Power Consumption in XOR-Based Circuits

Yibin Ye Kaushik Roy

School of Elect. and Comp. Eng., Purdue University
West Lafayette,IN 47907, USA
email: <ye, kaushik>@ecn.purdue.edu

Rolf Drechsler

Inst. of Comp. Science, Albert-Ludwigs-University
79110 Freiburg im Breisgau, Germany
email: drechsle@informatik.uni-freiburg.de

Abstract— The use of XOR gates has shown several advantages in modern circuit design, e.g. smaller representation size and better testability. In this paper we consider power consumption in XOR dominated circuits and compare such designs with traditional AND/OR logic. We investigate the suitability of using different delay models such as unit delay, fanout delay, and random delay in power estimation of XOR dominated logic. Due to different possible implementations of XOR gate, we model the XOR gate as a basic gate and a complex static CMOS gate, respectively. Power dissipation due to (charging and discharging) internal node capacitances is also considered.

I. Introduction

Due to the increasing use of portable systems, e.g., notebook computers, personal digital assistants, and cellular phones, power is rapidly becoming as important a parameter as area and speed in the design of such systems [1]. At the system level, increasing complexity and higher density on boards result in previously unseen levels of power dissipation. At the IC level, larger die sizes and increasing device density along with higher clock frequencies lead to the undesirable effect of escalating power dissipation to levels that can no longer be ignored. In case of mobile CPUs, both the limitation of battery size as well as unavailability of special cooling mechanisms are the two major constraints requiring power reduction. Desktops, on the other hand, due to increased functionality, dissipate such levels of power that they require new thermal solutions. Hence, there is clearly a need for new technologies and design techniques that would achieve ultra low power dissipation under performance and cost constraints.

Recently, there is a growing interest in XOR based design styles (see e.g. [7]). The resulting realizations are often smaller in size for large classes of circuits, e.g., arithmetic circuits, error correcting circuits, and circuits for tele-communication. Additionally, XOR based circuits have good testability properties [3] - at least if they are restricted to specific subclasses of AND/XOR forms - and thus are well suited for design for testability. The circuits can easily be mapped to Field Programmable Gate Arrays (FPGA's), where XOR gates, e.g. in table-lookup FPGA's, do not cause any extra cost in term of chip area. While circuit size, performance, and testability of XOR based circuits have been studied, none of the approaches presented so far has taken power consumption into account.

In this paper we discuss methods of modeling the power consumption in XOR dominated circuits. It turns out that the result significantly depends on the delay models and the implementation styles for XOR gates. We use different delay models: zero delay, unit delay, fanout delay, and random delay subject to a given distribution. Hence, power dissipation due to glitches can be included in our analysis. Glitches are usually more significant in XOR dominated circuits than in AND/OR circuits. For a specific XOR based design style we report experimental results that underline our theoretical considerations. We first derive AND/OR logic from BDD representations. An XOR-based synthesis technique that we described in [9] is then applied. The resultant circuits are XOR-dominated and have smaller size. We investigate the behavior of the original AND/OR circuits and the synthesized XOR dominated circuits with respect to power dissipation.

II. Power estimation

During design of VLSI circuits, the designers attempt to accurately estimate the silicon area required and the expected performance before the circuit goes into fabrication. However, with the requirements for low-power, the designers are faced with the added burden of qualifying their designs with respect to power dissipation.

Power dissipation in CMOS circuits is mainly due to three components of current: (i) the *leakage current*, (ii) the *short circuit* or *rush through* current due to the presence of current carrying path from the supply voltage to ground when certain PMOS and NMOS transistors are simultaneously ON for a short period due the signal transitions at the input to the logic gates, and (iii) the *switching current* due to charging and discharging of the load capacitance. Of the three sources of power dissipation, the switching current component is by far the most dominant in the current day technology.

Ignoring the internal capacitances of logic gates, the average power dissipation for a logic gate due to charging and discharging of load capacitance is given by

$$P_{avg} = \frac{1}{2} V_{dd}^2 C f \tag{1}$$

where V_{dd} is the supply voltage, C is the load capacitance of the gate under consideration, and f is the frequency of operation. For aperiodic signals, the frequency of operation can be estimated by the average number of signal transitions per unit time and is represented by *signal activity, A*. Assuming a constant supply voltage, accurate estimate of power dissipation involves determination of signal activities and capacitances at circuit nodes.

Power estimation involves accurately estimating switching activity and the capacitances at the internal nodes of a circuit. Estimation of each of these quantities at the architectural or behavioral level is quite difficult, while they can be more accurately estimated at the transistor circuit level. Glitches occur due to different delays through two different paths of the circuit. Glitches introduce unwanted signal transitions, and hence, dissipate power. If one can accurately predict the glitching activity at various nodes of a circuit, the power dissipation due to the glitches can be accurately estimated.

The problem of determining when and how often transitions occur at a node in a digital circuit is difficult because they depend on the applied input vectors and the sequence in which they are applied. Hence, the easiest solution to the problem of estimation of power dissipation is to use circuit simulation. Based on a given set of inputs, the power supply current can be monitored to determine the average power dissipation. The major problem with this approach is that circuit simulation is too slow for large circuits. Besides being slow, the technique is strongly input pattern dependent. Hence, pattern independent probabilistic techniques are required to quickly estimate the average number of transitions per circuit node. There are two main techniques to accurately estimate switching activities in CMOS circuits – probabilistic and statistical approaches. Different delay models can be easily incorporated in statistical techniques and hence, can more accurately estimate power in CMOS circuits. In all our analysis, we use a statistical power estimator described in [2]. In the following section we briefly review the models that will be considered and discuss their influence on XOR design styles.

A. Delay models

As mentioned earlier, minor delay model inaccuracies may lead to large errors in estimated activity. Therefore, delay models are crucial to the statistical estimation of activity. Probabilistic delay models used in the estimation will be introduced to capture the uncertainty of gate delays. Based on the probabilistic delay models, we will generalize the Monte Carlo approach.

In the design phase, a designer is faced with different sources of uncertainty that affect the delays of a circuit. These sources can be grouped into two classes: systematic and random [4]. The systematic class includes approximations made to simplify the model for improving simulation time, approximations made to estimate device and interconnect parasitics prior to layout, and uncertainty in the final process center and distribution when design proceeds in parallel with process development. On the other hand, the random class includes uncontrolled variations in photo-lithography, die to die variations, wafer to wafer variations, lot to lot variations, operating temperature, power supply voltage, etc. In [6], it has been shown that a circuit node where two re-convergent paths with different delays meet may have a large number of spurious transitions. However, even in a tree-structured circuit with balanced paths (without re-convergent fanout) there can be a large number of spurious transitions due to slight variations in delays. These variations can be caused by any of the above sources of uncertainty.

Example 1 *Let us consider an XOR circuit example as shown in Fig. 1. All gates are assumed to have the same delay. Because the tree has perfectly balanced paths, there are no glitches at all. The final output has normalized activity of* 0.5 *when all the primary inputs are assumed to be synchronous and have activity of* 0.5*. However, due to sources of uncertainty, the gate delays may have variations. As a result, glitches do occur and the values of activities at individual nodes change. The delay of a logic gate can be modeled as the sum of the inertial delay and the transport delay, where the inertial delay is the intrinsic delay of a logic gate and the transport delay is due to the output load. We performed simulations, where the inertial delays are assumed to be half of the values of transport delays. By this, the final output normalized activity becomes* 1.30 *rather than* 0.5*. Also note that XOR circuits are prone to glitching activity.*

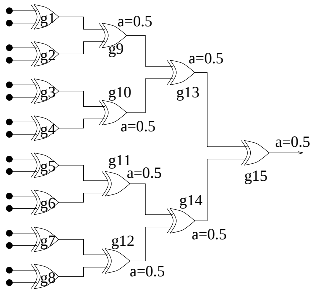

Fig. 1. A circuit with unit delays

In this paper, we use four delay models: zero delay, unit delay, fanout delay (gate delay is proportional to fanout) and random delay, to investigate the power dissipation in XOR-dominated circuits. For the random delay model, we choose transport delay (d) with inertial delay (d_I). However, it should be noted that the technique is not restricted to such a delay model. The idea is to model the parameters of chosen delay models as random variables in order to capture the probabilistic behavior of gate delays. The transport delay is modeled as a random variable of truncated normal distribution with mean μ_d and standard deviation σ_d. The mean is the nominal value of transport delay d and the deviation is either assigned by users or determined by feedback from the fabricated chips. Moreover, if a random delay is less than a minimum value Min, it is discarded since in real circuits it must be larger than some positive value. Similarly, if a random delay is greater than a maximum value Max, it is truncated since it can be considered as a delay fault.

B. Statistical estimation

Recall that in Monte Carlo based technique the primary input patterns are generated conforming to a given activity and probability of the input signals. In a more abstract view point, we can think of activity a at a node as a function of primary input vectors PI. Each component of PI is a stochastic process [5]. Therefore, a is also a stochastic process and can be expressed as follows,

$$\mathbf{a} = F(\mathbf{PI}). \tag{2}$$

Monte Carlo based techniques to estimate the expected value of a, $E(\mathbf{a})$ can be easily applied. However, what is missing in this approach is the information about the delay. In other words, the delays of the logic gates of the circuit are assumed to be some constants (deterministic). Now assume that gate delays are not deterministic and each gate delay can be represented by a random variable d_i. If D is a random vector consisting of all the random variables of gate delays, $\mathbf{a}$ can be represented as follows:

$$\mathbf{a} = F(\mathbf{PI}, D). \tag{3}$$

Therefore, when applying Monte Carlo based techniques to estimating $F(\mathbf{PI}, D)$, delays are modeled as random variables and should be generated from time to time throughout the simulation. The rationale behind this is that whenever we generate a new set of delays, they correspond to another die or even the same die but with different operating conditions such as temperature and power supply voltage.

C. Power consumption in XOR-based circuits

Let us consider power consumption in XOR dominated logic. A basic property of the XOR gate is that all inputs are sensitizing inputs. Unlike AND or OR gates, there does not exist controlling input value for XOR gates. Hence, they are more testable. However, this property can be a drawback from power dissipation point of view. A switching transition at an input of an XOR gate always results in a transition at its output, unless another input is also having a transition simultaneously (which also

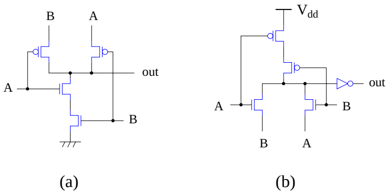

Fig. 2. Implementations of XOR gate in a style mixing pass logic and static CMOS (size is comparable to a two-input NAND or NOR gate)

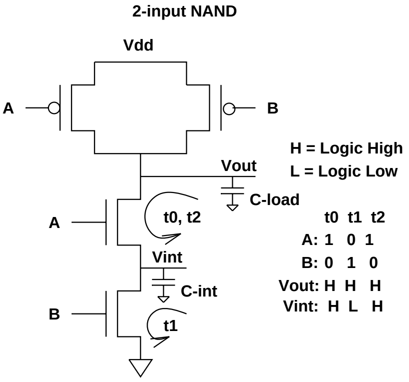

Fig. 3. Application of three consecutive input vectors results in the charging and discharging of internal node V_{int} in a static CMOS NAND gate

explains why the delay model is particularly important for XOR dominated circuits, as has been already demonstrated by Example 1). In general switching activity is higher for XOR heavy circuits.

Furthermore it should be pointed out that the power dissipation depends on the actual implementation of the logic gates. It makes a large difference when XOR gates are modeled as basic gates or complex ones. The 2-input static CMOS XOR gate has 8 transistors. However, other design styles such as complementary pass logic (CPL) design, in which 4 transistors are also popular. Fig. 2 shows two implementations of XOR gate in which pass logic and static CMOS styles are mixed. The size of XOR gate in both implementations is comparable to a two-input NAND or NOR gate. Therefore, in some cases it is still reasonable to model the XOR gates as basic ones.

When complex gates are used in a design, we should not only consider the power consumption in charging and discharging the capacitance at the output nodes, but also the internal capacitances within a logic gate, especially for large fanin complex logic gates. Fig. 3 shows the application of three consecutive input vectors to a static CMOS NAND gate. At time t_0, let us assume $A = 1$ and $B = 0$. Both the output node V_{out} and the internal node V_{int} are charged to a *HIGH*. At t_1, $A = 0$ and $B = 1$ is applied to the gate. The internal node V_{int} is then discharged to *LOW* while the output node V_{out} remains *HIGH*. When the third input vector $A = 1$ and $B = 0$ is applied at t_2, the internal node V_{int} is again charged *HIGH*. Thus, the internal capacitance is charged and discharged in the three clock cycles even though there is no transition at the output node.

III. Experimental Results

In this section we report experimental results that have been obtained by applying statistical power estimation method on circuits mapped from BDDs and XORDDs. The synthesized circuits considered in our experiments are derived from decision diagrams. We consider circuits resulting from a one-to-one mapping of reduced XORDDs (as introduced in [9]) in the SIS environment [8].

For all experiments we used benchmarks from LGSynth91. We first consider every node in BDD and XORDD as a basic logic gate. The experimental results are summarized in Table I, where *# nodes* is the number of nodes for the BDD and XORDD. As expected, the size of the synthesized XORDDs are smaller than the corresponding original BDDs in most cases. We have used three delay models in the simulations: *unit delay*, *fanout delay* and *random delay* given in columns *unit*, *fanout* and *random*, respectively. For simplicity, both signal probability and switching activity of primary inputs are assumed to be 0.5. The estimated power given in the table has been normalized as switched capacitance per clock cycle, which is independent of the clock frequency. In our computation, the node capacitance C in Eq. 1 equals the number of the fanout. Hence, the load capacitance for driving one gate is considered as one unit of capacitance, which implicitly assumes that every gate is considered as a basic gate. For most circuits considered, the estimated power of XORDD's is lower than BDD's. There is 5% to 15% difference in the estimated power due to the three different delay models.

When we map every node in XORDDs into 2-input logic gates, the size (number of logic gates) of the resultant circuits increases significantly. A large number of 2-input XOR gates are then generated. We consider that every gate is implemented in static CMOS, hence a 2-input XOR is assigned 2 units of capacitance while a 2-input NAND or NOR gate is assigned 1 unit of capacitance. Results after splitting the nodes are summarized in Table II. The second column gives the power under *zero* delay model. In the third column, we give the power dissipation including the contribution of internal nodes within a logic gate. We have assumed the total internal capacitances is always 1/2 unit capacitance. For most of the circuits, the internal power accounts for 15% to 20% of the total power dissipation, which is quite significant considering the value of internal capacitance we have assumed. Column 5 and 7 gives the power under *unit* and *random* delay model, respectively. Their ratio to the zero delay power is also shown in column 6 and 8. There are dramatic increases in the estimated power, mainly due to the large number of glitches. Chances of simultaneous transitions at the inputs also decrease under non-zero delay models. Note that simultaneous transitions to the two inputs of an XOR gate always prevent the output node from switching.

IV. Conclusions

While several XOR-based synthesis styles have been presented in the last few years, in this paper we discussed for the first time the modeling of power consumption of XOR dominated circuits. We considered several delay models. Experimental results show that accurate delay models are critical in estimating the circuit power dissipation.

Due to different possible implementations styles for XOR gates, we consider XOR gates as basic gates as well as complex gates. For large classes of circuits, size can be greatly reduced by using XOR-based synthesis techniques. Results in our experiments show that the the power consumption is also reduced compared to AND/OR logic, as long as the XOR gates are considered as basic gates. When XOR gates are implemented as complex static CMOS gates, the power consumption is much larger. Power consumption due to (charging and discharging) internal node capacitances in complex XOR gates is significant.

TABLE I
Average power when XOR as a basic gate

Circuit	BDD Circuits				XORDD (XOR basic)			
	# nodes	unit	fanout	random	# nodes	unit	fanout	random
apex6	795	1060	1108	1123	739	744	759	777
apex7	456	632	716	723	466	423	428	432
ALU2	171	238	251	260	132	160	164	167
ALU4	711	1777	2139	2160	506	678	743	768
b1	12	8.93	8.93	9.43	11	8.93	8.93	9.43
b9	166	204	217	220	176	173	176	178
c8	114	110	122	123	104	92.1	92.4	92.6
count	111	111	121	121	96	94.1	96.2	96.2
dalu	1440	1865	1971	2007	1375	1698	1792	1823
frg1	105	237	290	324	144	138	145	146
frg2	2349	3789	3825	3879	2382	2125	2211	2293
i1	84	84.1	86.2	87.1	76	53.1	54.3	54.8
i2	207	117	118	119	208	118	120	122
rd53	20	27.9	28.7	29.0	16	22.6	23.5	23.9
rd73	36	58.7	59.6	59.9	35	52.1	53.1	53.7
rd84	53	84.1	86.2	87.1	41	64.1	66.4	67.1
t481	27	41.2	42.5	44.9	23	25.7	26.5	28.5
ttt2	165	185	188	190	145	134	136	137
vda	546	801	820	827	518	730	740	745
x2	46	50.9	52.3	52.8	47	46.6	48.0	48.2
x3	810	1005	1042	1056	718	729	750	758
x4	595	757	768	771	588	576	588	590

TABLE II
Average power of XORDD circuits under different delay models (only 2-input static CMOS gates are used)

	XORDD mapped into 2-input static CMOS gates						
Circuit	zero-delay			unit-delay		random-delay	
	power	power	w/internal	power	%	power	%
apex6	461	558	17.6%	1208	262%	1343	291%
apex7	235	271	13.4%	610	260%	643	274%
ALU2	114	144	20.4%	385	336%	460	402%
ALU4	456	569	19.8%	1671	366%	2053	450%
b1	4.63	5.23	11.4%	10.2	220%	10.7	231%
b9	92.5	110	15.5%	252	273%	265	286%
c8	60.7	71.4	15.0%	140	231%	147	242%
count	58.0	68.2	14.9%	136	235%	144	248%
dalu	978	1170	16.5%	2783	285%	3169	324%
frg1	94.8	115	17.2%	349	368%	397	418%
i1	24.5	26.4	6.99%	53.9	220%	53.9	220%
i2	60.9	63.4	3.89%	129	212%	131	214%
rd53	16.7	20.9	20.1%	53.3	320%	71.6	429%
rd73	42.1	53.4	21.1%	169	402%	227	538%
rd84	54.9	69.4	20.9%	290	528%	429	781%
t481	14.9	18.4	19.3%	48.2	323%	59.7	401%
vda	452	557	18.8%	1407	311%	1528	338%
x2	23.5	27.2	13.6%	63.1	269%	67.2	286%
x3	463	561	17.6%	1140	246%	1303	281%
x4	392	480	18.3%	1189	303%	1382	352%

V. Acknowledgment

This research was supported by DARPA (F33615-95-C-1625) and NSF CAREER award (9501869-MIP).

References

[1] A.P. Chandrakashan, S. Sheng, and R. Brodersen, "Low Power CMOS Digital Design," *IEEE Trans. on Solid-State Circuits.*, vol. 27, No. 4, pp. 473-483, 1992.

[2] T.-L. Chou and K. Roy, "Statistical Estimation of Combinational and Sequential CMOS Digital Circuit Activity Considering Uncertainty of Gate Delays," *1997 Asia & South Pacific Design Automation Conf.*, 1997.

[3] R. Drechsler, et al., "Testability of AND/XOR Expressions," *IEEE European Design & Test Conference*, pp. 548-553, 1997.

[4] S. G. Duvall, "A Practical Methodology for the Statistical Design of Complex Logic Products for Performance." *IEEE Trans. on VLSI Systems*, pp. 112-123, 1995.

[5] F.N. Najm, "Transition Density, A New Measure of Activity in Digital Circuits," *IEEE Trans. on Computer-Aided Design*, vol. 12, No. 2, pp. 310-323, 1993.

[6] F.N. Najm and M.Y. Zhang, "Extreme Delay Sensitivity and the Worst-Case Switching Activity in VLSI Circuits," *ACM/IEEE Design Automation Conf.*, pp. 623-627, 1995.

[7] C.C. Tsai and M. Marek-Sadowska, "Multilevel Logic Synthesis for Arithmetic Functions," *ACM/IEEE Design Automation Conf.*, pp. 242-247, 1996.

[8] E. Sentovich, et al., "SIS: A System for Sequential Circuit Synthesis," *Technical Report, University of California at Berkeley*, 1992.

[9] Y. Ye and K. Roy, "Graph-based Synthesis Algorithms for AND/XOR Networks," *ACM/IEEE Design Automation Conf.*, pp. 107-112, 1997.