Reducing Power Consumption during Testing of VLSI Circuits by Proper Subsequence Selection and Modification

MICHAEL DIMOPOULOS, PANAGIOTIS LINARDIS*
Department of Informatics
Aristotle University of Thessaloniki
GR-54006 Thessaloniki
GREECE

Abstract: - A method for minimizing power dissipation in CMOS sequential circuits during test application is presented. Initially, the set of test sequences with the transition counts corresponding to each detected fault is mapped on a Transition Covering Matrix. On this matrix three reduction techniques are applied in cycles in order to reduce the size of the selection problem and thus speed-up the subsequence selection process that follows. After the reductions, on the remaining sequences a Branch and Bound method is applied to select proper subsequence parts so as to reduce transitions while maintaining the fault coverage. Additionally, by exploring the don't care conditions of certain circuit inputs better subsequences are generated. Substantial savings in power dissipation were observed in the experimental results.

Key-Words: - Sequential Digital Circuits, Low Power Consumption, Sequence Compaction, Test Generation.

1 Introduction

With the proliferation of portable battery operated devices and the trend toward low power VLSI circuits, the power issues for testing these circuits are becoming increasingly important. In [1] it is shown that the power consumption during test application may be significantly higher than during normal circuit operation. The increased power consumption can cause problems with heat dissipation and this fact must be taken into consideration when testing circuits designed for low-power. Several approaches [4-9, 11] have been proposed for minimizing power (circuit activity) during testing.

In [4] the reduction of the switching activity in the circuit is effected by reordering the initial test vectors, while exploiting the don't care input values that exist in the test set. For the proper reordering *TSP* (Traveling Salesman Problem) techniques are recursively applied.

In [5] the number of circuit transitions is reduced by producing new test patterns, using a PODEM-based [2] ATPG to assign logic values to unassigned primary inputs.

We must note that the above methods [2, 4, 5] are applied to combinational or full scan sequential circuits. For sequential circuits, which are our objective here, it is more complicated to find don't care (Xs) input values since a change of only one value in a test vector may affect the detection of faults for vectors applied later. Also TSP techniques or PODEM based ATPG methods are more time consuming in the sequential circuit case.

In [6] a low power test method is applied to sequential circuits. This method assumes that an external reset signal is available and the test set is composed of several independent sequences (each starting from the reset state). Initially, a large set of test sequences is generated and then a set of sequences is selected by a Genetic Algorithm so that the power consumption is reduced while at the same time the fault coverage is retained.

In [7] the test vectors within a test sequence are modified by inverting the values of the primary inputs one by one, while trying to preserve the original fault coverage. Although this method may be also applied to sets of test sequences it is more beneficial, as was experimentally asserted by us, to first find the don't care input values and then change them.

In [8] the *essential sequences* within the test set are exploited. The initial test set is enriched by addition of new GA generated test sequences by only targeting multiple essential faults. Afterwards a Branch and Bound method is applied in order to select a set of subsequences with reduced number of transitions.

In this paper the power minimization problem, during testing, for CMOS sequential circuits is

modeled as a *set covering* problem [14]. Given a set of independent test sequences (i.e. sequences that start from the unknown state or sequences that start from the reset state) a set of subsequences is generated and from these subsequences are selected those that cover all the initial detected faults with minimal power consumption (minimal number of circuit transitions). The problem of "subsequence selection vs. circuit transitions" is, initially, formulated with the help of a *Transition Covering Matrix* (section 3) on which efficient *reduction rules*, devised for other purpose [10], are applied to simplify the matrix. Then with the help of a Branch and Bound algorithm a set of subsequences is selected (section 4) from the reduced matrix (from the smaller set of candidate subsequences). A final attempt to further reduce the transitions is made by identifying bits in the sequences that may be modified (section 4).

Our proposed method though it uses the formulation of the *Transition Covering Matrix* as in [8] it is different from [8] in that, here, (a) are applied stronger reduction rules that lead to a more compact matrix and (b) instead of the time consuming GA of [6, 8] a more dedicated method is used for generating better sequences (modification of selected bits).

The paper is organized as follows: In sections 2 and 3 the power model is explained and the main lines of our method are analyzed. In section 4 the proposed methodology is presented. In section 5 experimental results are given, supporting the potential of the proposed method.

2 Power Dissipation Model

It is assumed here that the sequential circuit is implemented in CMOS technology and the zero delay fault model [11] is considered. The power dissipation in CMOS circuits can be divided into static, short circuit, leakage and dynamic [12]. Static power dissipation is negligible for correctly designed circuits. The power consumed by leakage currents and short-circuits (during switching) contributes up to 20% to the total power dissipation. The remaining 80% is attributed to the dynamic power dissipation during switching, caused by the transient currents required to charge or discharge the high load capacitances of the CMOS [12].

In CMOS circuits the average dynamic power consumed by a gate g is:

$$P_g = 0.5 V_{dd}^2 C_L E_g(sw) f_{switch} \qquad (1)$$

where V_{dd} is the supply voltage, C_L is the physical capacitance at the output of the gate, Eg(sw) is the average number of output transitions per $1/f_{switch}$ time i.e. the *switching activity factor* [13] of the output of the gate g, and f_{switch} is the clock frequency.

For a test pattern of length N, the power consumption during test application is [6, 13]:

$$P = \frac{1}{2} V_{dd}^2 C_L f_{switch} \sum_{t=1}^{N} \sum_{g} E_g^t(sw) \qquad (2)$$

where $E_g^t(sw)$ is the switching activity factor for gate g at time t. The switching activity factor for our model is:

$$E_g^t(sw) = (val(g)^{t-1} \oplus val(g)^t)$$

with $val(g)^t$ the logic value of gate g at time t. Since the only variable part in (2) is the term:

$$Q = \sum_{t=1}^{N} \sum_{g} E_g^t(sw)$$

i.e. the total number of transitions, this term becomes our quantitative measure of power dissipation.

3 Proposed Methodology

Let the test set $T=[S_1, S_2,\ldots, S_n]$ consisting of the *n* test sequences detecting the m circuit faults from the set $F=[f_1, f_2,\ldots,f_m]$. The problem of test sequence selection for low power dissipation is to find from T a set of subsequences $T_C=[S_{C1}, \ldots, S_{Cn}]$ that will cover the set F with the minimum collective number of transitions.

We define, here, the *transition count* Q(v) of vector v within sequence S as a (monotonically increasing) function counting the transitions (switching changes in the circuit) from the first vector of S up to v. For example in Fig. 1, vectors v_1 and v_2 in sequence S_1 have $Q_1(v_1)=4$ and $Q_1(v_2)=13$ so $Q_1(v_2)>Q_1(v_1)$.

Each sequence may be analyzed into several subsequences, where every one has its own number of transitions e.g. S_3 (Fig. 1) consists of 4 subsequences namely: $S_{31}(v_1, v_2)$ with $Q_3(v_2)=12$, $S_{32}(v_1,..,v_3)$ with $Q_3(v_3)=18$, $S_{33}(v_1,..v_5)$ with $Q_3(v_5)=33$, and $S_{34}(v_1,..v_6)$ with $Q_3(v_6)=41$. Obviously, only subsequences covering at least one fault at their tails are worth considering.

With the analysis into subsequences the selection problem is formulated here as a *set covering* [14] problem. However, the complexity of the problem is high (NP-complete) and it becomes desirable to try, first, to minimize the size of the initial problem. As explained in section 3.1 the proposed formulation makes the present problem formally

similar to other problems [10] for which reduction techniques exist. Following the simplification, a Branch and Bound algorithm is applied to reduce the transition count. Finally, inputs that don't affect fault coverage are identified and a subset of them is modified in order to save transitions. The complete algorithm is presented in section 4.

3.1 Transition Matrix Formulation

Our method will be illustrated with the following example.

Let the test set T=[S_1, S_2, S_3] of Fig. 1 where sequence S_1 covers the faults f_1, f_2, f_4, f_6, f_7, S_2 the faults f_2, f_5, f_6, f_7, and S_3 the faults f_1, f_2, f_3, f_4, f_5.

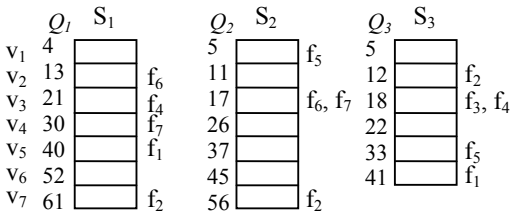

Fig.1. A set of test sequences

For the test set of Fig.1 we build the *Transition Covering Matrix* of Fig.2a, whose element $q_{ij}=Q_i(v_k)$ where vector v_k detects fault f_j (i.e. q_{ij} is the respective number of transitions until f_j is detected). If S_i does not detect f_j then it is set $q_{ij}=q_\infty$ = a very large integer.

	S_1	S_2	S_3
f_1	40	-	41
f_2	61	56	12
f_3	-	-	18
f_4	21	-	18
f_5	-	5	33
f_6	13	17	-
f_7	30	17	-

(a) Initial Matrix

S_1	S_2	S_3
40		23
-	-	-
-	-	-
-	-	-
	5	15
-	-	-
30	17	-

(b) Intermediate Matrix

Fig. 2. Transition Covering Matrix

After the formulation of the Transition Covering Matrix, as in Fig 2a, the problem of selecting a subset of subsequences that cover all faults with the minimum number of transitions is formally similar to the problem presented in [10], where the *detection costs* of [10] are replaced with the *transition counts* in the present case. In [10] a set of efficient *Reduction Rules* were proposed to reduce the size of the Matrix so that the remaining, smaller, problem (matrix) becomes easier to be solved by a Branch and Bound algorithm. Here, similar reduction rules, modified and extended for the present case, are applied to the Transition Covering Matrix as the first phase in our minimization problem. The important characteristic of these rules is that their reductions preserve the optimality of the solution.

Rule 1 (Essentiality)

A column j of the Transition Covering Matrix is an *essential* column if it is the only column that covers a row i (row i is called essential).

For every column j that is identified as essential:

- Set Z_j= max(q_{ij} , row i is essential).
- Remove every row i with $q_{ij} < Z_j$.
- Set $q_{ij}=q_{ij}-Z_j$ when $q_{ij} \geq Z_j$

In this procedure given the initial test set T and the covered fault set F′, the essential faults (if any) and their respective *minimum-transition* detection subsequences are identified.

The cost of applying Rule 1 is linear on the number of rows and gradually becomes smaller as rows are removed.

Rule 2 (Row elimination)

Given rows i and p, row p may be removed, without affecting the optimality of the solution, if and only if:

- For all columns j it is: $q_{ij} \geq q_{pj}$.
- For at least one column k it is: $q_{ik} < q_\infty$.

Proof: We have two cases:

a) The Set of columns J={j, where $q_\infty > q_{ij} \geq q_{pj}$}: i.e. only the columns covering row i. Selection of a column j in order to cover f_i will result in covering of fault f_p with no additional increase of transition counts, because $q_{ij} \geq q_{pj}$.

b) The Set K={k, where $q_\infty = q_{ik} \geq q_{pk}$}: i.e. the columns not covering row i. Since column k, does not cover row i, the only choice to cover row i is to select a column from subset J (case a). Columns from J cover also f_p (row p), so selecting a column from K to cover row p will not lead to a minimum transition count.

The cost of applying Rule 2 on a Transition Covering Matrix with dimensions m.n is $O(m^2n)$ and becomes smaller as the matrix is reduced.

Rule 3 (Set column dominance)

Let column j is covering the set of faults F_j={f_0, f_1, f_2,...,f_n}, having transition counts: $q_{0,j} \leq q_{1,j} \leq q_{2,j} \leq ...q_{n,j}$.

Let the set of columns C={k_0, k_1,..., k_q} ($j \notin C$) is covering at least F_j and let c_i be the minimum transition count for fault f_i within C.

Then we say that set C dominates j if and only if:

- $q_{r,j} \geq \sum_{i=0}^{r} c_i$, for r=0, 1,...,n

In this rule, if column set C dominates column j then column j may be removed without affecting the optimality of the solution.

Proof: From the above relations, we have that every fault and every subset of faults covered by column j may be also covered by a proper combination of subsequences from C with an equal or smaller collective transition count. Therefore, by removing column j solution optimality is retained.

Rule 3 is applied as follows (cost O(mn)):

For j=1 to n

1. Let C consist of the n-1 columns ($j \notin C$): $k_1, k_2,\ldots,k_{n-1}$
2. The n-1 columns are replaced with a temporary column c with elements c_i such that for every fault i covered by the column set C it is c_i = min $\{q_{i,1}, q_{i,2},\ldots,q_{i,n-1}\}$.
3. Column j is checked against c for possible dominance.

Rules 1, 2 and 3 are applied iteratively until a cyclic core is reached (no further reductions are possible). The remaining matrix is the *Reduced Transition Covering Matrix*. The total cost of applying Rules 1, 2 and 3 is $O(m^2n)$.

As an example of applying the rules, from fig 2a we see that fault f_3 is covered only by sequence S_3 and therefore it is an essential fault. We retain subsequence $S_3(v_1,\ldots,v_3)$ with the corresponding transitions q_{33}=18 (Fig. 2b). Next we update the Transition Covering Matrix by setting every element with $q_{i3} \geq q_{33}$ to $q_{13}=q_{13}-q_{33}$ and removing all rows with $q_{13}<q_{33}$ (Fig 2b). By applying Rule 2 (row elimination) fault f_6 is removed due to fault f_7. Next, by applying Rule 3 (set column dominance) column S_1 is removed due to the set of columns S_2, S_3. Now, faults f_1, f_7 become essential and we retain the respective subsequences $S_2(v_1,\ldots,v_3)$ and $S_3(\ldots,v_6)$. Finally, the Reduced Transition Covering Matrix contains $S_2(v_1,\ldots,v_3)$ and $S_3(v_1,\ldots,v_6)$. After restoring the values subtracted due to the removal of essential faults the final transition count is Q=17+(23+18)=58 which, in this case, is the minimum that can be obtained from the matrix alone (no Branch and Bound algorithm is needed for further selection because $S_2(v_1,\ldots,v_3)$ and $S_3(\ldots,v_6)$ are essential so both must be included in the final test set).

4 The *LpTestSeq* algorithm

The goal of our algorithm is to retain fault coverage while minimizing the transition count during testing. The proposed algorithm *LpTestSeq* consists of the following four procedures:

1) Formulation of the Transition Covering Matrix (section 3.1) from the given test sequences by fault simulating each sequence to count the transitions.

2) Reduction of the initial Transition Covering Matrix by applying the Reduction Rules of section 3.1, maintaining the solution optimality of the smaller problem. This procedure produces the Reduced Transition Covering Matrix. If the Reduced Transition Covering Matrix is empty then the next procedure, i.e. procedure 3), is skipped.

3) A Branch and Bound algorithm [18] is applied on the reduced Transition Covering Matrix. This Branch and Bound algorithm tries to select a subset of subsequences, which cover all faults with the smallest possible number of transitions. Since the size of the matrix is now smaller (reduced) the cost of the Branch and Bound algorithm is reasonable. However, a flag is raised whenever the Branch and Bound method fails to produce a solution within a time limit of 1 min.

4) Modification of the selected subsequences in order to generate better subsequences. In this procedure inputs that don't affect fault coverage are identified [15] and a subset of them is modified in an attempt to further minimize transition count, as is explained in subsection 4.1.

4.1 Modification of subsequences

A bit in a selected subsequence is randomly selected and inverted and then the modified subsequence is fault simulated to establish if (a) there are no faults sensitive to this bit position and (b) the transition count becomes smaller.

Because the simulation cost, in our case, may be considerable we apply the method from [15] according to which the set of subsequences is classified into "easy" and "difficult" ones, where a subsequence is considered as "difficult" if it covers faults covered only by a very limited number of subsequences. In our case the essential subsequences are the most "difficult" ones.

To further speed up the process, the condition that a modified test subsequence should cover the same faults is relaxed here. The resulting (modified) test subsequence is acceptable even when the number of faults a sequence detects is decreased, as long as the deleted faults are covered by some other test sequence in the solution (i.e. the total fault

coverage is maintained).

The actual process, as applied here, proceeds as follows: First are processed the subsequences that cover essential faults, then the remaining subsequences in order of decreasing "difficulty". In case of a tie, the subsequences are processed in decreasing number of transitions. A modified subsequence, after a user-defined limit on the performed number of fault simulations becomes accepted if: a) the total fault coverage is maintained, b) it has a reduced number of transitions with respect to the original (unmodified) subsequence. The accepted (modified) subsequence is added to the solution together with the covered faults which are crossed out from the remaining fault list. In this way, the remaining subsequences need to cover fewer faults and the fault simulation cost, which is analogous to the size of the fault list, is reduced.

5 Experimental Results

Our algorithm *LpTestSeq* has been implemented in C. The efficiency of the algorithm was measured by running the ISCAS'89 benchmark circuits [16] and several synthesized benchmarks circuits from Illinois [17] on a Pentium PC with 256 Mb.

In Table 1 are presented the initial test sets (#seq. is the number of sequences, #vectors is the total amount of vectors and #faults are the detected faults) that were obtained by the method of [3]. Under columns #transit are presented for every test set the respective number of total transitions.

As a first experiment we computed the size of the Reduced Transition Covering Matrix for the method of [8] (column 'Reduced Transition Matrix' under LpCompSeq) by applying the essentiality reductions to the initial test sets, without considering the new GA generated test sequences. The corresponding sizes of the Reduced Transition Covering Matrix when the proposed reduction rules (section 3.1) are applied to the initial test sets are presented in column 'Reduced Transition Matrix' under LpTestSeq. It is noted that both methods (LpCompSeq and LpTestSeq) start from the same matrices with the same essential faults (column #EssS denotes the number of essential sequences within the test set). We see that the application of the proposed, here, reductions results in very small Transition Matrices and thus speeds-up the Branch and Bound Subsequence selection step (in most examples the reduced matrix becomes 0x0 eliminating the need for the Brach and Bound step).

The final results for LpTestSeq and LpCompSeq [8] are presented, respectively, under columns '#transit'. Both methods achieve significant reductions in the number of transitions, when compared to the initial test set, but LpTestSeq is slightly better (average 59.5%) compared to LpCompSeq (average 55.1%).

Table 1. Problem size and results for LpTestSeq

	Initial Test Set					LpCompSeq [8]		LpTestSeq	
						Reduced Transit Covering Matrix		Reduced Transit Covering Matrix	
circuit	**#seqs**	**#vectors**	**#faults**	**#transit**	**#EssS**	**#flts x #seqs**	**#transit**	**#flts x #seqs**	**#transit**
s298	8	402	265	9 674	3	3 x 4	2 733	0 x 0	2 325
s344	10	183	329	7 415	8	2 x 2	3 166	0 x 0	2 836
s349	9	137	335	5 419	7	2 x 3	2 253	0 x 0	2 021
s382	3	844	357	22 764	3	0 x 0	12 141	0 x 0	9 845
s386	7	606	314	20 952	6	0 x 0	18 907	0 x 0	15 770
s641	17	468	404	34 607	13	1 x 3	17 086	0 x 0	14 476
s713	20	418	476	33 619	14	2 x 4	10 806	0 x 0	9 874
s1196	51	1 866	1 223	243 048	40	5 x 8	135 992	0 x 0	112 260
s1238	87	1 501	1 233	171 025	57	22 x 29	69 104	0 x 0	64 040
s1423	29	1 519	1 314	178 564	24	2 x 2	117 508	0 x 0	100 343
s1488	35	1 258	1 439	186 140	23	5 x 6	103 109	0 x 0	97 223
s1494	23	1 055	1 447	137 484	17	3 x 4	92 528	0 x 0	82 285
s5378	30	1 008	3 091	441 676	20	9 x 10	259 003	3 x 3	212 403
s6669	35	621	6 670	501 828	15	15 x 15	184 834	9 x 10	170 360
s4863	38	1 643	4 615	1 442 768	23	20 x 19	510 550	2 x 3	510 550
s3384	25	813	3 154	451 249	20	2 x 5	232 244	0 x 0	200 313
s3330	60	1 606	2 094	552 721	40	1 x 3	264 720	0 x 0	224 510
s3271	54	2 152	3 227	1 334 128	21	30 x 33	554 375	5 x 7	508 579
am2910	59	1 339	2 101	153 736	46	4 x 6	93 863	3 x 4	80 445
div16	37	1 155	1 702	165 288	11	22 x 21	48 310	7 x 7	45 120
mult16	22	434	1 465	32 697	11	7 x 9	20 036	0 x 0	17 360
pcont2	14	245	6 815	197 231	10	21 x 7	82 698	6 x 5	76 480

6 Conclusion

The problem of reducing power dissipation during the testing of CMOS sequential circuits is formulated as a set-covering problem. The test sequences with the corresponding transition counts (switching changes in the circuit), which are proportional to the induced power dissipation, are set into the form of a *Transition Covering Matrix*. This matrix formulation is shown to make the problem formally similar to other problems for which techniques for reducing the size of the initial problem (matrix) have been devised, while these techniques preserve the optimality of the solution.

After the reduction of the size of the initial problem, with a cyclic application of the proposed Reduction Rules, a Branch and Bound algorithm is applied to solve the smaller problem. A further minimization of the circuit transitions is obtained by identifying don't care conditions on certain inputs of the circuit and used to generate better sequences.

Experimental results support the usefulness of the proposed method.

References:

[1] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices", in 11h IEEE VLSI Test Symposium, April 1993, pp. 4-9.

[2] M. Abramovici, M. Breuer, A. Friedman, "Digital Systems Testing and Testable Design", *IEEE Press*, 1990.

[3] M. Dimopoulos, P. Linardis, "Improving a GA-based ATPG for Sequential Circuits by Exploiting Dynamically Generated Essential Sequences", in "Advances in Scientific Computing, Computational Intelligence and Applications", *Ed. N. Mastorakis etc, WSES Press, 2001*, pp 373-377.

[4] P. Flores, J.Costa, H. Neto, J. Monterio, and J. Marquessilva, "Assignment and Reordering of Incompletely Specified Pattern Sequences Targeting Minimum Power Dissipation", in 12th International Conference on VLSI Design, January 1999, pp. 37-41.

[5] S. Wang, S. Gupta, "ATPG for Heat Dissipation Minimization during Scan Testing", in 34th ACM/IEEE Design Automation Conference, July 1997, pp. 614-619.

[6] F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda, "A Test Pattern Generation Methodology for Low Power Consumption", in 16th IEEE VLSI Test Symposium, April 1998, pp. 453-457.

[7] Yoshinobu Higami, Shin-ya Kobayashi, and Yuzo Takamatsu, "Modifying Test Vectors for Reducing Power Dissipation in CMOS Circuits", in 1st IEEE International Workshop on Electronic Design, Test Applications (DELTA), 2002.

[8] M. Dimopoulos, P. Linardis, "Improved Selection of Test SubSequences in Sequential Circuits for Reduced Power Consumption", Proc IMACS/IEEE - The 7th Int. MultiConference on: Circuits, Systems, Communications and Computers (IMACS/IEEE-CSCC'2003), Corfu Greece, 7-10 July 2003.

[9] S. Wang, S. Gupta, "ATPG for Heat Dissipation Minimization during Test Application", in Proc. IEEE International Test Conference, October 1994, pp. 250-258.

[10] M. Dimopoulos, P. Linardis, "Accelerating the Compaction of Test Sequences in Sequential Circuits through Problem Size Reduction", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 22, no. 10, pp. 1443-1449, October 2003.

[11] A. Gosh, S. Devadas, K. Kuetzer, J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits", in 29th ACM/IEEE DAC, June 1992, pp. 253-259.

[12] A. P. Chandrakasan, R. W. Brodersen, "Low power digital CMOS design", Kluwer Academic Publishers, 1995.

[13] F. N. Najm, "Transition Density, a Stochastic Measure of Activity in Digital Circuits", in 28th ACM/IEEE DAC, June 1991, pp.644-649.

[14] O. Coudert, *"Two-level logic minimization: An overview"*, Integration, VLSI journal, Oct. 1994, pp. 97-140.

[15] Y. Higami, S. Kobayashi, Y. Takamatsu, S. Kajihara, I. Pomeranz, "A Method to Find Don't Care Values in Test Sequences for Sequential Circuits", *Proc. of the ICCD, 2003*, pp. 397-399.

[16] F. Brglez, D. Bryan and K. Kozminski, "Combinational profiles of sequential benchmark circuits", *Int. Symposium on Circuits and Systems*, 1989, pp. 1929-1934.

[17] E. M. Rudnick, "Simulation-based techniques for sequential circuit testing", Ph.D. dissertation, Dept. Electrical and Computer Engineering, *Tech. Report CRHC-94-14/UILU-ENG-94-2229*, University of Illinois, August 1994.

[18] O. Coudert, "Two-level logic minimization: An overview", Integration, VLSI journal, Oct. 1994, pp. 97-140.