Technology Mapping for Hot-Carrier Reliability Enhancement*

Zhan Chen and Israel Koren

Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003

ABSTRACT

As semiconductor devices enter the deep sub-micron era, reliability has become a major issue and challenge in VLSI design. Among all the failure mechanisms, hot-carrier effect is one of those which have the most significant impact on the long-term reliability of high-density VLSI circuits. In this paper, we address the problem of minimizing hot-carrier effect during the technology mapping stage of VLSI logic synthesis. We first present a logic-level hot-carrier model, and then, based on this model, we propose a technology mapping algorithm for hot-carrier effect minimization. The proposed algorithm has been implemented in the framework of the Berkeley logic optimization package SIS. Our results show that an average of 29.1% decrease in hot-carrier effect can be achieved by carefully choosing logic gates from cell libraries to implement given logic functions for a set of benchmarks. It has also been observed that the best design for hot-carrier effect minimization does not necessarily coincide with the best design for low power, which has long been considered as a rough measure for VLSI reliability.

Keywords: Hot-Carrier Effect, Design for Reliability, Technology Mapping.

1. INTRODUCTION

Reliability has become a major issue and challenge in the design and manufacturing of next generation deep-submicron VLSI circuits.[3,6,21,28,29] Among all the failure mechanisms, hot-carrier effect (HCE) is one of those which have significant impact on the long-term reliability of high-density VLSI circuits.[3,5,6,18] The hot-carrier-induced damage in MOS transistors is caused by the injection of high-energy electrons and holes into the gate oxide near the drain region. Those injected carriers may be trapped in the oxide, which results in the degradation of the MOS transistor characteristics and may lead to the failure of the circuit. Previous research in this area includes HCE failure mechanism analysis,[5] and based on this, a large number of HCE simulation and estimation tools have been developed.[7,9,12,22,26] Several techniques have also been proposed to improve hot-carrier reliability in various stages of the VLSI design. For example, at the device level, Takeda et al.[20] found that a lightly doped drain can be used to offset HCE. At the circuit level, hot-carrier resistant redesign techniques have been developed by Li et al..[13] Leblebici has presented some design considerations for hot-carrier reliability enhancement in CMOS circuits.[11] At the switching level, Dasgupta et. al[1] introduced methods to improve hot-carrier reliability by reordering inputs to logic gates and re-sizing transistors. At the logic level, Roy et al.[17] proposed algorithms for factoring logic expressions during multi-level logic optimization to reduce hot-carrier susceptibility.

In this paper, we address the problem of minimizing HCE during the technology-dependent stage of logic synthesis. The advantage of dealing with HCE at this stage instead of at other technology-independent stages of logic synthesis, is that various gate information like delay and loading capacitance, is available and therefore a more accurate reliability model can be applied. In our research, it is assumed that a logic network has already been optimized and now we need to map the optimized network onto a cell library while keeping the HCE of this mapped circuit as small as possible. Traditionally, the objective of technology mapping is area minimization or performance optimization.[2,8,19,24] Recently, there were several reports on technology mapping techniques for low power.[14,23,25] It has long been believed that a power optimized logic network would also be reliability optimized since both power and reliability measures reflect the use of the circuit. However, our results show that these two objectives do not necessarily yield the same result, i.e., the best design for reliability is not necessarily the best design for low power.

The paper is organized as follows. Section 2 presents the problem of logic-level hot-carrier measurement. The cost function for technology mapping for hot-carrier reliability is derived from this measurement technique. Section 3 describes technology mapping techniques for hot-carrier reliability. In this section we first present an exact algorithm,

*Support in part by NSF under contract MIP-9305912.

using dynamic programming, for tree mapping, then extensions of tree mapping to handle non-tree circuits are studied. Experimental results are presented in Section 4 on a large set of benchmark circuits indicating a potential for reduction in HCE by an optimal technology mapping. Conclusions are summarized in Section 5.

2. LOGIC-LEVEL HCE MEASURE

Hot-carrier effect in an MOS transistor is caused by the processes of charge trapping in the oxide and/or interface trap generation at the Si/SiO_2 interface. These processes result in a shift in the threshold voltage as well as degradation in the transconductance and electron mobility in the channel. For current semiconductor process technologies, hot-carrier induced degradation is much more severe in NMOS transistors than in PMOS.[29] It has been shown that hot-carrier effect in an NMOS transistor is dominated by interface trap generation which occurs mostly when the transistor is operating in or near the saturation region,[5,27] and the relative damage in NMOS transistors can be determined by the *bond-breaking current*, I_{BB}, which is defined as[5]

$$I_{BB} = (1/W_n) I_{SUB}^m / I_{DS}^{m-1} \tag{1}$$

where W_n is the width of the transistor, I_{SUB} and I_{DS} are substrate current and drain current, respectively, and $m \simeq 3$.

I_{BB} can be expressed as a function of time if a ramp-type input signal is applied to the transistor[10,11]

$$I_{BB}(t) = K_1 (at)^2 (V_{DD} - \frac{1}{3}\mu_n C_{ox} (\frac{1}{2} L_n) \frac{W_n}{C_L} a^2 t^3 - at)^3 \cdot exp(\frac{-K_2}{V_{DD} - \frac{1}{3}\mu_n C_{ox} (\frac{1}{2} L_n) \frac{W_n}{C_L} a^2 t^3 - at}) \tag{2}$$

where C_L is the output capacitance, K_1 and K_2 are process-dependent constants, μ_n is the electron mobility, C_{ox} is the gate oxide capacitance, L_n is the length of the transistor, a is the slope of the input signal, and V_{DD} is the power supply voltage.

Notice that input signal slope a and output load C_L define the environment under which the transistor operates while all other parameters are intrinsic to the transistor.

Given an NMOS transistor, the average bond-breaking current over one cycle period, $\overline{I_{BB}}$, is found to be a simple expression of a and C_L [10]

$$\overline{I_{BB}} = A \cdot a^{-0.8} \cdot C_L^{0.3} \tag{3}$$

where A is a constant whose value depends on the transistor geometry and the manufacturing process.

The degradation of a logic gate caused by HCE is assumed to be equal to that of the most susceptible NMOS transistor in the gate. So, in a CMOS inverter G_i, the HCE degradation measure can be represented as

$$\begin{aligned} HCE(G_i) &= \overline{I_{BB}} \cdot A_{G_i} \\ &= A_{G_i} \cdot a_{G_i}^{-0.8} \cdot C_{G_i}^{0.3} \end{aligned} \tag{4}$$

where a_{G_i} is the input signal slope, C_{G_i} is the output load and A_{G_i} is the gate switching rate. This HCE degradation measure has a unit of Ampere/second.

Figure 1 shows the degradation of an inverter as a function of the input slope and output load. From this figure and equation (4) we can see that reducing C_L or increasing a can improve the hot-carrier reliability of an inverter.

Complex gates can be reduced to inverter circuits for HCE analysis. Take a 2-input NOR gate for example. Suppose that the inputs to both NMOS transistors are at logic 0 at first. If only one of them switches on, the current will flow through that transistor and cause hot-carrier damage. If the two transistors switch on simultaneously, the current as well as the damage, will be shared by the two transistors. Accurate estimation of HCE degradation in the NOR gate requires detailed information about the switching rates and timing of the two input signals, but we can use the following approximation in practice. Assuming that the output of the NOR gate and the input of NMOS transistor i ($i = 1$ or 2) have signal switching rates of $A_{G_{nor}}$ and A_i, respectively, the two transistors can be considered as switching on one at a time with probabilities roughly equal to $\frac{A_1}{A_1+A_2} \cdot A_{G_{nor}}$ and $\frac{A_2}{A_1+A_2} \cdot A_{G_{nor}}$,

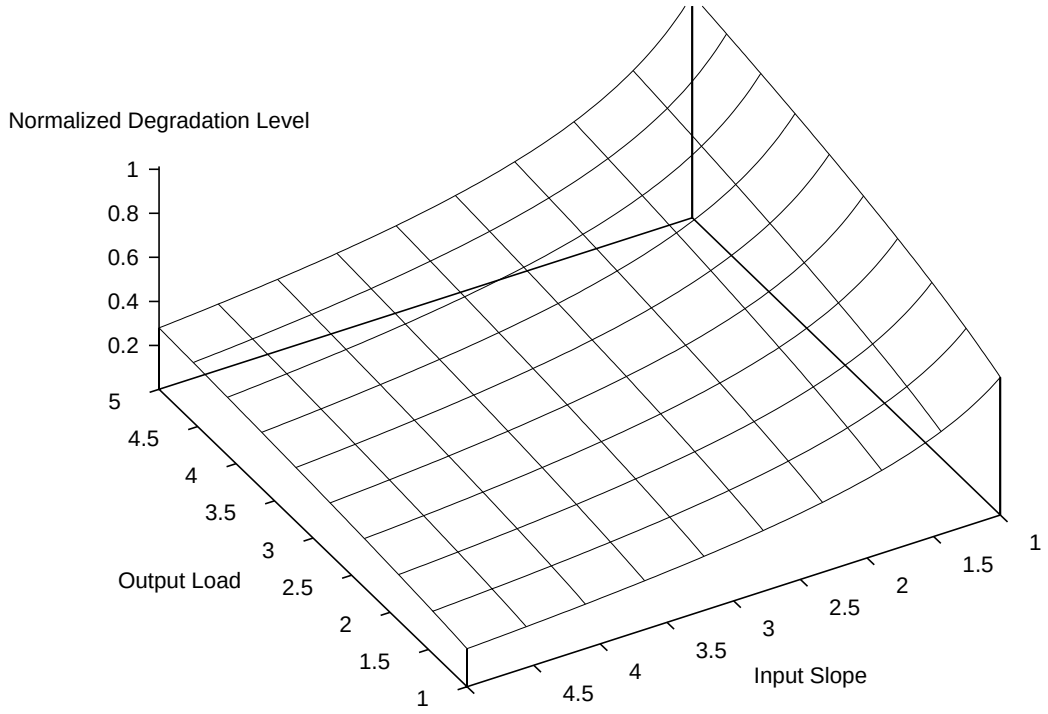

Figure 1. Degradation of an inverter as a function of input signal slope a and output capacitance C_L

which are also the probabilities when these transistors take all the hot-carrier damage. So we can use the following expression to calculate the hot-carrier degradation in a NOR gate

$$\begin{aligned} HCE(G_{nor}) &= max(\frac{A_1}{A_1+A_2} \cdot a_1^{-0.8} \cdot C_{G_{nor}}^{0.3} \cdot A_{G_{nor}}, \frac{A_2}{A_1+A_2} \cdot a_2^{-0.8} \cdot C_{G_{nor}}^{0.3} \cdot A_{G_{nor}}) \\ &= max(S_1 \cdot HCE(G_1), S_2 \cdot HCE(G_2)) \end{aligned} \quad (5)$$

where A_i and a_i are the transition rate and slope for input i $(i = 1, 2)$, respectively, $A_{G_{nor}}$ is the output signal switching rate, $C_{G_{nor}}$ is the output load of the NOR gate, S_i is a scalar which is equal to $\frac{A_i}{A_1+A_2}$ $(i = 1, 2)$, and $HCE(G_1)$ and $HCE(G_2)$ are the hot carrier degradation in inverters G_1 and G_2, respectively. G_1 and G_2 are the equivalent inverters used in the HCE analysis for the NOR gate. Thus, a NOR gate can be divided into two separate inverters as shown in Figure 2, and the HCE of the NOR gate can be represented by the worst of the two inverters multiplied by a constant determined by the signal transition rates of the two input signals. NAND gates and other complex gates can also be reduced to inverters for HCE analysis[22] in a similar way.

The HCE in a logic gate is determined by the switching rate of the input and output signals, the slope of input signals and the output capacitance. We now consider the calculation of these three parameters. The output capacitance can be easily obtained by summing up the input capacitance for each fanout branch. The switching rate of a node in a logic network is determined by the switching rates at the primary inputs and the delay on paths that lead to the node. In this paper, we adopt the zero-delay model. Under this model, the switching rate A_{G_i} is a product of N, the number of clock cycles per unit time, and PT_{G_i}, the transition probability in one clock cycle. If we further assume that all the primary inputs are statistically independent, then the signal transition probability of a specific node can be calculated as $PT_{G_i} = 2P_{G_i}(1 - P_{G_i})$,[16] where P_{G_i} is the signal probability at the output of gate G_i and it is defined as the probability that the output of G_i is equal to logic 1. A modified binary decision diagram (BDD) based on a procedure proposed by Najm[15] can be used to calculate P_{G_i}.

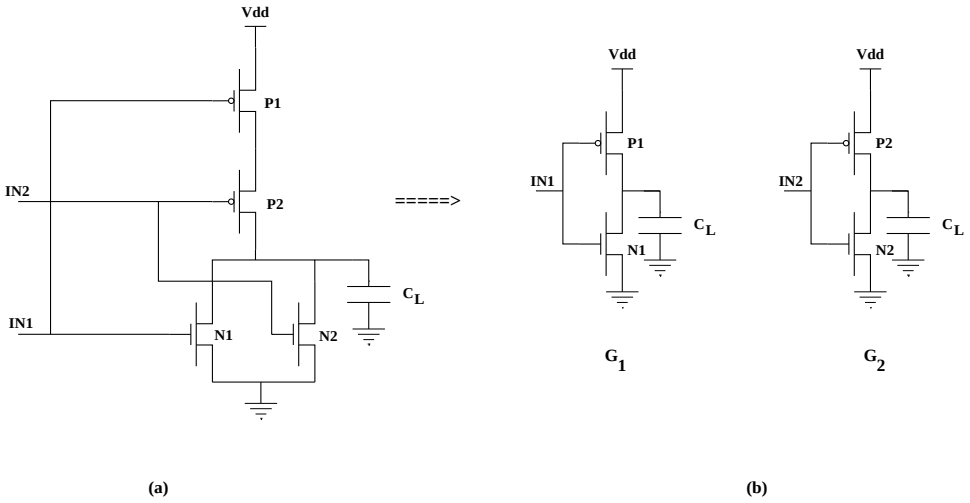

Figure 2. A NOR gate (a) is divided into two inverters (b) for HCE analysis

The input slope of a gate is determined by its input capacitance, the driving capacity of the previous gate and the input signal slope of the previous gate. For a ramp input signal ($V = at, t \leq V_{DD}/a$), the output signal slew rate is[4]

$$a_{output} = (\frac{t_{P,step}}{V_{DD}} + \frac{1 + \frac{2V_{TN}}{V_{DD}}}{6a})^{-1} \tag{6}$$

where a_{output} is the output signal slope, $t_{P,step}$ is the step response delay of this gate, and V_{TN} is the threshold voltage of an NMOS transistor.

Since we intend to compare the solutions for hot-carrier reliability and those for power minimization, we now consider the power consumption of CMOS logic gates. In CMOS circuits, the charging/discharging current is dominant and the leakage current and direct-path circuit current only play a limited role. The drain current of a transistor is therefore mainly determined by the switching rate of the output signal and the load capacitance. The same rule applies to the current of a logic gate G_i, i.e., the gate current I_{G_i} is a function of its output switching rate A_{G_i} and its output load C_{G_i}[16]:

$$I_{G_i} = \frac{1}{2} V_{DD}\, C_{G_i}\, A_{G_i} \tag{7}$$

The power consumed by a single gate is equal to $V_{DD} I_{G_i}$ and it can be calculated as

$$POWER(G_i) = I_{G_i} V_{DD} = \frac{1}{2} V_{DD}^2\, C_{G_i}\, A_{G_i} \tag{8}$$

The total power consumed by a circuit is the sum of the power consumption for all logic gates in the circuit

$$\begin{aligned} POWER &= \sum_{G_i \in circuit} \frac{1}{2} V_{DD}^2\, A_{G_i}\, C_{G_i} \\ &= K \sum_{G_i \in circuit} A_{G_i}\, C_{G_i} \end{aligned} \tag{9}$$

where K is a constant equal to $\frac{1}{2} V_{DD}^2$.

In summary, in technology mapping for hot-carrier reliability enhancement, our objective function is

$$Min \;\; Max\, (A_{G_i} \cdot a_{G_i}^{-0.8} \cdot C_{G_i}^{0.3}) \quad \text{for all gates } G_i \tag{10}$$

while in power minimization, the objective function is

$$Min \sum_{G_i \in circuit} A_{G_i} \cdot C_{G_i} \tag{11}$$

That is to say, in power minimization, we attempt to minimize the average current for all gates, while in hot-carrier reliability enhancement, we target only those gates with the worst hot-carrier reliability. We will illustrate later that the difference in their objective functions leads to different optimal solutions.

3. MAPPING FOR HOT-CARRIER RELIABILITY

The general technology mapping problem can be formulated as follows: given a Boolean network, which is usually represented as a directed acyclic graph (DAG), and a target cell library, find a binding of nodes in the network to cells in the library such that some predefined cost functions are optimized. To facilitate the mapping process, a canonical representation (subject DAG) is created for the Boolean equations using the base functions, which usually consist of 2-input NAND/NOR and inverters, and canonical representations (patterns) are also obtained for each of the gates in the cell library using the same base functions. Then, we try to cover all the nodes in the subject DAG by using the patterns in the library to optimize the cost function of hot-carrier reliability. Since DAG-mapping is NP-hard, we therefore study first tree-mapping, a sub-problem of DAG-mapping, and then extend our tree-mapping algorithm to DAG-mapping.

3.1. Tree mapping

In tree mapping, the Boolean network to be mapped as well as all the gates in the cell library are represented by trees. The use of tree mapping for technology mapping was originally proposed by Keutzer[8] and this was mainly motivated by the existence of efficient dynamic programming algorithms for optimum tree mapping.

To minimize hot-carrier effect, we follow a dynamic programming approach. Given a match, m, to a node n in the subject graph, $HCE(m,n)$, the hot-carrier reliability cost of this match is

$$HCE(m,n) = Max\ (A_{G_m} a_{G_m}^{-0.8} C_{G_m}^{0.3},\ MIN\_HCE(v_i)) \quad \text{for all } v_i \in \text{inputs of } G_m \tag{12}$$

where v_i are the nodes in the subject graph input to the match m and $MIN\_HCE(v_i)$ is the match with minimum hot-carrier cost at node v_i.

The first item, $A_{G_m} a_{G_m}^{-0.8} C_{G_m}^{0.3} = HCE(G_m)$, in equation (12) represents the HCE cost of the current mapping for this gate. However its value cannot be known before the output capacitance of this gate is available. The output slope, which turns to be the input slope for the gate in the next stage, also depends on C_{G_m}. The value of C_{G_m} can only be obtained after the fanout node has been mapped. Without a C_{G_m} value it is difficult to decide which mapping is the best for HCE. There are several solutions to this problem. One simple solution is to assume that the output capacitance is equal to a constant. For example, we can use the minimum input capacitance for all the gates in the library as the output capacitance at each step of the tree-mapping. Obviously, this approach can introduce errors and in some cases, especially when we have a a rich cell library with a lot of gates and a large range of input capacitance, the error introduced by this simple approach may be unacceptable. Another approach is opposite to the previous one and it calculates the HCE value and output slope at node i for every possible input capacitance in the library. Though this can solve the problem caused by the previous approach, the high computation and memory overhead make this option unattractive in practice. We adopted a different approach. Instead of considering every possible input capacitance value, we group them into several sets and use an average value to represent the capacitance in each set. For instance in *lib2.genlib*, a cell library which is part of the SIS package, the input pin capacitance ranges from 0.0541 units to 0.1897 units. By dividing this into 3 sets and using 0.0659, 0.1244, and 0.1632 as reference values for these sets, we find that the error by adopting this kind of approximation is quite small.

Our optimal tree mapping algorithm for minimum HCE first traverses the tree in topological order from the leaves to the root, visiting each node once, finding all possible mappings and computing the HCE cost and the output slope for different output capacitances. Notice that we cannot just minimize the HCE value at each node, since this may

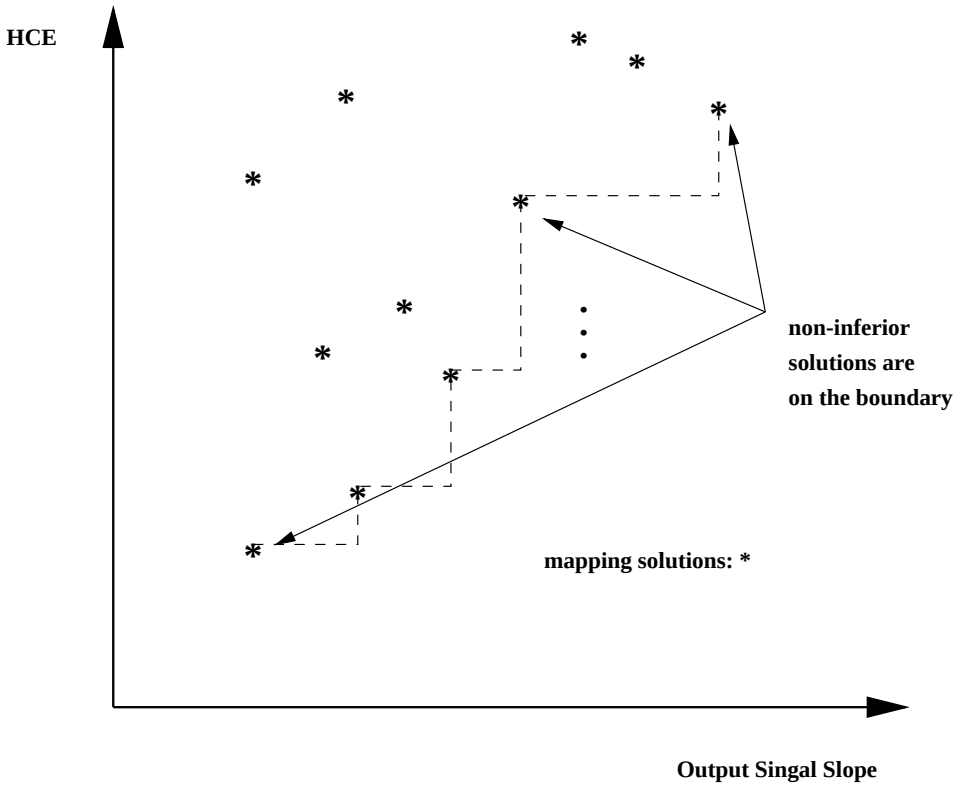

Figure 3. Non-inferior mappings are those on the lower-right boundary of the set of all possible mapping solutions.

result in a large output slope and cause a very large HCE in later stages of the logic network. Thus, we need to keep all the non-inferior solutions at each node. We say that a mapping m_a is inferior to a mapping m_b at node n iff

$$HCE(m_a, n) \geq HCE(m_b, n) \tag{13}$$

$$and \quad \text{output slope of } m_a \leq \text{output slope of } m_b \tag{14}$$

The inequality relation shown in (13) means that solution m_b is better than m_a for all the nodes that have been mapped, while (14) means that m_b is also better than m_a for the nodes that are to be mapped since m_b can provide a faster output signal slew rate. The non-inferior mappings can be obtained by applying a simple scheme. Assuming we have a set of possible mappings at a node, we use a pair of numbers to represent a mapping, where the pair of numbers are the values for HCE and output signal slope. A mapping can be represented as a point in a 2-dimensional plane if we use the x-axis for the HCE value and the y-axis for the output signal value. Only those points on the lower-right boundary are non-inferior and this is illustrated in Figure 3.

After we have obtained all the possible non-inferior mappings at each node, we then traverse the tree once again from the root to the leaves to determine the best mapping solution.

Sometimes, the hot-carrier reliability is not the only objective function we want to optimize. For example, we may want to get a design with high hot-carrier reliability as well as low power. This objective can be achieved by first doing technology mapping for hot-carrier reliability to get the optimal HCE value, then optimizing the logic network again for low power using the optimal HCE value as a constraint.

Tree mapping for low power is similar to that for hot-carrier reliability. In low power mapping, the cost for selecting a match at a node is[14,23,25]

$$POWER(m, n) = A_{G_m}\, C_{G_m} + \sum_{v_i \in inputs\ of\ G_m} MIN\_POWER(v_i) \tag{15}$$

where A_{G_m} C_{G_m} is the power contribution of gate m when implementing node n, and the term $\sum_{v_i \in inputs\ of\ G_m} MIN\_POWER(v_i)$ is the sum of the minimum power cost for the corresponding subtrees rooted at the input pins of G_m. At each node, we select the match that can minimize the power cost function and store it. The hot-carrier reliability constraint is maintained by selecting only the minimum power-cost matches that satisfy the constraint at each node.

3.2. DAG mapping

Most practical circuits are DAG's but not trees. But the problem of mapping a DAG is NP-hard. The main problem here is that the best mappings at the inputs of a matching gate are no longer independent of each other. Due to this reason, no exact polynomial algorithms are available, and we need to resort to heuristics. One heuristic is to decompose the DAG into a number of trees and then do a tree mapping for each tree separately. This heuristic is quite simple and easy to implement but its disadvantage is that it does not allow mapping across tree boundaries and thus tree overlapping can not occur. To get a better result, we adopted an approach that uses heuristics similar to those used in SIS delay mapping.[19,24] In this approach, we avoid decomposing the DAG into trees by not restricting the algorithm to trees. The library can also have non-tree patterns, such as XOR gate, and the subject graph can be a general DAG instead of a forest of trees. Then, starting from the primary inputs, we traverse the DAG subject graph in a depth first manner. At each node, all matched patterns including those which have multiple-fanout nodes are evaluated and the minimum-cost match is stored as in a tree mapping. Tree overlapping is sometimes allowed,[19,24] and it is applied wherever it can improve the hot-carrier reliability. Though tree overlapping may increase circuit area, the overhead is minimal since this technique is used in a few multi-fanout points where hot-carrier reliability becomes the bottleneck for the whole circuit.

DAG mapping for low power under a hot-carrier reliability constraint follows a similar approach as in tree mapping. In each node, we try to select a match with minimum power cost while controlling the hot-carrier measure not to exceed the pre-calculated value.

4. NUMERICAL RESULTS

The tree mapping and DAG mapping algorithms for hot-carrier reliability enhancement have been implemented and integrated with the technology mapping package in SIS.[19] Table 1 shows some results of our algorithms on the MCNC combinational benchmark examples. Column 2 to column 4 are the area, power and HCE measures, respectively, for the circuits that have been mapped for power. The area, power and HCE measures for the circuits optimized for hot-carrier reliability using our DAG mapping algorithm are shown in column 5 to column 7, respectively. They are represented as percentage of their corresponding values in power minimization. The last three columns show the results when a tree-mapping approach is applied for hot-carrier minimization. These results are also represented as percentage values as in DAG-mapping. In each case, the signal probability of each input is assumed to be 0.5 and all the input signals are assumed to be statistically independent. The signal probability at any internal node in the circuit is calculated using the BDD-package in SIS. The circuits were initially synthesized using a standard logic optimization script. They are then mapped using the hot-carrier reliability enhancement mapper and the low power mapper described in the previous section. The library used is a subset of *lib2.genlib*, a cell library included in SIS. In the case of hot-carrier reliability optimization, we first map the circuit for hot-carrier reliability and obtain the best value for this objective. Then, we use this value as a constraint, and re-do the technology mapping of the original logic network to optimize power. The HCE measure of a circuit is assumed to be equal to that of the most susceptible logic gate and is determined using equation (10). Since in the *lib2.genlib* library there is no information about the NMOS transistor size, we assume that they are the same. The power measure for a circuit is determined using equation (11). Both HCE and power in Table 1 are relative values. On an average, the percentage improvement in the HCE when compared to power optimized designs is 29.1% if a DAG-mapping is applied. The average penalties we paid in terms of area and power are 4.7% and 6.2%, respectively. Compared with tree mapping, our DAG mapping algorithm can achieve an extra 12.8% reduction in HCE with a cost of 2.4% more area and 2.1% more power consumption. From this, we can see that the DAG-mapping algorithm is usually a better choice than the tree-mapping algorithm. From these results, it can also be seen that the circuits optimized for hot-carrier reliability are not necessarily identical to the circuits optimized for power. This is because in power minimization, we want to reduce the average drain current for every gate, while in hot-carrier reliability enhancement, we only target those gates with the most serious hot-carrier effect.

Examples	Low Power			Hot-Carrier Reliability Enhancement					
				DAC Mapping			Tree Mapping		
	area	power	HCE	area(%)	power(%)	HCE(%)	area(%)	power(%)	HCE(%)
alu2	316350	17.699	1.403	104.7	104.8	62.7	100.1	104.0	70.4
alu4	606068	31.235	1.268	108.3	103.6	77.3	104.0	103.4	89.4
apex6	593240	33.965	1.157	107.1	104.6	78.9	102.6	105.0	94.2
b9	113752	8.810	0.622	104.8	106.5	64.7	101.1	107.2	74.7
c8	152048	11.842	0.710	108.2	104.7	77.4	102.5	103.1	90.1
C432	205518	11.741	0.590	104.9	107.5	73.9	104.1	103.9	90.9
C1355	571434	30.016	0.399	101.4	107.0	76.4	101.1	101.8	85.2
C1908	506604	33.163	0.882	106.9	104.5	63.2	104.1	103.3	84.1
C2670	774388	53.028	1.377	104.0	104.7	67.1	102.6	103.8	84.6
cc	67960	3.814	0.508	104.6	101.6	76.8	105.7	101.4	89.3
cm138a	31528	1.529	0.299	103.5	105.4	71.3	103.3	104.9	77.6
cm162a	46812	2.593	0.384	111.2	105.9	64.6	104.1	105.2	78.5
cmb	53060	3.291	0.288	108.6	104.6	56.6	105.5	103.8	80.1
cordic	10864	5.146	0.636	105.7	115.6	64.0	103.2	107.3	81.7
count	156820	6.516	0.879	102.8	106.7	80.1	102.3	105.4	75.3
dalu	728424	97.539	1.112	102.6	109.8	72.0	101.2	102.1	89.2
f51m	140692	10.927	0.798	104.9	110.1	70.3	104.4	106.5	82.6
frg1	121040	8.953	0.411	105.1	108.7	76.2	103.9	104.6	89.3
i3	94794	10.080	0.464	106.7	108.3	72.4	100.7	105.1	80.9
i4	196798	15.736	0.114	108.0	102.4	64.2	101.7	102.1	84.8
i7	716030	51.863	1.033	102.9	106.9	74.7	100.8	103.0	82.6
sct	120896	9.937	0.724	102.7	102.4	74.2	102.6	101.3	77.9
unreg	100368	10.702	0.624	105.4	104.9	73.1	103.8	104.7	85.2
x1	301110	22.237	1.172	104.5	107.9	76.0	102.0	101.9	86.3
x2	62346	4.203	0.284	101.9	102.0	60.9	101.3	101.7	87.5
x3	782174	57.572	1.138	108.5	105.1	76.6	101.6	104.3	83.3
x4	476018	29.232	1.037	102.6	109.7	62.3	101.1	109.0	79.0
z4ml	80780	4.822	0.389	106.2	108.5	77.7	105.4	106.8	90.3
average				104.7	106.2	70.9	102.3	104.1	83.7

Table 1. Experimental results

5. CONCLUSIONS

In this paper, we have studied the problem of technology mapping with hot-carrier reliability as the objective. A logic level HCE model is derived and based on this a tree mapping exact algorithm and DAG covering heuristics were proposed. Low power mapping under a hot-carrier reliability constraint was also presented. Our experimental results show that a substantial reduction in HCE can be achieved by applying our DAG-mapping algorithm. We get an average of 29.1% improvement in HCE with 4.7% penalty in area and 6.2% penalty in power consumption. From these results, we conclude that a design with the minimal power is not necessarily a design with the optimal hot-carrier reliability.

REFERENCES

1. A. Dasgupta and R. Karri, "Hot-Carrier Reliability Enhancement via Input Reordering and Transistor Sizing," *Proc. of DAC,* pp. 819-824, 1996.
2. E. Detjcus, G. Gannot, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang, "Technology Mapping in MIS," *Proc. ICCAD*, pp. 116-119, 1987.

3. R. B. Fair, "Challenges to Manufacturing Submicron, Ultra Large Scale Integrated Circuits," *Proceedings of IEEE*, Vol. 78, No. 11, pp. 1687-1705, November 1990.
4. N. Hedenstierna and K.O. Jeppson, "CMOS Circuit Speed and Buffer Optimization," *IEEE Tran. on Computer-Aided Design,*, Vol. CAD-6, No.2, March 1987.
5. C. Hu, and M. Horowitz, "Hot-Electron-Induced MOSFET Degradation - Model, Monitor and Improvement," *IEEE Transactions on Electronic Devices*, Vol. ED32, No. 2, pp. 375-384, February 1985.
6. C. Hu "Future CMOS Scaling and Reliability," *Proceedings of IEEE*, Vol. 81, No. 5, pp. 682-689, May 1993.
7. T. Karnik, C. C. Teng and S. M. Kang, "High-Level Hot Carrier Reliability-Driven Synthesis Using Macro-Models," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 65-68, 1995.
8. K. Keutzer, "DAGON: Technology Binding and Local Optimization by DAG Matching," *Proc. ACM/IEEE DAC*, pp. 341-347, 1987.
9. S. B. Kuusinen and C. Hu, "Hot-Carrier Induced Degradation of Critical Paths Modeled by Rule-based Analysis," *Proc. IEEE Custom Integrated Circuits Coonference*, pp. 69-72, 1995.
10. Y. Leblebici, W. Sun and S.M. Kang, "Parametric Macro-Modeling of Hot-Carrier-Induced Dynamic Degradation in MOS VLSI Circuits," *IEEE Trans. on Electronic Devices,* Vol. 40, No. 3, pp. 673-675, March 1993.
11. Y. Leblebici, "Design Considerations for CMOS Digital Circuits with Improved Hot-Carrier Reliability," *IEEE Journal of Solid-State Circuits,* Vol. 31, No. 7, pp. 1014-1023, July 1996.
12. P. C. Li, G. I. Stamoulis, and I. N. Hajj, "A Probabilistic Timing Approach to Hot-Carrier Effect Estimation," *Proc. ICCAD*, pp. 210-213, 1992.
13. P.C. Li and I.N. Hajj, "Computer-Aided Redesign of VLSI Circuits for Hot-Carrier Reliability," *IEEE Trans. on CAD,* Vol. 15, No.5, pp. 453-464, May 1996.
14. B. Lin and H. D. Man, "Low-Power Driven Technology Mapping under Timing Constraints," *Proc. ICCD*, pp.421-427, 1993.
15. F. Najm, "Transition Density, a Stochastic Measure of Activity in Digital Circuits," *Proc. DAC*, pp. 644-649, 1991.
16. F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Transactions on VLSI*, Vol. 2, No. 4, pp. 446-455, December 1994.
17. K. Roy and S. Prasad, "Logic Synthesis for Reliability - An Early Start to Controlling Electromigration and Hot Carrier Effects," *Proc. European DAC*, pp. 136-141, 1994.
18. A. G. Sabnis, *VLSI Electronics: Microstructure Science, Vol. 22, VLSI Reliability*, Academic Press, 1990.
19. E. M. Sentovich, K. J. Singh, C. Moon, H. Savoj, R. K. Brayton and A. Sangiovanni-Vincentelli, " Sequential Circuit Design Using Synthesis and Optimization," *Proc. ICCD*, pp. 328-333, 1992.
20. E. Takeda, et al., "New hot-carrier injection and device degradation in submicron MOSFET's," *IEEE Proceedings*, Vol. EDL-4, pp. 111-113, 1983.
21. E. Takeda, et al., "VLSI Reliability Challenges: From Device Physics to Wafer Scale Systems," *Proceedings of IEEE*, Vol. 81, No. 5, pp. 653-674, May 1993.
22. C. C. Teng, W. Sun, S. M. Kang, P. Fang, and J. Yue, "iRULE: Fast Host-Carrier Reliability Diagnosis Using Macro-Models," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 421-424, 1994.
23. V. Tiwari, P. Ashar, and S. Malik, "Technology Mapping for Low Power," *Proc. DAC*, pp. 74-79, 1993.
24. H. Touati, *Performance Oriented Technology Mapping*, Ph.D thesis, University of California, Berkeley, 1990.
25. C. Y. Tsui, M. Pedram, and A. M. Despain, "Technology Decomposition and Mapping Targeting Low Power Dissipation," *Proc. DAC*, pp. 68-73, 1993.
26. R. Tu, E. Rosenbaum, W. Chan, C. Li, E. Minami, K. Wuader, P. Ko, and C. Hu, "Berkeley Reliability Tools - BERT," *IEEE Trans. on CAD*, Vol. 8, No. 9, 1993.
27. W. Weber, M. Brox, T. Kuenemund, H.M. Muehlhoff, and D. Schmitt-Landsiedel, "Dynamic Degradation in MOSFET's – Part II: Application in the Circuit Environment," *IEEE Trans. Electron Devices,* Vol. 38, pp. 1859-1867, Aug. 1991.
28. M. H. Woods, "MOS VLSI Reliability and Yield Trends,"*Proceedings of IEEE*, Vol. 74, No. 12, pp. 1715-1729, December 1986.
29. P. Yang and J. H. Chern "Design for Reliability: The Major Challenge for VLSI," *Proceedings of IEEE*, Vol. 81, No. 5, pp. 730-744, May 1993.