LOW POWER SYNTHESIS OF DUAL THRESHOLD VOLTAGE CMOS VLSI CIRCUITS

Vijay Sundararajan *Keshab K. Parhi*

Dept. of ECE University of Minnesota
Minneapolis, MN 55455
E-mail: vijay@ece.umn.edu, parhi@ece.umn.edu

ABSTRACT

The use of dual threshold voltages can significantly reduce the static power dissipated in CMOS VLSI circuits. With the supply voltage at 1V and threshold voltage as low as 0.2V the subthreshold leakage power of transistors starts dominating the dynamic power. Also, many times a large number of devices spend a long time in a standby mode where the leakage power is the only source of power consumption. We present a near-optimal approach to synthesize low static power CMOS VLSI circuits with two threshold voltages that reduces power consumption compared with a previous approach by upto 29.45%. Also, presented is a technique which finds static power optimal configurations for CMOS VLSI circuits when arbitrary number of threshold voltages are allowed.

1. INTRODUCTION

Supply voltage reduction is the most effective technique to lower power consumption in CMOS VLSI circuits. Dynamic power depends quadratically on the supply voltage and static power is directly proportional to it. Ideally, portable devices should be operated with a 1V supply voltage, [1], so that commercially available battery cells can be used as a source for power supply. Unfortunately, when supply voltage is reduced there is a penalty on performance,[2]. In order to maintain performance threshold voltage scaling is currently employed [2]. With a 1V supply voltage, adequate performance is often only possible when the threshold voltage is scaled down to 0.2V. Unfortunately at such a low threshold voltage the static power dissipated in CMOS VLSI circuits, which has an exponential dependence on the negative of the threshold voltage, blows up. In fact at threshold voltage levels $\approx$ 0.2V the static power dominates the dynamic power dissipated in CMOS VLSI circuits. The chief component of the static power dissipated in CMOS VLSI circuits is due to the subthreshold leakage current present between V_{DD} and GND in NMOS and PMOS transistor stacks. This leakage takes place through transistors which are turned off due to 0V (5V) at the gate of the NMOS (PMOS). Therefore, the leakage current (power) becomes an issue when portions of a circuit are in an idle state, i.e. in standby mode, for a considerable length of time. The length of time when subcircuits are idle can be quite substantial in applications like X-servers. It is therefore imperative to devise techniques to reduce the static power dissipated in such applications.

Multithreshold CMOS [1], [3] has been recommended as a solution to this problem. There are two strategies that are commonly employed to reduce static power dissipation.

One strategy involves the use of a high threshold sleep transistor, typically an NMOS transistor, in series with low threshold transistor stacks. During normal operation or active state of the circuit the sleep transistor is turned "on" so that it presents a path to GND with low resistance. When the circuit is in standby mode the sleep transistor is turned "off" thereby increasing the path resistance to GND; this considerably limits the magnitude of the leakage current [1]. Unfortunately there are several design issues that crop up in using this strategy for complex circuits, [4], [5], that complicate the development of automation tools for the synthesis of multithreshold CMOS VLSI. Yet another strategy involves using low threshold transistors for gates on the critical path and high threshold transistors for gates off the critical path [3], [6]. This way significant savings of leakage power can be obtained as large number of gates off the critical path now exhibit substantially less leakage power. The best strategy might be to combine the two approaches by switching as many gates as possible to a high threshold and using low threshold transistors and high threshold sleep transistors for the remaining gates. This way the number of gates that use a sleep transistor can be reduced considerably making the job of design automation tools, [4], easier. For practical reasons, in gate level circuits it is currently feasible to have only two different threshold voltages, one high V_t^H and the other low V_t^L. Having selected these two threshold voltages, beginning with a circuit with all gates at the low threshold voltage V_t^L, not all gates off the critical path can be switched to the high threshold voltage, V_t^H. Only some gates have sufficient slack to be switched to V_t^H. Of these only a subset of gates can be switched to V_t^H as switching a gate to V_t^H increases the delay of that gate and affects the slack of other gates in the circuit. In [6] a heuristic algorithm was presented which, for a given circuit, a fixed supply voltage (1V) and a fixed V_t^L (0.2V) for gates on the critical path, finds a static power "optimum" V_t^H, and a subset of gates off the critical path which can be switched to V_t^H. This method demonstrated significant savings in leakage power for the optimized circuits. However, this technique employed simple backward breadth first search heuristics to identify the subset of gates that can be switched to V_t^H. Therefore, it is reasonable to assume that better heuristics and/or formal methods can be developed for this problem that exhibit substantially higher power savings.

In this paper we address the same problem as [6] but present a near optimal strategy for the solution. Also, presented is an exact algorithm to determine a power optimal configuration of the circuit when there is no limit on the number of distinct threshold voltages. In other words every gate can have a possibly different threshold voltage. While this technique may not be practical for current fabrication processes it helps to establish an upper bound on the power savings obtainable using multiple threshold voltages. The problem formulation followed here has lot of similarities with retiming, [7], which nevertheless is an optimization technique for synchronous circuitry based on relocation of registers. The problem formulation also borrows from the ideas in [8] which presents a relatively new strategy for low power gate resizing.

2. BACKGROUND INFORMATION

2.1. Delay Model

In [6] a model is developed for estimating the delay of a CMOS VLSI gate based on the commonly used Elmore delay model [9]. We use a simpler delay model but any complicated delay model can be used in conjunction with our proposed technique. The delay of a gate is assumed to have the following behavior:

$$Delay = \frac{2C_{Load}V_{DD}}{\beta(V_{DD} - V_T)^{\alpha}}. \quad (1)$$

The value of $\alpha \approx 1.3$ for short channel devices and ≈ 2 for long channel devices.

*This research has been supported by the Army Research Office under grant number DA/DAAG55-98-1-0315.

2.2. Subthreshold Leakage Power Estimation

Assuming the BSIM2 model [10] the subthreshold current of a MOS transistor is approximated as:

$$I_{sub} = Ae^{\frac{q}{n^{'}kT}(V_G - V_S - V_{TH0} - \gamma^{'}V_S + \eta V_{DS})}(1 - e^{-\frac{qV_{DS}}{kT}}), \quad (2)$$

where A = $\mu_0 C_{ox}\frac{W_{eff}}{L_{eff}}(\frac{kT}{q})^2 e^{1.8}$, C_{ox} is the gate oxide capacitance per unit area, μ_0 is the zero bias mobility of the carrier, $n^{'}$ is the subthreshold swing coefficient of the transistor. Also, V_{TH0} is the zero bias threshold voltage, the body effect is represented by the linearized coefficient $\gamma^{'}$, and η is the drain induced barrier lowering (DIBL) coefficient.

The standby leakage power of a logic circuit can be expressed as follows [11]

$$P_{stdby} = (\sum_i I_{stdby_i})V_{DD}, \quad (3)$$

where I_{stdby}, is the standby leakage current through each node i. It depends on the gate topology as well as the input signal levels. Note that the standby leakage current refers to the leakage current that flows after all charge stored in "isolated" internal nodes has been discharged and therefore the magnitude of this current can be determined completely by the input signal levels. An internal node is isolated if there is no path to V_{DD}/GND from this node which passes through only "on" transistors.

Let us consider a static CMOS logic gate and assume that the transistors that are turned "on" are short circuits. Under this scenario if the output is a "1" then the leakage current is determined by the NMOS transistors in the "pulldown" network that are "off". On the other hand if the output is a "0" then the leakage current is determined by the PMOS transistors in the "pullup" network that are "off". For a CMOS NAND gate with output "1" suppose that there are n NMOS transistors in the pulldown network that are "off". The quiescent subthreshold current through each of them must be identical. By equating the subthreshold current through each of the transistors given by (2) and using the fact that the net voltage drop across the pulldown network is V_{DD} we get a nonlinear system of equations. In [6] an approximate closed form solution is derived for this system of equations, which can however be solved with a greater degree of accuracy by using the nonlinear solver in MATLAB and we followed this approach. The leakage current through any logic gate can similarly be computed for all possible combinations of signal levels at the input to the logic gate.

The average leakage power of a circuit can be evaluated with random patterns applied to the primary inputs and averaging the sum total of leakage power for all the gates in the circuit.

3. NOTATION

A gate-level combinational circuit can be represented by a directed acyclic graph, G = (V, E), referred as a circuit-graph. Every circuit-graph node $u \in$ V represents a logic gate or a primary input and every directed edge $e_{uv} \in$ E denotes the wire that connects the output of gate u to the input of gate v. The *fanin* of a logic gate refers to the number of gates whose output has a wired connection to the input of the given gate. The *fanout* of the gate is the total number of gates which take the output of the given gate as an input. The input wires to gates with a fanin of 0 are fed externally and these external input junctions constitute the primary inputs, PI. The PIs are also modeled as nodes $\in$ V. The gates with a fanout of 0 constitute the primary outputs, PO. The delay of a node/gate u is denoted by $delay(u)$. The maximum propagation time for a signal through the circuit-graph, including input arrival times and the path gate delays, for any path from a primary input node to a primary output node constitutes the *critical path*, $CP(G)$, of the circuit-graph. We now define three attributes for every node in G. For a node u these are namely, the arrival time $AT(u)$, the required time $RT(u)$ and the slack, $sl(u)$. Additionally, every wire $e_{uv} \in$ E has the attribute *edge-slack*, $esl(e_{uv})$. We will now define all the attributes mentioned previously.

$$\begin{cases} AT(u) = external\ time\ of\ arrival,\ u \in PI, \\ else, \\ AT(u) = max_{v \in fanin(u)}(AT(v) + delay(v)), \end{cases}$$
$$\{\ CP(G) = max_{u \in V}(AT(u) + delay(u)),$$
$$\begin{cases} RT(u) = CP(G) - delay(u),\ u \in PO, \\ else, \\ RT(u) = min_{v \in fanout(u)}RT(v) - delay(u), \end{cases}$$
$$\begin{cases} sl(u) = RT(u) - AT(u), \\ esl(e_{uv}) = RT(v) - AT(u) - delay(u). \end{cases} \quad (4)$$

We call a circuit safe when all nodes $u \in$ V have $sl(u) \geq 0$ and all wires have $esl(e_{uv}) \geq 0$.

3.1. Delay Balancing

A given circuit-graph G can be transformed to a functionally equivalent circuit-graph G' by introducing buffers of appropriate delay into the circuit in such a manner that for every $e_{uv} \in$ E $esl(e_{uv}) = 0$ and $CP(G') = CP(G)$ this process is known as delay balancing. Delay balancing with real buffers can lead to a glitch free circuit. However, the dynamic power dissipated by these buffers may offset the benefits of glitch power reduction. For our purposes we use delay balancing as a tool to capture all the slack in the circuit. The delay buffers we use are fictitious entities whose sole purpose is to model the slack present in the circuit. We refer to these fictitious buffers as SDFs (Specific Delay Fictitious-Buffers). Fig. 1 shows a gate level circuit and Fig. 2 shows its delay balanced counterpart; the "boxed" numbers in the wires of the circuit in Fig. 2 represent the delay of the SDF on that wire.

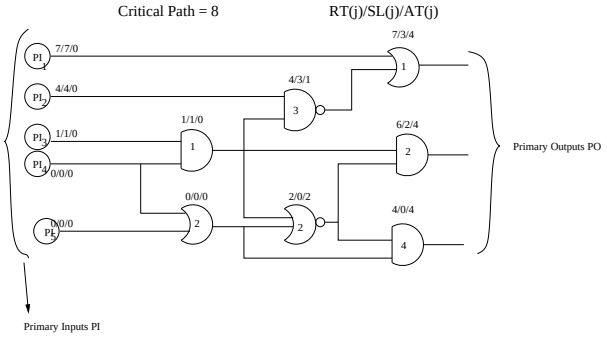

Figure 1. An example of a circuit-graph the integer numbers within each node/gate represent its delay and each gate u has the triplet (RT/SL/AT) above it.

Starting with a given circuit-graph there are several possible ways to produce a delay balanced graph. Any such delay balanced graph will from now on be referred to as a delay balanced configuration.

3.2. SDF-Displacement

We define *SDF-Displacement,* a circuit-graph transformation technique, as a mapping r:V$\longrightarrow$R, {R: the set of real numbers}; such that the delay of the SDF in the wire e_{uv}, $SDF^r(e_{uv})$, after *SDF-Displacement* is related to the delay of the SDF before *SDF-Displacement*, $SDF(e_{uv})$, by,

$$SDF^r(e_{uv}) = SDF(e_{uv}) + r(v) - r(u). \quad (5)$$

A *SDF-Displacement* is legal if and only if $SDF^r(e_{uv}) \geq 0$ for all wires $e_{uv} \in$ E.

We state the following without proof.

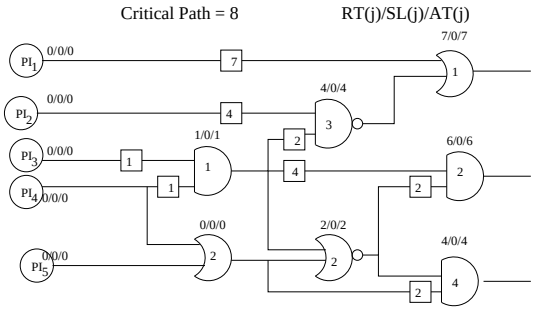

Figure 2. The circuit in Fig. 1 after delay balancing. The boxed integers on wires represent the delay of the SDFs added to the wires for delay balancing.

Theorem 1 *All legal delay balanced configurations for a given circuit-graph G are SDF-Displaced versions of each other.*

Theorem 2 *The net change in delay of SDFs in any structural path from a node/gate u to another node/gate v after* SDF-Displacement *is always $r(v) - r(u)$.*

The above theorem gives rise to the following corollary.

Corollary 1 *If we connect all the gates ∈ PO (primary outputs) of a given combinational circuit to a common dummy node O through dummy wires then, if we restrict r(O) to be exactly 0 and also r(I) for every input node I ∈ PI to be exactly 0, then the critical path of the transformed circuit-graph after SDF-displacement remains unaltered.*

Theorem 2 can be proved by considering any path $u \rightarrow v$ in the circuit-graph, breaking it down to its constituent wires and then using (5) on the individual wires to find the net change in delay of the SDF for each wire. The net change in delay of SDFs for the given path is then obtained as the sum of the change in SDF delays for these wires. Corollary 1 follows from theorem 2 as the delay of SDFs between any node I ∈ PI and the dummy node O remains unaltered as $r(O) = 0, r(I) = 0 => r(O) - r(I) = 0$. Hence the *critical path* CP(G) which depends on the maximum delay through such a path remains unaltered.

4. THRESHOLD VOLTAGE ALLOCATION FOR LOW POWER

The delay of a logic gate increases with its threshold voltage. The static power dissipated by the logic gate, on the other hand, decreases with an increase in the threshold voltage. The static power dissipated by the logic gate can also be expressed as a function of its delay provided the geometry of the logic gate, i.e. its $\frac{W}{L}$ ratio, and that of other gates in its neighborhood are unchanged. In addition we assume that all other variables like the supply voltage etc. are also unchanged. It turns out that under these conditions the static power dissipated by a CMOS gate can be approximated with a high degree of accuracy with a (generally non-differentiable) convex function of its delay, as shown in Fig. 3. We will now present a technique which uses *SDF-displacement* to exploit this convex relationship to model threshold voltage allocation for low power.

4.1. SDF-displacement for Threshold Allocation

Under the conditions of invariant gate and neighborhood geometry and the invariance of other variables like the supply voltage, the delay of the gate can be expressed as a function of its threshold voltage and conversely the threshold voltage of the gate can be expressed as a function of its delay. For a gate satisfying the conditions stated previously, let the threshold voltage V_{t_g} of gate g be expressed as a function $V_{t_g}(delay_g)$ of its delay $delay_g$. Note that the nature of this functional dependence varies from gate to gate depending among other things on the gate geometry its supply voltage and the geometry of its neighborhood. Once the supply voltage and geometry of the circuit are fixed governed largely by circuit performance/power consumption in active mode, the function $V_{t_g}(delay_g)$ can be computed for each gate. Also, for every gate we can determine the average leakage power dissipated by that gate g, $pow_g(delay_g)$, as a decreasing convex function of its delay $delay_g$.

Beginning with a circuit where all gates are at a low threshold voltage V_t^L, we can use any delay balanced configuration to switch some gates to higher threshold voltages. This can be done in the following manner:

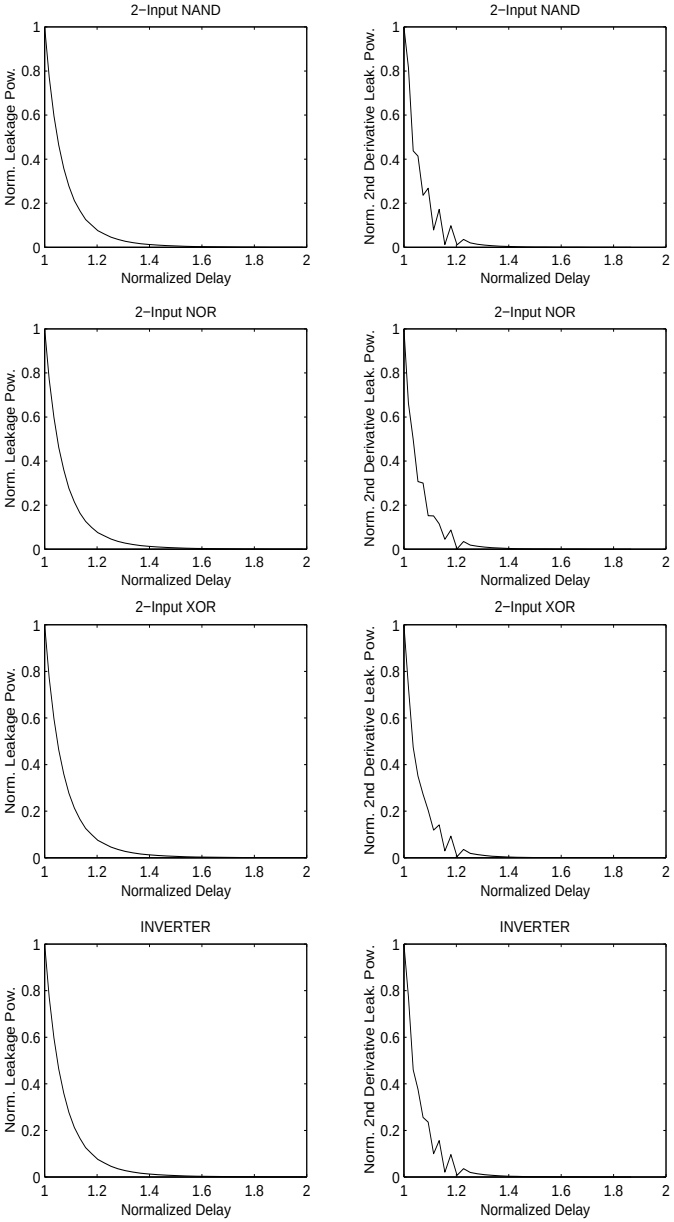

Figure 3. Demonstration of the Convexity of the functional dependence of leakage power on gate delay. The non-negative nature of the 2nd derivative demonstrates convexity. The results shown here were obtained from representative gates by simulation for the ISCAS85 benchmark circuit c432.

0)assign an attribute $Assigned(u)$ to all gates u in the circuit, set $Assigned(u) = 0$ for all the gates.
1)Starting from the POs, do a breadth-first traversal of the gates in the circuit.
2)For the current gate, u, if $Assigned(u) = 0$, identify a new threshold voltage as $V_{t_u}(delay_u + min_{v \in Fanout(u)} SDF(e_{uv}))$. If this threshold voltage is greater than $0.5V_{DD}$ then fix the threshold voltage at $0.5V_{DD}$ set $Assigned(u) = 1$. Move on to the next gate.
3)Once all the gates are visited if possible displace excess SDF delays in the fanout wires of gates u with $Assigned(u) = 1$ to gates v with $Assigned(v) = 0$. If no such displacement is possible terminate the process and output a new threshold voltage for each gate. Otherwise repeat 1-3 *till the process terminates.*

The fact that *SDF-displacement* can generate all possible delay balanced configurations of a circuit-graph leads to the following.

Objective:To employ SDF-displacement to identify that particular delay balanced configuration which identifies a new threshold voltage for all the logic gates while providing a maximum reduction in leakage power in standby mode while also maintaining the critical path in active mode.

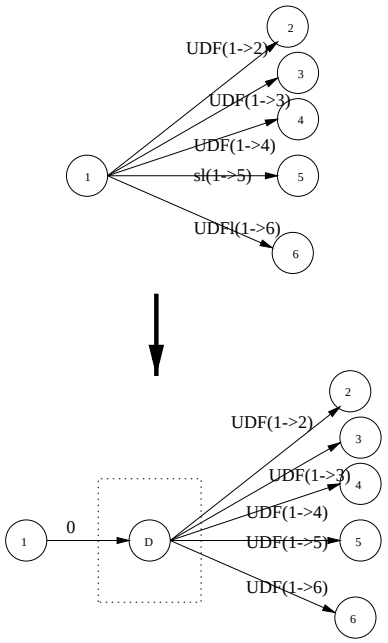

Figure 4. The artifice used to model fanout nodes. Node 1 is a fanout node and node D is the dummy node corresponding to it.

Before we develop a formal mathematical model, we first modify the circuit-graph by adding a $dummy$ node $dummy_u$ of delay 0 units for every fanout node u in the circuit-graph. A $dummy$ wire connects fanout node u to its dummy node. All fanout arcs which originated from node u now originate from node $dummy_u$. Fig. 4 illustrates this circuit-graph transformation with an example. Note that there is a 1-1 correspondence between fanout wires in the original circuit-graph and dummy fanout wires in the modified circuit-graph. We will now present a formal way to address achievement of our previously stated objective.

Power-Optimal-Threshold-Voltage-Assignment
1)Levelize the circuit with PIs at level "0" and recursively calculate levels for all gates u as $level(u) = 1 + max_{v \in Fanin(u)} level(v)$.

2)Generate a delay balanced configuration.
i) We use a backward breadth-first SDF delay assignment heuristic for this. We begin with the POs and visit every node. While at a node we change its delay to $delay_{new}(u) = delay_{old}(u) + slack(u)$. We then recompute the arrival time, required time and slack for every gate and the edge slack for every wire in the circuit.
ii)Now compute $delay_{new}(u) - delay_{old}(u)$ for each gate u and this gives the value for $SDF(u \rightarrow dummy_u)$. For fanout nodes v of u the value of $SDF(dummy_u \rightarrow v)$ is given by the value of $esl(dummy_u \rightarrow v)$ at the end of i). Now reset all the gate delays to $delay_{old}(u)$.

3)Now starting from the delay balanced configuration in 2) solve the following convex minimization problem,

minimize $\sum_{u \in V} p_u(d_u + SDF^r(u \rightarrow dummy_u))$
subject to:

$$\begin{aligned} SDF^r(u \rightarrow dummy_u) &= SDF(u \rightarrow dummy_u) + \\ & r(dummy_u) - r(u) \geq 0, u \in V, \\ SDF^r(u \rightarrow dummy_u) &= SDF(u \rightarrow dummy_u) + \\ r(dummy_u) - r(u) &\leq MAXDEL(u) - delay(u), \\ & u \in V, \\ SDF^r(dummy_u \rightarrow v) &= SDF(dummy_u \rightarrow v) + \\ & r(v) - r(dummy_u) \geq 0, v \in Fanout(u), \end{aligned} \quad (6)$$

where MAXDEL(u) is the delay of gate u when its threshold voltage is $0.5V_{DD}$.

We will refer to the above technique as PRACTIC for Power Reducing Allocation of Threshold Voltages for Integrated Circuits. The above problem is modeled as a convex minimization problem with linear constraints. Fast polynomial time interior-point solutions to this problem are possible that can obtain a solution arbitrarily close to the optimal solution [12]. The constraints governing the above minimization ensure that after SDF-displacement each wire in the circuit has a non-negative delay value for the SDFs on that wire. In addition for every logic gate (dummy gates are excluded) the maximum delay of the SDF on the wire that connects it to its dummy gate is the amount by which the gate gets delayed if its threshold is kept at $0.5V_{DD}$. Also, it is possible to linearize such a problem and solve it with arbitrary accuracy within a Linear Programming LP framework. This can be done by approximating the convex objective function with a piecewise linear convex objective function [13]. In fact the nature of the constraints, difference constraints that have variables with 1, -1 as coefficients makes it possible to model the problem as the dual of a minimum cost network flow problem, [14]. This makes it possible to solve the problem with extremely fast run-times, [14]. We followed this approach with three piecewise linear segments for the objective function. In this way we are able to determine an optimum threshold voltage within technology limits for every gate in a circuit so that the circuit as a whole dissipates minimal leakage power in standby mode.

Unfortunately, the above technique may not lead to practical solutions for gate level circuits for current process constraints. This is because current process constraints limit the number of distinct threshold voltages to a small value typically two. Given this restriction we need to modify the optimization framework developed so far.

We will first assume that we are given the two threshold voltages V_t^L and V_t^H and we are required to allocate a subset of gates to V_t^H in such a manner that the leakage power is minimized in standby mode. This problem is a special (NP-hard) case of the *basic circuit implementation problem* [15]. Due to the NP hard nature of this problem the best we can do is build near optimal heuristics to solve this problem. We will first present an exact model to this problem using *integer linear programming* ILP techniques. We will then provide a heuristic solution for this exact ILP model.

Let us assume that the power dissipated by gate u when its threshold voltage is V_t^L (V_t^H) is P_u^L (P_u^H). Note that $P_u^L > P_u^H$. Starting with a circuit where all gates have threshold voltage V_t^L do the following:

High-V_{th}-Assignment-Exact
1) Do 1) and 2) from PRACTIC.
2) Solve the following optimization problem:

$$minimize \sum_u (high(u)P_u^H + (1 - high(u))P_u^L)$$

subject to:

$$\begin{aligned} SDF(u \rightarrow dummy_u) + r(dummy_u) - r(u) \geq high(u), \\ SDF(u \rightarrow dummy_u) + r(dummy_u) - r(u) \leq (delay_{V_t^H} - delay_{V_t^L})high(u), \\ SDF(dummy_u \rightarrow v) + r(v) - r(dummy_u) \geq 0, \\ u \in V, v \in Fanout(u), high(u) \in \{0,1\}, u \in V. \end{aligned} \quad (7)$$

Since the above problem is an ILP problem solving it exactly may take inordinate amount of CPU time for large circuits. We therefore present a practical heuristic which can be performed in polynomial time while taking reasonable amount of CPU time even for large circuits.

We will now present our final heuristic for allocating threshold voltages to gates for minimizing leakage power. Start with all the gates at threshold voltage V_t^L.

High-V_{th}-Assignment-Heuristic
0)Associate an attribute $Alloc(u)$ with every gate u.
1)Set $Alloc(u) = 1$ for gates u which satisfy $slack(u) > delay_{V_t^H}(u) - delay_{V_t^L}(u)$ for other gates set $Alloc(u) = 0$.
2)Set up the ILP problem mentioned earlier. From the ILP remove the constraints for gates which have $Alloc(u) = 0$ for such gates set up the following constraints:

$$\begin{aligned} SDF(u \rightarrow dummy_u) + r(dummy_u) - r(u) = 0, \\ SDF(dummy_u \rightarrow v) + r(v) - r(dummy_u) \geq 0, \\ u \in V, v \in Fanout(u). \end{aligned} \quad (8)$$

Now solve this problem as a linear programming problem (LP) with the relaxation $0 \leq high(u) \leq 1, u \in V$.
3)For all gates u with $high(u) = 1$ set $Alloc(u) = 1$ and set $delay(u) = delay_{V_t^H}(u)$, now recompute the arrival time, required time and slack for each gate and the edge slack for each wire. Now iterate 1)-3) till all gates u have $Alloc(u) = 1$.

We will now give the final heuristic for determination of an optimal V_t^H.

Optimal-High-Vth (){ i = 1 and $V_t^H(i) = V_t^L + \Delta V_t$
while ($V_t^H(i) < 0.5V_{DD}$){
High-Vth-Assignment-Heuristic($V_t^H(i)$)
Estimate standby leakage power P_{stdby}
if($P_{stdby_{min}} < P_{stdby}$){
$P_{stdby_{min}} = P_{stdby}$
opt_Vth = $V_t^H(i)$
++i and $V_t^H(i) = V_t^L + i \times \Delta V_t$
}
}
}

In this way we are able to find a near-optimal high threshold voltage V_t^H for minimizing leakage power of slack-gates in the given circuit. As this technique is a practical application of the basic idea in PRACTIC we will refer to it as PRACTICAL. In the next section we will present simulation results on ISCAS85 benchmark circuits for both PRACTIC and PRACTICAL.

5. SIMULATION RESULTS

In order to prove the effectiveness of our approach we compare it with the approach presented in [6]. For this both the techniques were used under identical conditions. For the purpose of leakage power estimation, like in [6] our simulations are based on a 0.5μ process. The effective channel length is 0.32μ and the gate oxide thickness used is $9.8nm$. All PMOS (NMOS) transistors are assumed to be of identical size, but in theory any transistor size profile can be employed. The effective channel widths for PMOS and NMOS are assumed to be 10.5μ and 3μ, respectively. The circuit temperature is assumed to be $25°C$. The supply voltage is assumed to be 1V and V_t^L is fixed at $0.2V$. The search for the optimum V_t^H is therefore between $0.2V$ and $0.5V$. We use a step-size $\Delta V = 0.01V$ in order to get accurate results. For simulation purposes we also assume a unit delay model for every gate and assume that all gates drive a unit capacitance, in general any complicated delay and capacitance model can be used as these models are used only to extract values of the parameter used in the optimization process. The variation of the delay of a gate with its threshold voltage is governed by (1). For our simulation we use $\alpha = 1.3$ in (1). As our results are comparative in nature and it is easy to build leakage power models for any kind of complex gate, the simulations were done on the ISCAS85 benchmark netlists without any modifications, i.e., we did not perform explicit technology mapping to a gate-limited library. In general our techniques can be used in conjunction with any technology mapping environment, e.g., the Berkeley SIS environment. We assume the critical path of a circuit to be the delay of the longest structural path in that circuit. This critical path delay is then used to compute the slack on every net and on every gate.

Table 1 compares the power consumption reduction obtained by using two threshold voltages, as opposed to a single threshold voltage, based on the proposed technique, PRACTICAL, and the approach presented in [6]. Since our simulations use simplistic values only relative improvements of power with respect to the original single V_{th} circuit are tabulated. As can be seen PRACTICAL improves upon the power savings exhibited by the method in [6] anywhere between 0%-29.45%. Also, we can observe that the best possible improvement using arbitrary number of threshold voltages obtained by PRACTIC and shown in last column of Table 1 is usually no more than 5% better than the savings due to PRACTICAL except in c6288 where the improvement is 10.34%.

6. CONCLUSIONS

A new formal heuristic was developed for the synthesis of low static-power CMOS VLSI with dual threshold voltages. Also, developed was a technique that can synthesize static-power optimal CMOS VLSI circuits when arbitrary number of threshold voltages are allowed. While the latter technique is impractical for the state of the art of fabrication processes, it serves as an upper bound

Table 1. The power benefits for buffer-redistribution for gate resizing with ISCAS85 benchmark circuits.

Circuit	# gates	% Pow. Impr. [6]	% Pow. Impr. PRACTICAL	% Pow. Impr. Upper Bound (PRACTIC)
c432	160	27.86%	28.83%	28.83%
c499	202	22.96%	22.96%	22.96%
c880	383	68.96%	82.67%	83.81%
c1355	546	21.50%	21.50%	21.50%
c1908	880	64.86%	84.92%	88.33%
c2670	1211	60.80%	90.25%	92.70%
c3540	1705	65.33%	83.36%	88.57%
c5315	2351	88.28%	91.56%	94.28%
c6288	2416	32.48%	61.75%	72.09%
c7552	3624	84.63%	90.90%	94.19%

for the maximal power benefits that can be obtained with multiple threshold voltages.

REFERENCES

[1] S. Mutoh and etal., "1-V Power Supply High-Speed Digital Circuits Technology with Multithreshold-voltage CMOS," *IEEE Journal of Solid State Circuits*, vol. 30, pp. 847–854, Aug. 1995.

[2] A. P. Chandrakasan and R. W. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits," *Proceedings of the IEEE*, vol. 83, pp. 498–523, April. 1995.

[3] Z. Chen and etal., "0.18μ Dual V_t MOSFET Process and Energy-Delay Measurement," *IEDM Digest*, pp. 851–854, 1996.

[4] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS Hierachical Sizing Based on Mutual-Exclusive Discharge Patterns," *Proc. 35th ACM/IEEE DesignAutomation Conf.*, pp. 495–500, 1998.

[5] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor Sizing Issues and Tool for Multi-Threshold CMOS Technology," *Proc. 34th ACM/IEEE DesignAutomation Conf.*, pp. 409–414, 1997.

[6] L. Wei, Z. Chen, M. Johnson, and K. Roy, "Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits," *Proc. 35th ACM/IEEE DesignAutomation Conf.*, pp. 489–492, 1998.

[7] C. E. Leiserson and J. B. Saxe, "Retiming Synchronous Circuitry," *Algorithmica*, vol. 6, no. 1, pp. 5–35, 1991.

[8] V. Sundararajan and K. K. Parhi, "Low power gate resizing using buffer-redistribution," in *Proceedings of the Twentieth Anniversary Conference on Advanced Research in VLSI*, (Georgia Institute of Technology), Mar. 1999.

[9] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective 2nd Edition*. Addison-Wesley, 1993.

[10] J. Sheu and etal., "BSIM: Berkeley Short-Channel IGFET Model for MOS Transistors," *IEEE Journal of Solid-State Circuits*, vol. 22, pp. 558–566, August 1987.

[11] J. M. Rabaey, *Digital Integrated Circuits*. Ne Jersey: Prentice-Hall, 1996.

[12] Y. Nesterov and A. S. Nemirovskii, *Interior-point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.

[13] V. Chvatal, *Linear Programming*. W. H. Freeman and Company, 1983.

[14] Y. Pinto and R. Shamir, "Efficient Algorithms for Minimum-Cost Flow Problems with Piecewise-Linear Convex Costs," *Algorithmica*, vol. 11, pp. 256–277, 1994.

[15] W. Li, A. Lim, P. Agrawal, and S. Sahni, "On the circuit implementation problem," *IEEE Transactions on Computer-Aided Design of Integrated Circ uits and Systems*, vol. 12, pp. 1147–1156, August 1993.