

SIGNAL-PATH-LEVEL DUAL- V_t ASSIGNMENT FOR LEAKAGE POWER REDUCTION

YU WANG, HUAZHONG YANG and HUI WANG

*Circuit and System Division,
Department of Electronic Engineering, Tsinghua University,
Beijing, 100084, People's Republic of China*

Revised 11 November 2005

Along with the fast development of dual-threshold voltage (dual- V_t) and multi-threshold technology, it is possible to use them to reduce static power in low-voltage high-performance circuits. In this paper, we propose a new method to realize CMOS digital circuits that are implemented with dual- V_t technology. We first present a new signal-path-level circuit model which effectively deals with the fact that there can be two threshold voltages assigned to a single gate. In order to assign proper threshold voltage to all the signal-paths in the circuit, our new algorithms introduce the concept of *subcircuit extraction* and include the hierarchy algorithms which are effective and fast. Experimental results show that our algorithms produce a significant reduction for the ISCAS85 benchmark circuits.

Keywords: Leakage power; dual-threshold voltage; static delay model; graph algorithm; signal-path-level threshold voltage assignment.

1. Introduction

With the growing scaling of integration and the increasing usage of battery-operated devices, power dissipation has become a critical issue of VLSI circuits and systems designs. It is especially true, in the design of portable and wireless electronic systems where power issues have already reached a bottleneck. The total power dissipation, consists of switching power, short circuit power and leakage power, can be expressed as:

$$\begin{aligned} P_{\text{total}} &= P_{\text{dynamic}} + P_{\text{leakage}} + P_{\text{short circuit}} \\ &= \sum_{i=1}^N \left(\frac{1}{2} \alpha_i f C_i V_{dd}^2 + I_{l,i} V_{dd} + \alpha_i f Q_{\text{short},i} V_{dd} \right), \end{aligned} \quad (1)$$

where f is the operation frequency, V_{dd} is the supply voltage, and N is the number of gates. α , C_i , $I_{l,i}$, and $Q_{\text{short},i}$ are transition probability, load capacitance, leakage current, and short circuit charge of the i th gate, respectively.

The behavior of the short circuit power dissipation remains at around 10% of the total power dissipation.¹ As we can see, lowering the supply voltage is the

most effective way to reduce the total power dissipation. However, to maintain the performance at the lower supply voltage, the threshold voltage of transistors must be decreased to the same degree as V_{dd} . Unfortunately, lowering the V_t will lead to an exponentially increase in leakage current thereby leading to a dramatic increase in the standby power dissipation.²

An approximate expression for the subthreshold current which is the main contributor to the total leakage current is given by³:

$$I_{\text{sub}} = Ae^{q(V_{GS}-V_{t0}-\gamma V_{SB}+\eta V_{DS})/nkT} (1 - e^{-qV_{DS}/nkT}), \quad (2)$$

where V_{GS} , V_{DS} , and V_{SB} are the gate–source, drain–source, and source–bulk voltages, respectively, V_{t0} is the zero bias threshold voltage, and A , γ , η , k , T and n are technology-dependent constants.

With the development of the fabrication technology, leakage power dissipation has become comparable to switching power dissipation.⁴ At the 90 nm technology node, leakage power may make up 42% of total power.⁵

The rest of the paper is organized as follows. In Sec. 2, the overview of leakage control methods is presented. In Sec. 3, we give out preliminaries including our new circuit model, delay models and leakage power models. The problem definition is provided in Sec. 4. The details of our algorithms are presented in Sec. 5. The implementation and experimental results are given in Secs. 6 and 7, respectively.

2. Overview of Leakage Control Methods

2.1. Leakage power control techniques

Inevitably, techniques are necessary for reducing the increasing leakage power. These leakage control methods can be broadly categorized into two main categories: process-level and circuit-level techniques.

At the process-level, leakage reduction can be achieved by controlling the dimensions (length, oxide thickness, junction depth, etc.) and doping profile in transistors. Here we mainly talk about circuit design techniques. There are several circuit design techniques, namely, input vector control,⁶ power gating^{7,8} and multi- V_t design.

The input vector control method suffers from inefficiency with large circuits and extra control logic which brings power and area overloads, and finding the minimum leakage vector is still an NP problem. In power gating method, the extra area and delay due to the insertion of sleep transistors have considerable influence on the circuit performance. Furthermore, with the supply voltage scaling down, it is becoming harder to turn the circuit on under a very low supply voltage. The multi- V_t design method includes VTCMOS,^{9,10} DVTS^{11,12} and dual- V_t assignment. Both VTCMOS and DVTS suffer from large area and power penalty due to the extra control logic. The circuit using DVTS also suffer from increasing substrate capacitance. The substrate noise becomes another problem.

Among these, the dual- V_t process, which allows both low- V_t and high- V_t transistors on the same chip, is commonly used.¹³ A dual- V_t assignment method^{14–17}

means that a higher threshold voltage can be assigned to some of the transistors in the noncritical paths, in order to reduce the leakage current, while the performance is maintained due to the low- V_t transistors in the critical paths. A source-to-well reverse bias can be applied to some transistors to achieve high thresholds. Furthermore, a dual- V_t MOSFET process was developed,¹⁸ which makes the implementation of dual- V_t logic circuits more feasible. Dual- V_t method results in a significant reduction in total power dissipation and energy. Therefore, determining which gate should be the high- V_t becomes a major emphasis in the research field.

2.2. Dual- V_t optimization review

The method described in Ref. 14 is, for the first time, using the idea that some high- V_t transistors are assigned in the noncritical paths. All the transistors within the gate are either at V_{THhigh} or at V_{THlow} . Each gate is checked where it can be changed to V_{THhigh} without decreasing the minimum slack over all the gates. This method finds a subset of gates which can be transformed to V_{THhigh} . In Ref. 15, a method is presented to gain a “near optimal approach” which has further reduction of leakage power.

While these two methods demonstrated significant savings in leakage power without degradation in performance, they have shown significant drawbacks too.¹⁶ Their selected gates to be transformed into V_{THhigh} are not sufficient. In fact there are more gates that can be assigned. Notice that after the assignment of V_{THhigh} to a gate, the critical path may change. This dynamic change in critical path has not been taken into consideration in Refs. 14 and 15.

In Ref. 16, a different idea is presented which initialized the circuit by assigning a high threshold voltage to all the gates of the circuit, i.e., it essentially configures the circuit to give the minimum power. The algorithm selects a gate which is on the critical path and then assigns V_{THlow} to it. Every time a gate changes, an update of the whole circuit is necessary. The algorithm iterates until there exists at least one gate on the critical path, which is yet to be assigned with V_{THlow} . This method gets better assignment than the two mentioned before, but it has to reiterate the whole circuit every time we decide whether a single gate can be changed into V_{THlow} or not.

Three algorithms are presented in Ref. 17. Algorithm 1 is very similar to the previous one in Ref. 16. Every edge in the circuit graph has a weight in order to decide which gate should be changed. Algorithm 2 considers the signal probability for each node, and reduces the delay subject while minimizing the increase in the standby power. The problem of finding an optimal V_{THhigh} gate assignment is NP complete, and in Algorithm 2, an iterative improvement procedure called *Swep* is carried out as an escape from a local optimal solution. However, the drawback in Ref. 16 that one has to reiterate the whole circuit every time one gate changes still remains in these two algorithms. Algorithm 3 brings an improved version of Algorithm 2. After initializing all the gates with V_{THhigh} , gates in critical subcircuits are changed

into V_{THlow} to meet the timing constraints. Algorithm 2 is used to decide which gate in the subcircuits can be changed into V_{THhigh} to consume less standby power dissipation. The experimental results are almost the same as Algorithm 2, while the CPU time is up to two times less.

The possibility of different transistors having different threshold voltages within a logic gate is not considered in any of the above algorithms. In Ref. 19, a methodology for MVT (mixed- V_t) CMOS circuit design is presented. For MVT (mixed- V_t) CMOS circuits, the transistors within a gate can have different threshold voltages with certain process constraints. Therefore, more transistors can be assigned to V_{THhigh} and larger leakage current reduction can be achieved. However, the algorithms to assign the V_{THhigh} encountered the same drawbacks with the methods described in Refs. 14 and 15.

2.3. *Our algorithms*

In this paper, we assume that all the gates are using the low threshold voltage in order to get the best performance (timing characteristic). The signal-path-level circuit model we used is different from the circuit model which consider a gate as a vertex in a graph. This is to make our algorithms useful for transistor-level leakage control. We use look up table method in our signal-path-level static timing analysis to get the critical paths and noncritical paths of the circuit much faster and with more accuracy. The gates in the critical paths will remain unchanged to maintain the performance; and the gates in the noncritical paths are extracted into several subcircuits. Without reiterating the whole circuit, we focus solely on the subcircuits in which we use new heuristic algorithms to get an optimal result faster.

3. Preliminaries

3.1. *Signal-path circuit model*

A combinational circuit is represented by a directed acyclic graph (DAG) $G = (V, E)$. Traditionally a vertex $v \in V$ represents a CMOS transistor network which realizes a single output logic function (a logic gate), while an edge $(i, j) \in E$, $i, j \in V$ represents a connection from vertex i to vertex j . In this way, the transistors within a vertex that are driven by the same logic signal will be assigned to the same threshold. The assignment of threshold voltages to the transistors in the circuit can be represented as assigning a threshold voltage to a vertex $v \in V$,^{14–16} or assigning a threshold voltage to an edge.¹⁷ Thus, this allows treating the dual- V_t optimization problem as a kind of graph problem. It greatly simplifies delay analysis and standby power estimation during V_t assignment. The effects on delay when a V_t change is made can be easily modeled by static timing analysis (STA). In Fig. 1, a combinational circuit is presented at the left side (Fig. 1(a)); the traditional circuit model is at the right side (Fig. 1(b)).

In our circuit model, a vertex $v \in V$ represents a pin of a CMOS logic gate or a primary input/output; an edge $(i, j) \in E$ represents a connection from vertex

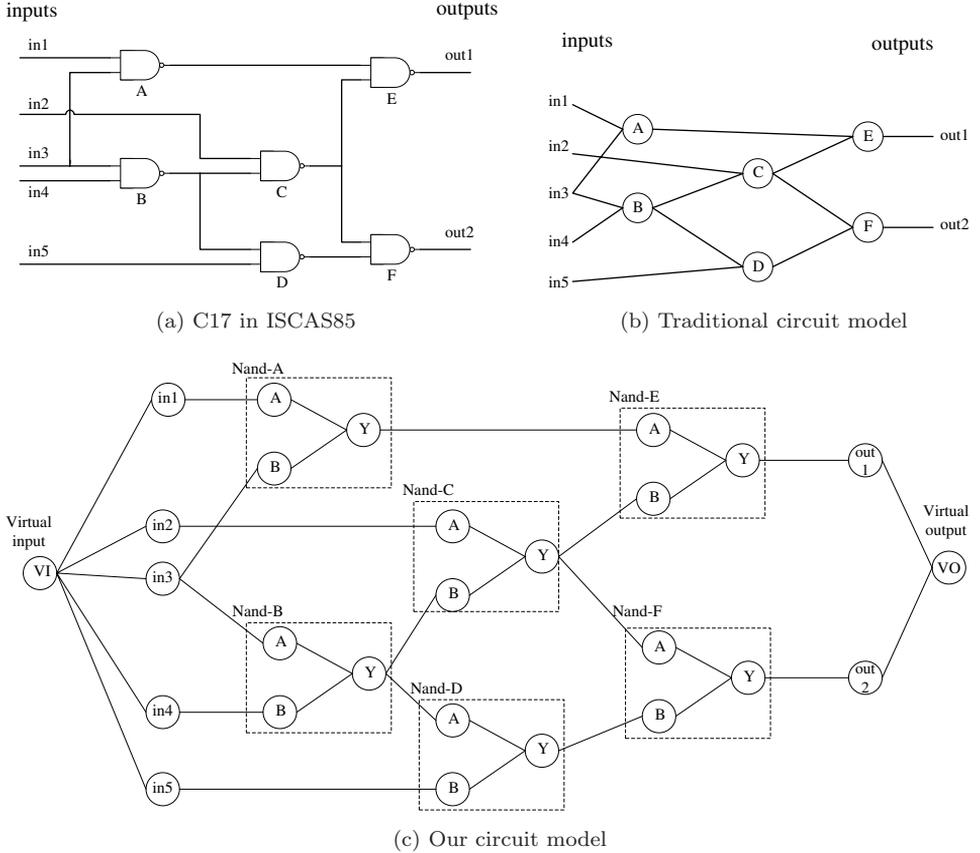


Fig. 1. Circuit and different graph abstraction: (a) original circuit C17 in ISCAS85, (b) traditional graph abstraction, and (c) signal-path circuit model.

i to vertex j . In our model, an edge is the abstraction of a wire connecting two gates or a signal-path in a logic gate from one of its input pins to an output pin. Furthermore, we have added a virtual input vertex and a virtual output vertex to our model. The virtual input vertex is connected to all the primary inputs (PIs) and the virtual output vertex is connected to all the primary outputs (POs). The fan-in of a logic gate’s input pin refers to the number of pins which connect this input pin. The fan-out of a logic gate’s output pin refers to the number of pins which is connected with this output pin. Pins which have a fan-in of zero constitute primary input pins; similarly pins which have a fan-out of zero constitute primary output pins. Figure 1 shows the traditional graph abstraction and our signal-path circuit model (Fig. 1(c)) of circuit C17 from ISCAS85 benchmark.

If vertex $i \in V$ represents one of input pins in gate A and vertex $j \in V$ represents gate A ’s output pin, we define edge $(i, j) \in E$ as a “*signal-path*” and this signal-path belongs to gate A . There are several reasons for using this new circuit model.

Firstly, the signal arrival time may be different for every input pin of a gate. More detailed delay information for every gate is presented since the delay information for every pin of the gate is computed by STA. Secondly, through the definition of signal-path, it is possible to have transistors with different V_t in a single gate at the same time, which means transistors in every signal-path of one gate may have different V_t . Thus, the dual- V_t optimization problem is changed into an assignment of high- V_t to the possible signal-paths. If we neglect the possibility of assigning different threshold voltage to signal-paths which belong to the same gate, it will get the same solution as previous methods.¹⁴ The edge E in the graph represents two kinds of connections. One is “signal-path”, the other is the connection of two pins belonging to different gates respectively which represents a wire between two pins in most cases. Hence, it is possible to consider the interconnect delay during STA in order to get more accurate model of the circuit.

3.2. Delay model

In order to get the delay attributes, we levelize the vertexes in the graph, make sure every two vertexes belong to the same level have no edges between them. Each pin’s fan-ins are not at the same level as itself, its fan-outs are not either; thus an edge $(i, j) \in E$ ’s two vertexes $i, j \in V$ are not at the same level. The delay of an edge $(i, j) \in E$, $i, j \in V$ is denoted by $d_{i,j}$.

We define three attributes for every vertex $v \in V$, they are namely, the arrival time $t_a(v)$, the required time $t_{\text{req}}(v)$, and the slack time $t_{\text{slk}}(v)$. The arrival time $t_a(v)$ is the worst case of delay from the primary inputs to pin v . $t_{\text{req}}(v)$ is the latest time the signal needs to arrive at pin v . We define them as:

$$t_a(v) = \begin{cases} \text{given time of arrival} & \text{if } v \text{ is the virtual input,} \\ \max_{i \in \text{fanin}(v)} \{t_a(i) + d_{i,v}\} & \text{otherwise,} \end{cases} \quad (3)$$

$$t_{\text{req}}(v) = \begin{cases} t_a(v) & \text{if } v \text{ is the virtual output,} \\ \min_{i \in \text{fanout}(v)} \{t_{\text{req}}(i) - d_{v,i}\} & \text{otherwise.} \end{cases} \quad (4)$$

By comparison to the traditional circuit model, the arrival time of a gate is the maximum of its input pins’ arrival time, and the required time of a gate is its output pin’s required time (if the gate is a CMOS transistor network which realizes a single output logic function). The slack time of a gate is also defined as the difference of its arrival time and the required time. The *critical path* of the circuits is constituted by the set of gates that has the minimum slack time value.

We define every edge $(i, j) \in E$, $i, j \in V$ in the graph G also has the attribute $s_{i,j}$ which represents the slack time of the edge:

$$s_{i,j} = t_{\text{req}}(j) - t_a(i) - d_{i,j}. \quad (5)$$

Finally, the slack time of a vertex $v \in V$ is defined as the minimum slack time of its fan-in edges:

$$t_{\text{slk}}(v) = \min_{i \in \text{fanin}(v)} s_{i,v}. \quad (6)$$

In our delay model, we define the *critical path* of the circuits as the set of edges that has the minimum slack time value. If there is no negative slack in the circuit, then timing constraints are satisfied.²⁰ The delay of a circuit is computed by STA tools under the signal-path-level.

Gate delay data are obtained by a table of the gate delay for standard cells which is provided by the IC manufacturers. We use a circuit scheme for the implementation of each signal-path. Consider a two-input NAND gate, Fig. 2 shows the four conditions of the threshold voltage changes. The original NAND gate with all the transistor having low threshold voltage is given in Fig. 2(a); Figs. 2(b) and 2(c) show how one of the two signal-paths' threshold voltage changes. If both signal-paths in the NAND gate can be changed, then all the transistors in this gate are changed into high threshold voltage and this is illustrated by Fig. 2(d).

Notice that every signal-path in the same gate can have different delay difference when it changes between high threshold voltage condition and low threshold voltage condition; and when several signal-paths can be simultaneously changed in one gate, the delay difference is even more complicated because of the interactions between the changed signal-paths. Here, we select the largest delay difference of all the signal-paths' change schemes as the reference delay difference of the signal-path in this kind of gate. The signal-path delay data are then derived from the look up table of the standard cells and HSPICE simulation.

3.3. Leakage power model

Leakage power of a large scale circuit can be estimated by the summation of every gate leakage power. As each gate may have several signal-paths, the leakage power change due to the signal-paths' threshold voltage change should be well estimated. Our circuit model makes it possible to assign different threshold voltages to each signal-path of one logic gate.

Using HSPICE and a typical library for each circuit scheme of the signal-path, we can create a table of leakage power for the signal-path's threshold voltage change. Consider the two inputs NAND gate again. It has three kinds of changes: no signal-path is changed, one of the two signal-paths is changed, and all the signal-paths' threshold voltage is changed. Table 1 shows the standby power for a two-input NAND for the four signal-path change schemes according to Fig. 2.

When all the signal-paths are changed in a gate with two signal-paths, the leakage power saving is larger than twice the leakage power saving of changing only one signal-path in that gate. We also find out that the leakage power change due to only one signal-path's change is always the same and furthermore, if there are k signal-paths which can change their threshold voltage in a gate with w signal-paths

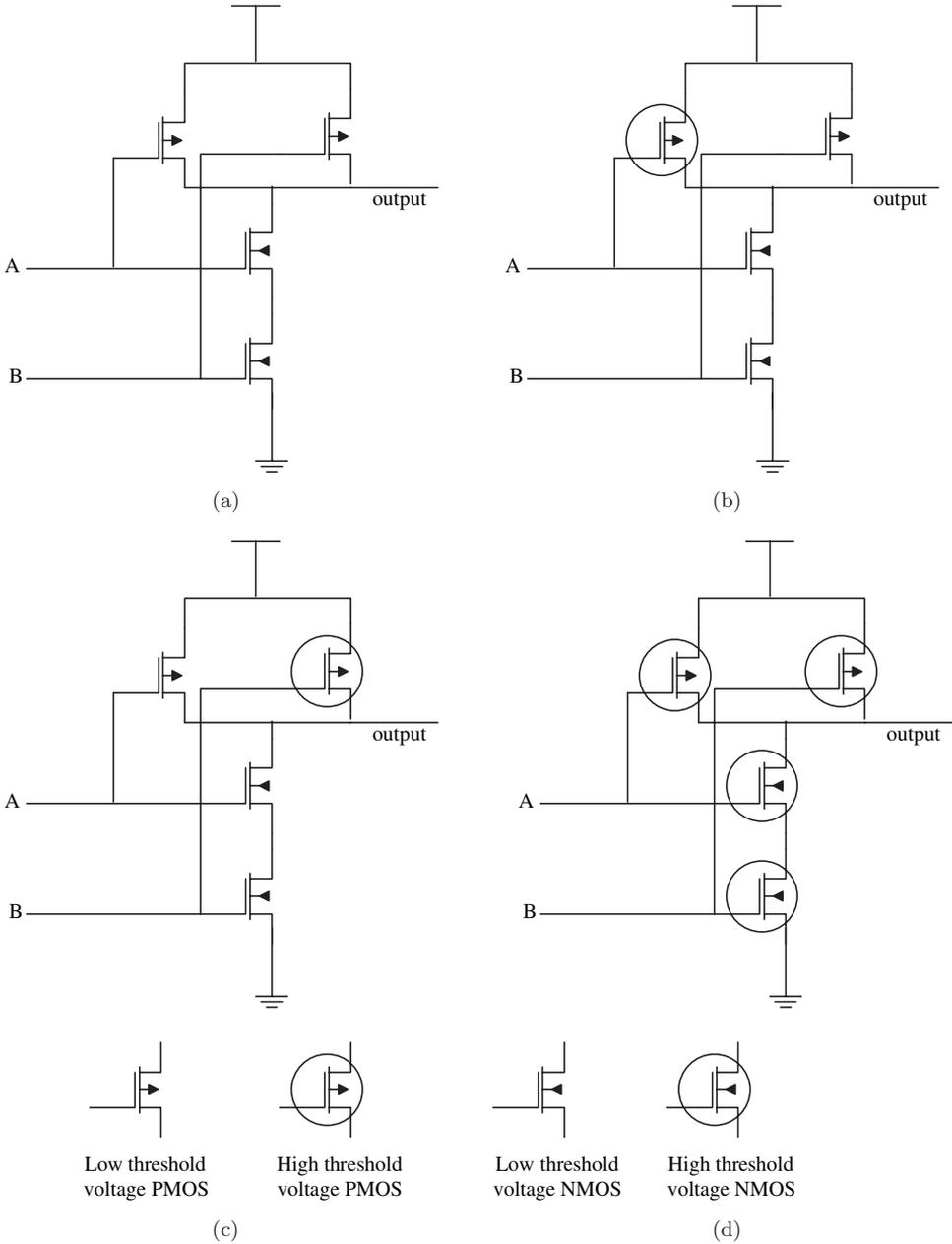


Fig. 2. Circuit schemes of signal-path's threshold voltage change in NAND2.

($k < w$), no matter how to choose the k signal-paths, the power change due to k signal-paths' threshold voltage change is always the same. The leakage power saving due to k signal-paths' threshold voltage change is nearly the same as k times the leakage power saving due to only one signal-path's change. However, if all the w

Table 1. Leakage power (nw) for a two-input NAND for the four signal-path change schemes.

Input A&B	Scheme (a)	Scheme (b)	Scheme (c)	Scheme (d)
00	0.1178	0.1178	0.1178	0.0026
01	0.6827	0.6827	0.6827	0.0039
10	0.6147	0.6147	0.6147	0.0061
11	3.3830	1.6982	1.6982	0.0133
Average	1.1996	0.7784	0.7784	0.0065

signal-paths in the gate is changed, the leakage power saving is larger than w times the leakage power saving due to only one signal-path's change. Finally, we use two values to represent each signal-path's leakage power attributes: the larger one is for all the signal-paths in that gate can change into high threshold voltage, and it equals to the leakage power saving due to the gate's threshold voltage change divided by the number of signal-paths in the gate; and the smaller one is for other conditions which equals to the leakage power saving due to only one signal-path's change.

We do not consider the signal probability at each pin of the gates, and we may use logic simulation or local probability propagation in our future work to make it possible to combine transistor stacking effects with the circuit analysis to get a more accurate leakage power estimation table.

4. Problem Definition

We first give some definitions to represent attributes of the above models. Transistors in every signal-path can have different threshold voltage V_t , thus different V_t is represented by labeling each signal-path by $x_{i,j}$, where $x_{i,j} = 0$ means that the transistors in signal-path $(i, j) \in E$, $i, j \in V$ have a low threshold voltage, i.e., $V_t = V_{THlow}$; $x_{i,j} = 1$ means that the transistors in signal-path $(i, j) \in E$, $i, j \in V$ have a high threshold voltage, i.e., $V_t = V_{THhigh}$. Assuming there are L kinds of gates in the given circuit. We define $\Delta d_{i,j}(k)$ as the difference between $d_{i,j}$ of signal-path $(i, j) \in E$ with V_{THhigh} and V_{THlow} . $1 < k \leq L$ represents the signal-path's type associated with the gate type. We use $\Delta P_{i,j}(k)$ to represent the signal-path's leakage power saving attribute where k also represents the signal-path's type. As we mentioned before, $\Delta P_{i,j}(k)$ may have two values under different circuit scheme.

The dual- V_t optimization is generally defined as a problem to assign one of two threshold voltages, V_{THhigh} and V_{THlow} , to each transistor, to satisfy the timing constraints. Thus, the problem can be formally expressed as:

$$\max_{x_{i,j}} \sum_{(i,j) \in E} x_{i,j} \Delta P_{i,j}(k) \quad (7)$$

or

$$\max_{\lambda(i,j,k)} \sum_{(i,j) \in E} \lambda(i,j,k) \Delta P_{i,j}(k), \quad (8)$$

where $\lambda(i, j, k)$ of a signal-path $(i, j) \in E$, $i, j \in V$ is defined as:

$$\lambda(i, j, k) = \begin{cases} 1 & \text{if } s_{i,j} \geq \Delta D_{i,j}(k), \\ 0 & \text{else.} \end{cases} \quad (9)$$

In order to select the signal-path which can lead to larger leakage power reduction, we also define the priority for signal-path $(i, j) \in E$, $i, j \in V$ whose slack time is not zero as following expression:

$$\text{Priority}_{(i,j)} = \frac{\Delta P_{i,j}(k)}{\Delta D_{i,j}(k)}. \quad (10)$$

Notice that this priority of a signal-path may have two values, since $\Delta P_{i,j}(k)$ may have two values. Changing high priority signal-paths to high threshold voltage will get high return because it achieves leakage power reduction at low delay penalty.

As we described before, if we neglect the possibility of assigning different V_t to signal-paths belongs to the same gate, we will get the same solution of the dual- V_t gate-level assignment problem.

5. The Algorithm

5.1. Initialization

We have assumed the DAG representation $G(V, E)$ of a signal-path-level combinational circuit. This graph is leveled to indicate the depth of the vertex in the graph. The level of the virtual input (source vertex) is defined to be 0 and it is also labeled as 0. Therefore, the level of any vertex $v \in V$, $l(v)$, is defined as:

$$l(v) = 1 + \max_{i \in \text{fanin}(v)} \{l(i)\}, \quad (11)$$

and the level of any signal-path $(i, j) \in E$, $i, j \in V$, $l(i, j)$ is also defined as:

$$l(i, j) = 1 + \max_{(u,i) \in \text{fanin}(i,j)} \{l(u, i)\}. \quad (12)$$

The algorithm for leveling a graph $G(V, E)$ is given below:

Levelize(G)

- 1 Set the virtual input vertex as level 0; $m = 1$;
- 2 While ($V \neq \text{virtual output}$) {
 - 1 Delete all the input vertex from V ;
 - 2 Find the new input vertex set $\{V_{\text{in}}\}$;
 - 3 Set $\{V_{\text{in}}\}$ as level m ;
 - 4 $m = m + 1$;
- }
- 3 Set the virtual output vertex as level m .

We initialize the circuit by assigning a low-threshold voltage ($V_{TH\text{low}}$) to all the signal-paths of the circuit, i.e., it essentially configures the circuit to have the

minimum delay. In the initialization procedure, we decide the delay attributes of every vertex and edge in the graph: the arrival time $t_a(v)$, the required time $t_{\text{req}}(v)$ and the slack time $t_{\text{slk}}(v)$, the edge slack time $s_{i,j}$, the edge propagation delay $d_{i,j}$. All these attributes can be calculated using static timing analysis and the formula we have denoted before. The fan-ins of a vertex are the former level vertexes which are connected with this vertex; the fan-outs of a vertex are the next level vertexes which are connected with this vertex.

Since every edge has a slack time, we extract all the nonzero slack time edges to construct a set of subgraphs $G_{\text{sub}1}, G_{\text{sub}2}, \dots, G_{\text{sub}n}$. The critical paths' delay attributes are not affected when the V_t of some signal-paths on noncritical paths are changed. Therefore, the assignment of the V_t in the whole circuit is decomposed into several small problems, which have much smaller solution space and thus are more easier to get the optimal assignment of V_t . When the $V_{TH\text{low}}$ in the circuit is given, we will just focus on the subgraph to decide the optimal high threshold voltage value without reiterating the whole circuit.

The algorithm for the initialization *Initialization* (G) is given below:

Initialization(G)

- 1 Assign $V_{TH\text{low}}$ to each signal-path;
- 2 Perform static timing analysis,
determine all the delay attributes for the circuits;
- 3 Extract Subgraph ($G_{\text{sub}1}, G_{\text{sub}2}, \dots, G_{\text{sub}n}$) of noncritical paths.

5.2. Assignment of dual- V_t

The assignment of dual- V_t to the whole circuit is converted into the assignment of dual- V_t to several subgraphs. Two methods are given to assign dual- V_t to signal-paths in a subgraph. In the third part of this section, we will consider gates other than signal-paths as the optimization object.

5.2.1. Algorithm 1: Forward depth-first low- V_t assignment

This first algorithm starts with all the signal-paths being high threshold voltage in the subgraph. Our purpose is to select the signal-paths, which can be assigned to low threshold voltage in order to decrease the delay of the circuit. Thus, we try to assign low threshold voltage to the signal-paths on the critical paths. Using the delay attributes which are gained by the former STA process, we will get the arrival time of the primary inputs and the delay constraints of the subgraph. When we perform the STA to the subgraph; we will get a new set of delay attributes for each vertex and signal-path. Algorithm 1 uses a depth first signal-path selection from the primary outputs to the primary inputs and then assign low threshold voltage to them. The algorithm to a subgraph is similar to the gate-level algorithm presented in Ref. 16 which can assign more high-threshold gates on the noncritical path than the algorithm presented in Ref. 14. We update the delay attributes every time a change occurs in order to get the correct delay attributes of other unvisited signal-paths.

The method is depicted below:

Forward Depth-First Low- V_t Assignment(G)

- 1 Assign V_{THhigh} to each signal-path of the subgraph;
- 2 Perform STA to subgraph;
- 3 Assume $t_{req}^{max} = \max_{v \in POs} t_{req}(v)$;
- 4 For each vertex $v \in POs$
 - If ($t_{req}(v) = t_{req}^{max}$)
 - $Enqueue(Q, v)$; //push v into queue Q
- 5 While ($Q \neq \emptyset$) {
 - $u = Dequeue(Q)$; //pop queue Q
 - $t_{req}(i) = \max_{v \in fanin(u)} t_{req}(j)$; //Get the vertex i with maximum
 - //required arrive time, $j \notin PIs$
 - If (i, u belong to the same gate){
 - If (signal-path (i, u) is V_{THhigh})
 - Change signal-path (i, u) into V_{THlow} ;
 - }
 - $Enqueue(Q, i)$; // push i into queue Q
 - Update the delay attribution of the subgraph G_{sub} ;
- 6 Do 2–5 until all the signal-paths in the critical paths are V_{THlow} .

5.2.2. Algorithm 2: Priority-based high- V_t assignment

The dual- V_t optimization problem in the subgraph can be regarded as an optimal slack distribution in a subgraph in which every signal-path has a positive slack time value. Levelize the subgraph based on the signal-paths and label every signal-path again. Assume the subgraph $G_{sub}(V_{sub}, E_{sub})$ has n levels. Since the subgraph is also a DAG, if we consider any signal flow from one primary input to one primary output, we will have:

$$\sum_m (s_{(i,j)_m} \times \lambda(i, j, k)) \leq t_{slk}(PO), \quad m = 1, 2, \dots, n; \quad m = l(i, j), \quad (13)$$

$$s_{(i,j)_m} = \begin{cases} \Delta D_{i,j}(k) & \text{if the path which passes the signal} \\ & \text{in } m\text{th level has high threshold,} \\ 0 & \text{else.} \end{cases} \quad (14)$$

$t_{slk}(PO)$ is the slack time of the primary output pin on the path which passes the signal. Therefore, the slack distribution in subgraph $G_{sub}(V_{sub}, E_{sub})$ can be expressed as:

$$\max_{\lambda(i,j,k)} \left\{ \sum_{m=1}^n \sum_{l(i,j)=m} (\Delta P_{i,j}(k) \times \lambda(i, j, k)) \right\}. \quad (15)$$

The second algorithm aims to find an optimal solution to satisfy the constraint above in a fairly fast way using a priority-based method. This method starts with all the signal-paths of the subgraph in V_{THlow} configuration. The objective here is to reduce the standby power as much as possible without increasing the delay. The main idea in the algorithm to achieve the objective is to change the signal-paths with high priority as much as possible without delay influence. So, we select the level with highest priority in the subgraphs and then change the threshold voltage of the signal-paths in that level. This procedure is clearly illustrated in Step 2 of algorithm *Deal_with_subgraph*(G_{sub}). Notice that we only add the priority of all the signal-paths with only one fan-out to gain the priority for each level. The reason is changing the V_t of a signal-path with only one fan-out has a much smaller effect on slack attributes on the graph compared to changing the V_t of a signal-path with multiple fan-outs.

When signal passes the p th level signal-path (i, j) to the primary outputs, the minimum slack time of these primary outputs is defined as t_{slk}^{min} . In the following algorithm, *Deal_with_subgraph*(G_{sub}) for each subgraph spends most of the computation time in the real time implementations. During Step 1 of *Deal_with_subgraph*(G_{sub}), if all the remaining signal-paths in the subcircuit have more than one fan-out, we would just check them level by level to see if it can be changed or not. Finally, all the signal-paths are visited and a near optimal dual- V_t assignment is given.

The basic steps of Algorithm 2: *Priority-Based High- V_t Assignment* are shown below and the main function *Deal_with_subgraph*(G_{sub}) is also depicted:

Priority-Based High- V_t Assignment(G_{sub})

- 1 Assign V_{THlow} to each signal-path of the subgraph;
- 2 *Enqueue*(Q, G_{sub}); // push G_{sub} into queue Q
- 3 While ($Q \neq \emptyset$) {
 - 31 $G_{sub} = Dequeue(Q)$; // pop subgraph queue Q
 - 32 *Deal_with_subgraph*(G_{sub});
 - 33 Do STA to extract the subgraphs set of G_{sub} : $sub\{G_{sub}\}$;
 - 34 *Enqueue*($Q, sub\{G_{sub}\}$); // push the subgraphs set of
// $G_{sub}, sub\{G_{sub}\}$ into queue Q
- }
- 4 Complete high threshold voltage assignment.

Deal_with_subgraph(G_{sub})

- 1 If (there is no single output signal-path in the subgraph G_{sub}) {
 - Check the all the signal-paths to see if they can be changed or not from low level to high level;
- }

```

2 else{
  21 Levelize( $G_{\text{sub}}$ ) ; // Levelize the subgraph
  22 Set the priority  $Priority_{\text{level}_p} = 0$  for each level  $p$  ;
      For each signal-path  $(i, j)$  with  $l(i, j) = p$  { //compute the priority of
                                                //each level
          If (fanout( $i, j$ ) = 1) {
              If ( $\Delta D_{i,j}(k) < t_{\text{slk}}^{\text{min}}$ ) {
                  Set the signal-path  $(i, j)$  into can be changed state;
                  Check other signal-path(s) which belongs to the same
                  gate with  $(i, j)$  if they can all be changed or not to
                  decide  $Priority_{(i,j)}$ ;
                   $Priority_{\text{level}_p} = Priority_{\text{level}_p} + Priority_{(i,j)}$ ;
              }
          }
      }
  }
  Select the level  $h$  with the highest level priority:  $Priority_{\text{level}_h}$ ;
  23 If ( $Priority_{\text{level}_h} \neq 0$ ) {
      Change all the signal-path that can be changed in level  $h$ ;
      Update the higher level's delay information, go back to 22;
  }
}

```

5.2.3. Gate-level optimization

If we do not consider the condition that signal-paths in the same gate can have different threshold voltages, we can get the solution for gate-level dual- V_t optimization. Therefore, during the subgraph extraction, we will only consider the gates in which all the signal-paths' slack times are positive. It could be easily realized by mapping a whole gate to a single vertex in the graph. The arrival time of the gate is the maximum of the arrival times of the gate's input pins. The required time of the gate is the output pin's required time. The slack time of the gate is the difference between the arrival time and the required time of the gate. Through a little change in Algorithms 1 and 2, we can get the gate-level optimization of the circuits.

5.3. Get optimal $V_{TH\text{high}}$

Due to the exponential relationship between threshold voltage and substrate leakage current, a higher threshold voltage will significantly reduce the leakage power. However, the higher threshold voltage will result in a higher propagation delay. The high threshold voltage is empirically assumed to be $0.2V_{dd} < V_{TH\text{high}} < 0.5V_{dd}$.¹⁶ Typical value of $V_{TH\text{low}}$ is $0.2V_{dd}$ due to the noise margin and other parameters constraints.²¹ Thus, it is important to decide the value of high threshold voltage. If the value of the $V_{TH\text{high}}$ is close to the value of the $V_{TH\text{low}}$, then there will be much

more signal-paths that can be changed into V_{THhigh} , and it also gives small leakage current improvement. On the other hand, if the V_{THhigh} is close to $0.5V_{dd}$, there might be less signal-paths that can be assigned with V_{THhigh} despite the fact that each of them will bring a large amount of leakage current reduction. Therefore, there must be an optimal V_{THhigh} corresponding the largest saving of the whole circuits.

The algorithm of obtaining the optimal V_{THhigh} is given blow:

Optimal- $V_{THhigh}(G)$

- 1 *Initialization*(G); // get subgraphs consisted of signal paths
// with nonzero slack time
 - 2 For all the subgraph $\{G_{sub}\}$ {
 - 21 $V_{THhigh} = V_{THhigh_start}$; $P_{min} = \infty$;
 - 22 While ($V_{THhigh} < 0.5V_{dd}$) ¹⁶ {
 - Assignment of dual- V_t ($\{G_{sub}\}$);
 - Estimate the standby leakage power $P_{leakage}$;
 - If ($P_{leakage} < P_{min}$) {
 - $P_{min} = P_{leakage}$;
 - $V_{THhigh_opt} = V_{THhigh}$;
 - $V_{THhigh} = V_{THhigh} + \Delta V_t$;
- 3 The V_{THhigh_opt} is the optimal value of V_{THhigh} .

V_{THhigh_start} and ΔV_t here depends on the technology. The algorithm is only dealing with the subgraphs. Thus, it is much simpler and uses less computation time and space compared to the methods stated in Refs. 14 and 16.

6. Implementation

The above algorithms have been implemented in C++ under signal-path-level static timing analysis environment. The value of various transistor parameters have been taken from the TSMC library, the effect channel length is $0.13\mu m$ and the gate oxide thickness is $2.4nm$. The circuit temperature is assumed to be $110^\circ C$. The leakage power table and delay look-up table is created by HSPICE simulation. In our analysis, the low threshold voltage and the supply voltage of the original circuits are assumed to be $0.2V$ and $1.2V$, and high threshold voltage during the dual- V_t optimization is assumed to be $0.3V$. In the optimal high- V_t acquirement algorithm, the high- V_t changed from $0.25V$ to $0.7V$.

7. Experimental Results

First, we can easily get the optimized circuit of C17 which belongs to ISCAS85 benchmark circuits. In Fig. 3, the signal-paths labeled in red can change their threshold voltage into high threshold voltage. If we perform gate-level optimization to C17, only NAND_A can be changed into high threshold voltage. The leakage power saving of C17 is respectively 16.3% and 28.7% for gate-level and signal-path-level optimization.

Figure 4 shows the leakage power savings for ISCAS benchmark circuits using Algorithms 1 and 2. Gate-level and signal-path-level optimization’s lead to different results for leakage power saving, and obviously more leakage reduction can be achieved through signal-path-level optimization since there are actually more transistors in the implementation of the circuit which can be assigned to high threshold voltage.

The shortage of gate-level algorithms^{14–16} was addressed in the introduction part, here we only compare the gate-level algorithm derived from Algorithm 2 (Priority-based(PB) High- V_t Assignment) with the signal-path-level Algorithm 2. Obviously the signal-path-level algorithm will take more memory and computational time, for the DAG extracted from the circuit is several times larger than the one in gate-level. Since we perform the same signal-path-level STA process, the gate-level algorithm is taking some extra time to extract gate-level timing attributes which leads to smaller gap comparing to the signal-path-level algorithm. As we can see from Table 2, by introducing the subcircuit extraction concept, signal-level algorithm takes approximately 4.2X times larger memory and 1.6X more time than the gate-level algorithm. Thus our signal-level algorithm is comparable with

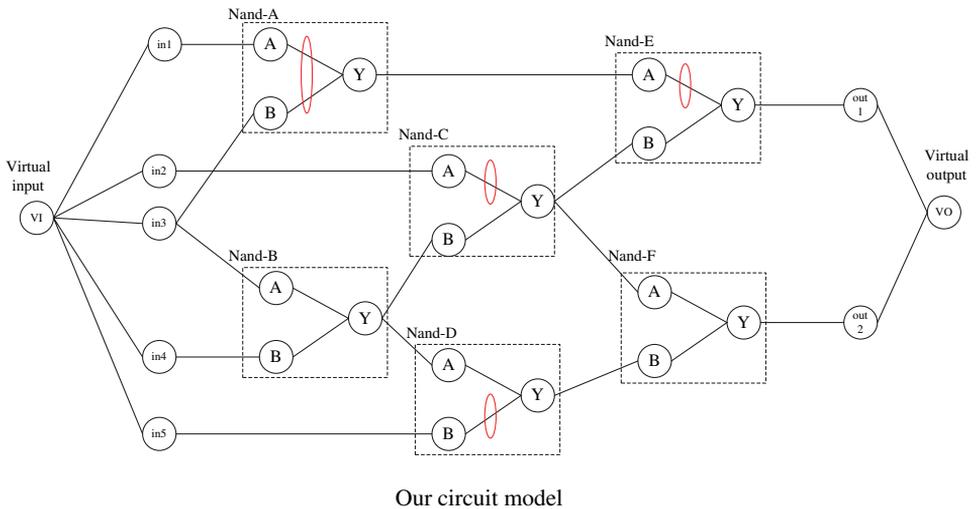


Fig. 3. C17’s signal-path change scheme.

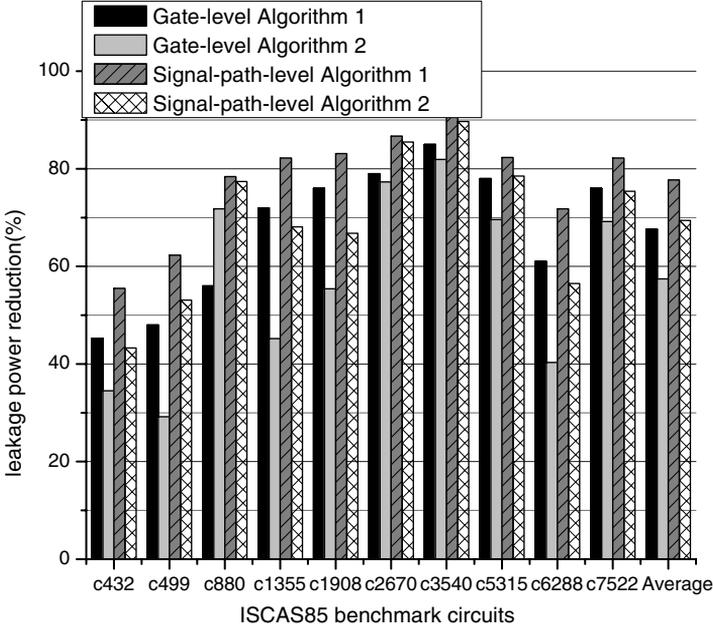


Fig. 4. Leakage power savings using the above algorithms.

Table 2. Comparison of gate-level and signal-path-level algorithms.

Benchmark	Gate-level algorithm			Signal-path-level algorithm		
	CPU time (s)	Memory usage (kb)	Reduction (%)	CPU time (s)	Memory usage (kb)	Reduction (%)
C432	2.5	7508	34.5	7.1	40996	43.3
C499	12.5	7624	29.2	16.4	43156	53.1
C880	16.1	8244	71.8	24.3	54780	77.4
C1355	19.4	13604	45.2	38.2	62272	68.1
C1908	23.5	20476	55.4	42.8	83336	66.8
C2670	35.3	21996	77.3	67.2	88800	85.5
C3540	57.6	23278	81.9	79.3	95888	89.7
C5315	63.6	26020	69.6	147.3	108204	78.5
C6288	309.5	33642	40.3	336.7	128624	56.5
C7522	168.3	42188	69.2	329.9	151940	75.4
Average	70.83	20458	57.44	108.92	85799.6	69.43

gate-level algorithm, meanwhile achieves about 12% more average leakage power reduction.²²

Table 3 reports the leakage power savings and CPU time of different algorithms for signal-path-level dual- V_t assignment. The results indicate that Algorithm 1

Table 3. Leakage power savings and CPU time for different algorithms.

Benchmark	PI//PO	Algorithm 1		Algorithm 2		
		FDF low- V_t assignment		PB high- V_t assignment		
		Reduction (%)	CPU time (s)	Reduction (%)	CPU time (s)	Optimal V_{THhigh} (V)
C432	36//7	55.5	17.2	43.3	7.1	0.359
C499	41//32	62.3	43.7	53.1	16.4	0.319
C880	60//26	78.4	45.2	77.4	24.3	0.359
C1355	41//32	82.2	94.7	68.1	38.2	0.339
C1908	33//25	83.1	139.6	66.8	42.8	0.279
C2670	233//140	86.7	153.5	85.5	67.2	0.359
C3540	50//22	92.5	200.3	89.7	79.3	0.339
C5315	178//123	82.3	273.6	78.5	147.3	0.299
C6288	32//32	71.8	388.5	56.5	336.7	0.329
C7522	207//108	82.2	464.9	75.4	329.9	0.309

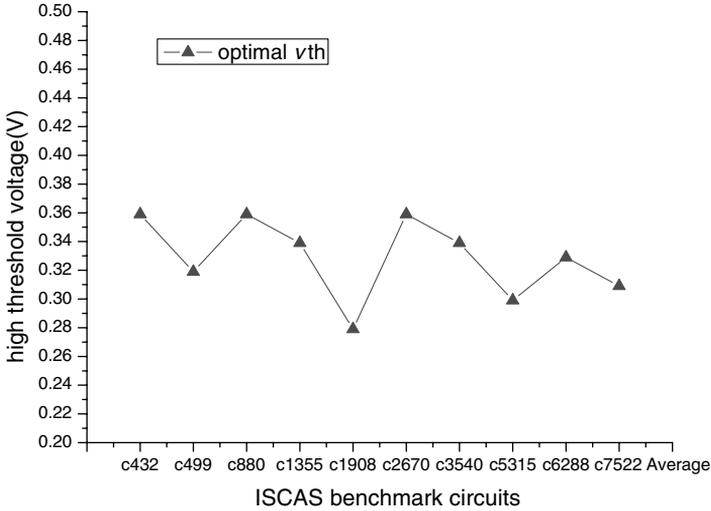


Fig. 5. Optimal V_{THhigh} for ISCAS85 benchmark circuits.

(Forward Depth-First (FDF) Low- V_t Assignment) takes more CPU time. On the contrary Algorithm 2 (PB High- V_t Assignment) spends much lesser CPU time with lower leakage reduction.

The optimal high threshold voltage for ISCAS85 benchmark circuits is shown in Fig. 5.

8. Conclusion

In this paper, we have proposed a new circuit model for combinational circuit. We have given two algorithms for the assignment of high threshold voltage to a

maximum number of signal-paths defined in our new circuit model without violating the delay constraints. The algorithms are sped up by the proper extraction of subgraphs. By using a delay look-up table and a leakage power table generated by HSPICE simulation, we find that approximately 12% more leakage power savings can be achieved under the signal-path-level optimization than the gate-level optimization.

Acknowledgments

This project was supported by National 863 project of China (Nos. 2004AA1Z1050 and 2005AA1Z1230) and National Natural Science Foundation of China (Nos. 90207001 and 60506010).

References

1. D. Duarte, N. Vijaykrishnan, M. J. Irwin and M. Kandemir, Formulation and validation of an energy dissipation model for the clock generation circuitry and distribution networks, *Proc. Int. Conf. VLSI Design* (2001), pp. 248–253.
2. R. X. Gu and M. I. Elmasry, Power dissipation analysis and optimization of deep submicron CMOS digital circuits, *IEEE J. Solid State Circuits* **31** (1996) 887–893.
3. B. J. Sheu *et al.*, BSIM: Berkeley short-channel IGFET model for MOS transistors, *IEEE J. Solid State Circuits* **22** (1987) 558–566.
4. G. Moore, No exponential is forever: But forever can be delayed, *IEEE ISSCC Dig. Tech. Papers* (2003), pp. 20–23.
5. J. Kao, S. Narendra and A. Chandrakasan, Subthreshold leakage modeling and reduction techniques, *ICCAD* (2002), pp. 141–149.
6. S. Mukhopadhyay, C. Neau, R. T. Cakici, A. Agarwal, C. H. Kim and K. Roy, Gate leakage reduction for scaled devices using transistor stacking, *IEEE Trans. Very Large Scale Integration (VLSI) Syst.* **11** (2003) 716–729.
7. S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukada and J. Yamada, 1 V multi-threshold CMOS DSP with an efficient power management technique for mobile phone application, *Proc. IEEE Int. Solid State Circuits Conf.* (1995), pp. 318–319.
8. J. W. Tschanz, S. Narendra, Y. Ye, B. A. Bloechel, S. Borkar and V. De, Dynamic sleep transistor and body bias for active leakage power control of microprocessors, *IEEE J. Solid State Circuits* **38** (2003) 1838–1845.
9. S. Narendra, A. Keshavarzi, B. A. Bloechel, S. Borkar and V. De, Forward body bias for microprocessors in 130-nm technology generation and beyond, *IEEE J. Solid State Circuits* **38** (2003) 696–701.
10. C. H. Kim, K. Jae-Joon, S. Mukhopadhyay and K. Roy, A forward body-biased-low-leakage SRAM cache: Device and architecture considerations, *Proc. 2003 Int. Symp. Low Power Electronics and Design, 2003 ISLPED'03*, 25–27 August 2003, pp. 6–9.
11. C. H. Kim and K. Roy, Dynamic VTH scaling scheme for active leakage power reduction, *Proc. Design, Automation and Test in Europe Conf. Exhibition, 2002*, 4–8 March 2002, pp. 163–167.
12. K. Nose, M. Hirabayashi, H. Kawaguchi, S. Lee and T. Sakurai, VTH-hopping scheme for 82% power saving in low-voltage processors, *Proc. IEEE Conf. Custom Integrated Circuits, 2001*, 6–9 May 2001, pp. 93–96.

13. V. De, Leakage-tolerant design techniques for high performance processors (invited paper), *Proc. 2002 Int. Symp. Physical Design*, 7–10 April 2002, San Diego, California, USA, p. 28.
14. L. Wei, Z. Chen and K. Roy, Design and optimization of dual threshold circuits for low voltage, low power applications, *IEEE Trans. VLSI Syst.* **17** (1999) 16–24.
15. V. Sundararajan and K. K. Parhi, Low power synthesis of dual threshold voltage CMOS VLSI circuits, *Proc. ISLPED* (1999), pp. 363–368.
16. N. Tripathi, A. Bhosle, D. Samanta and A. Pal, Optimal assignment of high threshold voltage for synthesizing dual threshold CMOS circuits, *Fourteenth Int. Conf. VLSI Design*, 3–7 January 2001, pp. 227–232.
17. W. Qi and S. B. K. Vrudhula, Algorithms for minimizing standby power in deep submicrometer, dual- V_t CMOS circuits, *IEEE Trans. Computer-Aided Design of Integrated Circuits Syst.* **21** (2002) 306–318.
18. Z. Chen *et al.*, 0.18 μm dual V_t MOSFET process and energy-delay measurement, *IEDM Dig.* (1996), p. 851.
19. L. Wei, Z. Chen, K. Roy, Y. Yibin and V. De, Mixed- V_{th} (MVT) CMOS circuit design methodology for low power applications, *Design Automation Conf. 1999. Proc. 36th*, 21–25 June 1999, pp. 430–435.
20. S. Devadas, A. Ghosh and K. Keutzer, *Logic Synthesis* (McGraw-Hill, New York, 1994).
21. H. Oyamatsu *et al.*, Design methodology of deep submicron CMOS devices for 1 V operation, *IEICE Trans. Electron* **E79-C** (1996) 1720–1724.
22. Y. Wang, H. Yang and H. Wang, Signal-path level assignment for dual- V_t technique, *Proc. IEEE PRIME 2005*, Lausanne, 25–28 July 2005, pp. 52–55.