Estimation of Circuit Activity Considering Signal Correlations and Simultaneous Switching *

Tan-Li Chou
Electrical Engineering
Purdue University
West Lafayette, IN

Kaushik Roy
Electrical Engineering
Purdue University
West Lafayette, IN.

Sharat Prasad
Integrated Systems Lab.
Texas Instruments, Inc.
Dallas, TX.

Abstract

This paper presents accurate estimation of signal activity at the internal nodes of CMOS combinational logic circuits. The methodology is based on stochastic model of logic signals and takes correlations and simultaneous switching of signals at logic gate inputs into consideration. In combinational logic synthesis, in order to minimize spurious transitions due to finite propagation delays, it is crucial to balance all signal paths and to reduce the logic depth [4]. As a result of balancing delays through different paths, the inputs to logic gates may switch at approximately the same time. We have developed and implemented an technique to calculate signal probability and switching activity of the CMOS combinational logic circuits. Experimental results show that if simultaneous switching is not considered the switching activities of the internal nodes can be off by more than 100% compared to simulation based techniques. In contrast, our technique is on the average within 2% of logic simulation results.

1 Introduction

With the recent trend toward portable computing and communication systems there has been an increasing thrust toward considering power dissipation during VLSI design [4, 3, 8]. In order to design circuits for low power and high reliability, accurate estimation of power dissipation is required. In CMOS circuits majority of the power dissipation is due to charging and discharging of load capacitance of logic gates. Such charging and discharging occurs due to signal transitions. 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. Therefore probabilistic techniques have been resorted to. Research directed at estimating signal activity for combinational logic are reported in [2, 5, 6]. However, such methods fail to consider the effect of "near simultaneous" signal switching at logic gate inputs. SPICE simulation result shown in Figure 2 for the circuit of Figure 1 shows that the spurious switching disappears at node V_6 and is negligible at node V_5 if the two primary inputs have a rising and a falling transition respectively, within 3ns of each other. The spurious pulses try to charge or discharge the capacitances associated with the nodes of a circuit. If such pulses are not wide enough to charge or discharge the capacitances, they disappear. The above example shows that if the inputs to a logic gate switch within a period of Δt, spurious transitions do not occur at the output. Δt is a function of the inertial delay of the gate and the load capacitances associated with the gate.

*This research was supported in part by the IBM Corporation under SUR program

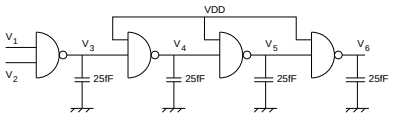

Figure 1: A Circuit for SPICE simulation

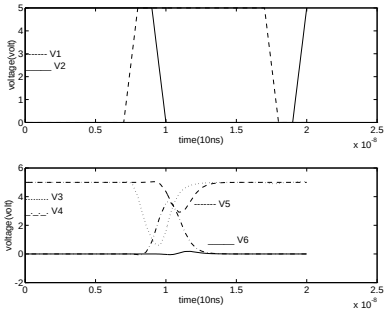

Figure 2: Timing diagram for SPICE simulation

The effect of simultaneous switching at the inputs of a logic gate can be best understood by considering the example of Figure 3. If the signals at the inputs of a two input XOR logic gate are switching as shown in the figure, the output switching activity will be zero, even though the signal transition rates at the inputs are high. In this paper, we consider such effects in estimating signal activities at the internal nodes of a multilevel circuit.

The paper is organized as follows. Section 2 gives the basic definitions and a brief review of *signal probability* and *activity* followed by a brief discussion on *power dissipation* in CMOS logic. Section 3 presents accurate estimation of signal activity considering signal correlations and simultaneous switching of inputs to logic gates. A technique to derive *activities* at the internal nodes of a circuit from their *symbolic signal probabilities* is given in Section 4. Section 5 gives the implementation details and experimental results to show that our technique is accurate and applicable to large circuits. The conclusions are given in Section 6.

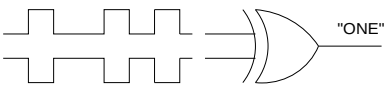

Figure 3: Example of simultaneous switching

2 Preliminaries and Definitions

In this section we describe the representation of multi-level circuits and review the concept of signal probability and activity, followed by a brief discussion on power consumption in CMOS logic circuits.

Multi-level Logic Representation: Multilevel logic can be described by a set $\mathcal{F}$ of completely specified Boolean functions. Each Boolean function $f \in \mathcal{F}$, maps one or more inputs and intermediate signals to a n output or a new intermediate signal. A circuit is represented as a *Boolean network*. Each *node* has a Boolean *variable* and a Boolean *expression* associated with it. There is a directed edge to a node g from a node f, if the expression associated with node g contains in it the variable associated with f in either true or complemented form. A circuit is also viewed as a *set of gates*. Each gate has one or more input pins and (generally) one output pin. Several pins are electrically tied together by a signal. Each signal connects to the output pin of exactly one gate, called the driver gate.

Signal Probability and Activity in CMOS:
We briefly describe the model used in [2] for estimation of signal activity of circuits in the following. The primary inputs to a combinational circuit are modeled as *mutually independent SSS (Strict-Sense Stationary) mean-ergodic 0-1 processes*. Under this assumption, the probability of the primary input logic signals $x_i(t), i = 1 \ldots n$, assuming the logic value ONE at any given time t becomes a constant, independent of time, and is called the *equilibrium probability* of the random signal $x_i(t)$. This is denoted by $P(x_i)$. The *activity* $A(x_i)$ at a primary input x_i of the module is defined as $\lim_{T\to\infty} \frac{n_{x_i}(T)}{T}$ and equals the expected value of $\frac{n_{x_i}(T)}{T}$. The variable n_{x_i} is the number of switching of $x_i(t)$ in the time interval $(-T/2, T/2]$. Since digital circuits can be thought of as non-linear but time-invariant systems, the signals at the internal and output nodes of the circuit are also *SSS* and *mean-ergodic*. Further, the Boolean functions describing the outputs of a logic module are decoupled from the delays inside the module by assuming the signals to have passed through a special delay module prior to entering the module under consideration. Therefore, the task of propagating *equilibrium probabilities* through the module is transformed into that of propagating *signal probabilities*. Also the *activities* $A(y_j)$ at the nodes y_j of the module are given by

$$A(y_j) = \sum_{i=1}^{n} P(\frac{\partial y_j}{\partial x_i}) A(x_i) \qquad (1)$$

Here $\partial y/\partial x$ is the Boolean difference of function y with respect to x and is defined by $\frac{\partial y}{\partial x} = y\mid_{x=1} \oplus y\mid_{x=0} = y(x) \oplus y(\bar{x})$.

Though equation 1 considers signal correlations within a logic module, it does not take simultaneous switching into account. Hence, this method results in errors in estimating activities of a circuit.

In order to better explain our approach, we introduce the concept of normalized activity, which will be used in Section 3. Assume all primary inputs to the module switch only at the leading edge of the clock and the module is delay-free. Every signal x at the internal or output node of the circuit become a *discrete-time* stochastic process. Therefore, $x(t)$ represents the logic value for the time interval $(t, t+T]$, where t is some leading edge of the clock cycle and T is the clock cycle of the circuit module. This is due to the fact that during the interval the signal $x(t)$ does not change value. We denote $P(x(t))$ as the probability of node x being logic ONE in the time interval $(t, t+T]$. If one selects a clock cycle at random, the probability of having a switching at t at node y is $A(y)/f$. Here $A(y)$ is the *activity* at node y and f is the clock frequency [7]. We define the *normalized activity* $a(y)$ as $A(y)/f$.

Power Dissipation in CMOS Logic Circuits
Of the three sources of power dissipation in digital CMOS circuits – switching, direct-path short circuit current, and leakage current – the first one is by far the dominant. Ignoring power dissipation due to direct-path short circuit current and leakage current, the average power dissipation in a CMOS logic is given by $POWER_{ave} = \frac{1}{2} V_{dd}^2 \sum_i C_i A(i)$, where V_{dd} is the supply voltage, $A(i)$ is the *activity* at node i, and C_i is the capacitive load at that node. The summation is taken over all nodes of the logic circuit. It should be observed that $A(i)$ is proportional to $a(i)$. C_i is approximately proportional to the fanout at that node. As a result, the *normalized power dissipation measure* Φ defined as $\Phi = \sum_i fanout_i \times a(i)$ is proportional to the average power dissipation in CMOS circuits. The parameter, $fanout_i$ is the number of fanouts at node i.

3 Accurate Activities

When more than one primary input, say x_i and x_j, are switching simultaneously, the Boolean differences $\frac{\partial y}{\partial x_i}$ and $\frac{\partial y}{\partial x_j}$ are undefined at those time instants. Hence, the proof in [2] that leads to equation 1 is no longer valid for this situation. Therefore, we need the following definitions to derive the accurate expression of activity considering simultaneous switching of signals. Let y be a Boolean expression and $x_i, i = 1 \ldots n$ be *mutually independent* primary inputs of y. We define, $\frac{\partial y^k\mid_{b_{i_1},\ldots,b_{i_k}}}{\partial x_{i_1}\cdots\partial x_{i_k}} = y\mid_{x_{i_1}=b_{i_1},\ldots,x_{i_k}=b_{i_k}} \oplus y\mid_{x_{i_1}=\bar{b}_{i_1},\ldots,x_{i_k}=\bar{b}_{i_k}}$, where k is positive integer, b_{i_j} is logic value ONE or ZERO and $x_{i_j}, j = 1 \ldots k$ are distinct *mutually independent* primary inputs of y. We define $Pc(\frac{\partial^k y\mid_{b_{i_1},b_{i_2},\ldots,b_{i_k}}}{\partial x_{i_1}\partial x_{i_2}\cdots\partial x_{i_k}})$ as the conditional probability of having a transition at time t while $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ are switching at time t and the rest of the signals are not. Here t is some leading edge of the clock. Under this condition, the probability of x_j ($\notin \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$) being ONE in the time interval $(t, t+T]$ is $\frac{P(x_j)-a(x_j)/2}{1-a(x_j)}$ rather than $P(x_j)$, where T is the clock cycle. This is due to the fact that we have assumed x_j is not switching at time t and can be explained as follows. $x_j(t-T)x_j(t) = 1$ and $\bar{x}_j(t-T)\bar{x}_j(t) = 1$ signify that x_j does not switch at time t and remain ONE and ZERO respectively. Similarly, $x_j(t-T)\bar{x}_j(t) = 1$ and $\bar{x}_j(t-T)x_j(t) = 1$ signify that x_j has a transition from ONE to ZERO and from ZERO to ONE respectively. Therefore, since x_j is *SSS*, the following equations hold,
$P(x_j(t-T)x_j(t)) + P(\bar{x}_j(t-T)\bar{x}_j(t)) = 1 - a(x_j)$
$P(x_j(t-T)\bar{x}_j(t)) + P(\bar{x}_j(t-T)x_j(t)) = a(x_j)$.
However, because $P(x_j(t-T)) = P(x_j(t))$ since x_j is *SSS*, and
$P(x_j(t-T)) = P(x_j(t-T)x_j(t)) + P(x_j(t-T)\bar{x}_j(t))$,
$P(x_j(t)) = P(x_j(t-T)x_j(t)) + P(\bar{x}_j(t-T)x_j(t))$,
therefore, $P(x_j(t-T)\bar{x}_j(t)) = P(\bar{x}_j(t-T)x_j(t)) = a(x_j)/2$.
Since $x_j(t-T)x_j(t) + \bar{x}_j(t-T)x_j(t) = x_j(t)$, it follows that $P(x_j) = P(x_j(t)) = P(x_j(t-T)x_j(t) + \bar{x}_j(t-T)x_j(t)) = P(x_j(t-T)x_j(t)) + P(\bar{x}_j(t-T)x_j(t))$. Hence, $P(x_j(t-T)x_j(t)) = P(x_j) - a(x_j)/2$. The conditional probability of x_j being ONE in the time interval $(t, t+T]$ while it does not switch at t, denoted as $Pc(x_j)$, is given by
$P(x_j(t) = 1 \mid \{x_j(t-T)x_j(t) + \bar{x}_j(t-T)\bar{x}_j(t) = 1\}) = \frac{P(x_j(t-T)x_j(t))}{P(x_j(t-T)x_j(t)) + P(\bar{x}_j(t-T)\bar{x}_j(t))} = \frac{(P(x_j)-a(x_j)/2)}{1-a(x_j)}$.

Under the assumption that the primary inputs are *mutually independent* and the logic signals can be modeled as *strict-sense stationary(SSS) mean-ergodic 0-1 discrete-time stochastic processes* with logic modules having zero delays, the following the-

orem holds.

Theorem 1 (n-inputs) *If y is a Boolean expression and $x_i, i = 1 \ldots n$ are* mutually independent *primary inputs of y. Then*

$$a(y) = \sum_{i=1}^{n} Pc(\frac{\partial y}{\partial x_i})(a(x_i) \prod_{\substack{j \neq i \\ 1 \leq j \leq n}} (1 - a(x_j)))$$
$$+\frac{1}{2}(\sum_{1 \leq i < j \leq n} (Pc(\frac{\partial^2 y|_{00}}{\partial x_i \partial x_j}) + Pc(\frac{\partial^2 y|_{01}}{\partial x_i \partial x_j}))[a(x_i)a(x_j)$$
$$\prod_{l \in \{1,2,\ldots,n\} - \{i,j\}} (1 - a(x_l))] \cdots$$
$$+\frac{1}{2^{n-1}}(Pc(\frac{\partial^n y|_{0\ldots0}}{\partial x_1 \cdots \partial x_n}) + Pc(\frac{\partial^n y|_{01\ldots1}}{\partial x_1 \partial x_2 \cdots \partial x_n}))(\prod_{l=1}^{n} a(x_l)),$$

where $Pc(\frac{\partial y}{\partial x_i}), Pc(\frac{\partial^2 y|_{00}}{\partial x_1 \partial x_2}), \ldots$ and $Pc(\frac{\partial^n y|_{01\ldots1}}{\partial x_1 \partial x_2 \cdots \partial x_n})$ are conditional probabilities under the condition that only some primary inputs switch and the rest do not at the leading edge of the clock cycle.

However, if we apply Theorem 1 to calculate the exact activity, we must compute
$Pc(\frac{\partial y}{\partial x_i}), Pc(\frac{\partial^n y|_{00\ldots0}}{\partial x_1 \partial x_2 \cdots \partial x_n}), \ldots$ *and* $Pc(\frac{\partial^n y|_{01\ldots1}}{\partial x_1 \partial x_2 \cdots \partial x_n})$.
This can be CPU time intensive. In the next section, we will show how to utilize symbolic probability to calculate exact activity.

4 Derivation of Activities from Signal Probabilities

Often times symbolic probability expression in terms of input symbolic probabilities or Binary Decision Diagram of a node is created before activity is computed [2, 3, 5, 8]. Therefore, it saves time and memory space by deriving the activity from its symbolic probability expression. We first show how to compute the exact activity, considering simultaneous switching.

We know from the last section that $P(y(t-T)y(t)) = P(y(t)) - \frac{1}{2}a(y) = P(y) - \frac{1}{2}a(y)$, where y is some node in a circuit. Hence, $a(y) = 2(P(y) - P(y(t-T)y(t)))$. Instead of computing $P(y(t-T)y(t))$ directly, we can utilize the symbolic form of $P(y)$ to do the same. Let $y = f(x_1, x_2, \ldots, x_n)$ be a Boolean expression in terms of the primary inputs. We can express $y(t-T)y(t)$ as follows,

$$y(t-T)y(t) = f(x_1(t-T), \ldots, x_n(t-T))f(x_1(t), \ldots, x_n(t))$$
$$= g(x_1(t-T), x_1(t), x_2(t-T), x_2(t), \ldots, x_n(t-T), x_n(t)),$$

where $g(.)$ is some Boolean expression. Similar to [1], $P(y(t-T)y(t))$ can be expressed as a sum of *primary input probability products* $\sum_{i=1}^{m} \alpha_i (\prod_{j \in I_i} P(s_j))$, where s_j is $x_j, \bar{x_j}, x_j(t-T)x_j(t), x_j(t-T)\bar{x_j}(t)$, or $\bar{x_j}(t-T)\bar{x_j}(t)$, and α_i is some integer. Note that s_j does not include $x_j(t-T), x_j(t)$ and $\bar{x_j}(t-T)x_j(t)$ since $P(x_j(t-T)) = P(x_j(t)) = P(x_j)$ and $P(\bar{x_j}(t-T)x_j(t)) = P(x_j(t-T)\bar{x_j}(t)) = \frac{1}{2}a(x_j)$.

Therefore, if we define
$P(x_j(t-T))P(x_j(t)) = P(x_j(t-T)x_j(t)) = P(x_j) - \frac{1}{2}a(x_j)$,
$P(x_j(t-T))P(\bar{x_j}(t)) = P(x_j(t-T)\bar{x_j}(t)) = \frac{1}{2}a(x_j)$, and
$P(\bar{x_j}(t-T))P(\bar{x_j}(t)) = P(\bar{x_j}(t-T)\bar{x_j}(t)) = 1 - P(x_j) - \frac{1}{2}a(x_j)$,
then $P(y(t-T)y(t)) = P(y(t-T))P(y(t))$.
Hence, we do not need to explicitly compute $P(y(t-T)y(t))$. The following example demonstrates this method.

Example 1 *$y = x_1 + x_2$. Given $P(y) = P(x_1) + \overline{P(x_1)}P(x_2)$, where $\overline{P(x_1)} = 1 - P(x_1)$, calculate $P(y(t-T)y(t))$.*

$$P(y(t-T)y(t)) = (P(x_1(t-T)) + \overline{P(x_1(t-T))}P(x_2(t-T)))$$
$$(P(x_1(t)) + \overline{P(x_1(t))}P(x_2(t)))$$
$$= P(x_1(t-T))P(x_1(t)) + (P(x_1(t-T))\overline{P(x_1(t))}P(x_2(t)) +$$
$$\overline{P(x_1(t-T))}P(x_1(t))P(x_2(t-T)) +$$
$$\overline{P(x_1(t-T))}P(x_1(t))P(x_2(t)))$$
$$= (P(x_1) - 1/2a(x_1)) + 1/2a(x_1)P(x_2) + 1/2a(x_1)P(x_2)$$
$$+(1 - P(x_1) - 1/2a(x_1))(P(x_2) - 1/2a(x_2)).$$

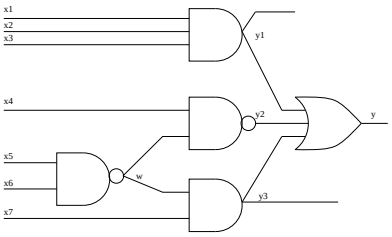

Figure 4: the minimum independent inputs to node y

After rearranging and simplifying terms, we have

$$a(y) = (1 - P(x_1))a(x_2) + (1 - P(x_2))a(x_1) - 1/2a(x_1)a(x_2). \square$$

Though $P(y(t-T))$ and $P(y(t))$ are expressed explicitly in terms of $P(x_j(t-T))$ and $P(x_j(t))$, it needs no extra memory space other than P(y). This is because they have the same symbolic form.

5 Implementation and Results

The proposed methodology uses signal probability measure to accurately estimate activity for CMOS circuits. Therefore, it is very important to accurately calculate signal probability for further use in estimating *activity*. We choose the general algorithm proposed in [1] and adopt a data structure similar to [3]. It should be noted that signal correlations have been taken into account in the algorithm.

However, the exact calculation of signal probability is $NP-hard$ [2]. Without properly partitioning the circuits, the method may not be applicable to very large circuits. This is due to the fact that the size of the symbolic probability expression grows exponentially with respect to the number of independent inputs to the circuit module. A partitioning algorithm that limits the number of inputs to a module has been proposed in [9]. However, it does not utilize the information of the circuit topology. For example, let us consider the circuit of Figure 4. Our goal is to calculate the signal probability at node y considering signal correlations. All we need is the set of 4 independent inputs, $\{y_1, x_4, w, x_7\}$ rather than the whole set of 7 primary inputs $\{x_1, x_2, \ldots, x_7\}$. This is because y_1, x_4, w, x_7 are independent. We, therefore, apply the heuristic similar to [9] to the smaller set and set the maximum number of independent inputs to be 10.

The Probability and Activity Simulator (PAS) to estimate activities at the internal and output nodes of a CMOS logic circuit has been implemented in C under the Berkeley SIS environment. We assume that the primary input signal probabilities and activities are available to us through system level simulation of the environment that the circuit will be working in. We first present a number of test cases that show the importance of considering simultaneous switching for circuits. Then we show the accuracy and efficiency of our technique on ISCAS benchmark circuits. In order to asses the accuracy of the results, we use a logic simulator with zero-delay model. We generated 10,000 random primary inputs (conforming to the given probabilities and activities) for logic simulation to determine the activities at the internal nodes of a circuit. The primary input signal probabilities and activities of the circuit were used in *PAS* to generate *probabilities* and *activities* at internal nodes of the circuits.

Table 1 shows the detailed results for an *MCNC* benchmark example (*parity*). All inputs were assigned a signal probability of 0.5. All primary inputs were assigned the same activity as

Table 1: Detailed result for *MCNC* benchmark example *parity*

act	*Sim*		*NSS*		*PAS*	
	Φ	CPU (sec)	CPU (sec)	Err %	CPU (sec)	Err %
0.05	3.15	232	0.02	23.5	0.01	0.0
0.1	5.35	270	0.02	42.6	0.01	0.0
0.18	8.42	407	0.02	72.1	0.01	-0.12
0.26	10.59	422	0.02	95.4	0.01	0.09

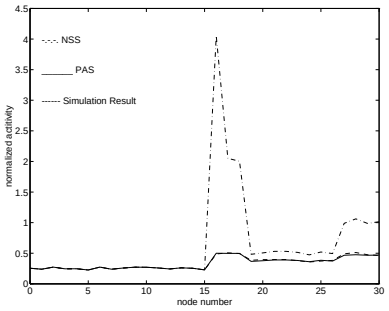

Figure 5: Node Activities for example *parity* (input activities = 0.26)

shown in the table (*act*). *PAS* denotes our technique for calculation of signal activity taking simultaneous switching and signal correlations into account. *NSS* (No Simultaneous Switching) denotes the technique that considers signal correlations but neglects simultaneous switching of signals at logic gate inputs. Φ represents the normalized power dissipation measure introduced in Section 2 and is used to compare the results with logic simulation technique (*Sim*). The percentage error (Err %) represents the deviation from the simulation results. The CPU times are also shown in the table for a SPARC 10 workstation. It can be observed that the power dissipation measure Φ can be off by more than 95% if simultaneous switching is not considered, while the results for our technique are remarkably close to the simulation results. Figure 5 shows the accuracy of activity calculation for the *parity* example. The x-axis represents the different nodes of the circuit, while the y-axis represents the *normalized activities* associated with each node. Node 0 through 15 are the primary inputs. Nodes 16 through 30 are either intermediate nodes or primary outputs. It can be easily observed that the *PAS* closely follows simulation results, while the errors introduced by *NSS* can be large.

Table 2 shows the results on 10 *ISCAS* benchmarks. Results show that the power dissipation measure Φ determined by *PAS* is on the average 2% of logic simulation results. On the other hand, *NSS* is at best 21% off the logic simulation results. It confirms again that simultaneous switching must be considered while estimating power dissipation.

6 Conclusions

By considering signal correlations and "near simultaneous" switching of inputs to static logic gates, we have shown that the activities at internal nodes are remarkably close to the simulation results. But if such switching is not considered, activities can be off by more than 100%. A technique that derive activity at a static logic gate from symbolic probability has been demonstrated. By partitioning the circuit properly, we have achieved large speed-up in our estimation algorithm while maintaining the accuracy. Hence, this technique can be efficiently used in a logic synthesis environment to estimate power dissipation.

Table 2: Results on ISCAS benchmarks

Ex.	*act*	*Sim*		*PAS*		*NSS*	
		Φ	CPU (sec)	CPU (sec)	Err %	CPU (sec)	Err %
C432	.1	37	273	2	6.4	2	39
	.26	82	554	2	0.6	2	71
C499	.1	102	529	4	0.0	16	143
	.26	144	872	4	-0.3	16	366
C880	.1	80	524	3	2.9	3	23
	.26	137	1068	3	1.2	3	37
C1355	.1	180	1033	22	0.2	34	106
	.26	292	1709	22	-0.0	34	245
C1908	.1	240	1508	27	0.7	26	63
	.26	358	2514	27	0.3	26	109
C2670	.1	268	1565	18	2.1	22	25
	.26	589	3255	18	1	22	55
C3540	.1	372	2308	498	1.8	454	53
	.26	740	4485	498	-1	454	110
C5315	.1	608	3839	68	-1.7	92	21
	.26	1314	7594	68	-0.9	92	52
C6288	.1	1028	5424	590	-10	2357	>300
	.26	1604	8596	590	-4.1	2357	>300
C7552	.1	892	7644	147	1.2	237	27
	.26	1892	11046	147	0.1	237	63

References

[1] K. P. Parker and E. J. McCluskey, "Probabilistic Treatment of General Combinatorial Networks," *IEEE Trans. on Computers.*, vol. C-24, Jun. 1975, pp. 668-670.

[2] F.N. Najm, "Transition Density, A Stochastic Measure of Activity in Digital Circuits," *ACM/IEEE Design Automation Conf.*, 1991, pp. 644-649.

[3] K. Roy and S. Prasad, "Circuit Activity Based Logic Synthesis for Low Power Reliable Operations," *IEEE Trans. on VLSI Systems*, Dec. 1993, pp. 503-513.

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

[5] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," *ACM/IEEE Design Automation Conf.*, 1992, pp. 253-259.

[6] J. Monteiro, S. Devadas, B. Lin, C-Y. Tsui, M. Pedram, and A. Despain, "Exact and Approximate Methods of Switching Activity Estimation in Sequential Logic," *Intl. Workshop on Low Power Design*, Napa Valley, 1994.

[7] S. Prasad and K. Roy, "Circuit Optimization for Minimization of Power Consumption under delay Constraint," *Intl. Workshop on Low Power Des ign*, Napa Valley, 1994.

[8] C. Tsui, M. Pedram, and A. Despain, "Technology Decomposition and Mapping Targeting Low Power Dissipation," *ACM/IEEE Design Automation Conf.*, 1993, pp. 68-73.

[9] B. Kapoor, "Improving the Accuracy of Circuit Activity Measurement," *ACM/IEEE Design Automation Conf.*, 1994, pp. 734-739.