A Leakage-aware Low Power Technology Mapping Algorithm Considering the Hot-Carrier Effect[1]

Chang Woo Kang and Massoud Pedram

Dept. of Electrical Engineering-Systems

University of Southern California, Los Angeles CA 90089

{ckang,pedram}@usc.edu

ABSTRACT - *Leakage power and hot-carrier effects are emerging as key concerns in deep sub-micron CMOS technologies with respect to their effects on the total power dissipation and reliability of VLSI circuits. Leakage power dissipation is rapidly becoming a substantial contributor to the total power dissipation as threshold voltage becomes small. Similarly, the hot-carrier effect is one of the most significant failure mechanisms in high-density VLSI circuits. In this paper, a technology mapping technique is presented for use in reducing the leakage power dissipation of the circuit by utilizing a dual-threshold voltage cell library and for minimizing the aged delay of the circuit by considering the effect of hot carriers on the cell speeds as the circuit ages. In addition, this paper presents two methods to reduce delay during technology mapping: primary output ordering and pin permutation. Experimental results show that the total power dissipation and leakage power dissipation can be reduced by up to 27% and 52% as a result of the leakage-aware technology mapping and that the circuit aging phenomenon can be reduced by up to 10.6% as a result of hot-carrier-aware technology mapping. Delay was also reduced by up to 13% using primary output ordering and pin permutation.*

1 INTRODUCTION

As device dimensions shrink to the 100 nm and below, leakage power dissipation as well as hot-carrier effect are becoming major concerns for low-power and reliable systems in the near future. Leakage power dissipation has already grown to be a substantial contributor to the total power dissipation of a VLSI circuit.

In the past, a general rule for high-performance transistor design was to maintain a V_{dd}/V_t ratio of at least four [1]. However, as the leakage power dissipation became an important portion of total power of a system, V_t could not be scaled down as aggressively as the supply voltage. This is because the leakage power dissipation increases exponentially as V_t decreases. To obtain reasonable device performance and for adequate circuit switching noise margins, V_{dd} must be no smaller than 2.3 times V_t [2]. The implication is thus leakage current concerns set the lower limit of V_{dd} scaling. In 130 nm technology node, the

[1] A preliminary version of this work has appeared in *Proceedings of ASP-DAC 2003*.

leakage power is estimated to be 15%-20% of total power dissipation in high performance micro processors [3]. The leakage power dissipation is expected to exceed the total power dissipation when the technology decreases beyond the 65nm node [4].

There have been several approaches to reduce leakage power dissipation in standby mode. By turning transistors off in a stack, leakage current can be reduced considerably [5]-[7]. This is called the stack effect. In order to utilize this effect, transistors were inserted to force transistors to be connected in series and to be turned off when in standby mode. A heuristic algorithm was proposed in [5]. The algorithm finds the best input that turns off the maximum number of devices in the transistor stacks in order to minimize the leakage power dissipation during standby mode. As an alternative approach, dual-threshold voltage assignment schemes have been proposed [8]-[10]. In a dual-threshold process, an additional mask layer is used to assign either a high or low threshold voltage to each transistor. Dual-threshold voltage technology was used in the Pentium 4 processor design [11]. In the library, low-V_t transistors were primarily used to gain speed for the same layout footprint. These cells were generated from the nominal versions by converting selected devices to low-V_t devices without changing the transistor sizes. By using dual-threshold technology, no additional layout complexity to insert transistors between logic and power supplies is required, which makes design procedure simple. Efficient utilization of dual-threshold technology is a key to succeed in low-power, high-speed design. The efficiency may change in which design phase the technology is applied. In this paper, we use the dual-threshold voltage technology during technology mapping phase to minimize leakage power dissipation while maintaining performance.

Long term, mission critical systems require highly reliable circuits. As the device dimensions shrink to the deep sub-micron ranges, the electric field in the transistor channel increases significantly. The hot-carrier-induced damage in MOS transistors is caused by the injection of high-energy electrons 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 can cause the degradation of the circuit [14]. Without considering this fact, timing paths will change their delay characteristic and cause problems of system reliability.

This paper presents a technology mapping technique that reduces total power dissipation, including leakage power dissipation, by utilizing both stack effect and a dual-V_t library. Previous researchers have primarily focused on leakage power dissipation in standby mode. However, by considering leakage power dissipation in technology mapping, the leakage power dissipation can be reduced in the active mode as well as the standby mode. This proposed technique reduces the total power dissipation in the circuit while maintaining its logic speed. Furthermore, a simple aging model of a logic cell for hot-carrier effect is proposed and used for technology mapping. This model considers the delay degradation caused by hot-carrier effects

so that it can optimize circuits for long-term reliability. On top of these schemes, two simple heuristic approaches are presented in which primary outputs of the circuit are ordered based on logic depth of their respective logic cones, and where logic cell input pins are permuted during the technology mapping algorithm. The objective of both heuristics is to reduce the circuit delay without impacting the total power dissipation.

This paper is organized as follows. The background work is discussed in Section 2. Two heuristic techniques to improve the circuit speed of a mapped netlist are presented in Section 3. Models for leakage power and hot-carrier effect are proposed in Section 4. A technology mapping algorithm that captures the leakage power cost and the aging effect in transistors is described in Section 5. In Section 6, the power-speed trade-offs of technology mapping with a dual-V_t library are discussed and results from heuristic speed-up techniques are reported. Conclusions are given in Section 7.

2 BACKGROUND

This section briefly reviews power-delay technology mapping, threshold voltage scaling effects and prior works on low power dual-V_t assignment algorithms, and the hot-carrier effect.

2.1 Power-Delay Technology Mapping

Technology mapping problem can be described as follows: binding nodes in a Boolean network representing a combinational logic circuit optimized by technology-independent synthesis procedures to logic cells in a target library such that cost of the final implementation is minimized and timing constraints are satisfied.

In [16][18], authors used a cost-delay curve to store all of the cost-delay trade-offs at a node to minimize the area or power costs during the mapping phase. For example, in [18], during a post-order traversal starting from primary inputs, a set of possible arrival times and power dissipations at each node of a network are produced. Starting at primary outputs, a pre-order traversal determines the best match for each node so as to minimize power dissipation of the network while satisfying the required time constraints. The post-order traversal comprises of two operations performed on each node: 1) *adding* the power-delay curves of the children to generate the power-delay curve of a parent node and 2) *merging* the power-delay curves for different candidate matches to generate the composite curve of the node. After adding and merging, a new curve must have only non-inferior points. A non-inferior point (t', p') is a point if and only if there does not exist a point (t, p) such that either $t \leq t', p < p'$ or $t < t', p \leq p'$, where t denotes the arrival time and p represents the power dissipation.

Figure 1 shows an example of *adding* curves of two child nodes to that of their parent, *C,* for a target match on node *C*. The algorithm selects a point from the power-delay curve of one child node, e.g., point *a* on power-delay curve of node *A*. Now

the corresponding point on the power-delay curve of child node *B* must have a delay less than 3 units and also have the minimum power, which is point *d*. Delay for this match is the sum of the maximum delay among delays to the child nodes and the delay of the match itself. Power dissipation for this match is the sum of power dissipations on points *a*, *d*, and the match itself. For power-delay curves generated after the *ADD* operation, a *lower bound MERGE* operation is performed. In Figure 2, node *C* has two alternative matches and two power-delay curves after addition. Since point *a* is inferior to point *d*, it must be removed in the merged curve. The power-delay curve addition and merging are repeated until the root of the tree is reached.

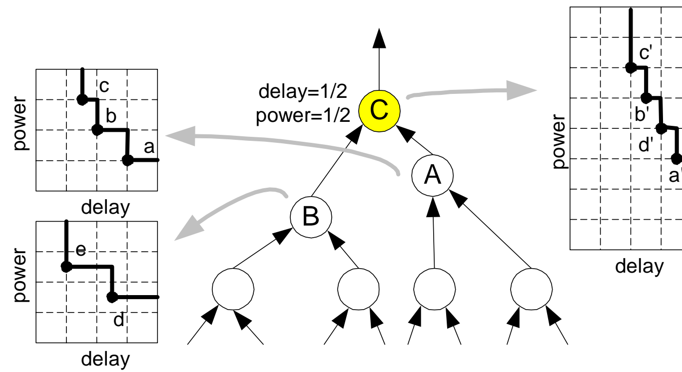


Figure 1: Adding two power-delay curves to generate one parent curve.

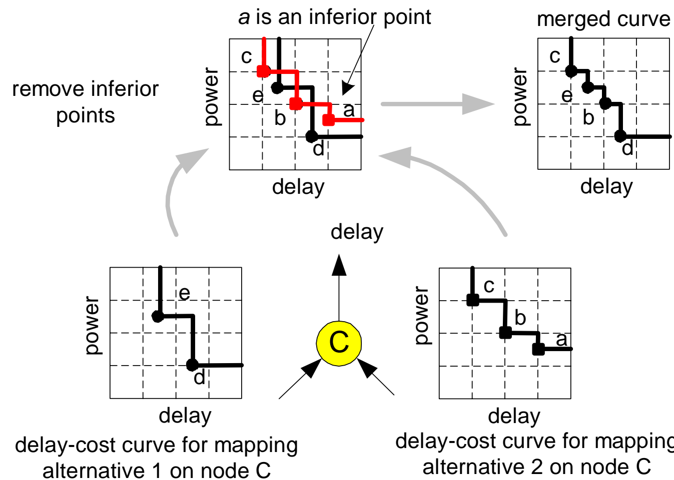


Figure 2: Merging power-delay curves for two mapping alternatives.

The user determines the arrival time-power trade-off. More precisely, during the pre-order traversal, given the required time at the root of the tree, a suitable point on the power-delay curve for the root node is chosen. The logic cell matching for the point at this root is identified and then required times for its inputs are computed. The preorder traversal resumes at its child nodes to satisfy the new required time and the minimum power dissipation.

The power-delay function may use a large memory size as the size of library set increases. A simple solution limits the maximum number of points, *K,* which can be kept in any delay-cost curve [16]. Notice that if *K* is chosen to be too small, some optimal points may be dropped. On the other hand, if *K* is chosen to be too large, it will slow the mapping procedure. Roy et al [17] introduced an α-approximate algorithm to exponentially compress a delay-cost function of [16] while keeping solutions within a constant bound of the optimal solution (cost may be area, power dissipation, etc.). In particular, they introduced the notion of the ith *α-break point* as defined by

$$\lambda_\alpha^i = c_1(1+\alpha)^i \tag{1}$$

where (c_1, t_1) denotes the cheapest design point of a delay-cost curve and the cost might be area or power. The ith α-interval is defined as the semi-open interval $[\lambda_\alpha^i, \lambda_\alpha^{i+1})$. The α-delay-cost curve contains at most two points in every α-interval; if a point in the curve belongs to some α-interval, then it is either the cheapest or the costliest design solution in that interval. The maximum number of points in the α-delay-cost curve is logarithmically proportional to the cost range $[c_1, c_n]$. The authors proved that the solution obtained by the exponential compression of the α-delay-cost curve is within α% of the optimal solution for a tree-structured subject graph if all input pin capacitances for the library logic cells are the same. Thus, this technique – under equal pin capacitance assumption - provides an exponential compression of the cost-delay curves.

2.2 Techniques for Low Leakage Power Dissipation

Demand for minimizing the dynamic power dissipation and push toward ever-shortening channel lengths in CMOS technology force the supply voltage to be scaled down. To maintain the circuit speed, the transistor threshold voltages must also be scaled down. This is easily seen from the first-order propagation delay equation of a transistor

$$\tau = \frac{CV_{dd}}{(V_{dd} - V_t)^\chi} \tag{2}$$

where C is the load capacitance, V_t is the threshold voltage, and χ (which is greater than 1 but less than 2) models the short channel effect [19]. The subthreshold leakage current can be modeled from the BSIM MOS transistor model as:

$$I_{subth} = \mu_o C_{ox} \frac{W_{eff}}{L_{eff}} \left(\frac{kT}{q}\right)^2 e^{1.8} e^{\frac{q}{kT}(V_{GS} - V_{t0} - \gamma' V_{SB} + \eta V_{DS})} \left(1 - e^{-\frac{qV_{DS}}{kT}}\right) \tag{3}$$

where μ_o is the zero bias mobility, C_{ox} is the gate oxide capacitance per unit area, kT/q is the thermal voltage, V_{t0} is the zero bias threshold voltage, γ' is the linearized body effect coefficient, and η is the DIBL (Drain-induced barrier lowering) effect

coefficient [5]. Clearly, the subthreshold leakage current exponentially increases as V_t is reduced. This principle of subthreshold current reduction is well explained in [20] and most of techniques for low leakage power are based on the principle.

Many different techniques have been developed for low leakage power dissipation. Those techniques have been well classified in [21]: input vector control, V_t control, and gating the supply voltage. We provide more background on multithreshold-voltage techniques to reduce leakage power dissipation. Multithreshold-voltage CMOS circuit technology emerged as one of best candidates to minimize leakage power dissipation. There can be two different techniques in utilizing the technology. The first technique is to insert high-V_t devices in series with the normal circuitry [8]. This reduces leakage current significantly since the high-V_t voltage transistor will allow only a small amount of leakage to flow. However, only standby leakage power can be reduced and the large inserted transistor will increase the area and delay. Sizing the high-V_t transistor is of utmost importance. Depending on the size, circuit performance can be degraded and noise margins in the circuits are reduced due to the increased voltage in virtual ground. In the worst case, the circuit can fail logically. This technique must be very effective for burst-mode-type applications. They may be acceptable to have large leakage currents during the active mode, but it is extremely wasteful to have large leakage currents during the idle stage because power will be drained continuously with no useful work being done.

In the second technique, a high V_t can be assigned to transistors of logic cells that are not on any timing critical paths. The idea is to reduce leakage current of logic cells with high V_t transistors, while maintaining the circuit performance by continuing to use low V_t transistors for logic cells that are on the timing critical paths. Therefore, no additional transistors are required, and both high performance and low power can be achieved simultaneously. A number of researchers have already used this techniques to reduce the chip leakage [9][13][10]. These techniques start from a target circuit, which has been mapped, placed, and routed with an initial choice of V_t. The method described in [13] consists of three steps. In the first step, it assigns low-V_t logic cells to a circuit. Then, it finds a slack for each node. Finally, from primary outputs, it assigns high-V_t logic cells to nodes if it does not cause any negative slack. Authors tried different high-V_t values to find the minimum total power dissipation. This technique chooses a logic cell in order of breadth first search from primary outputs, thus, limiting the possibility of improvement by forcing an order of nodes to be tested for the change of V_t. Dynamic power dissipation is not considered when V_t for a logic cell is selected. An approach considering both the dynamic power dissipation and leakage power dissipation was presented in [9]. All logic cells in the circuit are initially set to the high V_t. The logic cells are then sized and a supply voltage is selected such that the circuit meets the timing criteria and the power dissipation is minimized.

After the power optimization step, the critical paths of the circuit are studied to extract a subset of the logic cells, which are set to a low V_t. Selecting logic cells for the subset is the most significant part. From the most critical path, the logic cell that creates the maximum reduction in the delay is chosen and its V_t is set to the low V_t. The circuit is subsequently re-optimized for low power operation.

The problem of V_t assignment was considered in [10]. The authors presented three algorithms. The first algorithm starts with all devices at high V_t. Each edge (i, j) is assigned with a weight, which increases as leakage power increases and decreases as the arrival time at logic cell j increases, when V_t of logic cell j is changed to high V_t. Thus, the algorithm needs to find the minimum cut to minimize power increase while arrival time increases. Then, the max-flow-min-cut algorithm can solve this problem. The second approach starts with configuration where all devices are at low V_t. The objective here is to reduce the standby power as much as possible without increasing the delay. Each edge (i, j) is assigned with a weight, which is the reduction in leakage power dissipation when the threshold voltages of logic cell i and j are changed from low V_t to high V_t. Then, the objective is to find a maximum weight subset such that changing V_t of the subset will not violate the delay constraint. Since the problem of finding a maximum weight cut of a graph is NP-complete, authors defined a special type of cut and found the maximum weight cut among them. Iterative improvement was applied to escape from the local optimal solution. The third approach starts from a combinational circuit with the threshold voltages of all of transistors set to high V_t. In this approach, subcircuits are selected, which are on critical paths. All logic cells in the subcircuits are set to low V_t and fed into the second algorithm. The main idea is to process only the critical subcircuits. The last algorithm provided the best results.

2.3 Hot-Carrier Effect

As MOSFET devices are scaled down to smaller feature sizes, hot-carriers injected into the MOS gate oxide layer begin to cause major reliability problems [14]. This is because the lateral electric field in the channel can become sufficiently large so that the average carrier energy significantly exceeds that of the silicon bandgap of 1.1eV. The carriers in the channel, electrons in nMOS and holes in pMOS transistor, will have energy distributions close to their respective barrier heights at the Si-SiO_2 interface. As a result, electrons and holes with sufficient energy from the lateral electric field can surmount the barriers. The barrier height for electrons is ≈ 3.1eV, while for holes is ≈ 4.8eV. Due to the lower barrier for electrons, the hot-carrier effect in nMOS transistors is more significant than it is in pMOS transistors. The hot carriers are injected into the MOS gate oxide and can become trapped at the interface. In addition, carriers with high energy can hit atoms near the drain and the atoms are separated into pairs of electrons and holes as shown in Figure 3(a). Some electrons and holes are injected into oxide and the majority of holes cause substrate current (I_{sub}). The electrons trapped in the thin oxide act as a shield for the electric

field created by the gate voltage as depicted in Figure 3(b). Therefore, to create strong inversion in the channel under the gate, a higher gate voltage is needed. In other words, V_t of the transistor increases. In summary, injection of carriers into the oxide results in a change of the I-V characteristics, an increase in Vt, and a decrease in transconductance. Without careful estimation of future damage caused by the hot-carrier effect and appropriate design to prevent it, circuits may have erroneous operation after a certain amount of stress.

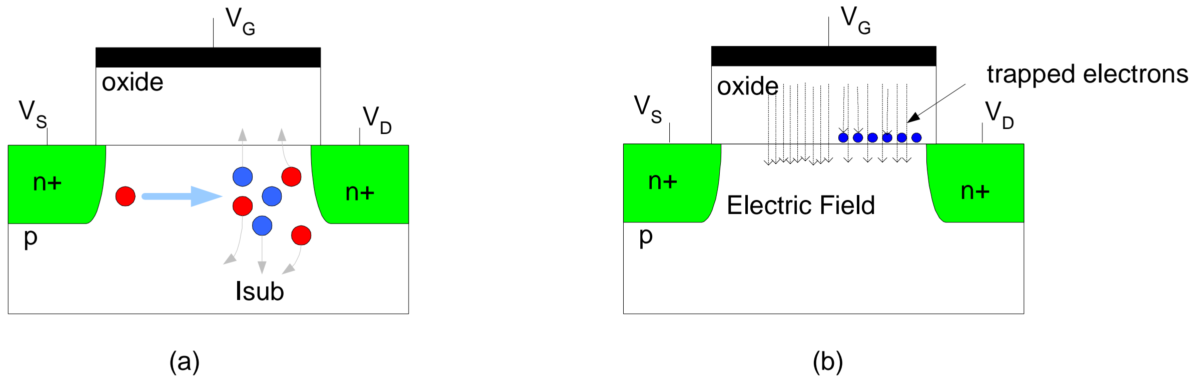


Figure 3: Impact ionization and charge trapping.

Hot-carrier degradation takes place when a transistor is in the saturation region. CMOS circuits can be in this region during transitions. Therefore, a slow slew rate of input signals and a large output load capacitance for a logic cell can stress the transistors in the logic cell. In addition, higher switching activity at the output pin of a logic cell can quickly wear out the driving capability of the cell.

There are a number of research activities to estimate the performance degradation of circuits due to the hot-carrier effect. In [27], the speed degradation of an inverter was expressed as a function of its input slope, the ratio of nMOS transistor channel width to the output load capacitance (W_n/C_L), and the ratio of the gate capacitance to the output load capacitance (C_g/C_L). It is seen that device speed degradation decreases with the increasing input signal slope (i.e., faster slew rates) and with an increasing (W_n/C_L) ratio. For larger C_g/C_L ratios, the hot-carrier induced speed degradation increases considerably. The presence of the parasitic gate-drain capacitance results in a significant output voltage overshoot, which causes more hot-electron effect. In addition, the authors provided systematic guidelines for the reliable design of inverter chains by using the parametric model they developed. In [35], the authors proposed a ratio-based hot-carrier degradation model for the aging-aware timing simulation of large-scale circuits. The degraded circuit delay is expressed as a multiplicative factor of the original fresh delay. Modeling the transistors as linear resistors, the authors described the output slew rate as well as the propagation delay as a function of this multiplicative factor. The resistance degradation was characterized in terms of the

input slew rate, the output load capacitance, and the expected number of output transitions. A technology mapping technique considering hot-carrier effect was presented in [24]. This work focused on minimizing the maximum degradation of a logic cell. Since the hot-carrier effect results in speed degradation, minimizing the maximum degradation may not improve the overall circuit performance. Rewiring, gate resizing, and pin reordering have also been used to minimize the aged delay of a circuit without increasing the fresh delay of the circuit [23]. In particular, the proposed algorithm generates all "generalized implication" supergates from the input netlist. Next all possible logic cell resizing and pin swapping choices inside a supergate are evaluated. Finally, the best supergate is chosen and timing information is updated with the new configuration of the supergate. This process repeats until there is no improvement.

3 SPEEDUP HEURISTICS

In this section, two heuristic techniques are presented to improve the circuit speed of mapped netlists.

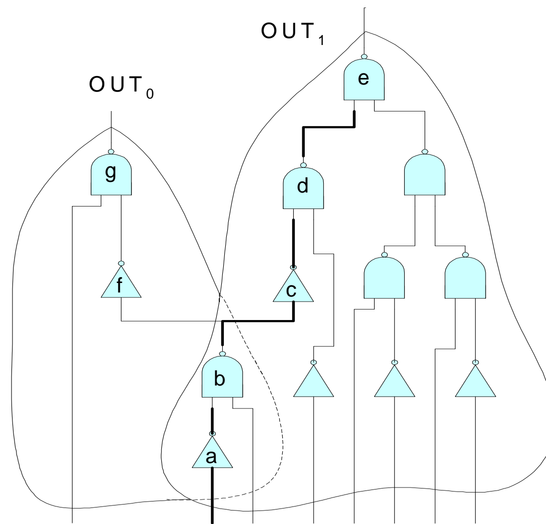


Figure 4: Decomposed network with transitive fanin cones from primary outputs.

3.1 Primary Output Ordering

A *logic cone* refers to the transitive fanin support of a primary output. There is one logic cone for each primary output of the circuit. The various logic cones may obviously be overlapping. The technology-mapping algorithm processes one primary output logic cone at a time. The order of mapping is important and produces different mapping results in terms of area, power dissipation, and circuit delay. For example, suppose that a network has been decomposed into inverter and NAND logic cells as shown in Figure 4 (this NAND-decomposed graph is often referred to as the subject graph in mapping literature). There are two logic cones; one corresponds to OUT_0 and the other to OUT_1. We assume that no logic duplication across the multiple fanout points is allowed. Furthermore, assume that the path comprising of primitive logic cells *a*, *b*, *c*, *d*, and *e* is a timing

critical path. If logic cells in the OUT_0 cone are mapped first, primitive logic cells *b*, *f*, *g* can be mapped to a three-input NAND cell in the library. Next when primitive logic cells in the OUT_1 cone are processed, logic cells *b*, *c*, and *d* cannot be mapped to a three-input NAND logic cell. This results in a sub-optimal mapping solution for the two logic cones. In general, logic duplication, on crossing multiple fanout points in the subject graph, is allowed. However, even in that general case, the observation that the technology-mapping solution depends on the order of processing logic cones still holds.

To achieve better mapping solutions, a simple heuristic approach is adopted whereby the primary outputs of the given circuits are sorted in descending order of their (maximum) logic depths. Next, the logic cones are processed so that the most timing-critical cone is mapped first. The intuition is that the delay of a mapped cone is roughly proportional to the logic depth of the corresponding primitive logic cell cone. Furthermore, by processing cones with larger logic depths first, maximum flexibility is provided for mapping these timing critical logic cones.

3.2 Pin Permutation

To reduce logic cell delay, it is well known that for sufficiently fast input transition times, the latest arriving input signal must be assigned to the top transistor of a pull-down section in a complex CMOS logic cell [26]. Namely, careful pin assignment for signals can result in reduced propagation delay through a CMOS logic cell. Therefore, during the technology mapping, this observation can be exploited in order to reduce the overall delay of the mapped circuit netlist.

3.2.1 Equivalent Pins

In an ASIC cell library, many of the logic cells have equivalent pins in the sense that the signal to pin assignments for these input pins may be interchanged without changing the functionality of the logic cell. The switching speed of the cell may however change as a function of the signal-to-pin assignment as explained above. Each cell in the library is annotated with its pin swapping information, that is, pins of the cell are assigned to a number of pin sets such that all pins belonging to the same pin set can be interchanged during the technology mapping procedure. For example, if a cell implements Boolean function $\overline{AB+C}$, the pin sets are *{A, B}* and *{C}*.

3.2.2 Pin Permutation and Delay Calculation

During the technology mapping procedure, when a cell match is found at the output of some node in the subject network, all valid input signal-to-pin assignments (as per *pin set equivalence* relationship) are enumerated, and the pin assignment that results in the lowest delay is selected to be included in the power-delay curve of the node in the subject graph.

As an example, we consider a four-input NOR logic cell:

$$Y = \overline{(A + B + C + D)}$$

which has 4! = 24 valid permutations of the input pins. During mapping, a four-input NOR match and its best pin assignment are found and stored. The power-delay curve data is modified accordingly. The pin assignment information is recovered from the solutions stored in these curves during the output-to-input traversal that generates the complete mapping solution.

4 LOGIC CELL MODELING

In this section, models for leakage power dissipation and logic cell aging due to the hot-carrier effect are proposed.

4.1 Power Dissipation

Power dissipation in a CMOS circuit consists of dynamic, short-circuit, internal, and leakage components. Charging and discharging the load capacitance of a logic cell causes dynamic power dissipation. Internal power dissipation results from charging and discharging the internal nodes of a logic cell without changing the output node. Short-power dissipation appears when PMOS and NMOS transistors are on during output transitions. Leakage power dissipation exists mainly because of the subthreshold current. (In this paper, we ignore the gate leakage currents because we believe the best solution for controlling the gate leakage is to use high-k dielectric material to realize the insulation layer under the gate. It is expected that the high-k material will become commonplace in 2006 [2].)

The average dynamic power dissipation of a logic cell in a synchronous CMOS circuit is given by:

$$P_{dyn} = 0.5 \times C_{load} \times \frac{V_{dd}^2}{T_{cycle}} \times sw \quad (4)$$

where C_{load} is the load capacitance of the logic cell, V_{dd} is the supply voltage, T_{cycle} is the clock cycle time, and sw is the switching probability per cycle at the output of the logic cell [28]. The amount of leakage current is strongly dependent on the binary pattern applied to the inputs of a CMOS logic cell. This phenomenon is shown in Figure 5. Therefore, signal probability, which is defined as probability for a signal to assume a value of one, must be considered when estimating the leakage power dissipation of a logic cell. The leakage power dissipation in a CMOS logic cell can be calculated as:

$$P_{leak} = V_{dd} \times \sum_{U} [I_{subth}(U) \times pr(U)] \quad (5)$$

where U is an input vector for a logic cell, $pr(U)$ is the probability that the input vector U is applied as an input pattern for the logic cell. For example, the probability for input vector $U\ (0,1,1)$ of a three-input logic cell, which has A, B and C as its input signals, is calculated as:

$$pr(U) = \left(1 - pr(A)\right) \times pr(B) \times pr(C)\,. \tag{6}$$

Note that the signal probabilities of intermediate variables in a Boolean network can be calculated from the signal probabilities of the primary input of the circuit by using well-known *signal probability propagation* techniques.

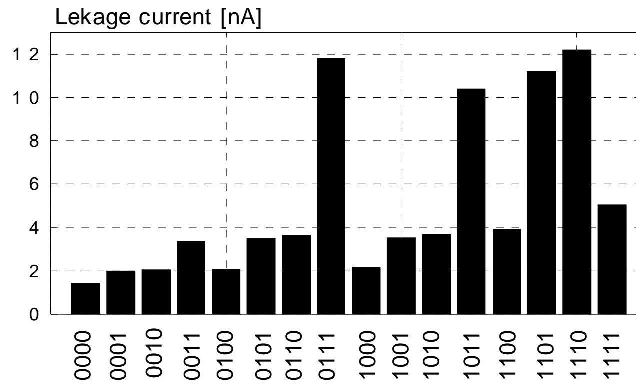


Figure 5: Leakage current of different input patterns for a four-input NAND logic cell. MSB refers to the input pin closest to the output port.

4.2 Logic cell Aging due to the Hot-Carrier Effect

The thin oxide damage by the hot-carrier effect impacts the delay of an MOS transistor because it shifts its V_t and decreases the drain current driving capability [14]. In this paper, we adopt a simple propagation delay equation to capture the aged delay of a logic cell. More precisely, the aged input-to-output propagation delay, T_{aged}, through a logic cell is calculated as:

$$T_{aged} = \left(1 + \delta \times sw \times C_{load}\right) \times T_{fresh} \tag{7}$$

where δ is the degradation factor, sw is the switching activity of the cell's output, C_{load} is the output load capacitance, T_{fresh} is the *fresh input-to-output delay*. The degradation factor is in fact inversely proportional to the driver strength of the logic cell, because the driver strength determines the duration of the transition time, and hence, the period during which the hot carrier effect is present.

In current semiconductor process technologies, hot-carrier induced degradation is much more severe in NMOS transistors than in PMOS. The impact-ionization rate due to holes (i.e., the number of electron-hole pairs generated by a hot hole) is one

to two orders of magnitude lower than that due to electrons at a given electric field [15]. Therefore, this paper only considers delay degradation for transistors in the pull-down network.

5 TECHNOLOGY MAPPING

Leakage power aware mapping with a dual-V_t library and hot-carrier effect aware mapping for long-term performance optimization is presented here.

5.1 Leakage Power Aware Mapping (LPAM)

A leakage-aware low power technology mapper has been implemented based on the algorithm presented in [18]. The leakage power cost of a logic cell has been accounted for in order to choose the logic cell that dissipates the least amount of power at a node in the Boolean network. The total power dissipation at node *n* with logic cell *g* matching at that node is given by:

$$P(n,g) = \frac{1}{2} C_{diff}(n,g) \frac{V_{dd}^2}{T_{cycle}} sw_n + V_{dd} \sum_U I_{subth}(U)\, pr(U) + \sum_{n_i \in inputs(n,g)} \left(\frac{1}{2} C_{load}(n_i) \frac{V_{dd}^2}{T_{cycle}} sw_{n_i} + \frac{P(n_i, g_i)}{fanout(n_i)} \right) \quad (8)$$

where $C_{diff}(n, g)$ is the diffusion capacitance on an output port of a logic cell g, $C_{load}(n)$ is the output load driven by node n, sw_n is the 0 to 1 and 1 to 0 transition probability of node n, $fanout(n)$ is the number of fanouts from node n, and $P(n_i,g_i)$ is the average power dissipated at input i.

5.2 Dual-V_t Mapping

A number of researchers [8][10] have proposed dual-V_t assignment schemes for low leakage power during standby mode. Generally speaking, these authors assign low-V_t devices on the timing-critical paths of a mapped circuit netlist, while assigning high-V_t to devices on non-critical paths. However, these approaches have a number of shortcomings. First, they put a serious constraint on the logic cell that can replace the current cell in the netlist because only logic cells with the same Boolean function can be candidates for substitution. Second, they do not consider signal probabilities, which in turn determine the expected leakage power dissipation of the circuit. Note that the amount of leakage power dissipation can vary significantly depending on the signal probabilities of the inputs to a logic cell. Third, the proposed algorithms are ad hoc in nature and they cannot even give an optimal solution even for a tree-structured Boolean network. Fourth, leakage power and dynamic power are not optimized simultaneously. Fifth, they do not effectively exploit dual-threshold standard cell libraries that are currently available (see for example [36].) Sixth, by performing V_t assignment during the technology mapping phase,

one will have much more flexibility in choosing the best set of logic cells and threshold voltages to meet the timing constraint with the least amount of power consumption.

5.3 Hot-Carrier Effect Aware Mapping

As in [18], the pin-dependent SIS library delay model [32] is adopted here for calculation of the arrival time. However, the delay calculation equation is modified to account for the aging effect due to the hot-carrier phenomenon as described below.

Suppose that logic cell g has matched at node n, and then the fresh pin-to-pin delay becomes:

$$T_{fresh}\left(n, g, C_{load}\right) = \tau_{i,g} + R_{i,g} C_{load} \tag{9}$$

where $\tau_{i,g}$ is the intrinsic cell delay from input i to output of g, $R_{i,g}$ is drive resistance of g corresponding to a signal transition at input i, and C_{load} is the load capacitance seen at n. Arrival time is calculated as:

$$arrival\left(n, g, C_n\right) = \max_{n_i \in inputs(n,g)} \left(T_{aged} + arrival(n_i, g_i, C_i)\right) \tag{10}$$

where T_{aged} is the aged time of node n given by equation (7), $arrival\ (n_i,\ g_i,\ C_i)$ is the arrival time at input i corresponding to load C_i seen at that input, and g_i is the best match found at input i.

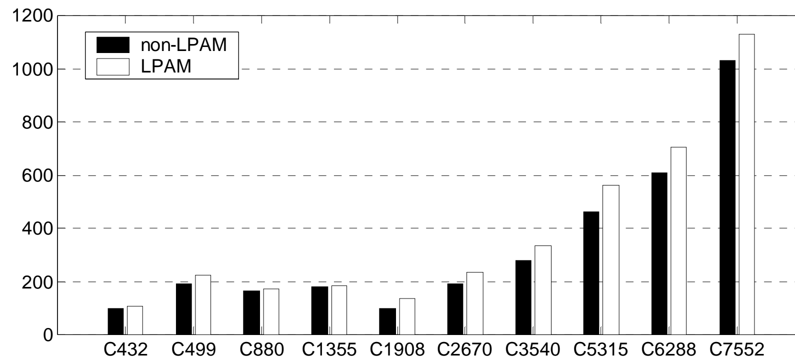
6 SIMULATION RESULTS

In this section, the simulation results for leakage power and for hot-carrier effect are described separately. SIS [31] was used as the logic synthesis environment and a 0.18µm CMOS technology logic cell library was employed. In this library the nMOS transistors have a V_t of 0.4V. Calculation of the signal probabilities by using the *global* ordered binary decision diagram (OBDDs) requires a large amount of memory and long computation time. Therefore, we used the *local* OBDD-based approach presented in [25]. In that scheme, nodes in the network are first levelized. Next the OBDD variables for each local OBDD (associated with some node n) are selected from the transitive fanins of n that are at least k levels away from n, where k is a user-supplied number (e.g., $k=4$).

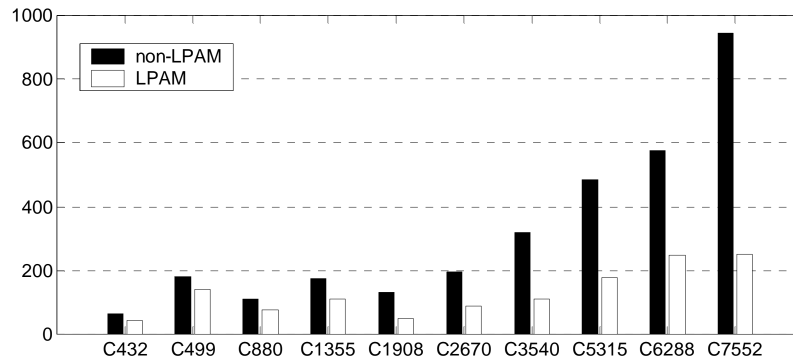
6.1 Leakage Power Reduction

First, a set of low-V_t library cells with 0.2V V_t was generated based on the original high-V_t library of cells. For each logic cell in the two (low-V_t and high-V_t) libraries, the leakage current was obtained and recorded for every input pattern. HSPICE simulations were used to obtain the leakage currents for each possible input combination. The circuits were simulated with a

supply voltage of 1.8V, and the primary input switching activity was set to 0.5. The substrate temperature was assumed to be 110 Celsius.



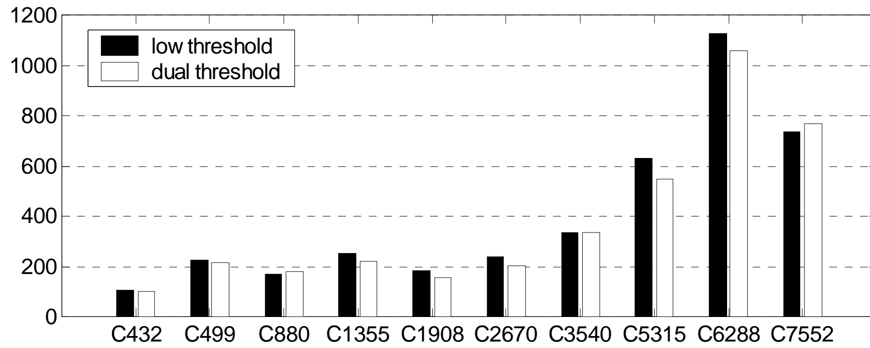
(a)



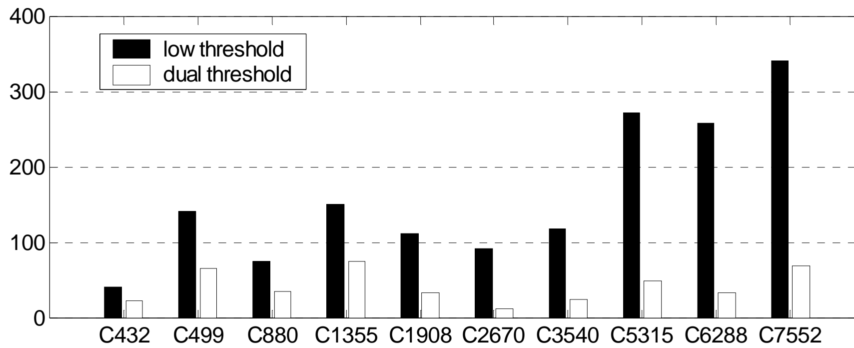
(b)

Figure 6: Power dissipation reduction by LPAM (leakage power aware mapping): (a) dynamic power dissipation in μW, (b) leakage power dissipation in μW.

Figure 6 shows that the proposed technology mapping procedure reduces leakage power dissipation, resulting in a reduction of the total power dissipation. In this figure, only low V_t logic cells are used so as to assess the effectiveness of the leakage-aware low power mapping algorithm. As can be seen, the leakage power is reduced by as much as 50%, while the dynamic power increases by only 16%. As a result, a total power saving of 17% is achieved. Therefore, although the dynamic power dissipation may increase after leakage power optimization, because the technology mapper uses total power consumption (summation of the leakage and dynamic components) as its objective function, it is able to provide a lower total power solution. Furthermore, as the circuit size increases, the average activity per node tends to decrease whereas the total leakage tends to increase. Therefore, the LPAM technique becomes even more effective as shown clearly in the experimental results of these tables.



(a)



(b)

Figure 7: Power dissipation reduction due to technology mapping with dual threshold cells: (a) dynamic power dissipation in μW, (b) leakage power dissipation in μW.

There are several advantages to employing high-V_t logic cells on non-critical paths and low-V_t cells on the critical paths of the circuit. In this way, leakage power can be reduced substantially while meeting a delay constraint. In Figure 7, leakage power is reduced by 68% on average compared to that dissipated in circuits mapped with low-V_t. Note that the longest path delay remains the same in both cases, this is, the results are reported when there is no circuit speed degradation. Dynamic power dissipation has been decreased by 5%, because the gate capacitance of a transistor increases as its V_t is lowered [26]. In fact, dynamic power can be reduced due to the reduction of an internal-node voltage swing for high-threshold cells [13]. In addition, as a part of dynamic power dissipation, short circuit power dissipation decreases as V_t increases [34]. These results point to an average of 24% reduction in total power dissipation of the circuits.

Based on our simulations, about 73 % of the logic cells can be mapped with high-V_t cells on non-critical paths. There is also an advantage of maintaining performance. Note that technology mapping was repeatedly performed to achieve the best circuit speed by starting with a loose timing constraint and gradually tightening it until the circuit could not meet the timing constraint. This tends to guarantee high circuit speeds while attempting to minimize the total power dissipation. This is

shown in Figure 8. By mapping low-V_t cells on critical timing paths, circuits regained most of the circuit speed that was achieved when using low-V_t cells everywhere. Finally, the area becomes larger than that of circuits mapped by low-V_t only, because low-V_t transistors have higher driving strength compared to high-V_t transistors of the same W/L ratio. The result is shown in **Figure 9**. Those areas are to achieve approximately the same delay by using different library sets.

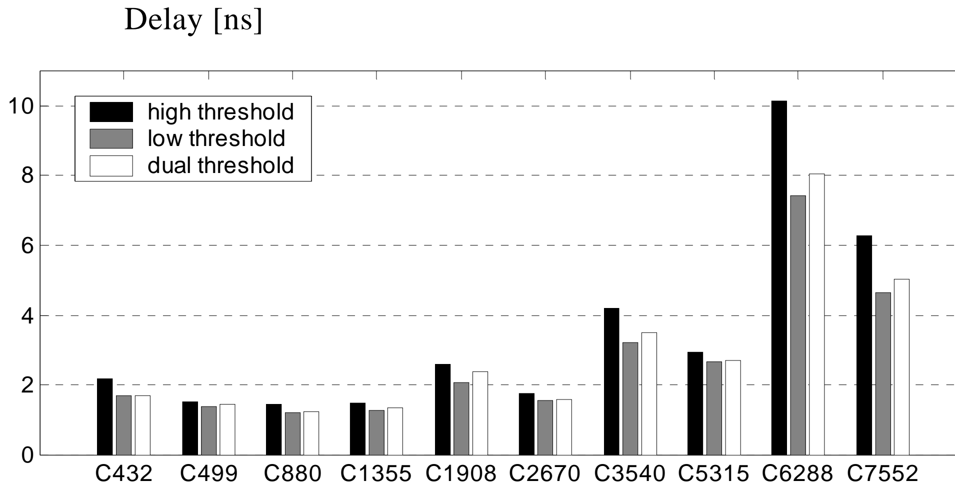


Figure 8: Speed-up by mapping low-V_t logic cells on critical paths.

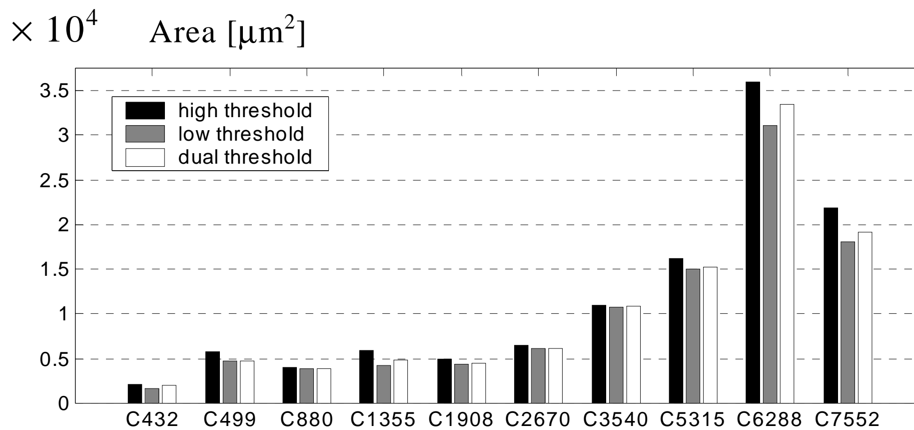


Figure 9: Area reduction by mapping high-V_t logic cells on non-critical paths.

6.2 Considering the Logic cell Aging Effect

Table 1 reports experimental results for the MCNC benchmark circuits. As seen in this table, by considering the aging effect during the technology mapping phase, the aging-aware delay calculator can identify the logic paths that are likely to become timing critical due to the hot-carrier induced aging process. The delay calculator then passes this information to the

technology mapping algorithm, which will in turn generate an appropriate mapping solution that will ensure that the circuit will operate within the required timing constraints even after some degree of aging. Notice that circuits C1355 and sqrt8ml are examples of cases where the fresh delay of a circuit mapped by the aging-aware mapping is longer than the fresh delay of the circuit obtained by the aging-unaware mapper. However, in these cases, as expected the aged delay of the circuits in the first case is lower than the aged delay of the circuits in the second case. Finally note that in many cases, the fresh delay of the circuit obtained by the aging-aware mapping happens to be shorter than the fresh delay of a circuit obtained by the aging-unaware mapping. This effect is due to the fact that the fresh and aged delays of circuit paths are correlated and thus by minimizing the aged delay (which is an upper bound on the fresh delay), we also tend to minimize the fresh delay of the circuit. From these results, we conclude that the critical path may change, as transistors get old, fast on currently non-critical paths.

6.3 Speed-up by Output Ordering and Pin Permutation

Table 2 shows the results of speed-up by primary output ordering based on the logic depth of circuits during technology mapping. The same timing constraint was used for the two different sets of experiments. As seen in this table, the circuit delay reduction is up to 9.2%. Furthermore, one can see that power and area were also reduced except in one case (*i7*). This general trend can be explained as follows. Logic cones with larger depths tend to have a larger number of nodes in the NAND-decomposed network, and thus, tend to account for a larger portion of the total circuit area and power dissipation in the mapped netlist. By performing technology mapping on the larger cones first, we provide maximum flexibility to the technology mapper.

In some circuits, the delay can also be reduced by as much as 12.8% by performing pin permutation during the dynamic programming-based mapping as shown in Table 3. Notice that in Table 3, the delay improvement for some circuits comes at the expense of an area and power increase (see for example, results for *C432*). This may appear strange since, at least on the surface, pin permutation should not have increased the circuit area. The key to understanding these results is that because the pin permutation is integrated with the dynamic programming-based technology mapping algorithm, it potentially changes the area-delay curve at each node in the subject graph, and hence, at the end of the dynamic programming algorithm, the mapping solution will be quite different from the solution that is obtained without dynamic pin permutation. Indeed, if we performed the pin permutation step on a mapped circuit, then only its delay will be impacted while its area will remain exactly the same. Notice however, that in such a case, the circuit delay reduction will be less than when pin permutation is intrinsically integrated within the dynamic programming algorithm as we have proposed and implemented here.

Table 1: Fresh and aged delays by aging-unaware and aging-aware mapper

Circuit	Aging-unaware mapping		Aging-aware mapping		Speed-up	
	Fresh [ns]	Aged [ns]	Fresh [ns]	Aged [ns]	Fresh (%)	Aged (%)
C432	2.21	2.54	2.08	2.34	6.3	7.9
C499	1.52	1.74	1.5	1.66	1.3	4.6
C880	1.39	1.41	1.33	1.34	4.5	5.0
C1355	1.61	2.09	1.72	1.94	-6.4	7.2
C1908	2.65	2.82	2.45	2.52	8.2	10.6
C2670	1.89	1.93	1.81	1.86	4.4	3.6
sqrt8ml	1.93	2.06	1.97	1.98	-2.0	3.9
f51m	2.01	2.04	1.89	1.95	6.3	4.4
alu2	2.1	2.14	2.03	2.05	3.4	4.2
i7	1.13	1.14	1.1	1.11	2.7	2.6

7 CONCLUSION

In this paper, a technology mapping technique was presented to reduce leakage power dissipation, as well as total power dissipation. By considering leakage power dissipation based on input signal probabilities, the reduction on total power dissipation became substantial as circuit size increased. We also presented the trade-offs of mapping a dual-V_t library with respect to leakage power dissipation, total power dissipation, delay, and area. The mapper mapped nodes on non-critical paths with high-threshold-voltage logic cells, resulting up to 52% reduction in leakage power dissipation and 27% in total power dissipation. In addition to the leakage power optimization, an aging model was proposed for the technology mapping to represent the transistor degradation due to the hot-carrier effect. The aging phenomenon was reduced by up to 10.6% in the test benchmark circuits. Two different methods for improving delay during technology mapping were presented: primary output ordering and pin permutation. They showed up to about 9% speed-up, compared to results without those schemes.

Table 2: Technology mapping with primary output ordering

	W/o PO ordering			W/ PO ordering			% Improvement		
Circuit	Area (μm^2)	Power (μW)	Delay (ns)	Area (μm^2)	Power (μW)	Delay (ns)	Area	Power	Delay
C432	2219	82.0	2.48	2128	78.59	2.36	4.3	4.2	4.9
C499	3881	136.9	1.65	3857	136.3	1.62	0.6	0.4	1.8
C880	4232	170.4	1.84	3986	155.7	1.67	6.2	8.6	9.2
C1355	6390	348.5	2.04	5988	330.7	2	6.7	5.1	2
f51m	1269	56	2.09	1255	54	2.05	1.1	3.6	1.9
alu2	4283	131.1	2.93	4270	125.4	2.84	0.3	4.3	0.3
i7	5271	142.4	1.51	5278	141.1	1.5	-0.1	0.8	0.7

Table 3: Technology mapping with pin permutation

	W/o Pin Permutation			W/ Pin Permutation			% Improvement		
Circuit	Area (μm^2)	Power (μW)	Delay (ns)	Area (μm^2)	Power (μW)	Delay (ns)	Area	Power	Delay
C432	3207	101.5	2.27	3331	103.5	2.04	-3.9	-2	10
C499	3395	125.8	1.53	3380	126	1.49	0.4	0	2.6
C880	2637	155.1	1.90	2865	162	1.7	-8.6	-4.5	10.5
C1355	3428	370.8	1.61	3380	377.1	1.54	1.4	-1.7	4.3
f51m	1026	79.82	1.81	997	73.04	1.71	2.9	8.5	5.5
alu2	2950	148.3	2.58	2958	147.1	2.43	-0.3	0.8	5.8
i7	4128	109.1	1.64	4134	107.9	1.43	-0.2	1.1	12.8

REFERENCES

[1] Thompson, S., P. Packan, and M. Bohr, "MOS Scaling: transistor challenges for the 21st century," *Intel Technology Journal*, 1998.

[2] "International Technology Roadmap for Semiconductors 2003 Edition," Semiconductor Industry Association, http://public.itrs.net.

[3] http://developer.intel.com/design/mobile/documentation.htm

[4] B. Doyle, R. Arghavani, D. Barlage, S. Datta, M. Doczy, J. Kavalieros, A. Murthy, and R. Chau, "Transistor elements for 30 nm physical gate lengths and beyond," *Intel Technology Journal*, vol. 6, no. 2, May 2002.

[5] Johnson, M.C., et al., "Models and algorithms for bounds on leakage in CMOS circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, pp. 714-725, 1999.

[6] Gu, R.X., and M.I. Elmasry, "Power dissipation analysis and optimization of deep submicron CMOS digital circuits*," IEEE Journal of Solid-State Circuits*, vol. 31, pp. 707-713, 1996.

[7] Johnson, M.C., et al., "Leakage control with efficient use of transistor stacks in single threshold CMOS," in *Proc. of the Design Automation Conference,* pp. 442-445, 1999.

[8] Kao, J.T., and A.P. Chandrakasan, "Dual-threshold voltage techniques for low-power digital circuits*," IEEE Journal of Solid-State Circuits, vol. 35, pp. 1009-1017, 2000.*

[9] Tripathi, N., *et al.*, "Optimal assignment of high threshold voltage for synthesizing dual threshold CMOS circuits," in *Proc. of the International Conference on VLSI Design,* pp. 227-232, 2000.

[10] Wang Q., and Sarma B. K. Vrudhula, "Algorithms for minimizing standby power in deep submicrometer, dual-Vt CMOS circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 3, pp. 306- 318, March 2002.

[11] Chappell B., et al., "Library architecture challenges for cell-based design," *Intel Technology Journal*, vol. 8, no. 1, Feb. 2004.

[12] Pant, P., et al., "Dual-threshold voltage assignment with transistor sizing for low power CMOS circuits*," IEEE Transactions on Very Large Scale Integration Systems,* vol. 9, pp. 390-394, 2001.

[13] Wei, L., et al., "Design and optimization of dual-threshold circuits for low-voltage low-power applications*," IEEE Transactions on Very Large Scale Integration Systems*, vol. 7, pp. 16-24, 1999.

[14] Amerasekera E. A., and F. N. Najm, *Failure Mechanisms in Semiconductor Devices*. Wiley& Sons, 1998.

[15] W. Wolf, *Silicon processing for the VLSI era volume 3 – the submicron MOSFET*, Lattice Press, 1994

[16] Chaudhary K., and M. Pedram, "Computing the area versus delay trade-off curves in technology mapping," *IEEE Trans. on Computer Aided Design*, vol. 14, no. 12, pp. 1480-1489, 1995.

[17] Roy, S., et al., "An alpha-approximate algorithm for delay-constraint technology mapping," in *Proc. of the Design Automation Conference,* pp. 367-371, 1999.

[18] Tsui, C.-Y., et al., "Technology decomposition and mapping targeting low power dissipation," in *Proc. of the Design Automation Conference,* pp. 68-73, 1993.

[19] Sheu, B.J., *et al.*, "BSIM: Berkeley short-channel IGFET model for MOS transistors*," IEEE Journal of Solid State Circuits*, vol. 22, pp. 558-566, 1987.

[20] Kawahara, T., M. Horiguchi, Y. Kawajin, G. Kitsukawa, T. Kure, and M. Aoki, "Subthreshold current reduction for decoded-driver by self-reverse biasing," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 1136 – 1143, Nov. 1993.

[21] Abdollahi, A., F. Fallah, and M. Pedram, "Leakage current reduction in CMOS VLSI circuits by input vector control," IEEE Transactions on Very Large Scale Integration Systems, vol. 12, no. 2, pp. 140 – 154, Feb. 2004.

[22] Kim C., and K. Roy, "Dynamic Vth scaling scheme for active leakage power reduction," in *Proc. the conference on Design, automation and test in Europe*, pp. 163 – 167, 2002.

[23] Chang, C-W., et al., "Layout-driven hot-carrier degradation minimization using logic restructuring techniques," in *Proc. of the Design Automation Conference,* pp. 97-101, 2001.

[24] Chen, Z., and I. Koren, "Technology mapping for hot-carrier reliability enhancement," in *Proc. of the SPIE - The International Society for Optical Engineering,* pp. 42-50, 1997.

[25] Ding, C-S., et al., "Gate-level power estimation using tagged probabilistic simulation," *IEEE Trans. on Computer Aided Design*, vol. 17. no. 11, pp.1099-1107, 1984.

[26] Rabaey, J., *Digital integrated circuits: a design perspective*, Upper Saddle River, NJ: Prentice Hall, pp. 198 – 199, 1996.

[27] Leblebici, Y., "Design consideration for CMOS digital circuits with improved hot-carrier reliability*," IEEE Journal of Solid Sate Circuits*, vol. 31, pp. 1014-1024, 1996.

[28] Pedram, M., "Power minimization in IC design: principles and applications," *ACM Transactions on Design Automation of Electronics Systems*, vol. 1, no. 1, pp. 3-56, 1996.

[29] Roy, K., and S.C. Prasad, *Low-power CMOS VLSI Circuit Design*. Wiley-Interscience, 2000.

[30] Sirichotiyakul, S., et al., "Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing," in *Proc. the Design Automation Conference*, pp. 436-441, 1999.

[31] Sentovich, E.M., *et al.*., SIS: A system for sequential circuit synthesis,. 1992, *ERL, University of California, Berkeley.*

[32] H. J. Touati, C. W. Moon, R. K. Brayton and A. Wang, “Performance-oriented technology mapping,” in Proc 6th MIT Conference, Advanced Research in VLSI, W. J. Dally ed., pp. 79-99, 1990.

[33] Sundararajan, V. and D.K. Parhi, "Low power synthesis of dual threshold voltage CMOS VLSI circuits," in *Proc. of the International Symposium on Low Power Electronics and Design,* pp. 139-144, 1999.

[34] Veendrick, Harryy J., “Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits,” *IEEE Journal of Solid-Sate Circuits*, vol. sc-19, pp. 468-473, 1984.

[35] Yonezawa, H., *et al.*, "Ratio based hot-carrier degradation for aged timing simulation of millions of transistors digital circuits," *IEEE Int. Electron Devices Meeting Technical Digest*, vol. pp. 93-96, 1998.

[36] http://www.viragelogic.com/products/tsmc/standard_cell