the gate either has a 1 value or is changing value.

Transients may also occur in gates which have differing rise and fall times. This can be illustrated by the exagerated example of a 2-input OR gate with a rise time of 10 units and a fall time of 1 unit. First, one input to the gate changes from 0⟶1 while the other input is 1, ten time units later the output will change to a 1. Physically, the change in the output occurs gradually until the valid 1 level is finally attained ten time units later. However, if 5 units later that input changes back to a 0, the output of the gate will revert back to a 0 in one time unit. Logically, the output of the gate never did change, but physically, a small spike may have occured. TSIM will generate a UP_SPIKE for this case. Note that the AND, OR, NAND and NOR gates are not limited to producing a particular type of transient; any type of gate may produce a UP_SPIKE or a DOWN_SPIKE, depending on the difference in the gate's rise and fall times.

Note that transients caused by simultaneous input changes and by differing rise and fall times are essentially the same; the difference is in the scale of the time-step compared to the scale of the rise-fall time differences.

In the TSIM circuit model, transients on input lines have no effect on a logic gate; they are treated just like a normal 1 or 0 value, as appropriate. Thus transients are not propagated through the circuit. As mentioned, transients only affect the power consumption computation.

1.3 Power Computation

TSIM uses a model for dynamic power dissipation similiar to that described in [?]. Each transition of a gate's output signal results in a fixed amount of power dissipation; the amount depends on the physical layout of the circuit. TSIM extends this model by allowing different amounts of power

dissipation for the rise and fall transitions of the gate's output. The rise and fall power constants (called `p01` and `p10` can be specified for each logic gate; they default to 1 unit of power.

As mentioned earlier, TSIM also also considers transients spikes in the dynamic power computation. In a real circuit, spikes occur in various shapes and sizes; the power dissipated by a spike strongly dependents on its shape, size, duration, and which inputs are changing. For simplicity, TSIM lumps all possible spikes into two categories, UP_SPIKE and DOWN_SPIKE. For each gate, all spikes are assumed to dissipate a fixed amount of power, depending on whether it is an UP_SPIKE or a DOWN_SPIKE. The exact amounts (called `pUP` and `pDOWN`) may be specified for each gate; these amounts may be viewed as the power dissipated by the "average" spike.

1.4 Result Tabulations

Gate Transitions For each gate, the simulator will output a history list of transitions. Each transition is of the form time value. Currently the printed list does not include Up-spikes and Down-spikes. After the gate's list of transitions the simulator prints the total number of transitions (not including hazards) and the total power dissipated in the gate (including hazards).

Transitions per Time-Step After the transitions of all gates are output the simulator outputs the number of transitions and power dissipated per time-step. Again the transition counts exclude hazards, while the power results include hazards.

Summary Results The summary results are printed after the time-step breakdown of the simulation. The Summary information includes the total number of transitions and total power dissipated over the entire simulation run.

The summary also provides information about some "peaks" discovered during simulation. It shows which gate exhibited the greatest and least transition activity, as well as the gate with the highest/lowest dynamic power dissipation. It also shows the time-step that exhibited the peak power dissipation.

Finally, the summary show the computed the average number of transitions per time-step and the average power dissipated per time-step.

2 Capabilities & Limitations

This section reviews the capabilities and limits of the TSIM simulator.

As described, TSIM is logic-based circuit simulator for estimating dynamic power dissipation. It uses a power modeled similiar to the PFA model described in [**?**]. The power constants for each gate in the circuit can be specified individually. Further, different constants for the rise and fall transitions may be specified.

Rise and fall time delays may be specified for each logic gate. Introduction of rise and fall times would allow the user to consider the effects of intermediate signal switching to the overall power consumption. Simulating a circuit with no delays would produce results which only consider the final (or lasting value) changes occuring in the circuit.

TSIM does provide for some consideration of the effects of hazards on the total power. Although hazards may vary greatly in terms of power dissipated, TSIM only distinguishes between $0 \rightarrow 1 \rightarrow 0$

and 1→0→1 spikes. The benefits of "balancing" input signals may be demonstrated by including the contribution of hazards in the power estimation. Also, TSIM does not propagate hazards to destination gates.

Signal changes are assumed to occur instantaneously at discrete time-steps. TSIM does not provide for consideration of the effects that slowly rising or falling waveforms may have on the total power.

Finally, the quality of any power estimation performed using TSIM will depend on the quality of the input sequence used to simulate the circuit.

3 Reference Manual

3.1 Command Line Arguments

The simulator is started with the command

```
tsim <circuit_file> <input_file> [-o <output_file>]
```

The syntax for the circuit file and the input event sequence file are described below. The `-o` option allows specification of a log file to capture the simulation results. The simulation results are printed to the standard output by default.

3.2 Circuit Description

The simulator borrows GSIM's parser. The circuit description file is divided into three sections: gates, primary inputs and primary outputs; each section begins with the keyword `gates`, `primary inputs` or `primary outputs`.

Gates Each gate in the circuit is described on a separate line in the form

```
name type output <input list> [<delays>] [<power>]
```

Input list is a whitespace separated list of input lines. Delays is an optional field of the form

```
[rise time] [fall time]
```

Power is an optional field of the form

```
[p01 value] [p10 value] [pUP value] [pDOWN value]
```

Like GSIM, the fan-out is limited to 1 for most gates. If a fan-out of more than 1 is required a buffer must be used. The syntax for a buffer is similiar to the other gates, except that it has a fan-out of up to 16 and a fan-in of one.

```
name type input <output list>  [<delays>] [<power>]
```

The types of gates are: {AND, OR, INV, NAND, NOR, XOR, and BUF}.

Primary Inputs and Primary Outputs The gates section is followed by the Primary Input specification. The Primary Input specification is followed by the Primary Output section. In their respective section, each primary input or output is specified on a separate line. The form is

```
io_name  line_name
```

where `io_name` is a description of the primary input/output and `line` is the name of the line used to connect this value to the inputs/outputs of other gates.

3.3 Input event Description

The input event description file specifies the input sequence of each primary input. The input sequence is described as

`input-line (` t_1 `` `v`$_1$ `:` `` t_2 `` `v`$_2$ `:` `` t_3 `` `v`$_3$ `:` `` t_4 `` `v`$_4$ `:` `` ... t_n `` `v`$_n$ `)`

where $t_1 < t_2 < t_3 < t_4 < ... < t_n$ and $v_i \in \{\texttt{1}, \texttt{2}, \texttt{X}\}$. Each primary input's sequence can be broken across several lines The file ends with a line containing the keyword `end` followed by the number of time-steps to be simulated.

Initial values for non primary-input gates may be specified as well.

3.4 Implementation limits

Circuit size = 500 gates

Simulation Length = 32000 time-steps

Gate Fan-in = 16 inputs

References

[1] I. Parulkar and M. Leaper, "GSIM" developed at USC, 1991.

[2] S. Novak, "A Tool to Measure and Analyze Gate Transitions in a Circuit," UCLA Computer Science Master Degree Project Report, 1993

[3] S.R. Powell and P.M. Chau, "Estimating Power Dissipation of VLSI Signal Processing Chips: the PFA Technique," *VLSI Signal Processing, IV*, Eds. H.S. Moscovitz, K. Yao, and R. Jain, IEEE Press, 1990, pp250-259.

A Circuit File Description Language Grammar

```
<circuit>       →  <gates> <primary_inputs> <primary_outputs>

<newline>       →  <NL>
                →  <newline> <NL>

<gates>         →  gates <newline> <gate_list>

<gate_list>     →  <gate>
                →  <gate_list> <gate>

<gate>          →  <gate_name> <gate_spec> <newline>

<gate_name>     →  NAME

<gate_spec>     →  and <oi_list> <delay> <power>
                →  or <oi_list> <delay> <power>
                →  inv <oi_list> <delay> <power>
                →  nand <oi_list> <delay> <power>
                →  nor <oi_list> <delay> <power>
                →  xor <oi_list> <delay> <power>
                →  buf <io_list> <delay> <power>

<delay>         →  <rise> <fall>

<rise>          →  rise NUM
                →

<fall>          →  fall NUM
                →

<power>         →  <p01> <p10> <pUP> <pDOWN>

<p01>           →  p01 NUM
                →

<p10>           →  p10 NUM
                →

<pUP>           →  pUP NUM
                →

<pDOWN>         →  pDOWN NUM
                →
```

```
<oi_list>           →  <output_line> <input_list>

<output_line>       →  NAME

<input_list>        →  <line_name>
                    →  <input_list> <line_name>

<line_name>         →  NAME

<io_list>           →  <input_line> <output_list>

<input_line>        →  NAME

<output_list>       →  <line_name>
                    →  <output_list> <line_name>

<primary_inputs>    →  primary inputs <newline> <pi_list>

<pi_list>           →  <pi_name> <pi_line> <newline>
                    →  <pi_list> <pi_name> <pi_line> <newline>

<pi_name>           →  NAME

<pi_line>           →  NAME

<primary_outputs>   →  primary outputs <newline> <po_list> <end>

<po_list>           →  <po_name> <po_line> <newline>
                    →  <po_list> <po_name> <po_line> <newline>

<po_name>           →  NAME

<po_line>           →  NAME

<end>               →  end <newline>
```

B Example Circuit File

```
/* A circuit description for a 3-bit ripple carry adder */
gates
bufx2 buf x2 x2a x2b
bufy2 buf y2 y2a y2b
bufx1 buf x1 x1a x1b
bufy1 buf y1 y1a y1b

bufd2 buf d2 d2a d2b
bufd1 buf d1 d1a d1b

ga3 nand a3 x3 y3
gb3 inv c3 a3 rise 1 fall 1

ga2 nand a2 x2a y2a rise 1 fall 1
gb2 nor b2 x2b y2b rise 1 fall 1
gd2 inv d2 a2 rise 1 fall 1
ge2 nor e2 d2a b2 rise 1 fall 1
gf2 nand f2 e2 c3 rise 1 fall 1
gg2 inv g2 d2b rise 1 fall 1
gh2 nand c2 g2 f2 rise 1 fall 1

ga1 nand a1 x1a y1a
gb1 nor b1 x1b y1b
gd1 inv d1 a1
ge1 nor e1 d1a b1
gf1 nand f1 e1 c2
gg1 inv g1 d1b
gh1 nand c1 g1 f1 rise 1 fall 1


primary inputs
x3 x3
x2 x2
x1 x1
y3 y3
y2 y2
y1 y1
clk clk
clkbar clkbar

primary outputs

c1 c1
end
```

C Example Input Sequence File

```
x3 (    0 0:     1 1:     2 0:     3 0:     4 0:     5 1:     6 0:
     7 0:     8 0:     9 1)
y3 (    0 0:     1 0:     2 0:     3 1:     4 0:     5 1:     6 0:
     7 0:     8 0:     9 0)
x2 (    0 0:     1 0:     2 0:     3 0:     4 0:     5 0:     6 0:
     7 1:     8 0:     9 1)
y2 (    0 0:     1 0:     2 0:     3 0:     4 0:     5 0:     6 0:
     7 0:     8 0:     9 0)
x1 (    0 0:     1 0:     2 0:     3 0:     4 0:     5 0:     6 0:
     7 0:     8 0:     9 0)
y1 (    0 0:     1 0:     2 0:     3 0:     4 0:     5 0:     6 0:
     7 0:     8 0:     9 0)
end 10
```

D Example Output

```
Simulation performed on Fri Mar 24 13:27:10 1995

Simulating circuit "circuits/sign3"
Input list "inputs/short.3"



line name d2 12  source gate INV   dest gate BUF

  0 0
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
averge trans/event, power/event = 0.090909, 0.090909

line name d1 15  source gate INV   dest gate BUF

  0 0
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
averge trans/event, power/event = 0.090909, 0.090909

line name a3 18  source gate NAND  dest gate INV

  0 1   5 0   6 1
total number of transitions for this gate is 3
total power disipated in this gate is 8 (5)
averge trans/event, power/event = 0.272727, 0.727273

line name c3 21  source gate INV   dest gate NAND

  0 0   5 1   6 0
total number of transitions for this gate is 3
total power disipated in this gate is 3 (0)
averge trans/event, power/event = 0.272727, 0.272727
```

```
line name a2 22  source gate NAND  dest gate INV

  0 1
total number of transitions for this gate is 1
total power disipated in this gate is 4 (3)
averge trans/event, power/event = 0.090909, 0.363636

line name b2 23  source gate NOR   dest gate NOR

  0 1   7 0   8 1   9 0
total number of transitions for this gate is 4
total power disipated in this gate is 4 (0)
averge trans/event, power/event = 0.363636, 0.363636

line name e2 24  source gate NOR   dest gate NAND

  0 0   7 1   8 0   9 1
total number of transitions for this gate is 4
total power disipated in this gate is 4 (0)
averge trans/event, power/event = 0.363636, 0.363636

line name f2 25  source gate NAND  dest gate NAND

  0 1
total number of transitions for this gate is 1
total power disipated in this gate is 6 (5)
averge trans/event, power/event = 0.090909, 0.545455

line name g2 26  source gate INV   dest gate NAND

  0 1
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
averge trans/event, power/event = 0.090909, 0.090909

line name c2 27  source gate NAND  dest gate NAND

  0 0
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
averge trans/event, power/event = 0.090909, 0.090909

line name a1 28  source gate NAND  dest gate INV

  0 1
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
averge trans/event, power/event = 0.090909, 0.090909

line name b1 29  source gate NOR   dest gate NOR

  0 1
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
```

```
averge trans/event, power/event = 0.090909, 0.090909

line name e1 30  source gate NOR   dest gate NAND

  0 0
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
averge trans/event, power/event = 0.090909, 0.090909

line name f1 31  source gate NAND  dest gate NAND

  0 1
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
averge trans/event, power/event = 0.090909, 0.090909

line name g1 32  source gate INV   dest gate NAND

  0 1
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
averge trans/event, power/event = 0.090909, 0.090909

line name c1 33  source gate NAND  dest gate PO

  0 0
total number of transitions for this gate is 1
total power disipated in this gate is 1 (0)
averge trans/event, power/event = 0.090909, 0.090909

the following is the number of transitions per input
event 0 has 16 transitions (power = 16) (0)
event 1 has 0 transitions (power = 1) (1)
event 2 has 0 transitions (power = 1) (1)
event 3 has 0 transitions (power = 1) (1)
event 4 has 0 transitions (power = 1) (1)
event 5 has 2 transitions (power = 3) (1)
event 6 has 2 transitions (power = 3) (1)
event 7 has 2 transitions (power = 4) (2)
event 8 has 2 transitions (power = 4) (2)
event 9 has 2 transitions (power = 5) (3)
event 10 has 0 transitions (power = 0) (0)
Total number of transitions for circuit is 26
Total power dissipation for circuit is 39 (13)

Line with most transitions is b2 with 4 transitions (0.363636)
Line with least transitions is d2 with 1 transitions (0.090909)
Line with most total activity is a3 with 8 power dissipation (0.727273)
Line with least total activity is d2 with 1 power dissipation (0.090909)
Peak power dissipation occurs at timestep 0 (16 power units)
Average number of transitions per input combo is 2.363636
Average power dissipation per input combo is 3.545455
```