Algorithm for Achieving Minimum Energy Consumption in CMOS Circuits Using Multiple Supply and Threshold Voltages at the Module Level

Yuvraj Singh Dhillon , Abdulkadir Utku Diril, Abhijit Chatterjee, and Hsien-Hsin Sean Lee
School of Electrical and Computer Engineering, Georgia Institute of Technology,
Atlanta, GA 30332 USA

Abstract

This paper proposes an optimum methodology for assigning supply and threshold voltages to modules in a CMOS circuit such that the overall energy consumption is minimized for a given delay constraint. The modules of the circuit should have large enough gate depths such that the delay and energy penalties of the level shifters connecting them are negligible. Both static and dynamic energy are considered in the optimization. Energy savings of up to 48% have been achieved on various example circuits. The first step in the optimization finds optimum supply and threshold voltages for each module in the circuit. If the circuit has a large number of modules, this step might yield a correspondingly large number of different supply and threshold voltages for minimum energy consumption. Since having a large number of different supply and threshold voltages on an IC is not feasible in current technologies, an additional step clusters the multiple voltages obtained from the first step into a fixed number of supply and threshold voltages (for example, 2 different supply voltages and 2 different threshold voltages). In addition to the application of this method to circuit optimization, it can also be applied to a wide range of problems with delay constraints, such as software tasks running on a dynamically variable V_{DD} and V_{th} processor.

1. Introduction

Energy consumption is recognized as one of the most important parameters in designing modern portable electronic systems. Dynamic energy has been the main component of total energy since it is proportional to the square of V_{DD}. However, with the shrinking of device sizes and reduction of supply voltages, static energy has become as important as dynamic energy. To obtain high gate overdrive (V_{DD} - V_{th}) for high speeds of operation, V_{th} is also decreased as V_{DD} is decreased. The decrease in threshold voltage increases the leakage current exponentially, which makes static energy consumption more significant in every new technology generation. Therefore, it has become essential to consider both supply and threshold voltage in any circuit optimization for low-energy consumption.

There has been significant research in the usage of dynamic supply voltage scaling [3, 4]; static assignment of different supply voltages to a system [1, 5, 6]; static multi-threshold voltage systems [7, 8, 16]; and dynamic threshold voltage scaling [9] for energy minimization. The optimization procedure that we propose in this paper can be used in all of the above scenarios. It can be used to determine the optimal supply and threshold voltages for tasks executing on a variable voltage processor; or it can be used to statically set supply and threshold voltages for a system, given the probable switching activity and timing requirements. The benefit of our method is that it considers both supply and threshold voltages *simultaneously*, an idea which is gaining importance as a means of saving energy without sacrificing performance [15]. Another important contribution of this work is that it gives a metric which can be used by circuit designers to test how close the energy consumption of their design is to the minimum possible. We use this metric to determine the stopping conditions of our optimization algorithm. If an unlimited number of supply and threshold voltages are available, the proposed algorithm *is optimum* in the sense that *no other voltage assignment for the given modules will give lower energy consumption for the given delay constraint*.

The complete procedure has two steps. The first step finds optimum supply and threshold voltage values for CMOS modules in a digital circuit that minimizes the total energy consumption. Considering a circuit as composed of modules allows energy optimization of much larger circuits than is possible with gate-level optimization algorithms. This is due to the significant reduction of problem complexity. We find the *exact* conditions for minimum energy using the Lagrange Multiplier Method. Then we iteratively find the supply and threshold voltage values for each module that satisfy the minimum energy condition. This step of the algorithm converges to an *exact solution in a small number of iterations* for a large and varied set of problems. If it is technologically feasible to assign the optimum (and perhaps all different) supply and threshold voltages to all the modules, then we stop here. Otherwise we continue to the next step.

The second step clusters the multiple voltages obtained from the first step into a fixed number of supply and threshold voltages (for example, 2 different supply voltages and 2 different threshold voltages). This step results in a feasible implementation of the system in current technologies. The organization of the paper is as follows. In Section 2, we provide the theoretical background of the paper. In Section 3, we explain the Lagrange Multiplier based optimization procedure. In Sections 4 and 5, we explain the algorithms for the first and second steps of the procedure. Section 6 shows experimental results. Finally, Section 7 concludes.

2. Theoretical background

In order to formulate the optimization, we need equations for delay, dynamic energy, and static energy in terms of V_{DD} and V_{th}.

For a CMOS circuit, delay can be approximated as being proportional to $V_{DD}/(V_{DD} - V_{th})^{\alpha}$ [2].

$$d_i = \frac{k_{0i} \cdot V_{DDi}}{(V_{DDi} - V_{thi})^{\alpha}} \quad (1)$$

Here d_i is the delay of the i^{th} module, k_{0i} is a constant for that module, V_{DDi} and V_{thi} are the supply voltage and threshold

This research was supported by NSF Information Technology Research Contract, CCR 022-0259.

voltage, respectively applied to that module, and α is the velocity saturation coefficient. For current technologies, α is between 1.2 and 1.5. The delay constant (k_{0i}) includes the effects of process, device sizes, load capacitance, and gate depth in that module. Gate depth can be included in the constant because of the additive characteristics of delays.

Having formed an equation for module delay, we now form an equation for the dynamic energy consumption in terms of supply voltage and threshold voltage. We can write the dynamic energy consumed in a module as:

$$E_{di} = 0.5 \cdot C_i \cdot V_{DDi}^2 \quad (2)$$

Here C_i is the term for all the capacitances that are switched during operation of the i^{th} module including possible multiple switching of some nodes. To simplify the derivation, we rewrite dynamic energy as:

$$E_{di} = k_{1i} \cdot V_{DDi}^2 \quad (3)$$

where k_{1i} stands for the circuit, process, and application dependent terms including switching activity. An average value for total switching activity in the module can be found by running several different tasks on the module and averaging the switching activity results. Short circuit power dissipation can also be included in k_{1i} because of the quadratic dependence of short-circuit power dissipation to V_{DD} [11].

For static energy consumption, we use a generalized model.

$$E_{si} = E_{subi} + E_{gatei} = k_{2i} \cdot V_{DDi} \cdot e^{k_{3i} \cdot V_{DDi} - k_{4i} \cdot V_{thi}} \cdot T_i + k_{5i} \cdot V_{DDi} \cdot e^{k_{6i} \cdot V_{DDi} - k_{7i} \cdot V_{thi}} \cdot T_i \quad (4)$$

E_{subi} stands for the sub-threshold leakage component of the static energy consumption. This component is strongly influenced by the threshold voltage. E_{gatei} stands for the gate leakage component of the static energy. This component is much smaller than E_{subi} when V_{th} is small. When V_{th} is large, the main contribution to the static energy comes from this component. Also, gate leakage increases as the gate oxide thickness becomes smaller. T_i is the period for which the circuit is idle. k_{2i} and k_{5i} are circuit-dependent parameters. k_{3i}, k_{4i}, k_{6i}, and k_{7i} are process-dependent parameters. The values of the process-dependent parameters can be found by fitting SPICE simulation results of a simple gate to Equation 4. The values for these parameters can be used for any circuit designed in the same technology.

Given these approximate models for delay and energy in terms of supply and threshold voltages, we state the energy-optimization problem for a digital system (assumed given to us) consisting of N modules and P paths from primary inputs to primary outputs as follows:

Minimize $\sum_{i=1}^{N} E_i$ under the constraints $\sum_{i \in P_j} d_i \leq T_d$ for all paths P_j where $E_i = E_{di} + E_{si}$, T_d is the time constraint and the variables are V_{DDi} and V_{thi} for each module. Note that we obtain the time constraint, T_d, for the optimized circuit from the initial delays of the modules of the unoptimized circuit.

In the following sections, we will consistently use i to index the modules and j to index the paths. An example circuit for N=4 and P=2 is given in Figure 1. In the next section, we derive the conditions on V_{DDi} and V_{thi} for minimum energy consumption.

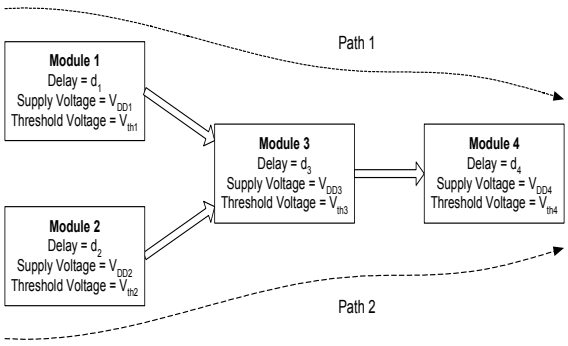


Figure 1. An example circuit with 4 modules and 2 paths (N=4, P=2).

3. Lagrange Multiplier based optimization

Consider a system of N modules and P paths from primary inputs to primary outputs. We form a binary matrix, **A**, of P rows and N columns as follows:

$$A_{ji} = 1 \; if \; Module \; i \; lies \; on \; Path \; j \\ = 0 \; otherwise \quad (5)$$

For example, the **A** matrix corresponding to the circuit in Figure 1 (N=4, P=2) is:

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}_{2 \; x \; 4} \quad (6)$$

If a path P_u is a subset of P_v (i.e. all modules on P_u also lie on P_v), then the row of **A** corresponding to P_u ($Row_u(\mathbf{A})$) is removed from **A** to reduce unnecessary computation. For the rest of the paper, we assume that the resulting **A** matrix has more columns than rows (i.e. N > P).

The total energy consumed by the system is given by:

$$E_{total} = (E_{s1} + E_{d1}) + (E_{s2} + E_{d2}) + \cdots + (E_{sN} + E_{dN}) \\ = E_1 + E_2 + \cdots + E_N \quad (7)$$

The initial delay of each path in the system is given by:

$$T_j = \sum_{i \in P_j} d_i \quad for \; all \; j \quad (8)$$

We can represent the above equation in vector form as follows:

$$\overline{T} = A \cdot \overline{d} \quad (9)$$

where $\overline{T} = [T_1 \; T_2 \; \ldots \; T_P]^T$ is the vector of path delays and $\overline{d} = [d_1 \; d_2 \; \ldots \; d_N]^T$ is the vector of module delays. The delay constraint is obtained from the initial delays of the modules as follows:

$$T_d = max(\overline{T}) \quad (10)$$

Following is a Lagrange Multiplier formulation with multiple constraints, where the function to minimize is total energy, the constraint for each path j is that its delay, T_j, should be less than T_d, and λ_j is the Lagrange Multiplier for the j^{th} path.

$$G(V_{DD1}, V_{th1}, \cdots, V_{DDN}, V_{thN}, \lambda_1, \cdots, \lambda_P) = \sum_{i=1}^{N} E_i - \sum_{j=1}^{P} \lambda_j \cdot [\sum_{i \in P_j} d_i - T_j] \quad (11)$$

Then, for minimum energy consumption, we have the following:

$$\frac{\partial G(V_{DD1},V_{th1},\cdots,V_{DDN},V_{thN},\lambda_1,\cdots,\lambda_P)}{\partial V_{DDi}} = 0 \quad for\ all\ i \tag{12}$$

$$\frac{\partial G(V_{DD1},V_{th1},\cdots,V_{DDN},V_{thN},\lambda_1,\cdots,\lambda_P)}{\partial V_{thi}} = 0 \quad for\ all\ i \tag{13}$$

Equations 12 and 13 become:

$$\frac{\partial E_i}{\partial V_{DDi}} = Row_i(A^T)\cdot\overline{\lambda}\cdot\frac{\partial d_i}{\partial V_{DDi}} \quad for\ all\ i \tag{14}$$

$$\frac{\partial E_i}{\partial V_{thi}} = Row_i(A^T)\cdot\overline{\lambda}\cdot\frac{\partial d_i}{\partial V_{thi}} \quad for\ all\ i \tag{15}$$

where $\overline{\lambda} = [\lambda_1\ \lambda_2 \ldots \lambda_P]^T$ and $Row_i(A^T)$ refers to the i^{th} row of A^T. We define two vectors, the Constant Threshold Energy Gradient Vector ($\overline{CTEG}$) and the Constant Supply Energy Gradient Vector ($\overline{CSEG}$), as follows:

$$\overline{CTEG} = \left[\frac{\partial E_1}{\partial V_{DD1}}\Big/\frac{\partial d_1}{\partial V_{DD1}}\quad \frac{\partial E_2}{\partial V_{DD2}}\Big/\frac{\partial d_2}{\partial V_{DD2}}\quad \cdots\quad \frac{\partial E_N}{\partial V_{DDN}}\Big/\frac{\partial d_N}{\partial V_{DDN}}\right]^T \tag{16}$$

$$\text{and } \overline{CSEG} = \left[\frac{\partial E_1}{\partial V_{th1}}\Big/\frac{\partial d_1}{\partial V_{th1}}\quad \frac{\partial E_2}{\partial V_{th2}}\Big/\frac{\partial d_2}{\partial V_{th2}}\quad \cdots\quad \frac{\partial E_N}{\partial V_{thN}}\Big/\frac{\partial d_N}{\partial V_{thN}}\right]^T \tag{17}$$

Following are the equations for the partial derivatives of the energy function, E_i. These equations are obtained using Equations 3 and 4. In these equations, we assume that each module is active for a small amount of time compared to the total deadline T_d. Then, we may write $T_i \approx T_d$ in Equation 4.

$$\begin{aligned}\frac{\partial E_i}{\partial V_{DDi}} &= 2\cdot k_{1i}\cdot V_{DDi} \\ &\quad + T_d\cdot V_{DDi}\cdot k_{2i}\cdot k_{3i}\cdot e^{(k_{3i}\cdot V_{DDi}-k_{4i}\cdot V_{thi})} \\ &\quad + T_d\cdot V_{DDi}\cdot k_{5i}\cdot k_{6i}\cdot e^{(k_{6i}\cdot V_{DDi}-k_{7i}\cdot V_{thi})} \\ &\quad + T_d\cdot\left(k_{2i}\cdot e^{(k_{3i}\cdot V_{DDi}-k_{4i}\cdot V_{thi})} + k_{5i}\cdot e^{(k_{6i}\cdot V_{DDi}-k_{7i}\cdot V_{thi})}\right) \\ &= \frac{2\cdot E_{di}}{V_{DDi}} + k_{3i}\cdot E_{subi} + k_{6i}\cdot E_{gatei} + \frac{E_{subi}+E_{gatei}}{V_{DDi}}\end{aligned} \tag{18}$$

$$\begin{aligned}\frac{\partial d_i}{\partial V_{DDi}} &= k_{0i}\cdot\frac{\left(V_{DDi}-V_{thi}\right)^{\alpha} - \alpha\cdot\left(V_{DDi}-V_{thi}\right)^{\alpha-1}\cdot V_{DDi}}{\left(V_{DDi}-V_{thi}\right)^{2\cdot\alpha}} \\ &= -\frac{d_i\cdot\left((\alpha-1)\cdot V_{DDi}+V_{thi}\right)}{V_{DDi}\cdot\left(V_{DDi}-V_{thi}\right)}\end{aligned} \tag{19}$$

$$\begin{aligned}\frac{\partial E_i}{\partial V_{thi}} &= -k_{2i}\cdot k_{4i}\cdot e^{(k_{3i}\cdot V_{DDi}-k_{4i}\cdot V_{thi})}\cdot T_d\cdot V_{DDi} \\ &\quad - k_{5i}\cdot k_{7i}\cdot e^{(k_{6i}\cdot V_{DDi}-k_{7i}\cdot V_{thi})}\cdot T_d\cdot V_{DDi} \\ &= -k_{4i}\cdot E_{subi} - k_{7i}\cdot E_{gatei}\end{aligned} \tag{20}$$

$$\frac{\partial d_i}{\partial V_{thi}} = \frac{\alpha\cdot k_{0i}\cdot V_{DDi}}{\left(V_{DDi}-V_{thi}\right)^{\alpha+1}} = \frac{\alpha\cdot d_i}{\left(V_{DDi}-V_{thi}\right)} \tag{21}$$

Finally, from Equations 18, 19, 20, and 21, we get:

$$\begin{aligned}CTEG_i &= \frac{2\cdot E_{di}+V_{DDi}\cdot\left(k_{3i}\cdot E_{subi}+k_{6i}\cdot E_{gatei}\right)+E_{subi}+E_{gatei}}{d_i\cdot\left((\alpha-1)V_{DDi}+V_{thi}\right)/\left(V_{DDi}-V_{thi}\right)} \\ CSEG_i &= \frac{k_{4i}\cdot E_{subi}+k_{7i}\cdot E_{gatei}}{\alpha\cdot d_i}\cdot\left(V_{DDi}-V_{thi}\right)\end{aligned} \tag{22}$$

Using Equations 16, 17, 22, Equations 14 and 15 can now be written concisely as follows:

$$\begin{aligned}&Minimum\ Energy\ Condition: \\ &\overline{CTEG} = \overline{CSEG} = A^T\cdot\overline{\lambda}\end{aligned} \tag{23}$$

We see that the initial energy optimization problem involving 2N variables (V_{DD} and V_{th} for each module) and a delay constraint has now been simplified to the form in Equation 23. We now need to solve N independent equations ($CTEG_i = CSEG_i = Row_i(A^T)\cdot\overline{\lambda}$) in 2 variables ($V_{DDi}$ and V_{thi}). However, doing this is not trivial since the Lagrange Multiplier Vector, $\overline{\lambda}$, is unknown. In the next section, we propose an iterative gradient search algorithm that yields a solution to this problem in a small number of iterations. After every iteration, the condition in Equation 23 will be used to check if minimum energy is achieved.

4. Gradient search algorithm for the optimization problem

We use an iterative algorithm to fulfill the conditions of Equation 23. The inputs to the algorithm are the initial parameters of all the N modules, such as the V_{DDi}s, the V_{thi}s, the module delays (d_is) and the circuit- and process-dependent parameters k_{0i}s, k_{1i}s, k_{2i}s, k_{3i}s, k_{4i}s, k_{5i}s, k_{6i}s and k_{7i}s.

To solve $CTEG_i = CSEG_i$ for the i^{th} module, we fix the delay, d_i, for that module. Then we can write V_{thi} in terms of V_{DDi} using Equation 1. This makes $CTEG_i$ and $CSEG_i$ functions of V_{DDi} only and the equation $CTEG_i = CSEG_i$ can be solved easily (we use MATLAB's FZERO function) to get V_{DDi} and V_{thi} values. Then with these V_{DDi} and V_{thi} values, we can find the energy consumption of that module (this will be the optimum energy consumption for that module, for the given delay, d_i). Hence, we can consider energy consumption of a module as a function of delay for that module. In vector form, we can write:

$$E_{total} = sum(\overline{E}) = E(\overline{d}) \tag{24}$$

where $\overline{E} = [E_1\ E_2 \ldots E_N]^T$.

First, we get intermediate values for module delays, $\overline{d}_{int}$, that make all path delays as close to T_d as possible. This step also makes sure that all modules have zero slack, so that we have an optimal starting point. A method similar to the Zero Slack Algorithm [13] is used in this step. Let $\overline{T}_{int} = A\cdot\overline{d}_{int}$ be the vector of path delays after this step.

Next, we minimize E_{total} by doing a gradient search on the delay vector, $\overline{d}$. But, the delay vector is constrained due to the path delay constraints ($A\cdot\overline{d} = \overline{T}_{int}$). So, in every iteration, we vary $\overline{d}$ by adding $\overline{\Delta}$ such that $A\cdot\overline{\Delta}=0$. This choice of $\overline{\Delta}$ satisfies the constraints as shown below:

$$A\cdot\overline{d} = A\cdot\left(\overline{d}_{int}+\overline{\Delta}\right) = A\cdot\overline{d}_{int} + A\cdot\overline{\Delta} = \overline{T}_{int} \tag{25}$$

In other words, $\overline{\Delta}$ has to lie in the nullspace of **A**.

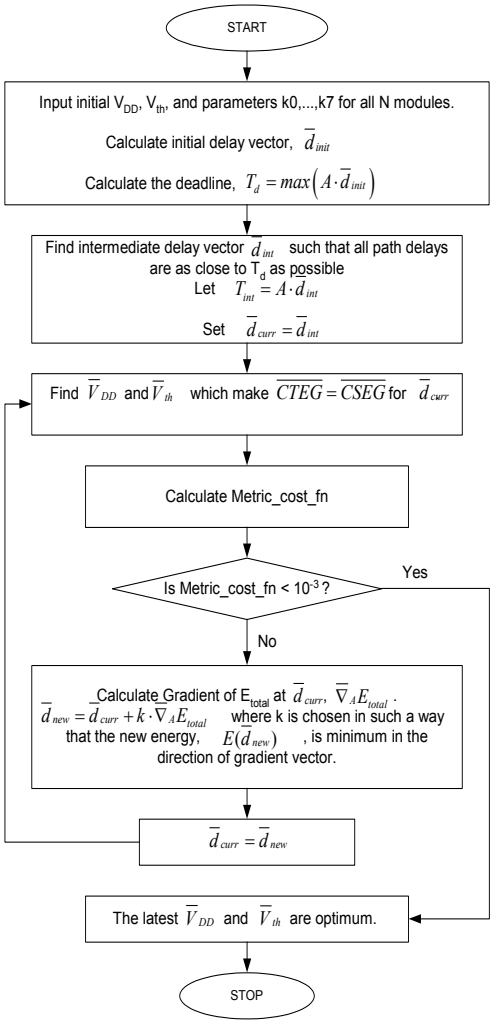


Figure 2. Algorithm for minimum energy consumption

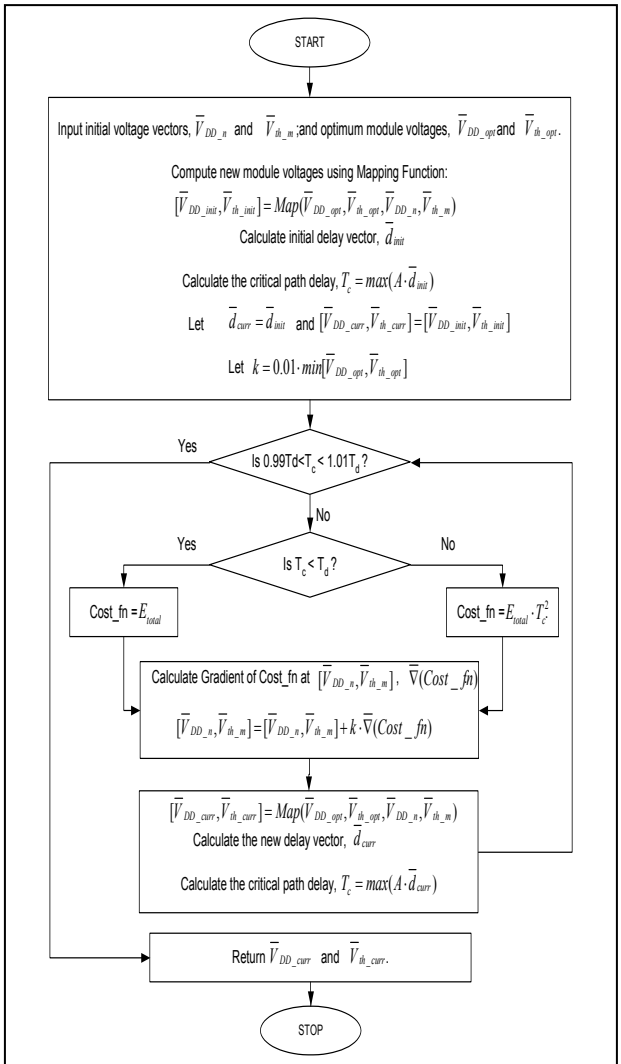


Figure 3. Algorithm for clustering

The delay vector, $\overline{d}_{new}$, for a new iteration is obtained from the current delay vector, $\overline{d}_{curr}$, as follows:

$$\overline{d}_{new} = \overline{d}_{curr} + k \cdot \overline{\nabla}_A E_{total} \tag{26}$$

where $\overline{\nabla}_A E_{total}$ is the gradient of E_{total} along the nullspace vectors of **A**. k is chosen in such a way that the new energy ($E(\overline{d}_{new})$) is minimum in the direction of gradient vector. We now derive the stopping condition for the gradient search.

The condition $\overline{CTEG} = \overline{CSEG}$ (Equation 23) is satisfied in every iteration of the search. To check how close $\overline{CTEG}$ (or $\overline{CSEG}$) is to $A^T \cdot \overline{\lambda}$, we define a Metric_cost_fn as follows:

$$\text{Metric\_cost\_fn} = norm\left(A^T \cdot A^{T\dagger} \cdot \overline{CTEG} - \overline{CTEG}\right) \Big/ norm\left(\overline{CTEG}\right) \tag{27}$$

where $A^{T\dagger}$ is the pseudo-inverse of A^T . At the minimum energy point, Metric_cost_fn should be zero as shown below:

$$\overline{CTEG}_{min} = A^T \cdot \overline{\lambda}$$

$$\begin{aligned}\text{Metric\_cost\_fn} &= norm\left(A^T \cdot A^{T\dagger} \cdot \overline{CTEG}_{min} - \overline{CTEG}_{min}\right) \Big/ norm\left(\overline{CTEG}_{min}\right) \\ &= norm\left(A^T \cdot A^{T\dagger} \cdot A^T \cdot \overline{\lambda} - A^T \cdot \overline{\lambda}\right) \Big/ norm\left(A^T \cdot \overline{\lambda}\right) \\ &= norm\left(A^T \cdot \overline{\lambda} - A^T \cdot \overline{\lambda}\right) \Big/ norm\left(A^T \cdot \overline{\lambda}\right) \\ &= 0\end{aligned}$$

Designers can use Metric_cost_fn to determine how close their design is to the optimum.

In our algorithm, we terminate the iterations when Metric_cost_fn goes below 10^{-3}. The overview of the optimization algorithm is given in the flowchart in Figure 2.

5. Clustering heuristic for limited number of supply and threshold voltages

The algorithm described in Section 4 yields optimum values of supply and threshold voltages for each module that minimize the overall circuit energy. But these voltages might all have different values, in which case a practical

implementation of the optimized circuit is difficult in current technologies. In this section, we propose a heuristic algorithm that clusters the optimum supply and threshold voltage values obtained into a limited number of supply and threshold voltages. The final solution meets the delay constraint at the expense of slightly higher total energy consumption than the optimum case.

Assume only n supply voltage planes and m threshold voltages are available (n<N, m<N). Note that the values of the available voltages are not fixed at the beginning, although their number is fixed. Let $\overline{V}_{DD\_opt}$ and $\overline{V}_{th\_opt}$ be the optimum supply and threshold voltage values (obtained in the previous section), respectively. Let $\overline{V}_{DD\_n}$ and $\overline{V}_{th\_m}$ be supply and threshold voltage vectors holding values for the limited number of supply and threshold voltages (n supply voltages, m threshold voltages) initialized as follows:

$$\overline{V}_{DD\_n}(p) = min\left(\overline{V}_{DD\_opt}\right) + \left(\frac{max\left(\overline{V}_{DD\_opt}\right) - min\left(\overline{V}_{DD\_opt}\right)}{n+1}\right) \cdot p \qquad for\ p = 1 \cdots n$$

$$\overline{V}_{th\_m}(q) = min\left(\overline{V}_{th\_opt}\right) + \left(\frac{max\left(\overline{V}_{th\_opt}\right) - min\left(\overline{V}_{th\_opt}\right)}{m+1}\right) \cdot q \qquad for\ q = 1 \cdots m$$

These vectors will finally hold the n supply voltage values and m threshold voltage values that will be used in the circuit. For any module i, the function "Map" finds the nearest pair $[V_{DD\_n}(p), V_{th\_m}(q)]$ to the pair $[V_{dd\_opt}(i), V_{th\_opt}(i)]$ and assigns it to $[V_{DD\_new}(i), V_{th\_new}(i)]$.

$$[\overline{V}_{DD\_new}, \overline{V}_{th\_new}] = Map(\overline{V}_{DD\_opt}, \overline{V}_{th\_opt}, \overline{V}_{DD\_n}, \overline{V}_{th\_m}) \quad (28)$$

In any iteration, the delay of the circuit (T_c) is calculated using $[\overline{V}_{DD\_new}, \overline{V}_{th\_new}]$. We use $E_{total} \cdot T_c^2$ as the cost function if T_c exceeds T_d by a fixed fraction (say 0.01). Doing this forces the critical path delay to go down in the next iteration, possibly increasing E_{total}. If T_c is less than T_d by a fixed fraction, E_{total} is used as the cost function. Doing this decreases the energy in the next iteration, possibly by increasing T_c. These cost functions were chosen because they yielded good results in experiments. The gradient, $\overline{\nabla}(Cost\_fn)$, is obtained by changing the entries of $\overline{V}_{DD\_n}$ and $\overline{V}_{th\_m}$ by a small amount, mapping these to new $[\overline{V}_{DD\_new}, \overline{V}_{th\_new}]$ and calculating the difference in the cost function. The new values of $\overline{V}_{DD\_n}$ and $\overline{V}_{th\_m}$, which lower the cost function, are obtained by searching in the direction of the gradient. The search terminates when the circuit delay is in 1% proximity of the deadline, T_d. The flowchart of the algorithm is given in Figure 3.

6. Experimental results

We synthesized the hierarchical Verilog descriptions of the combinational ISCAS'85 circuits and a 16-bit Wallace Tree Multiplier using Synopsys Design Compiler (with the TSMC 0.25μ library) to get the delay, dynamic energy and static energy consumption values for the modules at the top level of design hierarchy. The modules at the top level of hierarchy in the Verilog description were directly mapped to the modules used in the optimization*. The values of the process-dependent parameters (k_3, k_4, k_6, k_7) were obtained from SPICE simulations as explained in Section 2. SPICE simulation of simple gates showed that k_5 is 6 orders of magnitude smaller than k_2 for this technology. Since k_2 and k_5 scale almost linearly with number of gates [10, 12], k_5 can be taken to be 10^{-6} times k_2 for any module. The circuit-dependent parameters (k_0, k_1, k_2) were then calculated for each module by using the delay, dynamic energy and static energy values obtained from Synopsys and the process-dependent parameters.

We use the following notation for describing the results: The symbol "I" denotes the initial circuit which has the standard 0.25μ TSMC voltages (V_{DD} = 2.5V, V_{th} = 0.5V). We obtain the delay of the initial circuit using Synopsys Design Compiler and use this value as the time constraint for the optimization i..e the optimized circuits (II, III, IV) will have the same delay as I. "II" denotes the baseline circuit (for energy comparisons) that has the *single V_{DD} and V_{th} values that give the minimum energy consumption for the given deadline*. "III" denotes the circuit having optimum (and possibly all different) V_{DD}s and V_{th}s for the modules. "IV" denotes the circuit in which the V_{DD}s and V_{th}s in III have been clustered into two V_{DD}s and one V_{th}. We only use one V_{th} in the final circuit because we found that having more V_{th}s only saved an additional 2-3% of energy in the benchmark circuits designed using 0.25μ technology. The need for multiple V_{th}s will become more pronounced as technology shrinks.

For the experiments, we used various switching activities for the input ports to observe their effects on the energy savings and the optimum voltages obtained. We noticed that for switching activities above 0.05, the optimum V_{th}s were of the order of 10 mV. This is due to the fact that the static energy in 0.25μ technology is very small compared to the dynamic energy for high switching activities. So for these cases, the optimization algorithm scales down V_{DD} aggressively and to achieve the delay constraint, it reduces V_{th} to very small values without incurring a significant increase in static energy. Since such small V_{th} values are not currently feasible, for these cases we fixed V_{th} at 0.1V and found the optimum V_{DD}s. This phenomenon is not expected ot occur for deep sub-micron technologies, where static energy is significant.

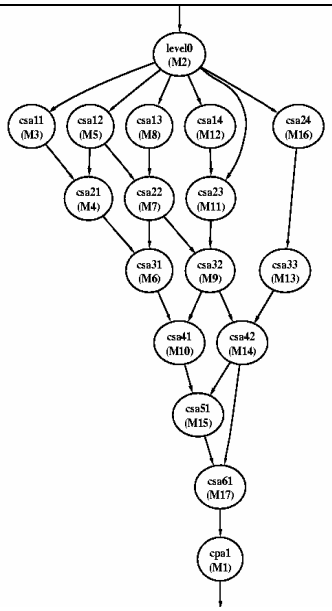
Table 1 provides detailed results for the Wallace Tree Multiplier circuit. The first column in the table shows the top level of the Verilog design hierarchy. The modules are a partial product generator (level0), Carry Save Adders (CSAs), and a Carry Propagate Adder (CPA). Also shown is the **A** matrix corresponding to the circuit. The second column gives the V_{DD}, V_{th} and energy consumption values for the baseline circuit (II) for two different input switching activities (SA=0.01 and SA=0.0001). Note that the delay for the baseline circuit is same as the delay of the initial circuit (I), which had V_{DD} = 2.5V, V_{th} = 0.5V. The third and fourth columns give the voltages for each module as well as the energy consumptions for circuits III and IV respectively.

Figure 4 shows the energy savings obtained for the various benchmark circuits as a percentage of the baseline energy consumption for an input switching activity of 0.01. The dynamic and static components of energy are also shown. It is observed that in II and III, static energy is ~10% of the total energy. This validates the fact that at the optimum, static energy is a fixed fraction of the total energy [14], although this fraction depends on the technology used.

Figures 5 and 6 show the savings for different input switching activities for circuits III and IV, respectively. The results show that the energy savings tend to increase as the input switching activity increases. Thus, accurate estimation of

* A power-aware partitioning of the circuit into modules could further improve the results, but that by itself is a very difficult problem to solve and is not handled in this work.

Table 1. Optimization Results for a Wallace Tree Multiplier (with two different input switching activities)

Wallace Tree Multiplier		II (Baseline System)		III (unlimited V_{DD}s, V_{th}s)		IV (2 V_{DD}s, 1 V_{th})	
	A^T	i	ii	i	ii	i	ii
	$\begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 1&1&1&0&0&1&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&1&1&1&0&0&0&0 \\ 0&0&0&0&0&1&1&1&0&0&0&0&0 \\ 0&0&0&1&1&0&1&1&1&1&1&0&0 \\ 1&1&1&1&0&1&1&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0&1&0&1&1&0&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$	SA=0.01 V_{DD}=1.62V V_{th}=0.11V T_d=13.7 ns E=71.6 pJ Ed=63.4 pJ Es=8.2 pJ	SA=0.0001 V_{DD}=2.18V V_{th}=0.35V T_d=13.7 ns E=0.35 pJ Ed=0.33 pJ Es=0.02 pJ	$\begin{bmatrix} V_{DD} & V_{th} \end{bmatrix}$ $\begin{bmatrix} 2.23 & 0.10 \\ 1.38 & 0.09 \\ 0.85 & 0.12 \\ 0.83 & 0.11 \\ 0.85 & 0.12 \\ 0.77 & 0.08 \\ 0.84 & 0.11 \\ 0.85 & 0.12 \\ 0.80 & 0.09 \\ 0.94 & 0.08 \\ 0.84 & 0.11 \\ 0.86 & 0.12 \\ 0.50 & 0.09 \\ 1.00 & 0.10 \\ 0.63 & 0.06 \\ 0.54 & 0.12 \\ 0.64 & 0.06 \end{bmatrix}$ T_c=13.7 ns E=36.9 pJ Ed=31.8 pJ Es=5.1 pJ Saving=48%	$\begin{bmatrix} V_{DD} & V_{th} \end{bmatrix}$ $\begin{bmatrix} 2.80 & 0.33 \\ 1.75 & 0.32 \\ 1.43 & 0.38 \\ 1.48 & 0.36 \\ 1.43 & 0.38 \\ 1.39 & 0.34 \\ 1.49 & 0.37 \\ 1.40 & 0.36 \\ 1.44 & 0.36 \\ 1.41 & 0.32 \\ 1.46 & 0.37 \\ 1.47 & 0.38 \\ 0.97 & 0.33 \\ 1.47 & 0.35 \\ 1.07 & 0.30 \\ 1.02 & 0.36 \\ 1.07 & 0.30 \end{bmatrix}$ T_c=13.7 ns E=0.22 pJ Ed=0.20 pJ Es=0.02 pJ Saving=39%	$\begin{bmatrix} V_{DD} & V_{th} \end{bmatrix}$ $\begin{bmatrix} 1.84 & 0.09 \\ 1.84 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \\ 1.84 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \\ 1.84 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \\ 0.91 & 0.09 \end{bmatrix}$ T_c=13.7 ns E=49.4 pJ Ed=41.9 pJ Es=7.5 pJ Saving=31%	$\begin{bmatrix} V_{DD} & V_{th} \end{bmatrix}$ $\begin{bmatrix} 2.42 & 0.34 \\ 2.42 & 0.34 \\ 1.45 & 0.34 \\ 2.42 & 0.34 \\ 1.45 & 0.34 \\ 1.45 & 0.34 \\ 2.42 & 0.34 \\ 1.45 & 0.34 \\ 1.45 & 0.34 \\ 1.45 & 0.34 \\ 2.42 & 0.34 \\ 2.42 & 0.34 \\ 1.45 & 0.34 \\ 2.42 & 0.34 \\ 1.45 & 0.34 \\ 1.45 & 0.34 \\ 1.45 & 0.34 \end{bmatrix}$ T_c=13.7 ns E=0.28 pJ Ed=0.25 pJ Es=0.03 pJ Saving=22%

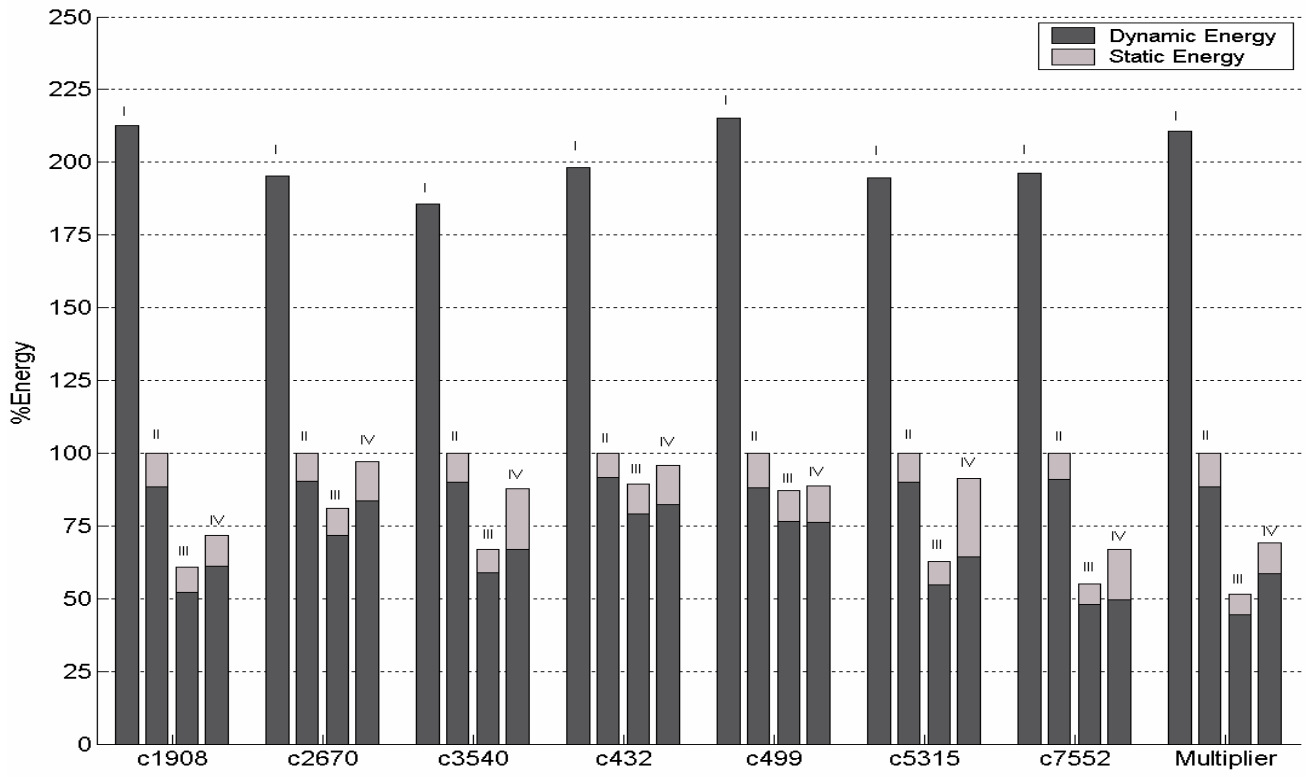


Figure 4. Energy consumption of benchmark circuits as a percentage of the baseline energy consumption when the input switching activity is 0.01.

I: Energy consumption of circuit with standard 0.25μ TSMC voltages (V_{DD} = 2.5, V_{th} = 0.5)
II: Energy consumption of circuit with optimum single V_{DD} and single V_{th} (baseline case)
III: Energy consumption of circuit with unlimited V_{DD}s and V_{th}s
IV: Energy consumption of circuit with two V_{DD}s and one V_{th}

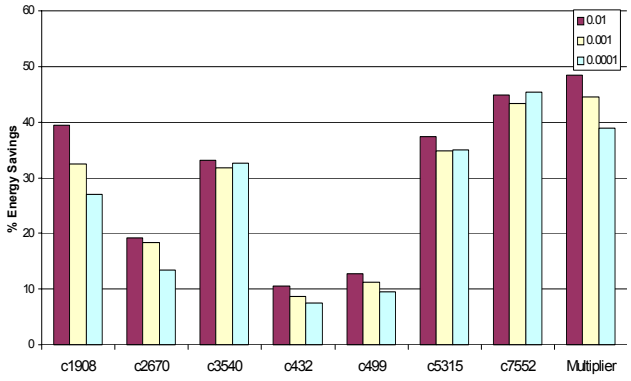


Figure 5. Percent energy savings with unlimited V_{DD}s and V_{th}s (III) for different input switching activities

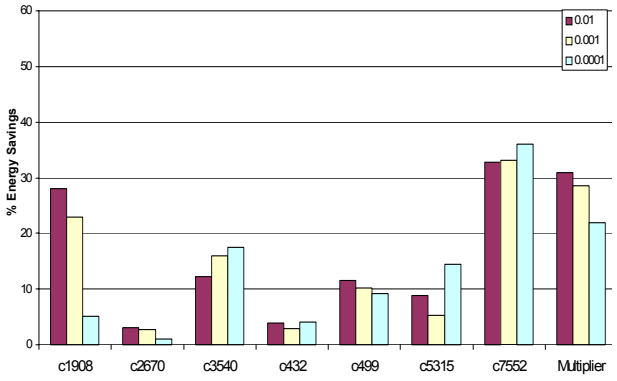


Figure 6. Percent energy savings with two V_{DD}s and one V_{th} (IV) for different input switching activities

Table 2. Optimization Results

Circuit	Input Switching Activity	E (I) pJ	E (II) pJ	E (III) pJ	E (IV) pJ	V_{DD} (II)	V_{th} (II)	V_{DD1} (IV)	V_{DD2} (IV)	V_{th} (IV)	% Energy Savings (III)	% Energy Savings (IV)
c1908	0.5	106	44.2	25.9	30.6	1.6	0.10	1.2	2.1	0.10	41.5	30.9
	0.1	44.1	18.7	11.2	13.2	1.6	0.10	1.2	2.1	0.10	40.2	29.4
	0.01	5.85	2.75	1.67	1.98	1.6	0.10	1.2	2.1	0.10	39.4	28.1
	0.001	0.62	0.37	0.25	0.28	1.8	0.20	1.4	2.4	0.19	32.5	22.9
	0.0001	0.07	0.05	0.04	0.05	2.0	0.28	1.7	2.1	0.27	26.9	5.1
c2670	0.5	238	100	92.5	100	1.6	0.10	1.6	1.6	0.10	7.5	0
	0.1	78.1	33.5	31.2	33.5	1.6	0.10	1.6	1.6	0.10	6.9	0
	0.01	8.95	4.58	3.71	4.45	1.7	0.14	1.3	1.7	0.12	19.2	3.0
	0.001	0.85	0.55	0.45	0.53	1.9	0.23	1.5	1.9	0.22	18.4	2.6
	0.0001	0.09	0.07	0.06	0.07	2.1	0.32	1.8	2.1	0.31	13.4	1.0
c3540	0.5	414	175	120	139	1.6	0.10	1.2	1.6	0.10	31.5	20.3
	0.1	130	57.0	39.1	45.3	1.6	0.10	1.2	1.7	0.10	31.4	20.6
	0.01	14.4	7.75	5.18	6.81	1.7	0.16	1.3	1.7	0.12	33.2	12.2
	0.001	1.29	0.87	0.60	0.73	2.0	0.25	1.5	2.0	0.22	31.8	16.0
	0.0001	0.09	0.07	0.05	0.06	2.2	0.36	1.7	2.3	0.35	32.6	17.6
c432	0.5	23.5	9.81	9.05	9.47	1.6	0.10	1.5	1.8	0.10	7.7	3.5
	0.1	6.77	2.88	2.65	2.77	1.6	0.10	1.5	1.7	0.10	8.0	3.9
	0.01	0.74	0.37	0.33	0.36	1.7	0.14	1.5	1.8	0.11	10.6	4.0
	0.001	0.09	0.053	0.049	0.052	1.9	0.22	1.7	2.1	0.20	8.7	3.0
	0.0001	0.01	0.0084	0.0078	0.0081	2.1	0.30	1.9	2.4	0.29	7.5	4.1
c499	0.5	81.4	34.0	26.9	27.6	1.6	0.10	1.2	1.9	0.10	20.8	18.8
	0.1	34.4	14.5	11.8	12.0	1.6	0.10	1.2	1.9	0.10	18.5	17.1
	0.01	4.81	2.24	1.95	1.98	1.6	0.10	1.3	1.8	0.09	12.8	11.5
	0.001	0.49	0.29	0.26	0.26	1.8	0.19	1.5	2.0	0.19	11.1	10.2
	0.0001	0.05	0.039	0.035	0.035	2.0	0.28	1.7	2.3	0.28	9.5	9.1
c5315	0.5	438	184	110	153	1.6	0.10	0.5	1.6	0.10	40.0	16.8
	0.1	143	61.5	37.3	50.7	1.6	0.10	0.5	1.6	0.10	39.3	17.5
	0.01	16.7	8.59	5.38	7.83	1.7	0.14	0.5	1.6	0.09	37.3	8.9
	0.001	1.59	1.03	0.67	0.97	1.9	0.23	0.6	1.8	0.18	34.8	5.3
	0.0001	0.15	0.12	0.07	0.10	2.1	0.33	0.8	2.1	0.29	35.0	14.5
c7552	0.5	861	361	259	285	1.6	0.10	1.0	1.6	0.10	28.1	21.0
	0.1	283	121	84.7	86.0	1.6	0.10	0.6	1.6	0.10	29.9	28.9
	0.01	32.3	16.4	9.04	11.00	1.7	0.14	0.7	1.6	0.10	44.9	32.8
	0.001	3.34	2.12	1.20	1.42	1.9	0.22	1.0	1.9	0.20	43.4	33.2
	0.0001	0.42	0.32	0.17	0.20	2.1	0.31	1.2	2.1	0.31	45.4	36.0
Multiplier	0.5	2890	1210	502	834	1.6	0.10	1.0	1.8	0.10	58.4	30.0
	0.1	1180	500	245	342	1.6	0.10	1.0	1.8	0.10	51.0	31.6
	0.01	151	71.6	36.9	49.40	1.6	0.11	0.9	1.8	0.09	48.4	30.9
	0.001	11.7	7.21	4.00	5.16	1.9	0.21	1.1	2.1	0.20	44.5	28.5
	0.0001	0.43	0.35	0.22	0.28	2.2	0.35	1.5	2.4	0.34	39.0	22.0

the input switching activity is crucial for obtaining good energy savings.

Table 2 summarizes the results of the experiments. V_{DD1} and V_{DD2} are the two voltages applied to the circuit after clustering. We obtained up to 48.4% savings for circuit III and up to 36% savings for circuit IV for switching activities below 0.05 (V_{th}s variable). For switching activities above 0.05 (V_{th}s fixed at 0.1V), we obtained up to 58.4% savings for circuit III and up to 31.6% savings for circuit IV. The average saving, for switching activities above 0.05, was 29% for circuit IIIand 18% for circuit IV. For switching activities below 0.05, the average saving was 28% for circuit III and 15% for circuit IV.

The optimization algorithms in Sections 4 and 5 were implemented in MATLAB and run on a PC with 1 GB RAM and PIII 800 MHz processor. To compare execution times for the different circuits, we terminate the optimal algorithm in Section 4 after 10 iterations of the loop in Figure 2. 10 iterations were enough to get near optimal results for most of the cases. Since we observed that the clustering algorithm (Section 5) takes only about 5-10% of the total execution time, we let it run to completion. Figure 7 shows the execution time of the program versus number of modules (N) multiplied with the number of null-space vectors of **A** (|Null(**A**)|). Execution time is roughly proportional to N x |Null(**A**)| because in each iteration of the loop in Figure 2, computation of each co-ordinate of the gradient vector, $\overline{\nabla}_A E_{total}$, requires computation of supply, threshold voltage and energy consumption values for each module.

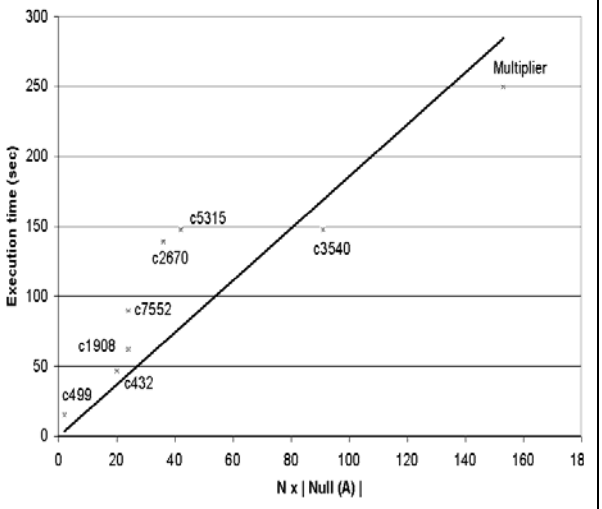


Figure 7. Execution time versus N x | Null(A)|

7. Conclusion and future work

In this paper, we presented an algorithm to find *optimum* values of supply and threshold voltages for circuit modules such that the energy consumption is minimized. The conditions for optimum energy were found mathematically and then a gradient search algorithm was presented which iteratively converges to the optimum values. An additional step clusters these optimum values into a limited number of supply and threshold voltages. The method can be applied to circuit modules of any kind, given the delay and energy parameters for the modules.

As a next step, we plan to apply our algorithm to deep sub-micron technologies, which we believe will give more energy savings than the results for 0.25µ. We are also investigating techniques for power-aware partitioning of circuits into modules.

References

[1] A. U. Diril, Y. S. Dhillon, K. Choi, A. Chatterjee, "An O(N) Supply Voltage Assignment Algorithm for Low-Energy Serially Connected CMOS Modules and a Heuristic Extension to Acyclic Data Flow Graphs," *ISVLSI,* pp.173-179, February 2003.

[2] T. Sakurai, A.R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, pp.584-594, April 1990.

[3] S. Lee, T. Sakurai, "Run-time Voltage Hopping for Low Power Real-Time Systems," *DAC*, pp.806-809, 2000.

[4] T. Pering, T. Burd, R. Brodersen, "The Simulation and Evaluation of Dynamic Voltage Scheduling Algorithms," *ISLPED*, pp.76-81, 1998.

[5] J. Chang, M. Pedram, "Energy Minimization Using Multiple Supply Voltages," *IEEE Trans. on VLSI Systems*, vol.5, no.4, December 1997.

[6] M. Johnson, K. Roy, "Optimal Selection of Supply Voltages and Level Conversions During Data Path Scheduling Under Resource Constraints," *ICCD '96*, pp.72 -77, 1996.

[7] P. Pant, R.K. Roy, A. Chattejee, "Dual-threshold voltage assignment with transistor sizing for low power CMOS circuits," *IEEE Trans. on VLSI Systems*, vol.9, no.2, pp.390 -394, April 2001.

[8] V. Sundararajan, K.K. Parhi, "Low power synthesis of dual threshold voltage CMOS VLSI circuits," *ISLPED*, pp.139-144, 1999.

[9] Y. Moisiadis, I. Bouras, A. Arapoyanni, "Dynamic back bias CMOS driver for low-voltage applications," *Electronics Letters*, vol.36, no.2, pp.135-136, Jan. 2000.

[10] J.A. Butts, G.S. Sohi, "A static power model for architects," *IEEE/ACM MICRO*, pp. 191-201, 2000.

[11] K. Nose, T. Sakurai, "Analysis and future trend of short-circuit power," *IEEE Trans. on CAD*, vol.19, no.9, pp.1023-1030, Sept. 2000.

[12] R. Kumar, C.P. Ravikumar, "Leakage power estimation for deep submicron circuits in an ASIC design environment," *DAC*, pp.45-50, 2002.

[13] R. Nair, C.L. Berman, P.S. Hauge, E.J. Yoffa, "Generation of performance constraints for layout," *IEEE Trans. on CAD*, vol.8, no.8, pp.860-874, Aug. 1989.

[14] K. Nose, T. Sakurai, "Optimization of V_{DD} and V_{TH} for low-power and high-speed applications," *ASP-DAC*, pp.469-474, 2000.

[15] R. Bai, S. Kulkami,W. Kwong, A. Srivastava, D. Sylvester, D.Blaauw, "An implementation of a 32-bit ARM processor using dual power supplies and dual threshold voltages**,**" *ISVLSI*, pp.149-154, 2003.

[16] L. Wei, Z. Chen, K. Roy, M.C. Johnson, Y. Ye, V.K. De, "Design and optimization of dual-threshold circuits for low-voltage low-power applications**,"** *IEEE Trans. on VLSI Systems*, pp.16-24, vol.7, no.1, March 1999.