Total Power Optimization Combining Placement, Sizing and Multi-Vt Through Slack Distribution Management *

Tao Luo, David Newmark†, and David Z. Pan
Department of ECE, University of Texas at Austin, Austin, TX
† Advanced Micro Devices, Austin, TX
tluo@ece.utexas.edu, david.newmark@amd.com, dpan@ece.utexas.edu

ABSTRACT

Power dissipation is quickly becoming one of the most important limiters in nanometer IC design for leakage increases exponentially as the technology scaling down. However, power and timing are often conflicting objectives during optimization. In this paper, we propose a novel total power optimization flow under performance constraint. Instead of using placement, gate sizing, and multiple-Vt assignment techniques independently, we combine them together through the concept of slack distribution management to maximize the potential for power reduction. We propose to use the linear programming (LP) based placement and the geometric programming (GP) based gate sizing formulations to improve the slack distribution, which helps to maximize the total power reduction during the Vt-assignment stage. Our formulations include important practical design constraints, such as slew, noise and short circuit power, which were often ignored previously. We tested our algorithm on a set of industrial-strength manually optimized circuits from a multi-GHz 65nm microprocessor, and obtained very promising results. To our best knowledge, this is the first work that combines placement, gate sizing and Vt swapping systematically for total power (and in particular leakage) management.

1. INTRODUCTION

For nanometer IC designs (90nm and below), power dissipation has become one of the most important limiting factors since leakage is increasing exponentially as CMOS technology scaling down. Both process and design technologies are being developed to conquer the leakage barriers. Among various design techniques, multiple-Vt assignment is very popular and effective. The idea is fairly straightforward. For a design starts with all regular-Vt (R_{Vt}) cells. Once the timing target is roughly met, one replaces non-critical cells with their high-Vt (H_{vt}) counter parts, as the sub-threshold leakage current of a gate is exponentially related to the threshold voltage. Meanwhile, one needs to fix the remaining failing paths by using a small number of low-Vt (L_{vt}) cells since they are faster (but leak much more).

The effectiveness of Vt swapping relies on the slack distribution. The slack distribution is heavily related with how timing closure is done during physical synthesis, e.g., placement and gate sizing. As timing and power are often conflicting objectives during optimization, traditionally, placement is mainly used for timing optimization. There is no existing work in placement that considers the leakage power reduction.

Gate sizing is used for both timing optimization and power reduction. Conventional gate sizing formulations either minimize the worst case delay or minimize the power under delay constraints [1, 2, 3, 4, 5, 6, 7]. However, gate sizing is never considered to help the Vt-swapping algorithm to maximize the power reduction overall, although Vt-swapping is known to be much more effective on leakage reduction.

To maximize the power reduction, the power saving opportunity in above physical design stages should be considered and utilized in a systematic manner. As we know the leakage current is exponentially related to the threshold voltage (Table 1), but linear to the cell size, multi-Vt assignment shall be a much more effective technique for leakage power reduction than gate sizing (i.e. by using high-Vt cells as much as possible). In other words, to reduce total power where leakage becomes prominent, it is more effective to use placement and gate sizing to *promote more effective Vt swapping afterwards* than using them independently for local power reduction. For example, we may size up some cells, which leads to less L_{vt} cells used finally. In that case, the amount of leakage saved could be much more than the power increased due to cells upsized.

Table 1: Normalized delay and leakage current for a cell with different threshold voltages in 65nm technology

Cell	L_{vt}	R_{vt}	H_{vt}
Delay	1	1.1	1.3
Leakage current	17.3	2.4	1

In this paper, we propose to use the slack distribution management to "glue" placement and gate sizing algorithms together to *boost* the Vt-swapping technique. The primary objective of our approach is to increase the sum of slacks on critical and near critical paths, i.e. to push the slack distribution curve (not the worst slack) of the circuit away from critical, even at the cost of up-sizing some cells slightly. Less total number of critical cells implies less L_{vt} cells and higher percentage of H_{vt} cells being used eventually. In other words, we trade small dynamic power increase for large leakage power reduction. In addition, we reduce the power directly by sizing down cells on non-critical paths when possible. Our methodology formulates a linear-programming (LP) based placement and two geometric programming (GP) based gate sizing algorithms to change the slack distribution.

In the rest of the paper, section 2 motivates our proposed approach. The LP based placement stage is introduced in section 3. The GP formulations are in section 4 and the Vt swapping algorithm is described in section 5. Experimental results are reported in section 6, and we conclude in section 7.

2. MOTIVATION & PROPOSED APPROACH

In a typical flow, a design starts with all regular Vt cells (R_{vt}). A few timing violating paths that are very difficult to optimize in other ways can be fixed by swapping in L_{vt} cells. All R_{vt} cells on non-critical paths with large slack will be swapped into H_{vt} cells to save power. As shown in Table 1, the leakage of a H_{vt} cell is significantly smaller, about 17 times compared with a L_{vt} version at 65-nm technology.

The results of Vt swapping is highly dependent on the slack distributions. If we can reduce the number of near critical cells, we may use fewer L_{vt} cells and more H_{vt} cells. Figure 1 plots the cell slack histogram of a circuit before and after placement plus gate

*This work is supported in part by NSF, SRC, and IBM Faculty Award.

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

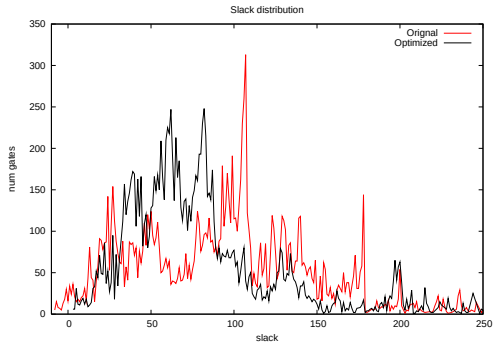

Figure 1: Slack distribution before and after optimization

sizing. The circuit is R_{vt} based. The slack histogram after optimization is tightened around a specified mean with a reduced deviation. Less near-critical cells implies less L_{vt} cells be used later, and less leakage power subsequently.

2.1 The proposed flow

Our strategy is to use the placement and gate sizing to optimize the slack distribution to promote Vt swapping. We formulate a LP program for placement and a GP program for gate sizing to maximize the sum of slacks on semi-critical paths. In addition, cells on non-critical paths may be oversized even if they are swapped to H_{vt}. We formulate a GP problem to reduce slack and power on non-critical cells directly.

Algorithm 1 The Overall Algorithm

```
1:  The slack distribution management algorithm
2:      Input: initial design (all R_vt cells)
3:      while ( less than max. iter. & improved) do
4:          Incremental placement optimization
5:          TimingAnalysis
6:          Cell sizing on critical path for slack
7:          TimingAnalysis
8:          Size down non-critical cells
9:          TimingAnalysis
10: The Vt-swapping algorithm (Algorithm 2)
11: Functon: TimingAnalysis
12:     Pre-routing, and timing analysis
13:     if(improved) accept solution, annotate the database
```

We use placement and gate sizing iteratively to improve the slack distribution. Algorithm 1 shows our proposed flow. Starting from a design, we do the placement and critical cell gate sizing iteratively until no further improvement. Finally, we employee the Vt swapping to use a few L_{vt} cells to fix the remaining critical paths, and replace as many R_{vt} with H_{vt} cells as possible. At the end of each stage in the flow, we run a fairly accurate timing analyzer. The timing tool pre-routes the circuit, extract the parasitics, and run the PrimeTime based timing analysis. The timing change from the previous stage is accurately updated and annotated back into the design databases, as the basis of the next stage.

2.2 Practical design constraints

In existing literature of power optimization, important practical design constraints, such as slew, noise, and short-circuit power, are often not considered, which makes the optimization algorithm impractical for realistic designs. For example, short circuit power is usually assumed small and ignored in most of existing power reduction algorithms. However, short circuit power may rise significantly if not explicitly controlled in the optimization framework.

2.2.1 *The slew and noise related constraints*

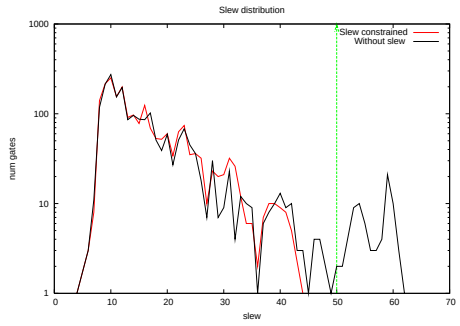

Figure 2: Slew rate distribution with and without explicit control

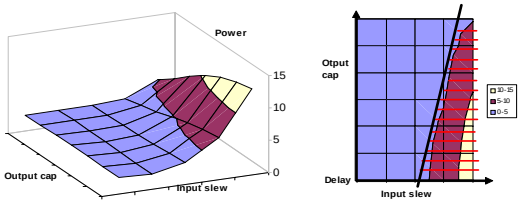

Figure 3: A simple yet effective short circuit power constraint model.

Without restricting the maximum slew rate, cells on short paths will be over-downsized. Figure 2 plots the slew rate histogram of the gate sizing results with and without restricting the slew rate. in Figure 2, a lot of instances violate the 50 pico-second slew limit if ignoring slew constraints. Our maximum slew rate constraints set an upper bound for the slew rate. Furthermore, cells have different sensitivities to slew for noise. Our model includes the effective fan-out constraints, which is an effective way to reduce the noise related issues.

2.2.2 *Short circuit power constraint*

Short circuit power is difficult to model and ignored in most existing power optimization algorithms. The inputs for internal power are the input slew and output capacitance, as shown in Figure 3. Imaging a scenario that the input slew of a cell is large and the load can be charged full quickly, the PMOS and NMOS will be both on for a longer period. The Vdd to ground current will consume a lot of power. Short circuit power is often assumed very small. However, in our experiments, it is comparable to leakage if not properly handled. Figure 3 is a SPICE simulation based look-up table to interpolate the short circuit power. Note that short circuit power could rise dramatically if the ratio of the input slew and output capacitive load falls into a certain range, e.g. input slew is large while the cell is driving a comparably small load.

In later sections, we will show how to handle above important design constraints in our proposed algorithm.

3. LP BASED PLACEMENT FOR POWER

The objective of our LP placement formulation is to reduce the power in Vt-swapping stage incrementally. Therefore, instead of reducing the worst case delay, our LP based placement is formulated primarily to reduce the total number of critical and semi-critical cells, i.e. to push the slack curve away from the critical point, which helps the Vt-swapping tool on leakage reduction.

Linear programming is commonly used for incremental timing driven placement [5, 8, 9, 10, 11]. In LP based incremental placement, a few critical paths are selected by a sign off timer, and crit-

ical paths are optimized incrementally. Existing LP based timing driven placement algorithms use the half parameter wire length for wire estimation, as HPWL can be formulated exactly in a LP framework.

Chen et.al. [5] proposed a simultaneous placement and gate sizing approach to optimize the delay. Because the unified placement and sizing GP formulation is not convex, the problem was formulated into a generic geometric program (GGP) and solve iteratively. However, the HPWL based wire load estimation is much less accurate compared with that in a stand alone gate sizing problem, which can be measured separately. A simultaneous formulation will make the wire load estimation less accurate for gate sizing, which often results in a sub-optimal solution.

3.1 The LP formulations

We assume the following gate delay DP_i and transition SP_i models for cell i

$$DP_i = dp_I + a1_i \cdot Slew_i + a2_i \cdot Cap_i \quad (1)$$

$$SP_i = sp_I + u1_i \cdot Slew_i + u2_i \cdot Cap_i \quad (2)$$

where $a1_i$, $a2_i$, $u1_i$, and $u2_i$ are the fitting coefficients. dp_I and sp_I denote the intrinsic delay and slew of the corresponding pin of the cell. $Slew_i$ denotes the input slew. Let $HPWL_j$ denotes the HPWL of net j, and Cap_j represents the capacitive load of the driver of net j. We have

$$Cap_j = c \cdot HPWL_j + Cpin_j$$

which is the sum of the wire capacitance $cHPWL_j$ plus the total pin capacitance driven by net j. c is the unit capacitance.

The LP placement algorithm selects a few critical paths selected from the timing report, which have slacks less than a threshold. The net delay sensitivity is computed for each critical net, and a LP program is formulated to minimize the sum of the weighted critical nets, which is an indirect method to increase the sum of total slack on those critical paths. The net delay sensitivity Sn_j is based on the delay propagation sensitivity computation in [11]

$$Sn_j = c \cdot (a2_i + a1_{i+1} \cdot u2_i)$$

Elmore delay [12] is used for wire delay modeling and the symbols related with net delay are omitted in the formulation for simplicity.

Similar to [13], the critical paths were counted to compute the criticality of each selected critical net, the criticality score of net j is denoted by Sc_j. Therefore, the combined timing weight $wt_t = Sc_j Sn_j$. The dynamic power is a function of the load capacitance of the net. If cell i drives net j, we have a power weight

$$wt_p = 0.5\alpha_i \cdot F \cdot V^2 \quad (3)$$

where α_i denotes the switching rate, F is the frequency, and V is the voltage. A control parameter β is used to adjust the ratio between the timing and power weight.

$$wt_j = \beta wt_p + (1-\beta) wt_t$$

β is a value between 0 and 1. The primary objective of our LP placement is to reduce the leakage power, thus, β is set to a relatively small value. A LP program is formulated to minimize the sum of the weighted critical nets, which indirectly increases the sum of pin slacks.

$$min \quad \sum wt_j L_j$$
$$\forall j \in Selected \quad critical \quad nets$$

The residual overlap created in this stage is carefully removed.

4. GP BASED GATE SIZING FOR POWER

Placement has a limited impacton slack distribution improvement if the cell sizes are not changed. To push the slack curve further, we use the effort based delay model, and formulate a *Geometric programming* based gate sizing problem. GP is a special type of the non-linear optimization problem that has been used for gate sizing since the 80s [14, 15, 16]. The standard GP problem has a posynomial objective and special format constraints. In last ten years, the solving efficiency of GP is approaching that of *Linear Programming*. We refer the reader to a tutorial for geometric programming [16].

Conventional gate sizing formulations minimize the worst case delay in a circuit with power or area constraints [2, 16, 5], or minimize the power directly under the delay constraints [7]. On the contrary, our first GP formulation increases the sum of slack on critical and near critical outputs instead. Our second GP program is related to the conventional formulation, which focuses on the non critical part of the circuit to "absorb" large slacks. Therefore, we treat cells differently depending on the criticality of the cell.

4.1 Cell classification

Cells are classified into two sets, the non-critical set *NC* and the critical set *CRIT*, based on the output pin slack. If the pin slack is larger than a threshold, we add the cell into *NC*. Similarly, if the slack is small enough, we add the cell into *CRIT*.

For the first GP program, we start from all outputs with slack less than δ, for example, $\delta = 70$ ps. We traverse the circuit in a reversed breath first order, and the reversed BFS traversal proceeds only on cell outputs with the slack smaller than $\delta + \gamma$ and stops at signal inputs, which are the inputs of the circuit or the outputs of sequential cells. Only cells with slack less than θ ($\theta < \delta$) are selected into the *CRIT*. The size of cells in *CRIT* are variables for the GP program. As the arrival time of all outputs with slack less than δ is controlled in the GP program, $\delta - \theta$ acts as a guard band to ensure that timing on other outputs are not disturbed too much. For the second GP program for non-critical cells, all cells with a slack larger than a threshold are sizable cells in *NC*, and all outputs are included into the GP problem. In other words, all arrival times are controlled.

4.2 The GP models

We model the gate as a resistor and a switch that drives a RC network. The gate delay and transition are the functions of the gate size W and the total capacitive load, Cap. The equation for the cell equivalent impedance is different for delay and transition equations, and the delay models for each pin of a gate and that for the falling or rising transition are different. We use the worst case models for a cell. The gate delay Dg_i and slew Sg_i are given by

$$Dg_i = dg_I + (h_i/W_i) \cdot Cap_i \quad (4)$$

Slew is not propagated. But slew is monitored and restricted by the following equation

$$Sg_i = sg_I + (v_i/W_i) \cdot Cap_i \quad (5)$$

Cap is the sum of the capacitive load and the gate capacitance a cell drives. The pin capacitance of a cell i is a linear function of the cell size W_i.

$$Cp_i = e_i + f_i W_i \quad (6)$$

In above equations, dg_I, h_i, sg_I, v_i, e_i, and f_i are all fitting coefficients to the cell library.

Assuming a cell i drives a sizable cells and b non-sizable cells. The total capacitance the cell i drives is

$$Cap_i = \sum_{k=1}^{a} (e_k + f_k W_k) + \sum_{l=1}^{b} (Cp_l) + Cap_{wire} \quad (7)$$

We add the wire delay in our formulation. An accurate pre-routing tool is used to estimate routs. The pin to pin wire delay is computed by a static timer and treated as a constant in the gate sizing formulation.

Three major source of power consumption, including dynamic, short circuit, and leakage power are considered in our approach. The dynamic power can be written as

$$P_i = 0.5\alpha \cdot F \cdot V^2 \cdot Cap_i \tag{8}$$

where α, F, and V are defined in equation (3). The leakage power is assumed proportional to the gate size, and the parameter $leak$ is extracted from the SPICE simulation based power library. The following linear leakage model is sufficient for the leakage estimation in the gate sizing stage.

$$L_i = leak_i \cdot W_i \tag{9}$$

The short circuit power is modeled as constraints in the GP formulation.

4.3 Gate sizing effectiveness analysis

Slack and power optimization are often contradictory objectives. To reduce the delay by sizing up cells will increase the dynamic and the leakage power. Whether or not and how to size a cell should be also determined by if such a chance has negative overall effect potentially. We do the following gate sizing effectiveness analysis to estimate a sizing range, i.e. we do not size a cell exceeding a limit that may have a negative effect. In the following, we will

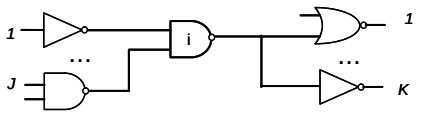

Figure 4: Gate sizing effectiveness analysis

derive the power and delay sensitivity to cell size. In Figure 4, the cell i has J inputs and drives K downstream cells. If we change cell i from R_{vt} to L_{vt}, the associated power will change by ΔPv_i, and delay will change by ΔDv_i. We have

$$\Delta Pv_i = \Delta Leak_i + \alpha_i FV^2 \sum_j f_j \Delta Cp_i, j = 1..J$$

$$\Delta Dv_i = ((h_i^{'} - h_i)/W_i)(\sum_k Cp_k + Cw) + max(\frac{h_j}{W_j}\Delta Cp_i), k = 1..K$$

Where ΔCp_i is the pin capacitance change. $h_i^{'}$ is the corresponding coefficient for L_{vt} cell, as in equation (4). Similarly, if changing the cell size by ΔW_i, the associated power change is denoted by ΔPg_i,

$$\Delta Pg_i = Leak_i \Delta W_i + \alpha_i FV^2 \sum_j h_j f_j \Delta W_i$$

The associated delay change is denoted by Dg_i

$$\Delta Dg_i = (\frac{h_i}{W_i + \Delta W_i} - \frac{h_i}{W_i}) \sum_k (Cp_k + Cw) + max(\frac{h_j}{W_j}\Delta W_i)$$

To solve the equation $\Delta Pv_i/\Delta Dv_i = \Delta Pg_i/\Delta Dg_i$, one zero and one non-zero ΔW_i solutions are generated. If the non-zero solution is negative, sizing up cell i will increase both power and delay, and the cell i is not allowed to be sized up. If one of the solutions is positive, and $W_i + \Delta W_i < Wmax_i$, we set the maximum sizable range of cell i as $Wmax_i = W_i + \Delta W_i$. Beyond this limit, gate sizing has a lower power and delay benefit compared to Vt swapping.

4.4 GP for near-critical cells

In brief, our first GP program creates more slacks for near critical cells, which maximizes the sum of slacks on critical outputs. The GP formulation is given by

$$\begin{aligned} minimize & \sum_j AT_j \\ & + \sum_i wt_i W_i (\Delta Pg_i/\Delta W_i), j \in PO \cap CRIT, i \in CRIT \end{aligned} \tag{10}$$

$$\begin{aligned} s.t.\ \ & Dg_i = d_I + \frac{b_i}{W_i} Cp_i \\ & AT_i \geq Dg_i + max(AT_{i-1}^p), p \in input\ pins\ of\ i \\ & AT_i = T_{start}, \forall i \in PI \\ & W_i \geq Wmin_i, W_i \leq Wmax_i \end{aligned}$$

In above, AT_i is the arrival time at the output of the $cell_i$. $\Delta Pg_i/\Delta W_i$ is the power sensitivity to cell sizes

$$\Delta Pg_i/\Delta W_i = Leak_i + \alpha_i FV^2 \sum_j h_j f_j \tag{11}$$

T_{slack} is a slack threshold. In the above GP formulation, we optimize the sum of the arrival time of all critical and near critical outputs. $Wmax_i$ and $Wmin_i$ are the sizing range for cell i. Wire delay is not shown for clarity, which is a constant computed by a static timer conjuncted with a pre-routing tool.

Let wt_i denotes the power weight. Without the power objective $\sum_i wt_i W_i$, the cell could be overly unsized, which will cause unnecessary increase on power. Before the optimizations, the sum of the arrival time on critical outputs and the sum of dynamic and leakage power on cells in $CRIT$ are evaluated. The power weight is computed to normalize the arrival time and power objectives, and the power weight is set to be associated with the power sensitivity of each cell. A 0 to 1 parameter is set to adjust the ratio between the arrival time and power objects.

4.5 GP for non-critical cells

The GP for non-critical cells is to optimize the total power on high slack cells, such that the arrival time does not violate timing constraints. The GP problem for non-critical cells can be written as

$$\begin{aligned} min. & \sum_i (\Delta Pg_i/\Delta W_i), i \in NC \\ s.t.\ \ & AT_i \leq max((T_{cycle} - T_{threshold}), AT\_orig_i), i \in PO \end{aligned} \tag{12}$$

where $\Delta Pg_i/\Delta W_i$ is from equation (11). $T_{threshold}$ is the slack guard band. We consider swapping non-critical cells with slack larger than $T_{threshold}$ to H_{vt} cells. AT_orig_i is the original arrival time of the output i. Constraint (12) implies that for each output i, the arrival time after the optimization may not violate the larger of a delay threshold and its original delay. The shared constraints in above GP problems are not shown here for simplicity.

4.6 Modeling important constraints

Besides the delay and power, there are a few constraints that are critical for industry practices, for example, the maximum slew constraint, the effective fan-out constraint for noise, and the short circuit power constraints, which were often ignored in previous studies. Our formulation considers those constraints and model them as follows in the GP framework.

4.6.1 The max slew constraint

Although adding the slew constraints will significantly limit the amount of power reducible, we should not ignore the slew constraints because slew rate violations are unacceptable for real world designs. The slew equation in (5) is used to estimate the slew rate, and we use the following to transform the slew constraint into sizing constraint in GP form.

$$\begin{aligned} S_i &= s_I + \frac{v_i}{W_i} Cp_i \\ S_i &\leq Slew_{max} \end{aligned} \tag{13}$$

where $Slew_{max}$ is the maximum slew rate acceptable.

4.6.2 Effective fan-out constraint for noise tolerance

The concept of effective fan-out (Efo) is related to but different from the conventional fan-out concept. Efo is the ratio of the effective capacitance a cell drives divided by the effect impedance ratio of the driver compared to a standard inverter. The effective impedance *Ratio* is the hold resistance of a cell divided by that of a standard inverter at a certain voltage level. The Efo constraint is given by

$$Efo_i = \frac{Cp_i}{Ratio_i \times C_{inv1}} \leq Efo_{limit} \quad (14)$$

Applying an effective fan-out constraint on each cell will avoid introducing large amount of noise issues during the optimization.

4.6.3 Short circuit power constraint

The short circuit power is non-trivial to handle, and mostly ignored in previous power optimization work. Since the short circuit power is not large unless the ratio of the input slew and output capacitive load falls into a certain range, as shown in Figure 3, we can specify a *do-not-enter* region by adding a linear constraint to restrict the ratio between the input slew and output capacitance to avoid large short circuit power consumption.

$$Cap_i \geq p_i + q_i S_i \quad (15)$$

where Cap_i is the capacitive load driven by cell i. p_i and q_i are the parameter of the linear function shown in Figure 3, which specify the boundary of the do-not-enter zone. Above constraint ensures that the input slew of a cell should not be much larger than its output slew.

The number of possible sizes for a gate varies depending on the gate type. An inverter could have over 20 different sizes. Our algorithm assumes the sizes are continuous. The solution of the GP solver are continuous gate sizes, which will be mapped into the closest discrete ones. The discrete size mapping stage may introduce less than 5 percent errors.

Algorithm 2 The Vt-swapping algorithm

```
1:  Input The design after placement and sizing opt.
2:  while (stopping criteria not meet) do
3:      foreach (all cells)
4:          if (slack > High) swap to H_vt
5:          if (slack < Low) swap to L_vt
6:      end
7:      TimingAnalysis
8:      Sort cells on Senstivity (critical and noncritical list)
9:      foreach (Sorted cells)
10:         Swap to L_vt or R_vt
11:         Propagate timing and evaluate
12:     end
13: end
```

5. VT SWAPPING ALGORITHM

We use a multiple pass sensitivity based Vt swapping algorithm, as shown in Algorithm 2 to swap cells. Cells with very large or small slacks are processed first. The rest are sorted on their sensitivity score. In each swapping pass, two hashes are created, one for R_{vt} cells and the other for L_{vt} cells. The sensitivity of a cell is computed by the original slack of the cell, the up-cone impact and the down cone impact of the cell. One top cell is selected at a time. The internal timer propagates the timing changes down stream and upstream to update the required times and the slacks. The process continues until the slack requirement is met. The swapping process will be performed multiple times for different supply voltages and performance corners. A solution that satisfies all corners will be adopted.

6. EXPERIMENTAL RESULTS

The placement and gate sizing algorithm are implemented in C++ and the Vt swapping algorithm is in perl. We use the commercial tool MOSEK [17] as the GP solver. Several modules from a multi-GHz micro-processor in 65nm process technology are used for experiments. The number of cells and nets are shown in table 2, which are typical in micro-processor designs. The circuits have been initially manually placed and timing optimized and taped out in a test chip. It is to be noted that the high performance microprocessor circuits have a stringent timing target and are very difficult for timing optimization. Therefore, the multi-Vt swapping technique has to be used to repair the remaining failing paths, in most of cases. All experiments are tested on a 2.4GHz 64-bit Opteron Linux server. We use an internal power evaluation tool to estimate the power consumption.

Table 2 shows the total power comparisons. Table 3 and 4 report the comparisons of leakage power and dynamic power respectively. In all tables, column *Base* shows the base-line optimization condition where cells are mostly R_{vt} cells. Column VT shows the power after the Vt swapping, and *BASE* stands for the baseline. *PV* shows the combined placement and Vt swapping, and column *PGV* stands for the combined placement, gate sizing and Vt swapping flow. We can see that the Vt swapping is very effective in reducing leakage power. The combined LP based placement and GP based gate sizing algorithm provides additional improvement and the flexibility to trade off on dynamic and static power through optimizing the slack distribution. We observe an additional 7.9% total power reduction, which is significant for manually optimized custom circuits. In current configurations, the placement optimization is configured to mostly help leakage power. The combined placement, gate sizing and Vt swapping gives the best results and helps to reduce 63.8% of leakage power and 32.9% of total power consumption.

Table 3: Leakage power comparison

	Base	VT	PGV	VT|*Base %*	*PGV*|*Base %*
ckt1	10.50	6.09	3.28	42.0	68.8
ckt2	11.49	4.79	3.67	58.3	68.1
ckt3	52.11	20.10	17.38	61.4	66.6
ckt4	45.76	30.42	28.06	33.5	38.7
ckt5	93.04	26.62	18.28	71.4	80.4
ckt6	99.29	24.78	19.64	75.0	80.2
ckt7	104.77	60.25	49.86	42.5	52.4
ckt8	215.24	108.46	96.28	49.6	55.3
				54.2	63.8

Table 4: Dynamic power comparison

	Base	VT	PGV	VT|*Base %*	*PGV*|*Base%*
ckt1	19.29	18.51	17.24	4.0	10.6
ckt2	18.77	17.62	16.02	6.1	14.7
ckt3	90.40	83.35	78.73	7.8	12.9
ckt4	65.10	63.15	58.32	3.0	10.4
ckt5	140.52	125.19	105.12	10.9	25.2
ckt6	141.98	131.11	120.79	7.7	14.9
ckt7	182.45	173.38	167.28	5.0	8.3
ckt8	283.91	268.88	258.30	5.3	9.0
				6.2	13.3

The break down of runtime is shown in table 5. Column *Timing* reports the runtime of the static timing analysis flow. Our sophisticated timing analysis flow pre-routes the circuit, extracts parasitics and run a PrimeTime engine to generate the timing report and annotates the timing information into the design database. We run the timing analysis at the end of every optimization stage to update the timing information. Therefore, multiple runs of the timing analysis flow took a lot of runtime.

Table 2: Total power comparison

	65 nm		Total power (mw)				Improvement %		
	Gates	Nets	Base	VT	PV	PGV	*VT|Base*	*PV|Base*	*PGV|Base*
ckt1	1765	2360	29.79	24.60	22.06	20.52	17.4	25.9	31.1
ckt2	2334	2881	30.26	22.41	21.93	19.69	25.9	27.5	34.9
ckt3	6640	8644	142.51	103.45	101.31	96.11	27.4	28.9	32.6
ckt4	9254	7928	110.86	93.57	93.57	86.38	15.6	15.6	22.1
ckt5	9541	9539	233.56	151.81	147.23	123.40	35.0	37.0	47.2
ckt6	12716	14042	241.27	155.89	154.14	140.43	35.4	36.1	41.8
ckt7	15486	18360	287.22	233.63	226.96	217.14	18.7	21.0	24.4
ckt8	27103	26991	499.15	377.34	372.51	354.58	24.4	25.4	29.0
							25.0	27.2	32.9

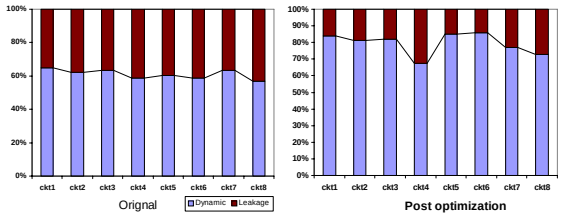

Figure 5: The leakage and dynamic power break up before and after optimizations

The break down of dynamic and leakage power in circuits before and after optimization is reported in Figure 5. We observe significant leakage reduction after applying our power optimization algorithm. The leakage reduction is due to higher percentage of H_{vt} cell used after the power optimization. The percentage of different Vt cells in all circuits are illustrated in figure 6. Originally a few circuits have negative slacks, and all circuits meet the timing target after the optimization. More L_{vt} cells is used in circuit 4 to close timing, and the percentage of L_{vt} cells is relatively small for the rest of testcases. Therefore, leakage power is reduced significantly.

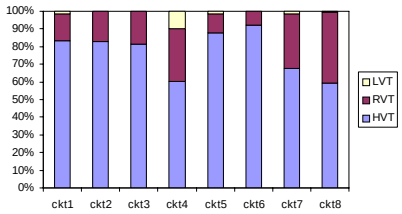

Figure 6: The percentage of different threshold voltage cells

Table 5: Runtime breakup(s)

Testcases	*Place*	*Size*	*Swap*	*Timing*
ckt1	4	14	55	168
ckt2	3	12	69	90
ckt3	30	75	124	453
ckt4	25	42	137	217
ckt5	26	68	149	324
ckt6	32	228	171	262
ckt7	16	193	186	342
ckt8	43	384	245	538

7. CONCLUSION

In this paper, we propose the methodology to combine placement, gate sizing, and multiple-Vt cell swapping algorithms for leakage and total power optimization. Our unified slack distribution management strategy makes it possible to combine different technique, the placement and gate sizing, to maximize the power reduction, while satisfying timing constraints. Our placement and gate sizing problems are formulated to optimize the slack distribution, which in turn transformed into power reduction through Vt-swapping. Our approach treats cells differently depending on their timing criticality. On a set of timing-closed multi-gHz 65nm custom microprocessor circuits, our approach reduced the leakage power by 63.8% and the overall power by 32.9%. Various practical design constraints are incorporated in our approach which were not considered before. Since power is becoming one of the most important design objectives, we believe there is a lot of room for future research on the total power reduction.

8. REFERENCES

[1] J. P. Fishburn and A. E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *Proc. ICCAD*, Nov. 1985.

[2] M. Berkelaar and J. Jess, "Gate sizing in MOS digital circuits with linear programming," in *Proc. European Design Automation Conf.*, pp. 217–221, June 1990.

[3] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, and S. M. Kang, "An exact solution to the transistor sizing problem for CMOS circuits using convex optimization," *IEEE TCAD*, 1993.

[4] G. Chen, H. Onodera, and K. Tamaru, "An iterative gate sizing approach with accurate delay evaluation," in *Proc. ICCAD*, pp. 422–427, Nov. 1995.

[5] W. Chen, C. Hseih, and M. Pedram, "Simultaneous gate sizing and placement," in *IEEE TCAD*, pp. 206–214, Feb. 2000.

[6] W. Chuang and S. S. Sapatnekar, "Power vs. delay in gate sizing: Conflicting objectives?," in *Proc. ICCAD*, pp. 463–466, Nov. 1995.

[7] S. S. Sapatnekar and W. Chuang, "Power-delay optimizations in gate sizing," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 5, no. 1, pp. 98–114, 2000.

[8] B. Halpin, C. Y. R. Chen, and N. Sehgal, "Timing driven placement using physical net constraints," in *Proc. DAC*, pp. 780–783, 2001.

[9] W. Choi and K. Bazargan, "Incremental placement for timing optimization," in *Proc. ICCAD*, p. 463, 2003.

[10] A. Chowdhary and et. al., "How accurately can we model timing in a placement engine?," in *Proc. DAC*, pp. 801–806, 2005.

[11] T. Luo, D. Newmark, and D. Z. Pan, "A new lp based incremental timing driven placement for high performance designs," in *Proc. DAC*, 2006.

[12] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *Journal of Applied Physics*, vol. 19, pp. 55–63, Jan. 1948.

[13] T. Kong, "A novel net weighting algorithm for timing-driven placement," in *Proc. ICCAD*, pp. 172–176, 2002.

[14] C. Chu and D. Wong, "Vlsi circuit performance optimization by geometric programming," in *Annals of Operations Research*, pp. 105:37–60, 2001.

[15] S. P. Boyd and S. J. Kim, "Geometric programming for circuit optimization," in *Proc. ISPD*, 2005.

[16] S. P. Boyd, S. J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Technical Report, ISL, Electrical Engineering Department, Stanford University*, 2004.

[17] MOSEK, "http://www.mosek.com," 2005.