Total Power Reduction in CMOS Circuits via Gate Sizing and Multiple Threshold Voltages

Feng Gao and John P. Hayes
Advanced Computer Architecture Lab.
University of Michigan, Ann Arbor, MI 48105, USA
{fgao, jhayes}@eecs.umich.edu

ABSTRACT

Minimizing power consumption is one of the most important objectives in IC design. Resizing gates and assigning different V_t's are common ways to meet power and timing budgets. We propose an automatic implementation of both these techniques using a mixed-integer linear programming model called *MLP-exact*, which minimizes a circuit's total active-mode power consumption. Unlike previous linear programming methods which only consider local optimality, *MLP-exact* can find a true global optimum. An efficient, non-optimal way to solve the MLP model, called *MLP-fast*, is also described. We present a set of benchmark experiments which show that *MLP-fast* is much faster than *MLP-exact,* while obtaining designs with only slightly higher power consumption. Furthermore, the designs generated by *MLP-fast* consume 30% less power than those obtained by conventional, sensitivity-based methods.

Categories and Subject Descriptors

B.6.3 Design aids

General Terms

Algorithms, design, experimentation

Keywords

Low power, linear programming, dual V_t, gate sizing

1. INTRODUCTION

Battery-powered, hand-held devices have become the biggest markets for integrated circuits (ICs). For such devices, low power consumption is perhaps the most important design objective. Various approaches have been proposed that aim at minimizing total power consumption; these include sensitivity-based methods [1] [13], linear programming [2][12], and genetic algorithms [10]. These methods typically employ power-reducing design techniques at the circuit level, especially dual threshold voltage assignment and gate sizing.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

Earlier work on gate sizing such as TILOS [6] focuses on delay rather than power optimization. It sizes the gates on critical paths based on the sensitivity of the circuit delay to gate size. A similar technique targeting power optimization using gate sizing and dual V_t assignment is proposed by Pant et al. [13]. Although such techniques are usually fast, their optimality is hard to determine.

Nguyen et al. [12] propose an iterative technique that employs a linear programming (LP) model for slack distribution based on the power delay sensitivity of each gate. However, the linearization technique used for the LP model is only accurate within a small range of delay. The optimization process hence has to be iterated, where the power delay sensitivities are re-computed. The optimality of the final solution remains hard to evaluate.

The major difficulty in constructing a successful LP model for delay-constrained power optimization lies in the fact that gate delay as a function of gate size and load capacitance, is hard to linearize. Specifically, the delay t of gate G with size S and load capacitance C is given by

$$t = a + b\,C/S \quad (1)$$

Berkelaar et al. [2] attempt to linearize C/S with piecewise linear functions of the form $c\,C + d\,S$. As in [12], this approximation is only valid over a very small range of values of S and C. Therefore, this application of the LP model only explores a limited range of cell sizes. An optimal way of exploring the design space while considering a large range of cell sizes provided by a cell library is still an open question.

In this paper, we propose to model the problem of total power optimization via gate sizing and V_t assignment as a mixed-integer linear program (MLP). However, we adopt a new way to linearize the delay function (1), which makes exploration of the entire design space possible. The proposed MLP model is able to obtain true global optima, and is hence named *MLP-exact*. We also modify *MLP-exact* to obtain a much faster approximate MLP model, referred to as *MLP-fast*. We present experimental results which show that *MLP-fast* is usually one or two orders of magnitude faster than *MLP-exact* while incurring just 3% power overhead in the designs obtained. Furthermore, the designs obtained using *MLP-fast*'s consume 30% less power than their

counterparts obtained with a TILOS-like [6] sensitivity-based method.

The paper is organized as follows. In Section 2, we present the delay and power consumption model, as well as our technique for linearizing the delay function. The exact MLP model for simultaneous gate sizing and V_t assignment, and the faster approximate model are described in Section 3. The experimental results are analyzed in Section 4, and Section 5 concludes this paper.

2. PRELIMINARIES

We start by defining our notation and describing our power and delay models. We assume a cell-based design flow with a given cell library, whose available V_t's are determined by the process technology. We assume that there is a basic unit cell for each cell type in the given cell (gate) library, and use $\overline{G}$ to denote the unit cell of the same type as cell G. The size $S(G)$ of G is the ratio of G and $\overline{G}$'s physical sizes. Let $I_l(G)$, $C_g(G)$, and $D(G)$ denote the leakage current, gate capacitance, and gate delay, respectively, of G. Neglecting high-order effects, we assume that $I_l(G) = I_l(\overline{G})S(G)$ and $C_p(G)=C_p(\overline{G})S(G)$.

Power consumption model. Since the switching time of a gate is much smaller than the system clock period, we assume that a gate stays in a stable state and keeps leaking for most of a cycle, no matter whether or not it switches during that cycle. Let $I_l(G,I)$ be G's leakage current under input pattern I. The transition probability $TP(G)$ is the probability that G's output switches, while the signal probability $SP(G,I)$ is the probability that input pattern I is applied to G. These probabilities can be calculated using BDD-based [8] or simulation-based [4] approaches. We resort to the latter approach because of its simplicity.

The average leakage power dissipation $P_l(G)$ of G can be expressed as the product of the average leakage current and power supply voltage V_{DD}.

$$P_l(G) = \sum_I P(G, I) I_l(G, I) V_{DD} = [\sum_I P(G, I) I_l(\overline{G}, I)] S(G) V_{DD} \quad (2)$$

The dynamic power dissipated in a cycle by G depends on its transition probability $TP(G)$, load capacitance $C_L(G)$, and V_{DD}. The load capacitance consists of the wire capacitances $C_w(G)$ and the gate capacitances of G's fanout gates U:

$$C_L(G) = C_w(G) + \sum_U C_g(\overline{U}) S(U).$$

As shown in [9], local wire delays scale with gate delays. We hence ignore the impact of local connections. Assuming a clock cycle t_c, we calculate the dynamic power dissipation of G as

$$P_d(G) = TP(G) \sum_U C_g(\overline{U}) S(U) V_{DD}^2/(2t_c) \quad (3)$$

The short-circuit power is usually controlled to a small percentage of the total power consumption, and the percentage is insensitive to load capacitance. We hence ignore this power component [5], and calculate the total power consumption of G as

$$P_{total}(G) = [\sum_I P(G, I) I_l(\overline{G}, I)] S(G) V_{DD} + TP(G) \sum_U C_g(\overline{U}) S(U) V_{DD}^2/(2t_c) \quad (4)$$

Delay model. We assume a conventional RC delay model. The gate delay $D(G)$ is hence linear in the ratio of load capacitance to cell size $C_L(G)/S(G)$ [16], so

$$D(G) = D_p(\overline{G}) + D_l(\overline{G})\, C_L(G)/S(G)$$

where $D_p(\overline{G})$ is the parasitic delay and $D_l(\overline{G})$ is the load delay coefficient, both of which are independent of G's size. The delay function of gate G is then

$$D_p(\overline{G}) + D_l(\overline{G}) \sum_U C_g(\overline{U}) S(U)/S(G) \quad (5)$$

We assume that two V_t's are available for each cell. We therefore define a binary variable $V_t(G)$ to represent G's V_t selection, where $V_t(G) = 1$ for V_t^L and $V_t(G) = 0$ for V_t^H. Accordingly, we augment the V_t-dependent variables $D_p(\overline{G})$, $D_l(\overline{G})$ and $I_l(G, I)$ with a new parameter $V_t(G)$, and obtain $D_p(\overline{G}, V_t(G))$, $D_l(\overline{G}, V_t(G))$ and $I_l(G, I, V_t(G))$. Linearization of these functions with respect to $V_t(G)$ will be considered during the MLP model construction in Section 3. Also note that for a given V_t selection, the above parameters can be measured beforehand for each gate type and will hence appear as constant coefficients in the MLP model. In fact, we can easily extend the model to $k > 2$ threshold voltage levels by specifying $V_t(G)$ as an integer between 0 and $k - 1$.

While higher-order delay and leakage models will be more accurate, the first-order estimation is still very accurate in the majority of cases [14]. Our Spice simulation data also justify this claim. For example, a comparison of our delay model with Spice simulation appears in Figure 1. Here we show the delay of an inverter of size 1 to 4 driving another inverter of size 1 to 5. The label Spice-S*i* refers to the delay from Spice simulation with a driving inverter of size *i*, where *i* = 1, 2, 3, 4. Similarly, Modeled-S*i* is the delay from the delay model in this paper. The figure shows that the delay

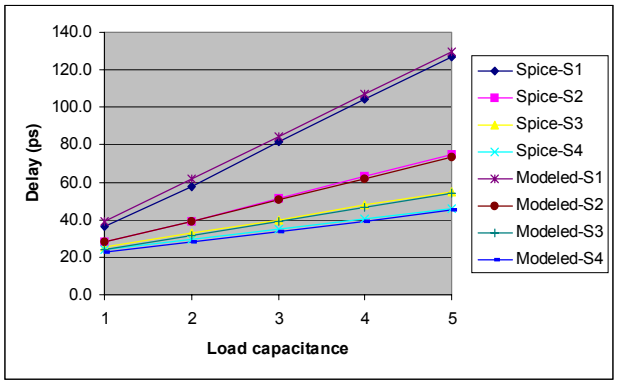

Figure 1. Delays obtained using our delay model and Spice simulation

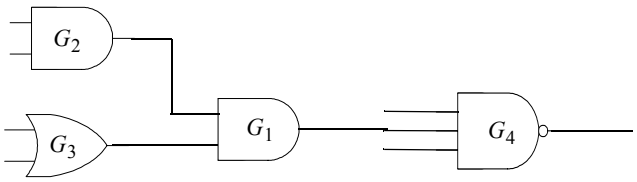

Figure 2. A circuit fragment

calculated using our model follows the Spice simulation very closely. In addition, we also neglect $C_g(G)$'s dependence on $V_t(G)$, which is negligible. Of course, the depedency can be linearized like the other V_t-related terms, if necessary.

Linearization of delay function. The delay function shown in Equation (5) is nonlinear in $S(G)$. We now consider the linearization of $S(U)/S(G)$. The technique given in [2] uses a set of linear functions defined on a number of subranges of $S(G)$ as follows.

$$S(U)/S(G) \geq a_1 + a_2\, S(U) + a_3\, S(G) \tag{6}$$

Coefficients a_i, $i \in [1,3]$ are determined by the subranges of $S(G)$. The error introduced by $a_3 S(G)$ can hence be controlled by adjusting the subrange size of $S(G)$. However, for a given subrange of $S(G)$, the error introduced by $a_2 S(U)$ still increases with $S(U)$. Since $S(U)$ depends on the total size range of U, this method is only capable of exploring a limited range of $S(U)$ values to limit the modeling error.

The disadvantage of the above method results from the fact that the coefficient of $S(U)$ depends on $S(G)$ instead of $S(U)$. Similar to [7], we solve this problem by defining two variables x and y such that $S(U) = 2^x$ and $S(G) = 2^y$. $S(U)/S(G)$ is thus transformed to 2^{x-y}. Piecewise linearization can then be applied to this exponential function. We use the lines determined by points $(m, 2^m)$ and $(n, 2^n)$ for this purpose:

$$S(U)/S(G) \geq ((2^n - 2^m)/(n - m))(x - y) + (n2^m - m2^n)/(n - m) \tag{7}$$

which is linear in x and y. Note that if we choose m as integers and $n = m + 1$, the above equation simplifies to

$$S(U)/S(G) \geq 2^m\,(x - y) + 2^m\,(1 - m) \tag{8}$$

All coefficients in this linearization method depend on both $S(U)$ and $S(G)$. The accuracy of this piecewise linear approximation is therefore completely determined by the subranges of $x - y$, i.e., by the choice of m and n. Consequently, the constraint on cell size ranges where such approximation is applicable is eliminated, enabling optimization over the full range of gate sizes. Note that the base of the exponential functions does not have to be two. Any positive number larger than one is a valid candidate. We discuss how to choose the base in the next section.

3. ILP MODEL

Using the assumptions and linearization techniques presented in the preceding section, we now construct an MLP model for simultaneous V_t selection and gate sizing. We assume that two threshold voltages are available, but extension to more voltage levels is straightforward.

Objective function. We use the sum of power dissipation of each gate as in Equation (4), as the objective function to be minimized. Note that $I_l(G,I)$ is V_t-dependent. Taking the threshold voltage $V_t(G)$ into consideration, we represent the leakage current by

$$I_l(G,V_t(G)) = \sum_I P(G, I) I_l(\overline{G}, I, 0)\, S(G) V_t(G) + \sum_I P(G, I) I_l(\overline{G}, I, 1)\, S(G) V_t(G) \tag{9}$$

Since $V_t(G)$ is a binary variable, we can hence replace Equation (9) with

$$I_l(G,V_t(G)) = I_l^0(G)\, 2^{rl(G)\mathrm{Vt(G)}} S(G) \tag{10}$$

where $I_l^k(G) = \sum_I P(G, I) I(\overline{G}, I_l, k)$ for $k = 0$, 1 and $rl(G) = \log_2(I_l^1(G)/I_l^0(G))$.

Note that Equation (10) is equivalent to Equation (9) for $V_t(G) = 0$ or 1 only. Now our objective function can be expressed in the following form:

$$I_l^0(G)\, 2^{rl(G)Vt(G)} S(G) \mathrm{V_{DD}} + TP(G) \sum_U C_g(\overline{U}) S(U)\, \mathrm{V_{DD}}^2/(2t_c) \tag{11}$$

Constraints. There are two classes of constraints in the MLP model: performance and linearization. The performance constraints guarantee that the performance

Minimize

$$\sum_{G_i,\, i \in [1,4]} I_l^0(G) 2^{rl(G)Vt(G)} S(G) VDD + \sum_{G_i,\, i \in [1,4]} \left(TP(G) \sum_U C_g(\overline{U}) S(U) VDD^2 \right) / (2t_c)$$

Subject to

{Performance constraints}

$T_a(G_o) \leq D_{max}$

$T_a(G_I) = 0$

{We only show constraints for G_1}

$T_a(G_2) + D(G_1, V_t(G_1) \leq T_a(G_1)$

$T_a(G_3) + D(G_1, V_t(G_1) \leq T_a(G_1)$

$D(G_1, V_t(G_1)) = D_p^0(G_1) + D_p^{10}(G_1)\, V_t(G_1) + C_g(G_4) SVR(G_4,G_1) D_l(\overline{G}_1, 0)$

$SVR(G_4,G_1) \geq 2^k (rd(G_1) V_t(G_1) + p(G_4) - p(G_1)) + (1- k)2^k,$

for $k = -p_{max}, -p_{max} + 2, \ldots p_{max} + 1$

$S(G_1) \geq 2^k p(G_1) + (1- k)2^k,\ k = 0, 2, \ldots, p_{max}$

$SV(G_1) \geq 2^k (rl(G_1) + p(G_1)) + (1- k)2^k,\ k = 0, 2, \ldots, p_{max}$

Bounds

$V_t(G)$s are binary variables

Figure 3. Summary of *MLP-exact* for the circuit in Figure 2

target is met for the size and V_t selection in the model. Non-linear terms appearing in the objective function and performance constraints are replaced with linear inequalities, which are the linearization constraints.

First consider the performance constraints. Let real variable $T_a(G)$ be the arrival time of G's output signal. For convenience, we insert a virtual gate G_I driving all primary inputs, and a virtual gate G_o driven by all primary outputs. Both virtual gates have zero delay. To satisfy the overall circuit delay D_{max}, we use constraints $T_a(G_o) \le D_{max}$, and $T_a(G_I) = 0$. We then derive constraints to relate the arrival times of G's fanin gates to that of G. Consider the circuit fragment in Figure 2, where G_1 has two inputs driven by G_2 and G_3. The arrival time of G_1's output signal satisfies

$$T_a(G') + D(G_1, V_t(G_1)) \le T_a(G_1)$$

where $G' \in \{G_2, G_3\}$.

We calculate $D(G, V_t(G))$ using linear functions as follows. Let $D_p(\overline{G}, V)$ be the parasitic delay of $\overline{G}$ with threshold voltage selection $V_t(G)$. and $D_p^{10}(G) = D_p(\overline{G}, 1) - D_p(\overline{G}, 0)$. Then, we can rewrite $D_p(\overline{G}, V_t(G))$ as

$$\begin{aligned} & D_p(\overline{G}, 0)\ \overline{V_t(G)}\ + D_p(\overline{G},1)\ V_t(G) \\ &= D_p(\overline{G}, 0) + (D_p(\overline{G}, 1) - D_p(\overline{G}, 0))\ V_t(G) \\ &= D_p(\overline{G}, 0) + D_p^{10}(G)\ V_t(G) \end{aligned} \quad (12)$$

On the other hand, we represent $D_l(G, V_t(G))$ similarly to $I_l(G,V_t(G))$:

$$D_l(\overline{G}, 0)2^{rd(G)Vt(G)} \quad (13)$$

where $rd(G)=\log_2(D_l(\overline{G}, 1)/D_l(\overline{G}, 0))$. Therefore, the total delay $D(G, V_t(G))$ of G is

$$\begin{aligned} D(G,V_t(G)) = {} & D_p^0(G) + D_p^{10}(G)\ V_t(G) \\ & + D_l(\overline{G}, 0)2^{r(G)Vt(G)} \sum_U C_g(\overline{U})S(U)/S(G) \end{aligned} \quad (14)$$

As discussed in Section 2, we define variable $p(G)$ for each gate, where $S(G) = 2^{p(G)}$. Therefore, we obtain

$$S(U)/S(G) = 2^{p(U) - p(G)} \quad (15)$$

The delay function $D(G,V_t(G))$ becomes

$$\begin{aligned} D(G,V_t(G)) = {} & D_p^0(G) + D_p^{10}(G)\ V_t(G) \\ & + D_l(\overline{G}, 0)\ \sum_U C_g(\overline{U})SVR(U, G) \end{aligned} \quad (16)$$

where $SVR(U,G) = 2^{rd(\mathrm{G})Vt(G)+p(U)-p(G)}$. We can now simply apply piecewise linear approximation to the exponential function $SVR(U,G)$, and obtain linearized relations between G's delay and size. Furthermore, the accuracy of the approximation is totally controlled by how we choose the piecewise linear functions.

Linearized objective function. Since we define $p(G) = \log_2 S(G)$ for each gate, the objective function is transformed accordingly to

$$I_l^0(G)\, SV(G) V_{\mathrm{DD}} + TP(G) \sum_U C_g(\overline{U}) S(U)\, V_{\mathrm{DD}}^2/(2t_c) \quad (17)$$

where $SV(G) = 2^{rl(G)Vt(G)+p(G)}$ and $S(G) = 2^{\mathrm{p(G)}}$. We also apply the foregoing linearization technique to these two exponential functions in the objective function.

MLP-exact. The final MLP model for the circuit in Figure 2 is presented in Figure 3. The objective function is the sum of total power consumption for each gate, given in the form of Equation (17), where $S(G)$ and $SV(G)$ are viewed as variables and determined by linearization constraints. The performance constraints relate gate arrival times to gate delays, and guarantee the performance target will be met. Similarly to $S(G)$ and $SV(G)$, the $SVR(U,G)$'s are also calculated using linearization constraints. Such MLP models can readily be imported to LP solvers such as *cplex* [11].

The exponential function base does not have to be two. For example, if the cell sizes are in a certain ratio s to one another, e.g. 1, s^2, s^3,..., we should use exponential functions with base s. Adding the constraints that $p(G)$'s are integers, we are able to obtain the exact solution using the MLP model. On the other hand, if there is no such relation between cell sizes, we can solve the MLP model with the $p(G)$'s being real variables, and use the smallest available cell size that is larger than the calculated value as the cell size assignment. The error caused by the cell size approximation is also controlled by carefully choosing the piecewise linear functions.

MLP-fast. We can solve the MLP model much faster if we are willing to pay some price in optimality. This is done using a two-step approach. First, we view the $V_t(G)$'s as real variables between 0 and 1 and solve the resulting linear programming model. Second, we fix the $p(G)$'s at the values obtained in the first step. The MLP model, with only V_t(G)'s as integer variables, is then solved as a new MLP problem. Intuitively, this MLP model has far fewer variables and so is much easier to solve. Our experimental data will justify this claim.

4. EXPERIMENTAL RESULTS

We performed experiments with the proposed methods using an extensive set of ISCAS and MCNC benchmark circuits. We used a small cell library with INV, AND2, NAND2, NAND3, NADN4, OR2, NOR2, NOR3, and NOR4. The maximum cell sizes for each cell type is 16X. The cell library uses 70nm CMOS technology, and was obtained using the Berkeley predictive technology model [3]. The higher V_t's for PMOS and NMOS are 220mV and 200mV, respectively. The lower ones for PMOS and NMOS are 190mV and 160mV, respectively. The low-Vt transistors have on-current which is ten times that of the high-Vt ones.

Runtime of *MLP-exact*. We first examine the runtime of *MLP-exact*. Usually cell sizes of powers of two are available in a library. We consider the special case where all cell sizes are powers of two. Consequently, we use base two in the exponential functions, and add the constraints that $p(G)$ are integers.

We linearize the exponential functions in question to guarantee accuracy when $p(G)$ are integers. Consider the linearization of $2^{p(G)}$ as an example. We use the following inequalities which are determined by $(k, 2^k)$ and $(k+1, 2^{k+1})$:

$$S(G) \geq 2^k p(G) + (1-k)2^k, \text{ where } k = 0, 2, 4, \ldots \quad (18)$$

Therefore, $S(G)$ is at least $2^{p(G)}$ when $p(G)$ is an integer. Furthermore, since we are trying to minimize the objective function (17), $S(G)$ will be assigned the minimum possible value. We can hence conclude that $S(G) = 2^{p(G)}$ in any valid solution. Other exponential functions can be linearized similarly to guarantee their accuracy. Hence, we do not sacrifice any optimality in linearizing the objective function and performance constraints, provided the $p(G)$'s are integers.

For each circuit, we first perform optimization using a TILOS-like sensitivity-based method (SBM) [6], and obtain the fastest design reachable with all low-Vt cells. We then set the target delay to 10% longer than the fastest design, and run *MLP-fast*. We stop the LP solver *cplex* after searching 10,000 nodes if no optimal solution is found. In fact, the solutions found are within 1% of the estimated optimal ones for all our MLP models. Since the LP solver spends the majority of its time improving the estimated lower bounds instead of improving the solution, the solutions obtained here are very close to, if not exactly, the optimal ones.

In addition to the power consumption of the designs obtained using *MLP-exact*, the corresponding runtime is shown in the MLP-exact column of Figure 4. Here we focus on runtime and leave the power reduction capability for later discussion. For most of the circuits, *MLP-exact* can produce optimal solutions quickly. The runtimes vary from 60 seconds (for c880) to 3,000 seconds (for c3540).

Circuits	MLP-exact		MLP-fast			
	Power P1 (mW)	Runtime T1 (sec)	Power P2 (mW)	Power ratio P1/P2	Runtime T2 (sec)	Speedup T1/T2
c432	0.15	297.90	0.16	0.95	2.01	148
c880	0.34	60.75	0.36	0.96	3.97	15
dalu	0.60	2708.51	0.60	1.00	15.33	177
rot	0.53	389.97	0.54	0.97	9.25	42
x3	0.90	1125.85	0.92	0.98	9.04	125
vda	0.41	1377.60	0.44	0.95	115.88	12
alu4	0.51	1950.03	0.53	0.98	199.90	10
apex6	0.98	1141.21	1.01	0.97	23.30	49
frg2	0.64	1405.73	0.64	1.00	31.55	45
I6	1.14	790.51	1.19	0.95	4.28	185
I7	0.92	1119.32	0.94	0.97	60.82	18
I10	0.80	445.03	0.80	1.00	81.56	5
c1908	0.35	1144.77	0.36	0.96	177.16	6
c2670	0.95	1630.37	0.96	0.99	110.90	15
c1355	0.46	1354.88	0.52	0.90	106.60	13
c3540	0.71	3072.73	0.72	0.98	238.99	13
c6288	N/A	N/A	1.98	N/A	1548.56	N/A
c7552	N/A	N/A	2.47	N/A	560.42	N/A
Average				**0.97**		**55**

Figure 4. Power consumption and runtime comparison between *MLP-exact* and *MLP-fast*

The runtime for larger circuits such as c6288 and c7552 is much longer due to the complexity of their MLP models.

Comparison of *MLP-exact* and *MLP-fast*. We examine the effectiveness of *MLP-fast* by comparing it with *MLP-exact* with the same performance constraints. The data for *MLP-fast* are presented in the MLP-fast column in Figure 4. *MLP-fast* achieves significant speedup over *MLP-exact*. In all except three of the benchmark circuits, *MLP-fast* is one or two orders of magnitude faster than *MLP-exact*. However, the designs from *MLP-fast* consume just 3% more power than those from *MLP-exact* on average. If the cell library provides more cell sizes in addition to powers of two, MLP-fast will provide even better solutions since it is able to choose cell sizes closer to their optimal values. We conclude that *MLP-fast* is a fast and accurate approximation to *MLP-exact*.

We next present results to show the effectiveness of the MLP approaches. Since *MLP-exact* and *MLP-fast* produces designs with very similar power consumption, we only compare *MLP-fast* with *SBM* and show its superiority in terms of power optimization capability.

Comparison of *MLP-fast* and *SBM*. We also re-ran *SBM* using all high-V_t and all low-V_t cells; we refer to these models as *SBM-H* and *SBM-L*, respectively. We collected the total power consumption of all resulting designs, and compared them with those of *MLP-fast*. In addition, we ran *SBM* assuming all-integer cell sizes of each type and all cells with low-V_t; this model is referred to as *SBM-I*. The experimental results are presented in Figure 5. For each design, we show the power consumption obtained using *SBM-L, MLP-fast, SBM-I,* and *SBM-H*. The first three designs meet the specified performance target while the latter one usually cannot. Therefore, the delays of designs of *SBM-H* are compared with their *SBM-L* counterparts.

A comparison between *SBM-L* and *MLP-fast* shows that *MLP-fast* usually generates designs requiring 20% to 50% less power than their *SBM-L* counterparts under the same performance constraints. On average, *MLP-fast* delivers 30% less power than SBM-L. This is not surprising since *MLP-fast* yields near-optimal designs, as shown earlier. The results also demonstrate that the all high-V_t designs usually are not able to achieve the same performance as the low-V_t designs by simply resizing the cells. They are on average 12% slower although consuming similar amounts of power as the *SBM-L* designs. In fact, most of the *SBM-H* designs consume the same amount of power as the *MLP-fast* designs. However, there are a few cases where the *SBM-H* designs consume much more power. This is because we have to oversize some gates to satisfy the timing requirement, resulting in high dynamic power.

Even when provided with a library having more cell sizes, *SBM* still produces designs with much higher power consumption than those from *MLP-fast*. In fact, the designs from *SBM-I* have slightly smaller power consumption that the *SBM-L* ones. As discussed

Circuits	SBM-L		MLP-fast		SBM-H				SBM-I	
	Delay D1 (ps)	Power P3 (mW)	Power P2 (mW)	Power ratio P2/P3	Delay D4 (ps)	Delay ratio D4/D1	Power P4 (mW)	Power ratio P4/P3	Power P5 (mW)	Power ratio P5/P3
c432	545.31	0.19	0.16	0.85	605.93	1.11	0.44	2.33	0.19	1.00
c880	449.42	0.49	0.36	0.72	497.07	1.11	0.34	0.68	0.49	0.99
dalu	474.75	0.98	0.60	0.61	534.24	1.13	0.56	0.57	0.95	0.96
rot	443.86	0.77	0.54	0.71	496.50	1.12	0.60	0.78	0.76	0.99
x3	221.24	1.13	0.92	0.81	254.87	1.15	0.80	0.71	1.12	0.99
vda	284.98	0.64	0.44	0.69	312.15	1.10	0.46	0.72	0.61	0.97
alu4	632.97	0.72	0.53	0.73	719.38	1.14	0.48	0.67	0.73	1.01
apex6	220.91	1.27	1.01	0.80	248.52	1.12	0.96	0.76	1.27	1.00
frg2	322.44	0.88	0.64	0.73	321.70	1.00	0.70	0.80	0.88	1.00
l6	125.98	1.35	1.19	0.88	143.59	1.14	1.21	0.89	1.32	0.97
l7	156.63	1.15	0.94	0.82	177.83	1.14	1.11	0.96	1.14	0.99
l10	906.54	1.52	0.80	0.52	1018.66	1.12	0.76	0.50	1.52	1.00
c1908	684.30	0.55	0.36	0.66	772.34	1.13	0.39	0.71	0.53	0.96
c2670	344.70	1.27	0.96	0.76	392.97	1.14	0.88	0.70	1.26	1.00
c1355	519.34	0.61	0.52	0.85	582.16	1.12	1.48	2.43	0.61	1.00
c3540	717.97	1.07	0.72	0.68	826.53	1.15	0.68	0.64	1.06	0.99
c6288	1291.00	3.16	1.98	0.63	1518.74	1.18	6.31	2.00	3.16	1.00
c7552	527.32	3.37	2.47	0.73	585.62	1.11	2.47	0.73	3.36	1.00
Average				**0.73**		**1.12**		**0.98**		**0.99**

Figure 5. Power and delay comparison of *SBM-L, MLP-fast, SBM-H* and *SBM-I*

previously, if provided with a library with more cell sizes, we expect *MLP-fast* to perform no worse, if not better than *MLP-fast* with power-of-two cell sizes. Consequently, when provided with a library having all-integer cell sizes, *MLP-fast* is also capable of obtaining designs with around 30% less power than *SBM*.

5. CONCLUSIONS

We have proposed an optimal MLP model (*MLP-exact*) for total power reduction during runtime under performance constraints. Unlike previous work where optimality can only be explored locally, a new approach to linearizing the delay function is proposed to enable exploration of the true global optima. An efficient way of finding near-optimal solutions (*MLP-fast*) is also proposed, which exhibits one or two orders of magnitude speedup for most benchmark circuits, with 3% power overhead. Compared with a sensitivity-based heuristic approach, the proposed methods can reduce the total power by almost 30%. The model can be readily extended to handle more threshold voltages and multiple V_{DD} levels.

Acknowledgement

This research was supported by National Science Foundation under Grant No. CCR-0073406.

6. REFERENCES

[1] S. Augsburger et al., "Reducing Power with Dual Supply, Dual Threshold and Transistor Sizing", *Proc. ICCD*, 2002.

[2] M. R. C. M. Berkelaar et al., "Gate Sizing in MOS Digital Circuits with Linear Programming", *Proc. DATE*, 1990.

[3] Berkeley Predictive Technology Model, http://www-device.eecs.berkeley.edu/~ptm/.

[4] R. Burch et al., "A Monte Carlo Approach for Power Estimation", *IEEE Trans. on VLSI*, 1993.

[5] A. Chatterjee et al., "An Investigation of the Impact of Technology Scaling on Power Wasted as Short-Circuit Current in Low Voltage Static CMOS", *Proc. ISLPED*, 1996.

[6] J. P. Fishburn, and A. E. Dunlop, "TILOS: A Posynomial Programming Approach to Transistor Sizing", *Proc. ICCAD*, 1985.

[7] F. Gao and J. P. Hayes, "Gate Sizing and Vt Assignment for Active-Mode Leakage Power Reduction", *Proc.* ICCD, 2004.

[8] A. Ghosh et al., "Estimation of Average Switching Activity in Combinational and Sequential Circuits", *Proc. DAC*, 1992.

[9] R. Ho et al., "The Future of Wires", *IEEE Proceedings*, 2001.

[10] W. Hung et al., "Total Power Optimization through Simultaneously Multiple-Vdd Multiple-Vth Assignment and Device Sizing with Stack Forcing", *Proc. ISLPED*, 2004.

[11] ILOG *cplex*. http://www.ilog.com/products/cplex/.

[12] D. Nguyen et al., "Minimization of Dynamic and Static Power Through Joint Assignment of Threshold Voltages and Sizing Optimization", *Proc. ISLPED*, 2003.

[13] P. Pant et al., "Dual-threshold Voltage Assignment with Transistor Sizing for Low Power CMOS Circuits", *IEEE Trans. on VLSI*, 2001.

[14] T. Pering, T. Burd, and R. Broderson, "Voltage Scheduling in the IpARM Micorprocessor System", *Proc. ISLPED*, 2000.

[15] A. Srivastava et al., "Concurrent Sizing, Vdd and Vth Assignment for Low-Power Design", *Proc. DATE*, 2004.

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