Measurement of the Switching Activity of CMOS Digital Circuits at the Gate Level

C. Baena, J. Juan-Chico, M.J. Bellido, P. Ruiz de Clavijo, C.J. Jiménez and M. Valencia[1]

Instituto de Microelectrónica de Sevilla-CNM / Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012-Sevilla, SPAIN
Phone: +34-95-505-66-66; Fax: +34-95-505-66-86;baena@imse.cnm.es

Abstract. Accurate estimation of switching activity is very important in digital circuits. In this paper we present a comparison between the evaluation of the switching activity calculated using logic (Verilog) and electrical (HSPICE) simulators. We also study how the variation on the delay model (min, typ, max) and parasitic effects affect the number of transitions in the circuit. Results show a variable and significant overestimation of this measurement using logic simulators even when including postlayout effects. Furthermore, we show the contribution of glitches to the overall switching activity, giving that the treatment of glitches in conventional logic simulators is the main cause of switching activity overestimation.

1 Introduction

Evaluating the switching activity in CMOS digital circuits is a key point to calculate its power consumption [1, 2]. In mixed-signal circuits, switching activity of the digital part creates a switching noise that is transferred to the analog part [3, 4, 5]. Furthermore, as digital circuits become faster and larger, the influence of glitches in the switching activity grows because there are more and more input collisions [6, 7, 8, 9]. Thus, evaluation of switching activity is today a major topic in the design process of both pure digital, and mixed-signal integrated circuits.

Measuring the switching activity in a digital circuit concerns three important questions: The first one is referred to determining the representative input stimuli that must be obtained in order to get an accurate estimation of the switching activity. The second one is concerned to the timing simulator. In timing simulation of digital circuits, standard gate-level logic simulators (like Verilog [10]) are able to handle very large circuits and they are commonly used by circuit designers. Otherwise, accurate evaluation of the switching activity is possible by using electrical simulators (like HSPICE [11]), but these simulators are limited to rather small circuits, they spend lots of computational resources, and they are not used in a typical digital design flow. The third issue focuses on the origin of the logic transitions at the nodes of the circuit. Input changes cause two types of logic transitions: First, proper operation generates

1. This work has been sponsored by MCYT of Spain under Projects TIC2000-1350 and TIC2001/2283

functional transitions and second, the generation and propagation of spurious transitory signal pulses (glitches) cause non-functional transitions.

The basic method to estimate the power consumption at logic levels consists in obtaining the final value of it by summing up the power contribution each node has every time it makes a transition. So, it is necessary to calculate the total number of transitions in the circuit besides the use of a power model to estimate the consumption at each node. Tools that use this method obtain an overestimation in the power consumption. In order to correct this result, new power models are proposed in [12,13].

In this communication, we demonstrate that the switching activity can be greatly overestimated when calculated with conventional logic simulators like Verilog. This overestimation is mainly due to an inaccurate propagation and elimination of glitches, which happens regardless the model used among those provided by the foundry, or the inclusion of postlayout information.

This contribution is organized as follows: The method used for switching activity computation applied to ISCAS'85 benchmark circuits is summarized in Section 2. Simulation results are presented and analysed in the section 3. Finally, in section 4, we draw some conclusions on switching activity evaluation.

2 Switching Activity Measurement Procedure

In this section, we describe a method to obtain the switching activity in a circuit. To illustrate the method the ISCAS´85 benchmark circuits are considered [14].

As said above, we compare different measures of switching activity of a circuit using two kinds of simulators, logic and electrical. The procedures are very similar in all of the cases and a scheme of them are presented in Fig 1. We start with the description of the circuit, provided by the ISCAS´85 benchmarks document. This description must be translated to another format suitable for the design environment Design FrameWork II (DFWII) in our case [15].

To do this translation, a software parser has been written using the PERL language [16]. The parser takes the original description of the circuit as supplied with the set of benchmarks, and produces the corresponding Verilog netlist. The parser also needs a simple mapping library which assigns the right cell for the current technology to each logic operator. In our case, circuits are implemented in a CMOS 0.35 μm technology. Once the circuit description is loaded in DFWII, we can generate HSPICE netlists in order to do electrical simulation, or run a Verilog logic simulation.

At this point, we follow three different paths, but before that, we need to study which and how many vectors of test must be applied to get an accurate and realistic evaluation of the switching activity.

Switching activity inside a circuit is highly input-pattern dependent [17], thus, simulation results are directly related to the specific input patterns used. The two main objectives when selecting a set of input patterns are to generate an "average" switching activity and to use a number of patterns that is small enough in order to limit the cost in computational resources. The method described in the following points accomplish both objectives:

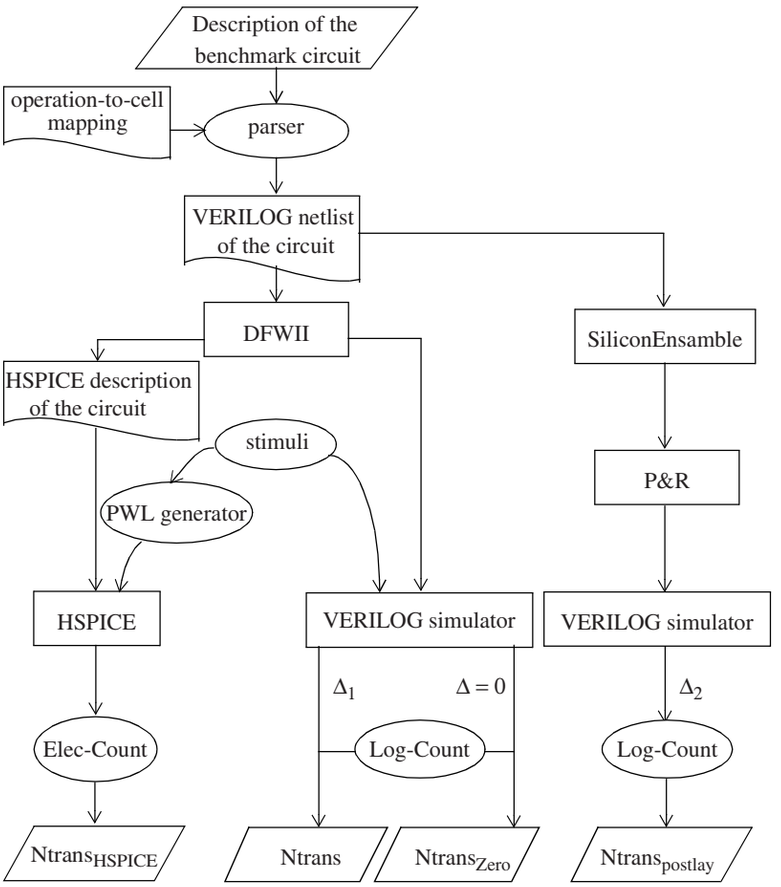

Figure 1 . A-scheme-of-the-method-for-switching-activity-computation

- First we run a Verilog simulation on 1000 random input patterns and get the number of transitions in the whole circuit. We have checked that for such a number of random patterns, similar switching activity is obtained (within 2%) for any set of patterns, thus, the result is an average measure of the switching activity.
- Several 1000 random vectors simulations are run, and the number of transitions per input vector calculated. The mean value of all measurements is taken as a standard value of the circuit's switching activity.
- Then, we simulate the circuit several times using only 50 random test vectors in order to find a set of input patterns that generated a number of transitions per stimuli within the 3% of the mean value previously determined. Thus, these 50 ran-

dom input patterns represent a generic input case and they are not expressly selected to get good results.

These set of vectors is then used to compare the switching activity using logic and electrical simulators. In this way, we significantly reduce the computational time when running electrical simulations on medium-large sized circuits.

After the selection of the stimuli we start with the logic simulation using the timing information of each cell and the Verilog standard simulator, which uses an inertial delay model. From the results of this simulation the global number of transitions (i.e., all of the nodes in the circuit, Ntrans) is computed. To do that, we have developed a program, Log-Count, that scans the Verilog output and returns the number of logic transitions in each node, as well as the total number of transitions.

Additionally, this same procedure is done using a zero delay model for each gate in the circuit. The results of this simulation provide a measurement of the minimum switching activity required by the logic functionality of the circuit. No glitch effects are considered. We will note this result $Ntrans_{Zero}$.

Another procedure considered to measure the switching activity is based on HSPICE simulation. This result is accurate and will be used as a reference in our comparison. For this purpose, we need to translate the same stimuli used before to piece-wise-linear functions for each input in the circuit. Manually translating the input vectors to PWL format is not feasible and this functionality cannot be found in the design environment. Hence, a general-purpose program that translates Verilog vectors to SPICE PWL format has been developed to generate the appropriate stimuli. These stimuli joined to the HSPICE netlist are all the necessary data for the HSPICE simulator. The files generated with these simulations are the input to a software program, Elec-Count. This program is dedicated to count the number of times each node in the circuit crosses the Vdd/2 voltage. The final result is the switching activity for the whole circuit and is noted as $Ntrans_{HSPICE}$.

Finally, we want to consider the effects of routing on each node in order to have a more realistic description of the circuit. To do that, we used Silicon Ensemble tool and following an automatic process we get the layout of the circuit. From this view we obtain a new set of delay values for each node which include the wire effects and new capacities values. After that, we run another logic simulation using this new information and, following the same procedure we used before for the logic simulation, we obtain the number of logic transitions in each node, as well as the total number of transitions and called it $Ntrans_{postlay}$.

3 Results

In this section, the whole method will be applied to nine of ISCAS'85 benchmark circuits, in order to compare the switching activity obtained with logic simulation includ-

ing pre and postlayout against the “intrinsic” switching activity (zero delay) and the “accurate” value obtained with HSPICE.

Table 1: ISCAS85 benchmark circuits

	no. of inputs	**no. of outputs**	**total gates**
c432	36	7	160
c499	41	32	202
c880	60	26	383
c1355	41	32	546
c1908	33	25	880
c2670	233	140	1193
c3540	50	22	1669
c5315	178	123	2307
c6288	32	32	2416
c7552	207	108	3512

In Table 1, we list the circuits selected and their complexity. For each one, the number of transitions for the simulation of 50 test vectors using Verilog considering inertial delay model (minimum, typical and maximum values) for each gate ($Ntrans_{min}$, $Ntrans_{typ}$, $Ntrans_{max}$), and Zero delay model ($Ntrans_{Zero}$) is shown in Table 2. The result of the simulation under the same conditions using HSPICE ($Ntrans_{HSPICE}$) is also included. As can be seen in the table, if we compare minimum and maximum delays to the typical, the differences in the total number of transitions are less than 1%. Furthermore, in some cases, the number of transitions using minimum value for the delay is bigger than the result using maximum delays but in other cases, it happens the opposite.

In Table 3, we represent the relative error between $Ntrans_{typ}$ and $Ntrans_{HSPICE}$ ($\%err_{\Delta 1\text{-}Hsp}$) and the relative error between $Ntrans_{Zero}$ and $Ntrans_{HSPICE}$ ($\%err_{Zero\text{-}Hsp}$). As can be seen, in each example, the relative error between $Ntrans_{typ}$ and $Ntrans_{HSPICE}$ is very different. It varies between 3% for c880 and 115% for c6288. Another important conclusion can be drawn from Table 3: when we compare $Ntrans_{Zero}$ to $Ntrans_{HSPICE}$, we observe that a high contribution to the switching activity is due to the glitches generated and propagated inside the circuit. In all of the cases, the contribution of the glitches is between 20% and 50%, except for the case of c6288 for which this contribution is even greater than 77%. This is mainly due to the size of this circuit and specially, the high number of levels (123, [13]) the circuit has.

Table 2: Number of transitions using 50 input vectors

Circuit	$Ntrans_{HSPICE}$	$Ntrans_{Zero}$	$Ntrans_{min}$	$Ntrans_{typ}$	$Ntrans_{max}$
c432	4517	3637	4719	4735	4753
c499	6196	4868	6423	6417	6421
c880	11033	7707	11353	11337	11331
c1355	13960	10420	16318	16190	15990
c1908	25873	18465	32393	32411	32441
c2670	38655	26279	45095	44979	45029
c3540	52303	28799	61044	60920	60376
c5315	79803	48899	100589	100295	100327
c6288	194784	44378	421909	418815	416729
c7552	144535	76315	174682	174292	174062

Table 3: Relative errors with respect to $Ntrans_{HSPICE}$

Circuit	$\%err_{\Delta 1\text{-}Hsp}$	$\%err_{Zero\text{-}Hsp}$
c432	4.8	19.5
c499	3.5	21.4
c880	2.7	30
c1355	16	25.3
c1908	25.2	28.6
c2670	16.3	32
c3540	16.5	45
c5315	25.7	38.7
c6288	115	77.2
c7552	20.6	47.2

Table 4: Number of transitions pre-postlayout

Circuit	$Ntrans_{min}$			$Ntrans_{typ}$			$Ntrans_{max}$		
	pre	**post**	**%**	**pre**	**post**	**%**	**pre**	**post**	**%**
c432	4719	4499	4.7	4735	4505	4.8	4753	4511	5.1
c499	6423	5937	7.6	6417	6260	2.4	6421	5977	6.9
c880	11353	11495	-1.2	11337	11493	-0.1	11331	11485	-1.3
c1355	16318	15447	5.3	16190	15383	2.2	15990	15297	4.3
c1908	32393	32976	-1.8	32411	32880	-1.4	32441	32782	-1
c2670	45095	50122	-11.1	44979	50106	-11.4	45029	50072	-11.2
c3540	61044	61465	-0.7	60920	60979	-1	60376	61005	-1
c5315	100589	108587	-7.9	100295	108441	-8.1	100327	108015	-7.7
c6288	421909	365702	13.3	418815	362870	13.3	416729	361019	12.1
c7552	174682	166908	4.4	174292	166806	4.3	174062	166709	4.2

In order to be more realistic doing logic simulation, in the Table 4 we show the number of transitions obtained after considering the parasitic effects in each node using minimum, typical and maximum delay for the gate and we compare them to the results obtained previously, before layout. For the three values of the delay we can say that the relative deviation between pre and postlayout are not very significant, less than 10% except in the case of c2670 (11%) and the case of c6288 (13%). As we said for table 2, in some of the circuits the error is positive and in other examples is negative. After analysing the results presented in tables 2 and 4, we can conclude that the post-layout information does not improves in general the computation of the switching activity when using logic simulators.

In table 5, we present the relative error in the number of transitions after a logic simulation having used postlayout typical delays versus HSPICE simulation[1]. Generic conclusions are similar we did when we compared the results with the prelayout logic simulation (Table 3). The more simple circuits (c432, c499 and c880) have a relative error is really close to the HSPICE value, but in others the difference can reach the 86% as the case of c6288. It is important to notice that there are cases in which the postlayout results are further to the reality. In effect, although c6288 decreases its relative error from 115% (prelayout) to 86% (postlayout), in the case of c2670 the change is from 16.3% (pre) to 29.6% (post) making postlayout worse than prelayout result. Then, postlayout values do not guarantee an accurate measurement of the switching activity.

1. Unfortunately, for this technology we haven't had available the necessary data to run postlayout electrical simulation.

Table 5: Relative errors number of transitions postlayout vs HSPICE

Circuit	$\%err_{postlay\text{-}Hsp}$
c432	-0.2
c499	1
c880	4.2
c1355	10.2
c1908	27.1
c2670	29.6
c3540	16.6
c5315	35.9
c6288	86.3
c7552	15.4

The results obtained in the different tables make us conclude that Verilog simulation is not an appropriate way to measure the switching activity in a circuit. Specially for two main reasons, the first one because the relative error can be very high in some cases, in the case of the circuit c6288, the result is not valid at all; and the second idea to emphasise is the great variation in the percentage among the different examples that makes the results for the switching activity not reliable in comparison to HSPICE.

From these results, it can be concluded that the deviation in the power consumption estimation of a circuit obtained from logic simulators is derived from the overestimation in the switching activity. So, the way to improve this result in this kind of tools is getting a more accurate switching activity estimation through the use of new delay models with a better treatment of glitch generation and propagation [18].

Finally, in Table 6 we show the approximate of CPU time spent in each simulation. From these results, we can point out, the well known conclusion, that electrical simulators are limited to rather small circuits because their cost is high in computational resources and CPU time. These kind of tools are restricted to critical parts of a digital circuit.

Table 6: Simulation CPU time

	CPU time (s) VERILOG simulation	**CPU time (s) HSPICE simulation**
c432	6.4	2714
c499	7.2	8087
c880	8.3	15240
c1355	8.3	28411
c1908	9.9	83989
c2670	16.7	260106
c3540	14.8	722935
c5315	24.1	1518849
c6288	34.2	836644
c7552	31.3	4577727

4 Conclusions

Some results of switching activity estimation in digital CMOS circuits when they are measured using standard simulators has been presented. In order to be impartial, benchmark circuits has been selected as circuits under test, and random medium length stimuli have been applied.

Generally, activity due to glitches (i.e., $Ntrans_{Zero}$) has a remarkable contribution (form 19% for c432 to 77% for c6288) to the overall switching activity. Thus, it can be emphasized the great importance of adequately handling the glitch generation and propagation effects by timing simulators.

When the results of standard logic simulation (Verilog) are compared to accurate data (HSPICE) (i.e. Ntrans vs. $Ntrans_{HSPICE}$), it is observed that the overestimation of the Ntrans varies appreciably, i.e. from 3% for c880 to 115% for c6288. That overestimation persists even when minimum and maximum values are used and postlayout effects are taking into account. The greatest variation between min/max is 1% and between pre and postlayout is 13%. Both deviations are much smaller than the average value of the overestimation. Hence, logic simulators are neither precise nor reliable at measuring switching activity. It is due to the fact that they are not accurate at simulating glitch propagation.

References

1. A. Ghosh, S. Devadas, K. Keutzer, and J. White: “Estimation of Average Switching Activity in Combinational and Sequential Circuits”. Proc. 29th Design Automation Conference, pp. 253–259. June 1992.
2. J. Monteiro, S. Devadas, and B. Lin: “A Methodology for Efficient Estimation of Switching Activity in Sequential Logic Circuits”. Proc. 31th Design Automation Conference, pp. 12–17. Jun. 1994.
3. X. Aragonès, J. L. González and A. Rubio, “Analysis and Solutions for Switching Noise Coupling in Mixed-Signal ICs”. Kluwer Academic Publishers, 1999.
4. D.J. Allstot, S-H. Chee and M. Shrivastawa, “Folded Source-Coupled Logic vs. CMOS Static Logic for Low-Noise Mixed-Signal ICs”, IEEE Trans. Circuits and Systems I, Vol. 40, pp. 553–563, Sept. 1993.
5. Y. Tsividis, “Mixed Analog-Digital VLSI Design and Technology”. McGraw-Hill, 1995.
6. E. Melcher, W. Röthig, M. Dana. “Multiple Input Transitions in CMOS Gates”. Microprocessing and Microprogramming 35 (1992) pp. 683–690. North Holland.
7. C. Metra, M. Favalli, B. Riccò. “Glitch power dissipation model”. In Proc. PATMOS'95. pp. 175–189
8. M. Eisele, J. Berthold. “Dynamic Gate Delay Modeling for Accurate Estimation of Glitch Power at Logic Level”. In Proc. PATMOS'95. pp. 190–201. 1995.
9. M.J. Bellido, J. Juan-Chico, A.J. Acosta, M. Valencia and J.L. Huertas: “Logical modeling of delay degradation effect in static CMOS gates”. IEE Proc. Circuits Devices Sist., Vol. 147, Nº2, pp. 107–117. April 2000.
10. “Verilog-XL Reference Manual”. Cadence Design Framework II V. 4.43, 1999.
11. “Star-Hspice Manual”. Avant! Corporation, June 1999.
12. G. Jochens, L. Kruse and W. Nebel: “Application of toggle-based power estimation to module characterization”. In Proc. PATMOS'97. pp. 161–170. 1997.
13. D. Rabe, G. Jochens, L. Kruse and W. Nebel. “Power-simulation of cell based ASICs: accuracy-and performance trade-offs”. In Proc. DATE'98. pp. 356–361. 1998.
14. D. Bryan and H. Fujiwara: “A neutral netlist of 10 Combinational Benchmark Circuits and a target Translator in Fortran”. IEEE International Symposium on Circuits and Systems, pp. 695–698. June, 1985.
15. Design Framework II. Version 4.4.3. Cadence, 1999.
16. L. Wall, T. Christiansen and R. L. Schwartz, “Programming PERL”. O'Reilly, 1996.
17. F. N. Najm: “A Survey of Power Estimation Techniques in VLSI Circuits”. IEEE Transactions on VLSI Systems, Vol. 2, num. 4, pp. 446–455. Dec. 1994.
18. J. Juan-Chico, M.J. Bellido, P. Ruiz-de-Clavijo, A.J. Acosta and M. Valencia: “Gate-Level Modeling of the Delay Degradation Effect”. Proc. 15th DCIS, pp. 537–542, Nov. 2000.