Maximum Circuit Activity Estimation Using Pseudo-Boolean Satisfiability

Hratch Mangassarian[1] Andreas Veneris[1,2] Sean Safarpour[1] Farid N. Najm[1] Magdy S. Abadir[3]

***Abstract*— Disproportionate instantaneous power dissipation may result in unexpected power supply voltage fluctuations and permanent circuit damage. Therefore, estimation of maximum instantaneous power is crucial for the reliability assessment of VLSI chips. Circuit activity and consequently power dissipation in CMOS circuits are highly input-pattern dependent, making the problem of maximum power estimation computationally hard. This work proposes a novel pseudo-boolean satisfiability based method that reports the exact input sequence maximizing circuit activity in combinational and sequential circuits. The method is also extended to take multiple gate transitions into account by integrating delay information into the pseudo-boolean optimization problem. An extensive suite of experiments on ISCAS85 and ISCAS89 circuits confirms the efficiency and robustness of the approach compared to simulation based techniques and encourages further research for low-power solutions using boolean satisfiability.**

I. Introduction

In the nanometer VLSI era, reliability analysis of digital VLSI circuits is taking a significant share of the design process. In view of the roughly doubling component failure rate for every 10^oC increase in operating temperature [1], overheating caused by excessive power dissipation can degrade performance and reduce chip lifetime [2]. Large instantaneous power dissipation can also lead to a temporary voltage drop on power supply lines, which can result in soft errors [3]. Hence, accurate estimation of maximum peak power is vital to the reliability analysis of a circuit.

Dynamic power in CMOS circuits is a nontrivial function of clock frequency, technology parameters, gate capacitances and delays, circuit topology and primary input vectors. With everything else held constant, the pair of consecutive primary inputs that maximizes the switched capacitance in the circuit also maximizes dynamic peak power. Finding this pair among an exponential number of possibilities, or equivalently finding the associated circuit activity, is a combinatorial optimization problem whose corresponding decision problem is NP-complete.

Recent advances [13, 14, 15] and ongoing research in boolean satisfiability (SAT) have made it an attractive tool for solving theoretically intractable problems in VLSI CAD, in areas such as testing [16], verification [20], debugging [21] and physical design [22]. Furthermore, any improvement to the state-of-the-art in SAT solving immediately benefits all SAT based solutions.

This work proposes a pseudo-boolean satisfiability (PB-SAT) based method for generating tight lower bounds on maximum circuit activity per clock cycle. Given enough time, the method is guaranteed to find the exact input sequence that maximizes the given circuit activity model. The described framework is applicable to both combinational and sequential circuits. It is also extended to take glitches into account by integrating delay into the pseudo-boolean (PB) optimization problem. The competitive experimental results in this paper confirm the robustness of SAT in low-power analysis techniques. Coupled with the wide-ranging modeling flexibility offered by SAT, this encourages further research in the use of SAT as a platform to solve other low-power problems.

The rest of the paper is organized as follows. Section 2 presents previous work. Section 3 contains background information on SAT and PB-SAT theory. Section 4 briefly discusses assumptions and preliminaries. Section 5 gives the PB formulations for the maximum activity problem in combinational and sequential circuits. Section 6 extends the method to incorporate unit gate delay. Section 7 presents optimizations and heuristics. Section 8 contains experiments and Section 9 concludes the paper.

[1]University of Toronto, ECE Department, Toronto, ON M5S 3G4 ({hratch, veneris, sean, najm}@eecg.toronto.edu)

[2]University of Toronto, CS Department, Toronto, ON M5S 3G4

[3]Freescale Semiconductor, Inc., Austin, TX 78729 ({M.Abadir}@freescale.com)

II. Previous Work

Several techniques have been proposed to estimate the maximum peak power dissipation of a CMOS circuit [6-12]. An analogous problem is that of finding the maximum instantaneous current [3-5]. In [3, 4], a loose upper bound on the maximum instantaneous current is generated in linear time by propagating the signal uncertainty through the circuit. The upper bound is subsequently tightened using a branch-and-bound algorithm by considering spatial signal correlations at gate outputs. Extending the characterization of signal correlations from [3, 4] and exploiting mutually exclusive gate switching, the authors in [5] manage to generate tighter upper bounds. However, for larger circuits, the gap between the generated upper bounds and lower bounds obtained using random simulations can remain considerable.

In [6], the authors present an Automatic Test Pattern Generation (ATPG) based greedy algorithm that strives to maximize the fanout-weighted gate flips of a circuit. In [7], the method is extended to cover sequential circuits as well as glitches. A continuous optimization method is put forward in [8], which treats the boolean input space as a real-valued vector space and makes use of a gradient based heuristic to estimate the maximum power. A genetic search algorithm proposed in [9, 10] generates more robust lower bounds. In [11], the authors describe a statistical method that draws on the theory of Asymptotic Extreme Order Statistics. The probability distribution of the maximum power in a random sample of fixed size is computed. The largest power value with a non-zero probability is estimated to determine the maximum peak power.

The approach that is closest to this work is given in [12], where the power dissipation of a circuit is modeled as a multi-output Boolean function in terms of the primary inputs. A disjoint cover enumeration as well as a branch-and-bound algorithm are used to maximize the number of weighted gate transitions. An approximation strategy for upper bounding maximum power is also proposed. However, the described techniques can become computationally expensive. Furthermore, sequential circuits are not covered.

III. Background

A. Boolean Satisfiability

A *propositional logic formula* Φ is a logic function over a set of boolean variables linked by boolean *connectives* such as $\neg$ (negation), $\cdot$ (conjunction), $+$ (disjunction), $\rightarrow$ (implication) and $\leftrightarrow$ (equivalence). Φ is said to be satisfiable or SAT if it has a *satisfying assignment*: a *truth assignment* Π of its variables that causes it to evaluate to `true`, denoted as $\Pi \models \Phi$. Otherwise, Φ is said to be unsatisfiable or UNSAT. The problem of boolean satisfiability consists of determining whether Φ is SAT. In modern SAT solvers, the logic formula Φ is given in *Conjunctive Normal Form* (CNF) as a conjunction of *clauses* where each clause is a disjunction of *literals*. A literal is an instance of a variable or its negation. In order for a formula to be SAT, at least one literal in each clause must evaluate to `true`. For example, the CNF formula given in (1) is SAT because $\{a = 1, b = 0, c = 1\} \models \Phi$.

$$\Phi = (a + b) \cdot (a + \overline{b} + \overline{c}) \cdot (c) \quad (1)$$

A logic circuit can be converted to a CNF formula in linear time [16], such that there is a one-to-one correspondence between the variables of the generated CNF formula and the gate outputs of the corresponding circuit, and such that satisfying variable assignments

978-3-9810801-2-4/DATE07 © 2007 EDAA

in the CNF formula correspond to valid gate output values in the circuit. Hence, a circuit and its corresponding SAT formulation are often referred to interchangeably in this paper.

Modern SAT solvers [13, 14, 15] are able to solve large SAT problems with millions of clauses and hundreds of thousands of variables by utilizing advanced branch-and-bound procedures such as intelligent decision making, conflict based learning, and non-chronological backtracking. During the solving procedure, SAT solvers strive to prune parts of the non-solution search-space by analyzing their mistakes and learning from them by appending *conflict clauses* to the original CNF formula. For example, consider the CNF formula in (1) and suppose that the solver has made the unsatisfiable variable assignments $\{a = 0, b = 1, c = 1\}$. A modern SAT solver might determine that the real cause of the conflict is the assignment $\{a = 0\}$, and hence add the conflict clause (a) to Φ in order to force $\{a = 1\}$.

B. Pseudo-Boolean Satisfiability

A *pseudo-boolean constraint* is a generalization of a CNF clause. A PB constraint over boolean variables $\{x_i\}_{i=0}^{n-1}$ is an inequality of the form:

$$\sum_{i=0}^{n-1} c_i l_i \geq c_n \tag{2}$$

where $\{c_i\}_{i=0}^{n} \in \mathbb{Z}$ and $\{l_i\}_{i=0}^{n-1}$ are the literals corresponding to $\{x_i\}_{i=0}^{n-1}$. *I.e.,* $l_i = x_i$ or $l_i = \overline{x_i}$. Note that a CNF clause is a PB constraint where $c_i = 1$ for $0 \leq i \leq n$.

A coefficient c_i is said to be *activated* if its corresponding literal l_i is assigned to `true`. A PB constraint is said to be *satisfied* if (2) holds. A PB formula Ψ is a conjunction of PB constraints. The problem of pseudo-boolean satisfiability questions the existence of a truth assignment to $\{x_i\}_{i=0}^{n-1}$ satisfying all the PB constraints in Ψ.

The pseudo-boolean *optimization* problem strives to find a satisfiable assignment to Ψ that also minimizes a given *objective function*:

$$\mathcal{F}(\mathbf{x}) = \sum_{i=0}^{n-1} d_i l_i \tag{3}$$

where $\mathbf{x} =< x_0, \ldots, x_{n-1} >$ and $\{d_i\}_{i=0}^{n-1} \in \mathbb{Z}$.

For example, given Ψ and $\mathcal{F}$ as shown in (4) below, both $\{a = 1, b = 0, c = 1\}$ and $\{a = 1, b = 0, c = 0\}$ are satisfying assignments. However, only the former minimizes $\mathcal{F}$.

$$\begin{aligned} \Psi &= (2a - 3b \geq 1) \wedge (a + b + \overline{c} \geq 1) \\ \mathcal{F} &= \overline{c} - a + 2\overline{b} \end{aligned} \tag{4}$$

There are two types of PB solvers: [18, 19] support PB constraints *natively*, while [17] translates the PB-SAT problem into a SAT problem and runs a state-of-the-art SAT solver [15] on the produced SAT instance. The latter approach is particularly suited to problems that are *almost* "pure" SAT [17] (*i.e.*, consisting of mostly SAT clauses and relatively few PB constraints), which is the case in this work. Furthermore, any advancements in SAT solving directly enhances such a strategy. Optimal translation of PB-SAT constraints into a set of CNF clauses is currently an area of active research.

The objective function is minimized as follows. The PB-SAT solver in [17] first runs the SAT solver without considering $\mathcal{F}(\mathbf{x})$ in order to get an initial `SAT` solution $\mathbf{x_0}$, with say $\mathcal{F}(\mathbf{x_0}) = k$, where k is the corresponding initial value of the objective function. The new PB constraint $\mathcal{F}(\mathbf{x}) \leq k - 1$ is subsequently added to the original problem. The SAT solver runs on the updated CNF formula and this process is repeated until the problem becomes `UNSAT`. The solution corresponding to the last k before the problem becomes `UNSAT` is the optimal solution minimizing the objective function.

IV. Assumptions and Preliminaries

In this work, latch-controlled *synchronous* digital circuits are considered. Primary inputs (PIs) and flip-flop (FF) outputs can only switch at the beginning of the clock cycle. This assumption is considered valid in related previous work as well.

The average dynamic power dissipation of a CMOS circuit is proportional to the *total switched capacitance*:

$$P \propto \sum_{i=0}^{m} C_i T(g_i) \tag{5}$$

where m is the number of circuit gates, C_i is the capacitive load on gate g_i and $T(g_i)$ is the output *transition count* of g_i per unit time.

Under the assumption that the clock period is sufficiently small, it is sound to interpret the average dynamic power dissipation over a clock cycle as the instantaneous dynamic power during that clock cycle [6-12]. Thus, letting $T(g_i)$ in Eq. (5) correspond to the transition count of g_i *during a clock cycle*, P can be viewed as the instantaneous dynamic power. In the remainder of this work, instantaneous dynamic power is simply referred to as *power*.

The following notation is used throughout this paper. $\mathcal{G}$ denotes the set of gates ($|\mathcal{G}| = m$), $\mathcal{I}$ the set of PIs ($|\mathcal{I}| = n$) and $\mathcal{S}$ the set of FFs ($|\mathcal{S}| = p$). $\textsc{fanouts}(g_i)$ ($\textsc{fanins}(g_i)$) denotes the set of fanouts (fanins) of g_i. Finally, in all the examples and experiments, it is assumed that $C_i = |\textsc{fanouts}(g_i)|$ for internal gates and $C_i = 1$ for primary output gates.

V. Zero-Delay Maximum Activity Computation using PB-SAT

A. Maximum Activity for Combinational Circuits

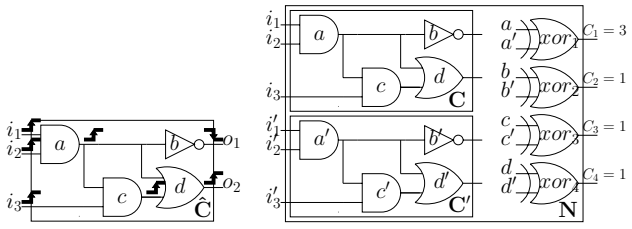
Under a zero-delay model, $T(g_i)$ becomes a boolean variable because g_i can flip at most once per clock cycle. Accordingly, Eq. (5) can be rewritten as:

$$P \propto \sum_{i=1}^{m} C_i \big(g_i(I) \oplus g_i(I')\big) \tag{6}$$

where I and I' are consecutively applied PI vectors and $g_i(X)$ denotes the steady-state value of g_i given PI vector X.

The problem then translates into finding the pair of consecutive PI vectors $< I^*, I'^* >$ that maximizes the right-hand-side of Eq. (6) and therefore P:

$$< I^*, I'^* >= \underset{<I,I'>\in\{0,1\}^{2n}}{\arg\max} \sum_{i=1}^{m} C_i \big(g_i(I) \oplus g_i(I')\big) \tag{7}$$



(a) Zero-delay gate switching

(b) Zero-delay PB-SAT formulation

Fig. 1. Combinational circuit

The solving procedure of Eq. (7) relies on the construction of a new circuit that will be used by the PB-SAT solver. The general procedure is illustrated with the use of an example. Consider the problem of finding the pair of PI vectors maximizing power for the circuit shown in fig. 1(a). First, the original circuit and its PIs are duplicated as shown in fig. 1(b). Next, every pair of corresponding gates, g_i in $\mathbf{C}$ and g_i' in $\mathbf{C}'$, is fed to an XOR gate, xor_i, in the new circuit $\mathbf{N}$. Since g_i and g_i' perform the same logic function, clearly $xor_i = g_i(I) \oplus g_i(I')$ becomes 1 if and only if the output of gate g_i flips under consecutive input vectors I and I'. The weighted sum of these XOR outputs, $\sum_{i=1}^{m} C_i xor_i$, is equal to the right hand-side-side of Eq. (6), which should be maximized. Given that the circuit $\mathbf{N}$ itself can be transformed to a SAT CNF which is a set of PB constraints, the following PB optimization problem strives to maximize the right-hand-side of Eq. (6) and therefore to solve Eq. (7):

$$\begin{aligned} \Psi &= \text{CNF}(\mathbf{N}) \\ \mathcal{F} &= -\sum_{i=1}^{m} C_i xor_i \end{aligned} \tag{8}$$

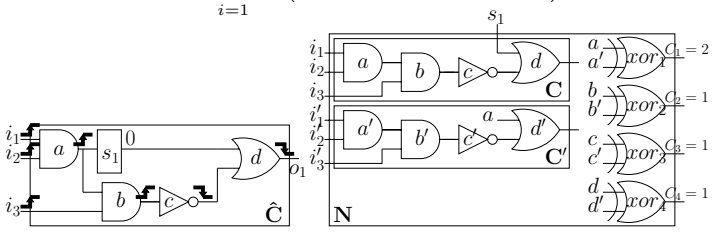
Remark that only the target function $\mathcal{F}$ in (8) is not in a pure SAT format. The PB formula Ψ is simply the CNF of $\mathbf{N}$, which markedly suits the choice of the PB-SAT solver [17].

Example 1 *Consider the original circuit $\hat{\mathbf{C}}$ and the corresponding construction $\mathbf{N}$ shown in Fig. 1. An optimal solution to the associated PB optimization problem given by (8) is $< I^*, I'^* >=<< 0,0,0 >, < 1,1,1 >>$, which amounts to a total switched capacitance of 6 units by flipping all four gate outputs as shown in Fig. 1(a).*

B. Maximum Activity for Sequential Circuits

Let $\delta : \{0,1\}^p \times \{0,1\}^n \rightarrow \{0,1\}^p$ denote the state transition function of a given sequential circuit. Also, let $g_i(S,X)$ denote the steady-state value of g_i given initial state S and PI vector X. Estimating the peak power per cycle for sequential circuits is equivalent to finding a triplet $< S^*, I^*, I'^* >$ consisting of an initial state S and consecutive PI vectors, I and I', that maximizes the right-hand-side of Eq. (9):

$$P \propto \sum_{i=1}^{m} C_i \Big(g_i\big(S,I\big) \oplus g_i\big(\delta(S,I),I'\big) \Big) \tag{9}$$



(a) Zero-delay switching (b) Zero-delay PB-SAT formulation

Fig. 2. Sequential circuit

The general solving procedure for this problem is illustrated with the use of an example. Consider the sequential circuit $\hat{\mathbf{C}}$ shown in Fig. 2(a). First, FF inputs (outputs) are transformed into circuit pseudo-outputs (pseudo-inputs). This *full-scanned* circuit $\mathbf{C}$ is subsequently duplicated similarly to the combinational case. Moreover, the pseudo-outputs of the first time-frame $\mathbf{C}$ are connected to the pseudo-inputs of the second time-frame $\mathbf{C}'$ as shown in Fig. 2(b). This iterative logic array (ILA) expansion of the original sequential circuit is referred to as *circuit unrolling*. In the new circuit $\mathbf{N}$, after feeding corresponding gates in $\mathbf{C}$ and $\mathbf{C}'$ to XORs, similarly to the combinational case, it is easily seen that

$$xor_i = g_i\big(I,S\big) \oplus g_i\big(I',\delta(S,I)\big).$$

The resulting PB optimization problem has the same form as (8).

Example 2 *Consider the circuit $\hat{\mathbf{C}}$ and the corresponding $\mathbf{N}$ shown in Fig. 2. Not counting flips at FF outputs (s_1), an optimal solution to the PB optimization problem given by (8) is $< S^*, I^*, I'^* >=<< 0 >, < 0,0,0 >, < 1,1,1 >>$, which amounts to a total switched capacitance of 5 units as shown in Fig. 2(a). However, this solution might be suboptimal if gate delays are considered. Section VI describes how delay is integrated into the PB optimization problem.*

The given problem formulation allows for any initial state to be returned in the optimal solution. Reachability analysis [20] can be subsequently performed to verify the reachability of the solution. SAT solvers offer a trivial way to discard unreachable solutions by adding conflict clauses. Similarly, combinations of invalid PIs can be eliminated using conflict clauses.

The illustrated framework can also be extended to generate peak n-cycle power ($n \geq 1$), which is the maximum average power over a contiguous sequence of n clock cycles, by unrolling the circuit $n+1$ times. Each pair of corresponding gates in adjacent cycles would then be fed to an XOR gate.

VI. Modeling Delay

Different input signal arrival times might cause a gate to flip several times during one clock cycle. In fact, glitches due to gate propagation delays can often dominate the maximum instantaneous power [7,9]. On the other hand, empirical results in [9] show that a *unit gate delay* model yields reasonably accurate power estimates. This section discusses the integration of unit gate delay into the problem formulation. It is also explained how this can be extended to *arbitrary delay* using a linear preprocessing step.

First, formal recursive definitions of *max-level* $L(g)$ and *min-level* $l(g)$ are given for $g \in \mathcal{G} \cup \mathcal{I} \cup \mathcal{S}$. $L(g)$ and $l(g)$ essentially denote the lengths of, respectively, the longest and shortest simple paths to g, in terms of number of gates, starting from a PI or a FF output.

Definition 1

$$L(g) = \begin{cases} \max\limits_{\{g_j \in \text{FANINS}(g)\}} L(g_j) + 1 & \text{if } g \in \mathcal{G} \\ 0 & \text{if } g \in \mathcal{I} \cup \mathcal{S} \end{cases}$$

Definition 2

$$l(g) = \begin{cases} \min\limits_{\{g_j \in \text{FANINS}(g)\}} l(g_j) + 1 & \text{if } g \in \mathcal{G} \\ 0 & \text{if } g \in \mathcal{I} \cup \mathcal{S} \end{cases}$$

Let $\mathcal{L} = \max_{g \in \mathcal{G}} L(g)$ designate the largest max-level in the circuit. Under a unit-delay model, time t is a discrete variable, meaningful in $\{0, \ldots, \mathcal{L}\}$. Moreover, the signal arrival time at the output of gate g following a certain path from a PI or a FF output is equal to the length of the traveled path to g.

Let $\mathcal{G}_t$ describe the set of all gates whose max-levels and min-levels bound t inclusively.

Definition 3 $\mathcal{G}_t = \big\{ g_i \in \mathcal{G} | l(g_i) \leq t \leq L(g_i) \big\}$

Lemma 1 *If every gate has a unit delay, any gate that could potentially flip at time-step t belongs to $\mathcal{G}_t$.*

Proof: The contrapositive is proved. If gate g does not belong to $\mathcal{G}_t$, then either $l(g) > t$ or $L(g) < t$. In the first case, the shortest signal arrival time from an input or a pseudo-input to a fanin of g takes at least t time-steps. So g can only flip strictly after time-step t. Similarly, g can only flip strictly before time-step t. ■

Consider a circuit whose gate logic values have stabilized given initial state S and PI vector I. If PI vector I' is applied at the start of a new clock cycle ($t = 0$), then let $g^t(S,I,I')$ denote the output of gate g right after time-step t. Note that the output value of g depends on both I and I' because if $t < l(g)$, $g^t(S,I,I') = g(S,I)$. Accordingly, the total switched capacitance is given as follows:

$$P \propto \sum_{t=1}^{\mathcal{L}} \sum_{g_i \in \mathcal{G}_t} C_i \big(g_i^{t-1}(S,I,I') \oplus g_i^t(S,I,I') \big) \tag{10}$$

The outer summation in Eq. (10) adds up the total switched capacitances across time-steps. The inner summation adds up the capacitances of the gates whose outputs flip at time t. Due to Lemma 1, one need not check all the gates at time-step t, but only the gates in $\mathcal{G}_t$.

The procedure to maximize the right-hand-side of Eq. (10) relies on the construction of a new circuit $\mathbf{N}$ that will be used by the PB-SAT solver. This is illustrated with the use of an example. Consider the sequential circuit $\hat{\mathbf{C}}$ shown in Fig. 2(a). First, FF inputs (outputs) are transformed into circuit pseudo-outputs (pseudo-inputs). Generating the sets $\{\mathcal{G}_t\}_{t=1}^{\mathcal{L}}$ for $\hat{\mathbf{C}}$ takes linear time using a Breadth First traversal starting from PIs and pseudo-inputs. For the circuit in Fig. 2(a), these sets are as follows:

$$\mathcal{G}_1 = \{a,b,d\}, \mathcal{G}_2 = \{b,c,d\}, \mathcal{G}_3 = \{c,d\}, \mathcal{G}_4 = \{d\}$$

Now, for each time-step t, for $0 \leq t \leq \mathcal{L}$, a *time-circuit* $\mathbf{C^t}$ is associated, containing the following *time-gates*:

$$\mathcal{G}(\mathbf{C^t}) = \begin{cases} \big\{ g_i^t | g_i \in \mathcal{G}_t \big\} & \text{if } t \geq 1 \\ \Big\{ g_i^0 | g_i \in \mathcal{G}(\hat{\mathbf{C}}) \Big\} & \text{if } t = 0, \end{cases} \tag{11}$$

as shown in Fig. 3. The new circuit $\mathbf{N}$ (Fig. 3) accommodates all these time-circuits $\{\mathbf{C^t}\}_{t=0}^{\mathcal{L}}$.

Next, the gate interconnections in $\mathbf{N}$ are discussed. The gates of $\mathbf{C^0}$ are interconnected identically to the original full-scanned circuit, given pseudo-input vector S and PI vector I, as shown in Fig. 3. For the remaining time-circuits $\{\mathbf{C^t}\}_{t=1}^{\mathcal{L}}$, there are three cases: The given time-gate's fanin was originally (in $\hat{\mathbf{C}}$) *i)* another gate, *ii)* a PI, or *iii)* a FF output. In case *i)*, the given time-gate must be connected to the most recent time-gate corresponding to the original

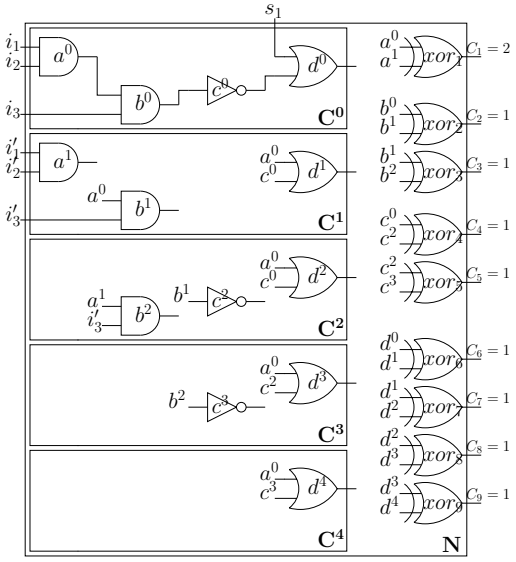


Fig. 3. Unit-delay sequential PB-SAT formulation

fanin gate *strictly before the current time-step*: No two time-gates in the same time-circuit can be connected because they can only change simultaneously. In case *ii)*, the given time-gate must be connected to the corresponding new PI in I'. In case *iii)*, the given time-gate must be connected to the pseudo-output in $\mathbf{C^0}$ corresponding to the FF to which it was originally connected.

Formally, consider a gate $g \in \hat{\mathbf{C}}$, such that $\text{FANIN}(g) = \{f, i, s\}$, where $f \in \mathcal{G}(\hat{\mathbf{C}}), i \in \mathcal{I}(\hat{\mathbf{C}}), s \in \mathcal{S}(\hat{\mathbf{C}})$. In the new circuit $\mathbf{N}$, for each time-step $t \geq 1$ where g^t exists, it will be connected to the following fanins:

$$\text{FANIN}(g^t) = \left\{ f^{\max\{j | f^j \in \mathcal{G}(\mathbf{C^j}), j<t\}}, i', \text{FANIN}(s)^0 \right\}$$

where each fanin corresponds to one of the different cases.

It can be shown that the output every time-gate g_i^t in $\mathbf{N}$ is consistent with its given definition in Eq. (10). In fact, the outputs of the time-gates in $\mathbf{C^0}$ represent the steady-state values of the original circuit $\hat{\mathbf{C}}$ given initial state S and PI vector I. This is consistent with the given definition of $g^0(S, I, I')$ used in Eq. (10). Furthermore, the construction is made such that in each time-circuit, from $\mathbf{C^1}$ to $\mathbf{C^{\mathcal{L}}}$, the new signals coming from the pseudo-outputs of $\mathbf{C^0}$ and the new PI vector I' propagate through exactly one additional gate. Therefore, the value at the output of time-gate g^t in $\mathbf{C^t}$ will be equal to that of gate g in $\hat{\mathbf{C}}$ after t time-steps, which is again consistent with Eq. (10).

The final step is to add an XOR gate for every pair of originally identical time-gates that are not separated by another identical time-gate between their respective time-circuits, as shown in Fig. 3. The weighted sum of these XOR gates yields:

$$\sum_{t=1}^{\mathcal{L}} \sum_{g_i \in \mathcal{G}_t} C_i \big(g_i^{\max\{j | g^j \in \mathcal{G}(\mathbf{C^j}), j<t\}}(S, I, I') \oplus g_i^t(S, I, I') \big)$$

which is in fact equivalent to Eq. (10) because even if $\max\{j | g^j \in \mathcal{G}(\mathbf{C^j}), j < t\} < t - 1$, time-step $\max\{j | g^j \in \mathcal{G}(\mathbf{C^j}), j < t\}$ is by definition the last time-step before t in which gate g_i could have flipped. Hence, $g_i^{\max\{j | g^j \in \mathcal{G}(\mathbf{C^j}), j<t\}}(S, I, I') = g_i^{t-1}(S, I, I')$.

Therefore, the problem of maximizing the right-hand-side of Eq. (10) can be formulated as a PB optimization problem of the same form as (8).

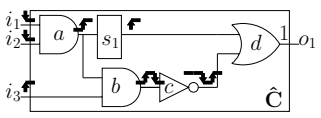


Fig. 4. Unit-delay gate switching in a sequential circuit

Example 3 *Consider the circuit $\hat{\mathbf{C}}$ in Fig. 4 and the corresponding $\mathbf{N}$ in Fig. 3. Using a unit-delay model and not counting flips at FF outputs (s_1), an optimal solution to the PB optimization problem given by (8) is $< S^*, I^*, I'^* >=<< 0 >, < 1, 1, 0 >, < 0, 0, 1 >>$, which amounts to a total switched capacitance of 6 units as shown in Fig. 4. It is notable that both the optimal solution and the associated circuit activity are different than those obtained for the same circuit with the zero-delay model in Example 2.*

The procedure outlined in this section can be extended to an arbitrary delay model as follows. A linear time preprocessing step is described in [7], which generates, for each gate, the sequence of time instants at which it might flip. For each gate g, let t_g^i and t_g^f respectively denote the first and last time instants at which g might flip. A circuit-level time sequence that includes *all* possible gate flipping time instants can be subsequently created. In order to apply the methodology described in this section to an arbitrary delay model, for each gate g, $l(g)$ and $L(g)$ should be respectively set to the *indices* of t_g^i and t_g^f in the circuit-level time sequence.

VII. Optimizations and Heuristics

In this section, optimization techniques to reduce the size of the PB problem and a heuristic to guide the search are presented.

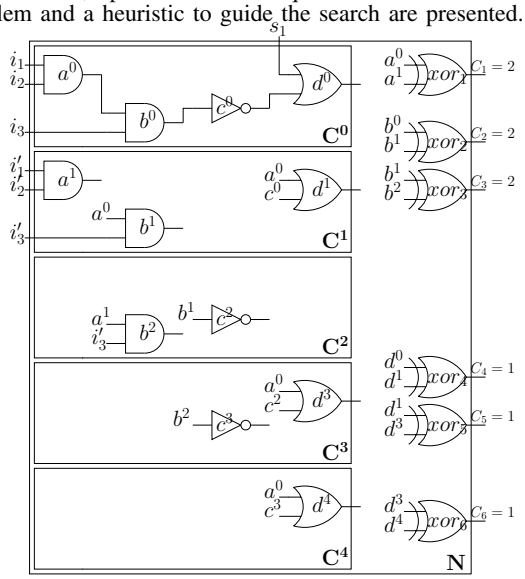


Fig. 5. Optimized unit-delay sequential PB-SAT formulation

Optimization 1. The definition of the sets $\{\mathcal{G}_t\}_{t=1}^{\mathcal{L}}$ in Def. 3 can be tightened. In fact, it is sometimes known in advance that a certain gate g will never flip at time-step t even though $g \in \mathcal{G}_t$. This can happen if $l(g) \leq t \leq L(g)$, but there exists no path p of length *exactly* t ($|p| = t$) from a PI or FF output to the output of g. For example, in the circuit of Fig. 2(a), although $l(d) = 1$ and $L(d) = 4$, d can never flip at time-step 2. Hence, in Fig. 3, the time-gate d^2 is redundant because its output will always be the same as that of d^1. The following is a tighter definition of $\mathcal{G}_t$.

Definition 4 $\mathcal{G}_t = \left\{ g_i \in \mathcal{G} | \exists p : x \overset{p}{\rightsquigarrow} g_i, x \in \mathcal{I} \cup \mathcal{S}, |p| = t \right\}$

The sets $\{\mathcal{G}_t\}_{t=1}^{\mathcal{L}}$ can be generated in linear time using a Breadth First traversal of the original circuit, starting from PIs and pseudo-inputs and memorizing the set of newly reached gates at each time-step. In Fig. 5, $\mathbf{N}$ is optimized to use Def. 4 for $\{\mathcal{G}_t\}_{t=1}^{\mathcal{L}}$.

Optimization 2. Suppose gate g is a buffer or an inverter. If the input of g flips, then the output of g flips. Therefore, for every sequence of buffers and/or inverters, it is sufficient to put only one XOR at the input of the first buffer/inverter and to add the load capacitances of the other gates to that XOR's original weight. In Fig. 5, this optimization is used to reduce the number of XORs. For large circuits with significant numbers of inverters and buffers, this can significantly reduce the size of the constructed circuit $\mathbf{N}$, and therefore the number of clauses in the CNF of the PB optimization problem.

Heuristic. As described in Subsection III-B, the PB-SAT solver gradually tightens the upper bound on the objective function, and therefore the lower bound on maximum circuit activity (8). This is done until either the absolute maximum is found or the solver is timed-out. However, instead of starting from an activity of 0, it is possible to first run random simulations for R seconds, record the generated maximum activity M and then force the solver to start from an activity of at least $\alpha \times M$, for some user-specified $\alpha \in [0, 1]$, using an appropriate conflict clause. If α is close to 1, this has the advantage of guiding the solver into parts of the search-space that might potentially yield higher circuit activities, and saves it the time of finding possibly many suboptimal solutions in other parts of the search-space. However, this will make the initial SAT problem harder. Therefore finding the first solution that yields a circuit activity greater than $\alpha \times M$ may take a longer time. Moreover, the PB-SAT solver may have a harder time learning from its mistakes.

VIII. Empirical Results

Both the zero-delay and unit-delay formulations of the proposed PB-SAT based approach for circuit activity estimation are implemented in C++ using MiniSat+ [17] as the underlying PB-SAT engine. The optimizations and the heuristic described in Section VII are also integrated. All experiments are conducted on a Pentium IV 2.8 GHz Linux platform with 2 GB of memory. Our approach (PB-SAT) is compared to parallel-pattern random simulations (SIM) with 32-bit words (32 simultaneous vector simulations). Since it is generally believed that the *switching probability* of the PIs of a circuit is positively correlated to that of internal gates [6], the switching probability of the PIs is set to 0.9 in SIM. All experiment runs, both PB-SAT and SIM, are timed-out after $10,000$ seconds, and the generated sequence of strictly increasing activities along with their corresponding run-times is recorded for each. Depending on the size of the circuit, roughly $1,000,000$ to $40,000,000$ vectors are simulated in $10,000$ seconds for SIM. On the other hand, two sets of PB-SAT experiments are performed, one for $\alpha = 0$ and one for $\alpha = 0.9$. For $\alpha = 0.9$, random simulations are first run for $R = 5$ seconds to extract the initial maximum activity estimate M.

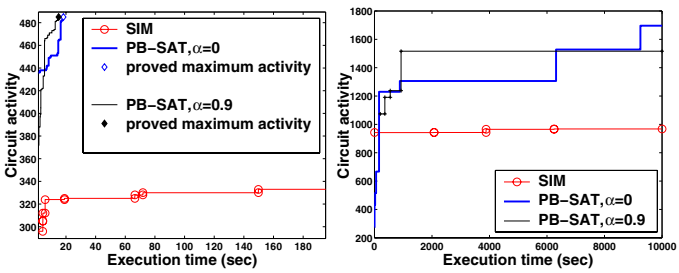
Table 1 shows the experimental results for ten ISCAS85 and twenty ISCAS89 circuits. The first and second rows respectively show the circuit names and the corresponding numbers of gates. The maximum circuit activities in Table 1 are in *units of switched capacitance*, where $C_i = |\text{FANOUTS}(g_i)|$ for internal gates and $C_i = 1$ for primary output gates. For each experiment, the generated maximum activity values are recorded after 100, $1,000$ and $10,000$ seconds. For each circuit and delay model, activities are compared between the three sets of experiments (PB-SAT,$\alpha = 0$), (PB-SAT,$\alpha = 0.9$) and SIM. The higher activity after each time-period is highlighted in bold. An empty table cell indicates that no bound is found up to that time. A bound that remains unchanged from the previous recorded time is highlighted in italic. Finally, a "$*$" next to an activity value indicates that the PB-SAT solver *proved* that the generated activity is in fact the absolute maximum. For instance, using a unit-delay model, in circuit s1488, (PB-SAT,$\alpha = 0$) yields the highest activity (1450) at 100 seconds, both (PB-SAT,$\alpha = 0$) and (PB-SAT,$\alpha = 0.9$) prove the maximality of 1450 by $1,000$ seconds, whereas the maximum activity generated by SIM remains unchanged (1250) after the 100 second mark.

In many circuits, the estimation improvement from simulations is considerably large. For instance, using a unit-delay model, in circuit s1423, (PB-SAT,$\alpha = 0$) and (PB-SAT,$\alpha = 0.9$) respectively record 196% and 177% improvements over SIM. c6288 constitutes a special case because of its disproportionately large number of levels ($\mathcal{L} = 164$), which causes $\mathbf{N}$, and subsequently the CNF of the SAT problem, to be very large.

Proving maximality is hard because it requires a virtual examination of the complete search-space. For instance, using a zero-delay model, in circuit c880, the solver converges to an activity of 482 before the $1,000$ second mark, but only later proves its maximality. In 53.3% of zero-delay experiments and 43.3% of unit-delay experiments, the PB-SAT solver proves maximality. To the best of our knowledge, this is the first work to compute the proven maximum activities for these circuits.

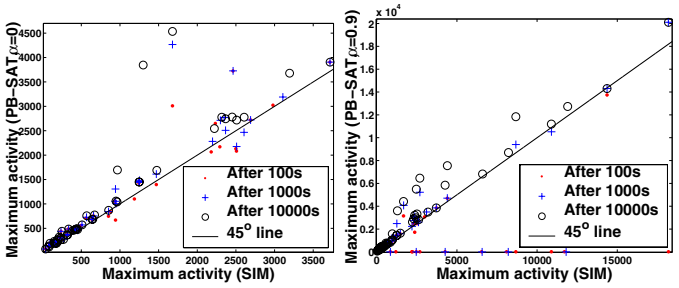
For combinational circuits, our approach yields an average of 13% improvement over simulations with a zero-delay model, and 18% with a unit-delay model, comparing both methods after $10,000$ seconds. For sequential circuits, our approach yields an average of 49% improvement over simulations with a zero-delay model, and 42% with a unit-delay model, comparing both methods after $10,000$ seconds. The greater improvements for sequential circuits are due to the larger and more intricate nature of the search-space.

Fig. 6 shows the sequences of strictly increasing activities generated by our methods and those generated using simulations, plotted against execution time, for the ISCAS89 circuit s713, under both zero-delay (Fig. 6(a)) and unit-delay (Fig. 6(b)) models. It can be noted that SIM results, in this case and in most other circuits shown in Table 1, tend to plateau after a while. In fact, as shown in Fig. 7, which plots (PB-SAT,$\alpha = 0$) (Fig. 7(a)) and (PB-SAT,$\alpha = 0.9$) (Fig. 7(b)) results against those of SIM, after 100 and $1,000$ seconds, a few points are still below the 45^o line, but after $10,000$ seconds, in virtually all the cases, the activities generated by our approach beat the ones generated by simulations during the same amount of execution time.



(a) Zero-delay model (b) Unit-delay model

Fig. 6. Activity vs. execution time for s713



(a) (PB-SAT,$\alpha = 0$) vs. SIM (b) (PB-SAT,$\alpha = 0.9$) vs. SIM

Fig. 7. SIM activities vs. PB-SAT solver activities

IX. Conclusion

This work proposes a pseudo-boolean satisfiability based framework, applicable to both combinational and sequential circuits, for finding the input sequence that maximizes single-cycle circuit activity. The method can also take into account multiple gate transitions during a clock cycle. The experimental results are promising for further research in low-power using SAT solvers, especially given the tremendous rate of advancement in SAT solvers and PB-SAT solvers.

References

[1] C. Small, "Shrinking devices put the squeeze on system packaging," *EDN*, pp. 41-46, Feb. 1994.

[2] F. Najm, "A survey of power estimation techniques in VLSI circuits," *IEEE Trans. on VLSI*, Vol. 2, No. 4, pp. 446-455, Dec. 1994.

[3] H. Kriplani, F. Najm and I. Hajj, "Maximum current estimation in CMOS circuits," *IEEE DAC*, pp. 2-7, 1992.

[4] H. Kriplani, F. Najm, P. Yang and I. Hajj, "Resolving signal correlations for estimating maximum currents in CMOS combinational circuits," *IEEE DAC*, pp. 2-7, 1993.

[5] C.-T. Hsieh, J.-C. Lin and S.-C. Chang, "A vectorless estimation of maximum instantaneous current for sequential circuits," *IEEE ICCAD*, pp. 537-540, Nov. 2004.

[6] C.-Y Wang and K. Roy, "Maximum power estimation for CMOS circuits using deterministic and statistical approaches," *IEEE Trans. on VLSI*, Vol. 6, pp. 134-140, 1998.

			$\hat{\mathbf{C}}$	**c432**	**c499**	**c880**	**c1355**	**c1908**	**c2670**	**c3540**	**c5315**	**c6288**	**c7552**
			$\|\mathcal{G}\|$	164	555	381	549	404	709	965	1579	3398	2325
zero delay	**PB-SAT**	$\alpha = \mathbf{0}$	$100s$	$\mathbf{193}^*$	**441**	470	447	430	**753**	**1053**	1395	3023	2064
			$1000s$	***193****	**493**	**482**	**480**	**445**	754	**1054**	**1611**	3191	**2282**
			$10000s$	***193****	*493*	$\mathbf{482}^*$	*480*	456	773	**1058**	**1689**	**3678**	2544
		$\alpha = \mathbf{0.9}$	$100s$	$\mathbf{193}^*$	462	**476**	**462**	**433**	744	979	**1578**	**3078**	
			$1000s$	***193****	471	481	471	441	**764**	1008	1593	**3497**	2236
			$10000s$	***193****	485	$\mathbf{482}^*$	479	**459**	**775**	1032	1638	*3497*	**2620**
	SIM		$100s$	176	421	414	424	389	657	938	1470	2980	**2179**
			$1000s$	178	*421*	437	426	*389*	659	*938*	*1470*	3107	2198
			$10000s$	*178*	444	*437*	*426*	*389*	671	950	1476	3195	2222
unit delay	**PB-SAT**	$\alpha = \mathbf{0}$	$100s$	838	2172	1330	**2647**	2133	2082	2633	7140	64720	6198
			$1000s$	**1006**	2713	2508	2770	2176	2467	5096	*7140*	*64720*	*6198*
			$10000s$	$\mathbf{1041}^*$	2779	2743	2781	2720	2779	6670	8034	*64720*	10477
		$\alpha = \mathbf{0.9}$	$100s$	879	**2494**	1712		**2691**	2466				
			$1000s$	941	**2741**	**3103**	**2974**	**2720**	**2605**				
			$10000s$	$\mathbf{1041}^*$	**2900**	**3196**	**2987**	**3329**	**2804**	**6813**	**8706**	101921	**12744**
	SIM		$100s$	**880**	2292	**2366**	2234	2500	**2508**	**6540**	**8100**	**118364**	**11699**
			$1000s$	949	2299	*2366*	2348	2509	**2605**	**6570**	**8199**	**129672**	**11811**
			$10000s$	964	2316	*2366*	2450	*2509*	2606	6596	8216	**139341**	11900

			$\hat{\mathbf{C}}$	**s208**	**s298**	**s344**	**s382**	**s386**	**s444**	**s526**	**s820**	**s832**	**s953**
			$\|\mathcal{G}\|$	113	148	191	201	172	224	236	432	300	298
zero delay	**PB-SAT**	$\alpha = \{\mathbf{0}, \mathbf{0.9}\}$	$100s$	$\mathbf{76}^*$	$\mathbf{139}^*$	$\mathbf{199}^*$	$\mathbf{194}^*$	$\mathbf{203}^*$	$\mathbf{201}^*$	$\mathbf{233}^*$	$\mathbf{364}^*$	$\mathbf{365}^*$	$\mathbf{274}^*$
			$1000s$	***76****	***139****	***199****	***194****	***203****	***201****	***233****	***364****	***365****	***274****
			$10000s$	***76****	***139****	***199****	***194****	***203****	***201****	***233****	***364****	***365****	***274****
	SIM		$100s$	38	79	178	94	167	149	149	310	319	266
			$1000s$	*38*	81	179	*94*	*167*	*149*	*149*	314	322	270
			$10000s$	*38*	*81*	*179*	*94*	*167*	*149*	*149*	315	*322*	***270***
unit delay	**PB-SAT**	$\alpha = \{\mathbf{0}, \mathbf{0.9}\}$	$100s$	$\mathbf{118}^*$	$\mathbf{195}^*$	$\mathbf{439}^*$	$\mathbf{265}^*$	$\mathbf{267}^*$	$\mathbf{321}^*$	$\mathbf{303}^*$	$\mathbf{465}^*$	$\mathbf{475}^*$	**556**
			$1000s$	***118****	***195****	***439****	***265****	***267****	***321****	***303****	***465****	***475****	$\mathbf{570}^*$
			$10000s$	***118****	***195****	***439****	***265****	***267****	***321****	***303****	***465****	***475****	***570****
	SIM		$100s$	83	154	237	192	237	214	233	401	426	496
			$1000s$	*83*	*154*	*237*	*192*	*237*	*214*	*233*	404	*426*	509
			$10000s$	*83*	*154*	250	*192*	*237*	*214*	*233*	410	*426*	*509*

			$\hat{\mathbf{C}}$	**s713**	**s1238**	**s1423**	**s1488**	**s1494**	**s9234**	**s13207**	**s15850**	**s38417**	**s38584**
			$\|\mathcal{G}\|$	454	545	806	666	660	6054	9290	10967	25452	22158
zero delay	**PB-SAT**	$\alpha = \mathbf{0}$	$100s$	$\mathbf{485}^*$	**444**	710	$\mathbf{684}^*$	$\mathbf{685}^*$	3010	**3727**	**2720**	**12077**	**11425**
			$1000s$	***485****	**460**	**726**	***684****	***685****	**4266**	*3727*	*2720*	***12077***	***11425***
			$10000s$	***485****	$\mathbf{474}^*$	757	***684****	***685****	**4533**	5181	6072	***12077***	***11425***
		$\alpha = \mathbf{0.9}$	$100s$	$\mathbf{485}^*$	432	**713**	$\mathbf{684}^*$	$\mathbf{685}^*$	**3161**				
			$1000s$	***485****	**460**	*713*	***684****	***685****	4074		**5225**	9401	10519
			$10000s$	***485****	$\mathbf{474}^*$	$\mathbf{770}^*$	***684****	***685****	4404	**5489**	**6451**	11852	11193
	SIM		$100s$	330	436	563	644	635	1679	2463	2690	8676	10895
			$1000s$	336	453	568	649	636	*1679*	*2463*	*2690*	*8676*	*10895*
			$10000s$	337	*453*	570	*649*	637	*1679*	*2463*	*2690*	*8676*	*10895*
unit delay	**PB-SAT**	$\alpha = \mathbf{0}$	$100s$	667	747	1104	**1450**	1430	3155	**4708**	**3906**	**21879**	**15522**
			$1000s$	**1306**	844	1483	$\mathbf{1450}^*$	$\mathbf{1450}^*$	*3155*	***4708***	***3906***	***21879***	***15522***
			$10000s$	**1696**	**870**	**3848**	***1450****	***1450****	**5922**	**10779**	***3906***	***21879***	***15522***
		$\alpha = \mathbf{0.9}$	$100s$	847			1440	**1449**		**4708**	3855		13742
			$1000s$	1187		**2484**	$\mathbf{1450}^*$	$\mathbf{1450}^*$		***4708***	*3855*	20109	14310
			$10000s$	1577	845	3596	***1450****	***1450****	5843	7546	*3855*	*20109*	*14310*
	SIM		$100s$	**942**	**852**	**1186**	1250	1246	**4268**	4403	3717	18185	14363
			$1000s$	942	***852***	1253	*1250*	*1246*	***4268***	*4403*	*3717*	*18185*	*14363*
			$10000s$	968	*852*	1300	*1250*	*1246*	*4268*	*4403*	*3717*	*18185*	*14363*

TABLE I

MAXIMUM ACTIVITIES PER CYCLE OBTAINED BY (PB-SAT,$\alpha = 0$), (PB-SAT,$\alpha = 0.9$) AND SIM, AFTER 100, 1,000 AND 10,000 SECONDS

[7] C.-Y Wang, K. Roy, "Estimation of maximum power for sequential circuits considering spurious transitions," *IEEE ICCD*, pp. 746-751, 1997.

[8] C.-Y Wang, K. Roy, "COSMOS: a continuous optimization approach for maximum power estimation of CMOS circuits," *IEEE ICCAD*, pp. 52-55, 1997.

[9] M. S. Hsiao, E. M Rudnick and J. H. Patel, "Effects of delay models on peak power estimation of VLSI sequential circuits," *IEEE ICCAD*, pp. 45-51, Nov. 1997.

[10] M. S. Hsiao, "Peak power estimation using genetic spot optimization for large VLSI circuits," *IEEE DATE*, pp. 175-179, Nov. 1999.

[11] Q. Wu , Q. Qiu and M. Pedram, "Estimation of peak power dissipation in VLSI circuits using the limiting distributions of extreme order statistics," *IEEE Trans. on CAD*, Vol. 20, No. 8, pp. 942-956, Aug. 2001.

[12] S. Devadas, K. Keutzer and J. White, "Estimation of power dissipation in CMOS combinational circuits using boolean function manipulation," *IEEE Trans. on CAD*, pp. 373-383, Mar. 1992.

[13] J. P. Marques-Silva and K. A. Sakallah, "GRASP – A search algorithm for propositional satisfiability," *IEEE Trans. on comput.*, Vol. 48, No. 5, pp. 506 521, May 1999.

[14] M. H. Moskewicz, C. F. Madigan, Y. Zhao and L. Zhang, "Chaff: Engineering an efficient SAT solver," *IEEE DAC*, pp. 530-535, jun. 2001.

[15] N. Eén and N. Sörensson, "An extensible SAT-solver," *SAT*, pp. 502-518, 2003.

[16] T. Larrabee, "Test pattern generation using boolean satisfiability," *IEEE Trans. on CAD*, Vol. 11, No. 1, pp. 4-15, 1992.

[17] N. Eén and N. Sörensson, "Translating pseudo-boolean constraints into SAT," *JSAT*, Vol. 2, pp. 1-26, 2006.

[18] F. Aloul, A. Ramani, I. Markov and K. Sakallah, "PBS: A backtrack search pseudo-boolean solver," *SAT*, 2002.

[19] H. Sheini and K. Sakallah, "Pueblo: A hybrid pseudo-boolean SAT solver," *JSAT*, Vol. 2, pp. 157-181, 2006.

[20] R. Drechsler, *Advanced Formal Verification*. Kluwer Academic Publishers, 2004.

[21] A. Smith, A. Veneris, M. F. Ali, and A. Viglas, "Fault diagnosis and logic debugging using Boolean satisfiability," *IEEE Trans. on CAD*, Vol. 24, No. 10, pp. 1606-1621, 2005.

[22] R. G. Wood and R. A. Rutenbar, "FPGA routing and routability estimation via Boolean satisfiability," *IEEE Trans. on VLSI*, Vol. 6, No. 1, pp. 222-231, Jun. 1998.