Estimation of the Weighted Maximum Switching Activity in Combinational CMOS Circuits

Fadi A. Aloul and Assim Sagahyroon
Department of Computer Engineering, American University of Sharjah, U.A.E.
{faloul, asagahyroon}@aus.edu

Abstract - **To achieve high reliability in VLSI designs, estimation of the maximum power dissipation during the design cycle is becoming important. In previous work, it was shown that maximizing dissipation is equivalent to maximizing gate output activity, appropriately weighted to account for differing load capacitances. Recent advances in Boolean Satisfiability (SAT) models and algorithms have made it tempting to use satisfiability-based techniques in solving various VLSI design-related problems such as verification and test generation. SAT solvers have also been extended to handle 0-1 integer linear programming (ILP) problems. In this paper, we present an ILP-based solution to compute the maximum weighted activity of combinational circuits. The problem is formulated as an ILP instance and the new SAT-based ILP solvers are used to find an estimate for the power dissipation. For performance comparison, the problem is also solved using generic ILP solvers. The validity of the proposed approach is demonstrated using benchmarks from the MCNC suite.**

I. INTRODUCTION

The continuing decrease in feature size and increase in chip density has raised concerns about power dissipation in VLSI chips. High or excessive power dissipation may cause run-time errors and chip destruction due to over heating. With increasing demands for high reliability in VLSI designs, accurate estimation of the maximum power dissipation during the design cycle is becoming critical. Estimation of maximum power helps designers in selecting the appropriate packaging and cooling techniques and in optimizing the sizes of the power and ground buses.

Power dissipation in CMOS combinational circuits arises from the following sources [20, 21]: dynamic power dissipation due to switching current from charging and discharging the parasitic capacitances, dynamic power dissipation due to short-circuit current when both *n*-channel and *p*-channel transistors are momentarily on at the same time and static power dissipation due to leakage current and subthreshold current.

The first source, namely dynamic power dissipation, is due to the signal switching activity that takes place during charging and discharging of the capacitors [21]. The signal switching activity depends on the input patterns applied to the circuit. Therefore to determine the maximum switching activity that a circuit can experience, it is necessary to search for a two-vector input sequence {V1, V2} that produces the maximum power dissipation [13, 11, 7]. The exact solution to this power maximization problem is NP-complete [7]. A possible scenario is to exhaustively simulate all possible patterns; however, this is practical for circuits with a small number of primary inputs. For large circuits, several heuristic-based approaches were reported in the literature. Devadas et al. [7] formulated the power dissipation of CMOS circuits as a Boolean function in term of the primary inputs. They attempted to maximize the function by solving a weighted max-satisfiability problem using exact and approximate algorithms. The technique proved to be practically applicable only to small circuits. In [18], an upper bound of maximum transition or switching density of individual nodes of combinational circuits was computed via propagation of uncertainty waveforms. Test generation based approaches have also been reported [24]. A test generation strategy is devised for finding test patterns that would produce the maximum node switching corresponding to the maximum power dissipation; the capacitive load of a gate was approximated using its fanout; nodes with large fanout are then assigned transitions which are justified backwards until the primary inputs are reached. In [13], the switching activity maximization problem is shown to be equivalent to a fault-testing problem on a transformed circuit. A maximum weighted activity is achieved by test vectors covering a selected set of faults of the transformed circuit. In [26], a statistical approach based on the asymptotic theory of extreme order statistics was presented. The method is based on the theory of extreme order statistics applied to the probabilistic distributions of the cycle-by-cycle power consumption, the maximum likelihood estimation, and the Monte-Carlo simulation.

Recent years have seen a remarkable growth in the use of Boolean Satisfiability (SAT) models and algorithms for solving various problems in Electronic Design Automation (EDA). This is mainly due to the fact that SAT algorithms have seen tremendous improvements in the last few years, allowing larger problem instances to be solved in different applications domains [2, 9, 14, 17, 27]. Such applications include formal verification [3], FPGA routing [19], global routing [1], logic synthesis [16], and sequential equivalence checking [4]. SAT has also been extended to a variety of applications in Artificial Intelligence including other well-known NP-complete problems such as graph colorability, vertex cover, and Hamiltonian path [6].

SAT solvers have traditionally been used to solve decision problems. Given a set of Boolean variables and constraints expressed in products-of-sum form (also known as conjunctive normal form (CNF)), the goal is to identify a variable assignment that will satisfy all constraints in the problem or prove that no such assignment exists. Recently, SAT solvers have been extended to handle Pseudo-Boolean (PB) constraints [1, 5, 8, 23, 25]. PB constraints are more expressive and can replace an exponential number of CNF constraints. Another key advantage of PB constraints is the ability to express optimization problems which are traditionally handled as integer linear programming (ILP) problems. Hence, SAT solvers can now handle both decision and optimization problems.

In this paper, we propose using the new SAT-based 0-1 ILP solvers to find an estimate for the maximum switching activity and therefore the maximum power dissipated by a CMOS combinational circuit. We transform the problem to an optimization instance and use SAT-based ILP solvers to find the vector pair that maximizes the power dissipation. Our approach provides a tight lower bound of power in a reasonable amount of time for large circuits. The problems are also solved using generic ILP solvers, e.g. CPLEX [12], for performance comparison. Experimental results are quite promising considering both, circuit complexity of the benchmarks used and CPU time required by the solvers to find a solution.

The rest of the paper is organized as follows: Section 2 provides background information on power dissipation and Boolean satisfiability; Section 3 explains how Boolean satisfiability was used to formulate the problem of estimating maximum dissipation in CMOS circuits; Section 4 gives a summary of the result and the paper is concluded in Section 5.

II. BACKGROUND

In this section, we review power dissipation in CMOS circuits and we introduce some key concepts related to Boolean satisfiability. This will serve in clarifying our proposed approach to solve the maximum power estimation problem.

2.1 Dynamic Power in CMOS Circuits

As explained in the introduction, dynamic power can be a major source of energy dissipation in CMOS circuits. The relationship between the logical behavior of a CMOS combinational circuit and the energy that the circuit dissipates is described using the following equation [7]:

$$E = 0.5CV_{DD}^2 S_G \quad (1)$$

E is the energy dissipated by the CMOS gate, C is the output capacitance for the gate and S_G is the total number of gate output transitions. V_{DD} is the voltage of the power source and also the assumed voltage swing of the node.

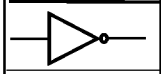
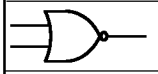
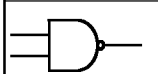
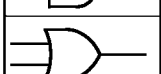
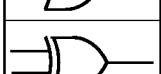
Gate Type	Gate Equation	CNF Expression
	$z = NOT(x)$	$(x + z) \cdot (\bar{x} + \bar{z})$
	$z = NOR(x, y)$	$(\bar{x} + \bar{z}) \cdot (\bar{y} + \bar{z}) \cdot (x + y + z)$
	$z = NAND(x, y)$	$(x + z) \cdot (y + z) \cdot (\bar{x} + \bar{y} + \bar{z})$
	$z = AND(x, y)$	$(x + \bar{z}) \cdot (y + \bar{z}) \cdot (\bar{x} + \bar{y} + z)$
	$z = OR(x, y)$	$(\bar{x} + z) \cdot (\bar{y} + z) \cdot (x + y + \bar{z})$
	$z = XOR(x, y)$	$(\bar{x} + y + z) \cdot (x + \bar{y} + z) \cdot (\bar{x} + \bar{y} + \bar{z}) \cdot (x + y + \bar{z})$

TABLE I. CNF formulas representing simple gates.

To a first degree of approximation, the capacitance C is assumed to be directly proportional to the fanout f of the gate [7, 13]. In this work, the effect of interconnect capacitance is ignored. It is clear from the equation above and the assumptions made so far, that the product $(f \cdot S_G)$ is directly proportional to the power consumed. Therefore, to maximize dynamic power dissipation, we need to search for an input vector pair {V1,V2}, that tends to maximize the *weighted* sum of the gates output transitions. The weights are determined by the capacitances or the fanouts of each gate in the circuit. The weighted switching activity (W) of the circuit can be approximated using the following equation:

$$W = \sum_{allgates} f_i(g_i(V1) \oplus g_i(V2)) \quad (2)$$

Where f_i is the fanout of gate g_i, and $g_i(V1)$ is the output of the gate when vector input V1 is applied while $g_i(V2)$ is its output when the vector V2 is applied at the primary inputs. Here, the summation of "XOR equal to 1" is the number of switching nodes when the input vector V1 is followed by input vector V2. Hence, to maximize W, one has to search for primary input vector pairs {V1,V2} that will induce maximum switching activity in the circuit. Note that a zero-delay model is assumed for all the gates in the circuit.

2.2 Boolean Satisfiability

Boolean Satisfiability (SAT) is often used as the underlying model in the field of computer aided designs of integrated circuits. A number of SAT solvers have been proposed and implemented [2, 9, 14, 17, 27]. These solvers employ powerful algorithms that are sufficiently efficient to deal with large-scale SAT problems that typically arise in the EDA domain. Most of these algorithms claim competitive results in runtime efficiency and robustness.

In SAT, given a formula f, the objective is to identify an assignment to a set of Boolean variables that will satisfy a set of constraints. If an assignment is found, it is known as a satisfying assignment, and the formula is called satisfiable. Otherwise if an assignment doesn't exist, the formula is called unsatisfiable. The constraints are typically expressed in conjunctive normal form (CNF). In CNF, the formula consists of the conjunction (AND) of m clauses $\omega_1, \ldots, \omega_m$ each of which consists of the disjunction (OR) of k literals. A literal l is an occurrence of a Boolean variable or its complement. Hence, in order to satisfy a formula, each of its clauses must have at least one literal evaluated to true.

As an example, a CNF instance $f(a, b, c) = (a + \bar{b}) \cdot (a + b + c)$ consists of 3 variables, 2 clauses, and 5 literals. The assignment $\{a = 0, b = 1, c = 0\}$ leads to a conflict, whereas the assignment $\{a = 0, b = 0, c = 1\}$ satisfies f.

Despite the problem being NP-Complete, there have been dramatic improvements in SAT solver technology over the past decade. This has lead to the development of several powerful SAT solvers that are capable of solving problems consisting of thousands of variables and millions of constraints in a few seconds [2, 9, 14, 17, 27].

Recently, SAT solvers [1, 5, 8, 23, 25] have been extended to handle pseudo-Boolean (PB) constraints which are linear inequalities with integer coefficients that can be expressed in the normalized form [1] of:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \geq b \quad (3)$$

where $a_i, b \in Z$ and x_i are Boolean variables. PB constraints can, in some cases, replace an exponential number of CNF constraints. They have been found to be very efficient in expressing "counting constraints" [1]. Furthermore, PB extends SAT solvers to handle *optimization* problems as opposed to only *decision* problems. Subject to a given set of CNF and PB constraints, one can request the minimization (or maximization) of an objective function which consists of a linear combination of the problem's variables. Note that each CNF constraint can be viewed as a PB constraint. For example the CNF constraint $(a + \bar{b})$ can be viewed as the PB constraint $a + \bar{b} \geq 1$. PB constraints represent ILP inequalities.

In this paper, we are interested in using SAT solvers to measure the maximum power dissipation in combinational circuits. Circuits are easily represented as a CNF formula by conjuncting the CNF formulas for each gate output. A gate output can be expressed using a set of clauses which specify the valid input-output combinations for the given gate. Hence, a CNF formula φ for a circuit is defined as the union of set of clauses φ_x for each gate with output x:

$$\varphi = \bigcup_{x \in Q} \varphi_x \quad (4)$$

where Q denotes all gate outputs and primary inputs in the circuit. Table I shows generalized CNF formulas for various gates.

III. PROBLEM FORMULATION AND IMPLEMENTATION

To estimate the maximum power dissipated, it is necessary to identify two-vector input sequence {V1, V2} that produces weighted maximum switching activity in the circuit under consideration. In this work, we assume that after the first vector V1 is applied, the circuit is allowed to stabilize before the application of the second vector V2. A SAT problem is created with following four groups of constraints:

1. A set of clauses representing the circuit's logical behavior after the application of input vector V1.
2. A set of clauses representing the circuit's logical behavior after the application of input vector V2. Note that the set of constraints in (1) and (2) are identical but the variables are renamed differently.
3. A set of clauses representing XOR gates between the outputs of gates in (1) and (2). The number of XOR gates equals the number of gates in the original circuit. An XOR gate output of logic 1 indicates that a transition (0 to 1 or 1 to 0) has occurred at the output of the gate in the original circuit upon the successive application of the vector V1 followed by vector V2.
4. An objective constraint which specifies the weights of the XOR outputs. Weights are computed based on the capacitance of the gate which is assumed to be proportional to the fanout of the gate.

Constraints (1) and (2) represent the circuit's logical behavior following the application of the two vectors respectively. The constraints are represented as explained in Section 2.2. Constraint (3) compares the output of the same gate for the two vectors. If a transition or a change in the output has occurred the XOR gate will produce an output of 1, else, the XOR gate output will be 0. Here also, the constraint is expressed using the prin-

Circuit A Consistency Function

$(a_1 + d_1) \cdot (\overline{a_1} + \overline{d_1})$

$(a_1 + e_1) \cdot (b_1 + e_1) \cdot (\overline{a_1} + \overline{b_1} + \overline{e_1})$

$(d_1 + f_1) \cdot (e_1 + f_1) \cdot (\overline{d_1} + \overline{e_1} + \overline{f_1})$

$(\overline{c_1} + g_1) \cdot (\overline{e_1} + g_1) \cdot (c_1 + e_1 + \overline{g_1})$

Circuit B Consistency Function

$(a_2 + d_2) \cdot (\overline{a_2} + \overline{d_2})$

$(a_2 + e_2) \cdot (b_2 + e_2) \cdot (\overline{a_2} + \overline{b_2} + \overline{e_2})$

$(d_2 + f_2) \cdot (e_2 + f_2) \cdot (\overline{d_2} + \overline{e_2} + \overline{f_2})$

$(\overline{c_2} + g_2) \cdot (\overline{e_2} + g_2) \cdot (c_2 + e_2 + \overline{g_2})$

Output Conditions

$(\overline{d_1} + d_2 + d) \cdot (d_1 + \overline{d_2} + d) \cdot$

$(\overline{d_1} + \overline{d_2} + \bar{d}) \cdot (d_1 + d_2 + \bar{d})$

$(\overline{e_1} + e_2 + e) \cdot (e_1 + \overline{e_2} + e) \cdot$

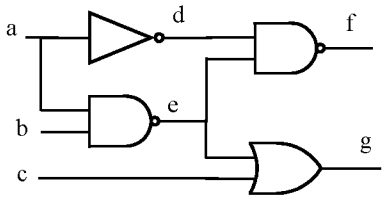
$(\overline{e_1} + \overline{e_2} + \bar{e}) \cdot (e_1 + e_2 + \bar{e})$

$(\overline{f_1} + f_2 + f) \cdot (f_1 + \overline{f_2} + f) \cdot$

$(\overline{f_1} + \overline{f_2} + \dot{f}) \cdot (f_1 + f_2 + \dot{f})$

$(\overline{g_1} + g_2 + g) \cdot (g_1 + \overline{g_2} + g) \cdot$

$(\overline{g_1} + \overline{g_2} + \bar{g}) \cdot (g_1 + g_2 + \bar{g})$



Transition Objective Function

$Maximize(d + 2e + f + g)$

Solution:

Max W. Transitions = 5

$\{a_1, b_1, c_1\} = \{1, 1, 0\}$

$\{a_2, b_2, c_2\} = \{0, 1, 1\}$

Fig 1. An illustrative example showing how to determine the weighted number of possible transitions in the given circuit.

ciples explained in Section 2.2. A new variable is declared for each XOR gate's output to indicate whether a transition occurred in the original circuit. Finally, the goal of the objective function in constraint (4) is to identify the two input vectors that would maximize the number of transitions in the circuit. This is expressed as a PB constraint consisting of the sum of the XOR gate's outputs. In other words, this can be viewed as a constraint representing the predicate, "*there exist two input vectors that can cause a weighted summation of gate transitions* > *k*" where *k* is an integer value. In formulating the problem, integer coefficients are used to represent the fanout (capacitance) of each gate.

3.1 An Illustrative Example

In this subsection we use the circuit shown in Figure 1 to provide the reader with an example that clearly illustrates the various steps of the proposed approach. The weighted number of possible transitions is determined.

The circuit shown in the example has three primary inputs *a, b* and *c*. To generate consistency functions for two circuit instances (Circuit *A* and Circuit *B* in the above example), the variables were renamed as $a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, etc$. The CNF clauses representing the circuits' consistency functions are generated. CNF clauses representing the XOR gates between the outputs of gates in circuits *A* and *B* are expressed as shown in the output conditions. The objective function consists of the sum of all XOR outputs. Each output is associated with an integer coefficient that is equal to the fanout of the gate. In the given example *d, f,* and *g* have a fanout of 1 and *e* has a fanout of 2. The optimization instance is passed to the ILP solver which returns the assignment: $\{a_1, b_1, c_1, a_2, b_2, c_2\} = \{1, 1, 0, 0, 1, 1\}$. The assignment yields the maximum possible switching activity that the given circuit can experience.

IV. EXPERIMENTAL RESULTS

In this section, we will report and discuss the experimental results obtained using the proposed power estimation technique. The results for the MCNC benchmark circuits [15] are presented in Table II. Given the limited space, we present results for 30 circuits of various sizes. Each circuit was sensitized using *sis* [22] to a circuit consisting of only 2-input NAND, NOR, and NOT gates. We used the SAT-based 0-1 ILP solvers PBS [1] and Galena [5]. In addition to the generic commercial ILP solver, CPLEX 7.0 [12]. The PBS and Galena experiments were conducted on a Pentium-IV 2.8 Ghz workstation running Linux and equipped with 500 MB of RAM. The CPLEX experiments were conducted on a SunBlade 1000 workstation with 2MB cache running SunOS 5.9. We used the default settings for PBS, Galena, and CPLEX. A time-out limit of 10,000 seconds was set for all experiments.

In order to speed up the ILP solvers, an initial objective goal was identified by generating 10K random primary input vectors and identifying the maximum weighted switching activity among all vectors [10]. The random approach helps eliminate significant parts of the search space as shown in Table II. The random approach runtime did not exceed one minute in all cases.

Table II lists the experimental results for PBS, Galena, and CPLEX. The first four columns describe the circuit; *#PI* is the number of primary inputs, *#Gates* represents the total number of gates in the circuit; column *MaxPos* gives the theoretical upper bound or the maximum weighted switching activity that can be attained only if all the gates switched simultaneously. The *Random* column represents the maximum weighted switching activity obtained using the random vector generation approach [10]. The *Time* column indicates the runtime (in seconds) for each solver. The column labeled *Val* represents the maximum weighted activity value obtained using the each solver. The *%-Max* column gives the percentage of the activity reported by the solver (*Val*) relative to the maximum upper bound (*MaxPos*). *m/o* indicates out-of-memory. Several observations are in order:

- In all but two cases, CPLEX was successful in computing the optimal weighted switching activity.
- PBS (Galena) was able to compute the optimal weighted switching value in all but 7 (8) cases. But for circuits where it timed-out, it did return the best weighted switching value seen so far. This can be viewed as a lower bound of the possible optimal weighted switching value. Giving any of the solvers extra time would have probably helped tighten the gap or even solve the problem by finding the optimal value.
- For the presented application, generic ILP solvers, e.g. CPLEX, perform much better than SAT-based 0-1 ILP solvers, e.g. PBS and Galena.
- The ILP solvers are fast for most small- and medium-size circuits but as circuits become larger, the ILP solvers become slower. Perhaps, larger circuits can be partitioned to speed up the search process.
- The random approach was unable of identifying the optimal weighted switching activity value in any of the presented instances.
- The proposed approach was able to improve on top of the random approach by a factor of 16% on average and in some cases by 55%. For example, PBS was able to improve on the value obtained by the random approach by a factor of 55% for the *i2* circuit. Detecting such a difference in the weighted switching activity value can be very useful.
- The ILP solvers can be used to verify and perhaps improve the

Circuit Name	#PI	#Gates	MaxPos	Random	PBS			Galena			CPLEX		
					Time	Val	%-Max	Time	Val	%-Max	Time	Val	%-Max
tcon	34	41	56	45	0.02	48	85	0.01	48	86	0.07	48	86
cmb	32	62	67	42	0.01	55	82	0.04	55	82	1.35	55	82
pm1	32	67	84	65	0.01	71	85	0.04	71	85	1.4	71	85
pcle	38	71	92	64	0.21	69	75	0.12	69	75	0.88	69	75
parity	32	75	89	60	379	60	67	14.3	60	67	78.9	60	67
cc	42	79	104	86	0.06	88	85	0.09	88	85	1.29	88	85
cm150a	42	79	89	74	0.02	84	94	0.05	84	94	1.03	84	94
pcler8	54	104	140	97	3.1	110	79	0.52	110	79	1.52	110	79
mux	42	106	130	99	0.18	115	89	0.3	115	88	1.9	115	89
cordic	46	124	152	97	8809	113	74	301	113	74	176	113	74
i3	264	132	132	73	0.59	132	100	0.02	132	100	0.05	132	100
sct	38	143	198	160	0.19	179	90	0.41	179	90	4.41	179	90
frg1	56	143	167	120	0	164	98	0.26	164	98	1.95	164	98
unreg	72	145	163	111	>10K	129	79	163	131	80	17	131	80
b9	82	147	183	130	0.25	164	90	0.42	164	90	3.12	164	90
count	70	161	221	134	>10K	173	78	>10K	170	77	3.86	173	78
lal	52	179	247	189	0.83	223	90	0.8	223	90	5.57	223	90
c8	56	211	289	207	495	235	81	220	235	81	24.4	235	81
i2	402	242	255	111	0.01	251	98	0.1	251	98	3.56	251	98
cht	94	249	328	229	>10K	288	88	>10K	291	89	13.5	295	90
C432	72	282	436	271	4468	358	82	>10K	348	80	46.2	358	82
apex7	98	295	414	247	2251	325	79	1181	325	79	98.9	325	79
ttt2	48	303	425	292	34.3	336	79	81.1	336	79	69.1	336	79
i4	384	308	308	158	0.02	308	100	0.01	308	100	0.11	308	100
ex2	170	351	493	280	>10K	345	70	m/o	282	57	48	353	72
i5	266	445	511	293	1.34	511	100	m/o	293	57	0.26	511	100
term1	68	525	739	508	>10K	608	82	m/o	562	76	505	608	82
vda	34	1417	2419	662	859	689	29	274	689	29	1434	689	29
pair	344	1955	2906	1437	>10K	1790	62	m/o	1437	50	>10K	2004	69
C6288	64	2400	4271	2267	>10K	2306	54	>10K	2584	61	>10K	2895	68

TABLE II. Experimental results using the SAT-based 0-1 ILP solvers PBS and Galena and the generic ILP solver CPLEX.

optimal weighted switching activity value identified by the random approach.

- In some cases, the optimal weighted switching activity value returned by the two solvers was very close to the theoretical value (*MaxPos*). In scenarios like these, the architectural aspects of the circuit can be modified or careful attention has to be paid to the cooling techniques used.
- It was also clear in other cases that the maximum possible weighted switching activity value is much smaller than the theoretical value. For example, while the maximum theoretical value for the *vda* circuit was 2419, all three solvers are able to prove that the maximum possible weighted switching value is only 689.

V. CONCLUSIONS

In this paper, we have presented an ILP-based technique to obtain an estimate for the maximum weighted switching activity that can be exercised by a CMOS combinational circuit. The difficulty in estimating the maximum dissipation possible stems from the fact that power dissipated is dependent on the input pattern. We formulated the problem in an ILP context and experimented with SAT-based 0-1 ILP solvers and generic ILP solvers to show that a solution can be found in most cases and in a reasonable amount of time. We showed that the proposed ILP approach produces better results than the random-based approach. Our results also showed that generic ILP solvers tend to be more competitive than SAT-based ILP solvers for the given application. In future work, the technique presented here will be extended by using a variable gate delay model. This will improve our estimate by taking into account the power dissipated due to any glitches that might occur in the circuit.

REFERENCES

[1] F. Aloul, A. Ramani, I. Markov, and K. Sakallah, "Generic ILP versus Specialized 0-1 ILP: An Update", in *ICCAD*, 450-457, 2002.

[2] R. J. Bayardo Jr. and R. C. Schrag, "Using CSP Look-Back Techniques to Solve Real World SAT Instances," in *AAAI*, 203-208, 1997.

[3] A. Biere, A. Cimatti, E. Clarke, M. Fujita, and Y. Zhu, "Symbolic Model Checking using SAT procedures instead of BDDs", in *DAC*, 317-320, 1999.

[4] P. Bjesse and K. Claessen, "SAT-based Verification Without State Space Traversal", in *Proc. of Formal Methods in CAD*, 372-389, 2000.

[5] D. Chai and A. Kuehlmann, "A Fast Pseudo-Boolean Constraint Solver", in *DAC*, 830-835, 2003.

[6] N. Creignou, S. Kanna, and M. Sudan, "Complexity Classifications of Boolean Constraint Satisfaction Problems", in *SIAM Press*, 2001.

[7] S. Devadas, K. Keutzer, and J. White, "Estimation of power dissipation in CMOS combinational circuits using Boolean function manipulation", in *IEEE Trans. on CAD*, 11(3), 373-383, 1992.

[8] H. Dixon and M. Ginsberg, "Inference Methods for a Pseudo-Boolean Satisfiability Solver," in *AAAI*, 635-640, 2002.

[9] E. Goldberg and Y. Novikov, "BerkMin: A fast and robust SAT-solver," in *DATE*, 142-149, 2002.

[10] J. Halter and F. Najm, "A gate-level leakage power reduction method for ultra-low-power CMOS circuits," *in Proc. of the IEEE 1997 Custom Integrated Circuits Conference*, 475-478, 1997.

[11] M. S. Hsiao, "Peak power estimation using genetic spot optimization for large VLSI circuits", in *Proc. of the European Design and Test Conf.*, 175-179, 1999.

[12] ILOG CPLEX, *http://www.ilog.com/products/cplex*

[13] S. Manich and J. Figueras, "Maximizing the weighted switching activity in combinational CMOS circuits under variable delay model", in *Proc. of the European Design and Test Conference*, 597-602, 1997.

[14] J. Marques-Silva and K. Sakallah, "GRASP: A Search Algorithm for Propositional Satisfiability", in *IEEE Trans. on Computers*, (48)5, 506-521, 1999.

[15] MCNC Benchmarks, *http://www.cbl.ncsu.edu/CBL_Docs/Bench.htm*

[16] S. Memik and F. Fallah, "Accelerated Boolean Satisfiability-Based Scheduling of Control/Data Flow Graphs for High-Level Synthesis", in *Proc. of the International Conference on Computer Design*, 2002.

[17] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik, "Chaff: Engineering an Efficient SAT Solver", in *DAC*, 530-535, 2001.

[18] F. Najm and M. Zhang, "Extreme delay sensitivity and the worst-case switching activity in VLSI circuits", in *DAC*, 623-627, 1995.

[19] G. Nam, F. A. Aloul, K. A. Sakallah, and R. Rutenbar, "A Comparative Study of Two Boolean Formulations of FPGA Detailed Routing Constraints", in *ISPD*, 222-227, 2001.

[20] M. Pedram, "Power minimization in IC design: principles and applications", in *ACM Trans. on Design Automation of Electronic Syst.*, 1(1), 3-56, 1996.

[21] K. Roy and S. Prasad, "Low-Power CMOS VLSI Circuit Design", *John Wiley & Sons Inc.*, 2000.

[22] E. Sentovich et. al, "SIS: A System for Sequential Circuit Synthesis," *Univ of California-Berkeley, UCB/ERL M92/41*, 1992.

[23] H. Sheini and K. Sakallah, "Pueblo: A Modern Pseudo-Boolean SAT Solver," in *DATE*, vol. 2, 684-685, 2005.

[24] C. Wang and K. Roy, "Maximum power estimation for CMOS circuits using deterministic and statistic approaches", in *Proc. of the 9th Intl. Conference on VLSI Design*, 364-369, 1995.

[25] J. Whittemore, J. Kim, and K. Sakallah, "SATIRE: A New Incremental Satisfiability Engine," in *DAC*, 542-545, 2001.

[26] Q. Wu, Q. Qiu, and M. Pedram, "Estimation of peak power dissipation in VLSI circuits using the limiting distributions of extreme order statistics", in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 20(8), 942 – 956, 2001.

[27] H. Zhang, "SATO: An Efficient Propositional Prover", in *Proc. of the International Conference on Automated Deduction*, 272-275, 1997.