Bounded Potential Slack: Enabling Time Budgeting for Dual-Vt Allocation of Hierarchical Design

Jun Seomun, Seungwhun Paik, and Youngsoo Shin
Department of Electrical Engineering, KAIST
Daejeon 305-701, Korea

***Abstract*— Time budgeting, which assigns timing assertion at block boundary, is a crucial step in hierarchical design. The proportion of high- and low-V_t gates of each block, which determines overall leakage power consumption, is dictated by timing assertion, yet dual-V_t allocation is not taken into account during conventional time budgeting. Bounded potential slack is introduced as a measure of dual-V_t allocation, and is experimentally shown to be strongly correlated with the percentage of high-V_t gates. A new time budgeting is proposed with objective of achieving bounded potential slack, which is formulated as a linear programming problem. In experiments with example hierarchical designs implemented in 45-nm commercial technology, the proposed time budgeting reduced leakage power by 32% on average compared to conventional time budgeting, when both are followed by the same dual-V_t allocation. The time budgeting is also applied to voltage island design, where each block can have its own V_{dd} with mix of high- and low-V_t gates.**

I. Introduction

Logical hierarchy is always used in VLSI design to cope with complexity and to facilitate reuse. It, however, is usually removed during automatic logic- and physical-synthesis, i.e. synthesis is done on flat design. This is because synthesizing each hierarchical block one by one, rather than synthesizing one whole design, inherently yields inferior result. Keeping hierarchy also requires additional design steps such as port assignment, wiring resource assignment, and time budgeting, which increases overall design time.

In spite of aforementioned disadvantages, hierarchical approach is used in some designs, mostly in very large and complex designs. Microprocessors are prime examples; in fact, they heavily rely on hierarchical approach [1] because each block or unit is developed by independent group of designers. System-on-a-Chip (SoC) also relies on hierarchical approach [2], because some cores are designed and characterized a priori in terms of timing or physical design. Hierarchical approach is also used in large ASIC designs to reduce run time for physical design [3] or for faster timing closure [4], [5].

The key design step in hierarchical approach is *time budgeting*, which generates timing assertion for each hierarchical block from given chip-level timing constraints. Timing assertion consists of arrival time (AT) at each block input and required arrival time (RAT) at each block output. AT is associated with limit on signal transition time; RAT is associated with limit on load capacitance. Time budgeting is critical for quality of overall design since it dictates the quality of each block in area, delay, power, and so on.

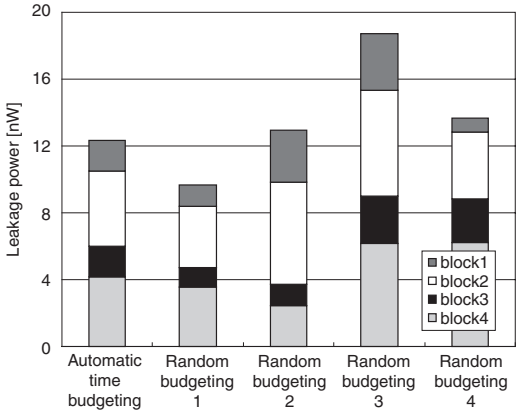

Fig. 1. Leakage power in each hierarchical block of `ps2` after dual-V_t allocation, with various time budgeting.

Time budgeting is performed by timing experts based on their experience, intuition, and knowledge of chip-level timing constraints [3]. It alternatively can be performed by automatic time budgeter [6], which relies on rough estimate of combinational delay (along timing path) within each hierarchical block. In hierarchical design, the use of dual-V_t, which is commonly deployed to manage leakage current, is typically planned on an ad hoc manner. For example, a maximum number of low-V_t gates that can be used is derived from requirement on standby current; it is distributed to each block based on block size, tightness on timing, and so on; each block is then designed with its timing assertion and the limit on the number of low-V_t gates. In this approach, timing assertion and maximum number of low-V_t gates are determined independently, even though actual number of low-V_t is strongly dependent on timing assertion.

A. Motivational Example

We consider `ps2`, which is taken from [7], to see the quantitative effect of time budgeting on dual-V_t allocation. It consists of 4 hierarchical blocks with 2255 combinational gates and 254 flip-flops in total, after mapping [8] to a commercial 45-nm gate library; all the gates are initially mapped to low-V_t. Automatic time budgeting [6] is performed, which is followed by block-by-block dual-V_t allocation. The leftmost bar in Fig. 1 shows leakage power with contribution from each block also identified.

978-1-4244-5766-3/10/$26.00 ©2010 IEEE

We then randomly generate timing assertions, out of which we pick only those that do not fail in timing analysis. For each set of timing assertions, dual-V_t allocation is performed and leakage power is obtained. The result is shown in Fig. 1 for four sets of timing assertions. It is clear that conventional time budgeting does not lead to minimum leakage power; the second bar from the left has 21.5% less leakage. There is also a large variation of leakage power depending on how timing assertions are generated; the fourth bar from the left, for example, has 52.0% more leakage than the leftmost bar.

B. Proposed Approach

We address a problem of time budgeting such that, after block-by-block dual-V_t allocation, total leakage power is minimized. Since dual-V_t allocation itself is a non-trivial process, our goal can be achieved only by introducing a measure that can predict dual-V_t allocation, which we can use during budgeting process. Bounded potential slack (BPS) is proposed for this purpose, which is shown to have a strong correlation with the percentage of high-V_t gates after dual-V_t allocation (Section III). Time budgeting using BPS is formulated as linear programming; experiments with commercial 45-nm technology demonstrate that leakage power is cut by 32% on average compared to conventional time budgeting, which is followed by the same dual-V_t allocation (Section IV). BPS-based time budgeting is applied to voltage island design, where each block can have its own V_{dd} and a mix of high- and low-V_t gates (Section V).

II. Preliminaries

A. Dual-Vt Allocation

There are many approaches to dual-V_t allocation. All possible allocations may be enumerated to search for the allocation with minimum leakage [9]; this approach, however, requires a circuit to be partitioned into a set of trees and run time is excessively high. Dual-V_t allocation can be transformed into slack assignment problem [10], i.e. assign timing slacks so that maximum number of gates can receive high-V_t; run time of this approach is also very high. The most popular approach is to use a function that heuristically determines the order of allocation [11], [12], [13]. Typical function is sensitivity, which is the change of leakage current divided by the change in timing of a circuit that would be caused by assigning high-V_t to a particular gate.

In this paper, we also use sensitivity-based dual-V_t allocation, except that flip-flops are handled differently. In particular, master and slave portions of each flip-flop are assumed to receive high- and low-V_t independently, leading to four choices of a flip-flop. Each choice is appropriately designed and prepared in a gate library.

B. Time Budgeting

Fig. 2(a) shows an example hierarchical design consisting of three blocks. A single flip-flop to flip-flop timing path spans all three blocks, with delay within each block shown in the figure. Let clock period be 10. Since the delay of timing path is 8 (delays of flip-flops and inter-block wires are ignored for simplicity of presentation), the path has timing slack of 2. In conventional time budgeting [6], this slack is distributed to each block in proportion to its own delay along the timing path. The arrival time of net n_1 is set[1] to $10 \times \frac{3}{8} = 3.75$. Similarly, the arrival time of net n_2 is set to $3.75 + 10 \times \frac{4}{8} = 8.75$. Negative slack, if it happens, is distributed in the same way.

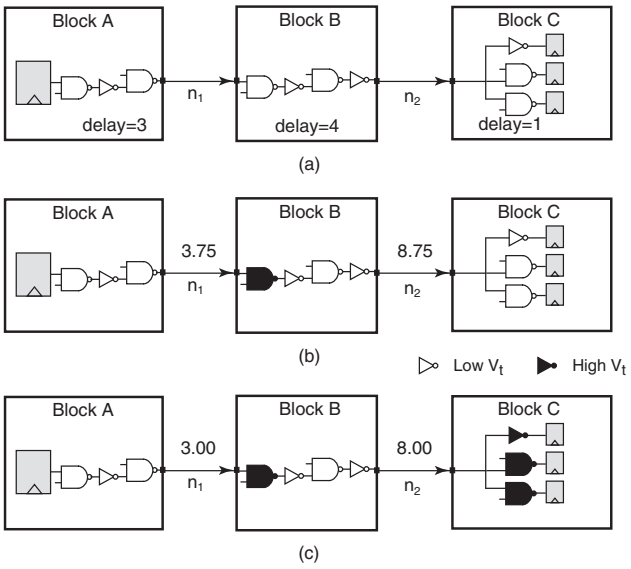

Fig. 2. (a) An example hierarchical design, (b) conventional time budgeting, and (c) alternative time budgeting for better dual-V_t allocation.

The arrival times of n_1 and n_2 allow us to allocate dual-V_t independently in each block. Assume that the delays of low-V_t gates are all 1 and those of high-V_t gates are 2. As shown in Fig. 2(b), only one gate from block B can use high-V_t. This is because block B receives the largest amount of slack, which is 1, since the proportion of path delay attributed by B, which is 4, is the largest.

If we assign more slack to block C, for example, set the arrival time of n_1 to 3 and that of n_2 to 8 as shown in Fig. 2(c), however, three more gates from block C can use high-V_t. Note that block C is a good candidate to use high-V_t, because many gates can be assigned to high-V_t while consuming small amount of slack, because those gates are in parallel. This is not exploited in conventional time budgeting, which is the motivation of our work.

III. Bounded Potential Slack

Fig. 2 illustrates that conventional time budgeting, which uses path delay as a measure, does not work well with dual-V_t allocation. This is understandable because timing-critical path alone is taken into account in budgeting process, but, in dual-V_t allocation, gates that are not in critical path are also important. As a search for a new measure, we review

[1]Precisely speaking, we want to set RAT at the output port of block A and AT at the input port of block B. For simplicity of presentation, we neglect inter-block wire delay: this allows us to simply set AT of n_1, which represents both RAT for block A and AT for block B.

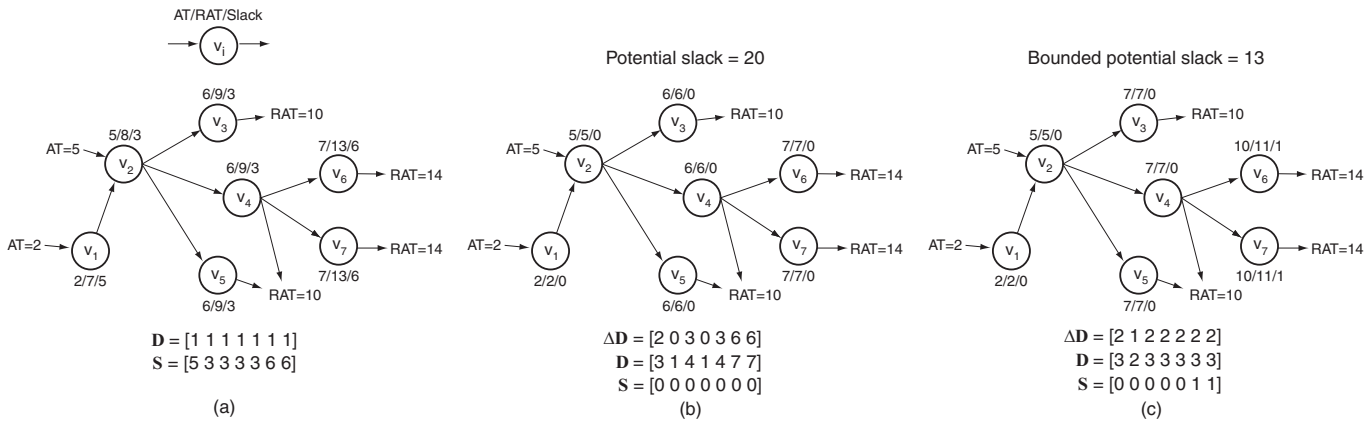

Fig. 3. (a) An example circuit, (b) deriving potential slack, and (c) deriving bounded potential slack.

potential slack in Section III-A, which we then extend to bounded potential slack in the following section.

A. Potential Slack

A combinational circuit is modeled as a directed graph $G = (V, E)$, where $v_i \in V$ corresponds to a gate with propagation delay d_i and $(v_i, v_j) \in E$ models a connection from v_i to v_j. For each vertex v_i, its AT, RAT, and slack at the output are denoted by a_i, r_i, and s_i, respectively, where $s_i = r_i - a_i$. A vector of delays $\mathbf{D} = [d_1\ d_2\ \ldots\ d_n]$ is called a delay distribution, and a vector of slacks $\mathbf{S} = [s_1\ s_2\ \ldots\ s_n]$ by a slack distribution. A slack assignment is to derive a vector of incremental delays $\Delta\mathbf{D} = [\Delta d_1\ \Delta d_2\ \ldots\ \Delta d_n]$, which updates the delay distribution from $\mathbf{D}$ to $\mathbf{D} + \Delta\mathbf{D}$. *Potential slack* [14] is a maximum value of $|\Delta\mathbf{D}| = \sum_{i=1}^{n} \Delta d_i$, such that the new slack distribution are non-negative, i.e. there is no timing violation in the circuit. Potential slack can be obtained by formulating the problem as linear programming (LP) [15] or by maximal independent-set formulation [14], where the former requires less run time [15].

Fig. 3(a) illustrates an example circuit [14], where ATs at primary inputs and RATs at primary outputs are assumed to be given as shown; AT, RAT, and slack of each vertex is calculated using delay distribution $\mathbf{D}$; slack distribution is also shown in the figure. With assigning incremental delays of $\Delta\mathbf{D} = [2\ 0\ 3\ 0\ 3\ 6\ 6]$, all slacks can be made 0, which yields potential slack of 20 as shown in Fig. 3(b).

B. Bounded Potential Slack

Potential slack maximizes the sum of incremental delays. As shown in Fig. 3(b), this results in $\Delta\mathbf{D}$, where some gates are not allowed to increase their delays (v_2 and v_4) while some other receive too much slacks (v_6 and v_7). This is not desirable in dual-V_t allocation. Instead, the number of gates that receive right amount of slack, thereby being assigned to high-V_t without leaving any slack, should be maximized.

We define *bounded potential slack* (BPS) as potential slack, while incremental delay of each gate is bounded, i.e. $\Delta d_i \leq B_i$. The bound B_i is set to the difference in delay when high- and low-V_t are assigned to v_i, respectively. Let $B_i = 2$ for all the gates of Fig. 3(a); bounded potential slack is shown in Fig. 3(c). Even though the value of bounded potential slack, which is 13, is much smaller than that of potential slack in Fig. 3(b), more gates can be assigned to high-V_t. This is because incremental delays $\Delta\mathbf{D}$ are more uniformly distributed due to the use of bound.

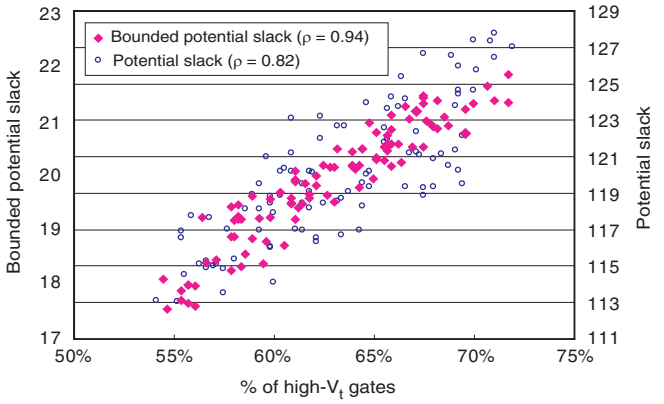

Fig. 4. Correlation between the percentage of high-V_t gates and bounded potential slack (y-axis on the left), and between the percentage of high-V_t gates and potential slack (y-axis on the right); ρ indicates a correlation coefficient.

To assess the effectiveness of bounded potential slack as a measure to be used in time budgeting that considers dual-V_t allocation, we carried out an experiment with `c2670`, which is one of the ISCAS benchmark circuits, in 45-nm technology. All the gates of the circuit were initially assigned to low-V_t. Timing constraints (ATs at primary inputs and RATs at primary outputs) were randomly generated, except that we pick only those that do not fail in timing analysis. For each timing constraint, bounded potential slack was obtained by LP formulation, which we address in Section IV; the circuit and timing constraint were submitted to dual-V_t allocation and the percentage of gates assigned to high-V_t was calculated. Fig. 4 shows a result, which illustrates a strong correlation between bounded potential slack and the percentage of high-V_t gates, with correlation coefficient (ρ) of 0.94. Potential slack was

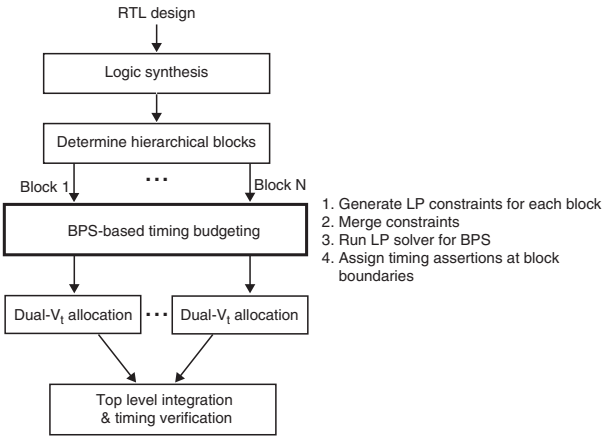

Fig. 5. Overall design flow based on BPS-based time budgeting.

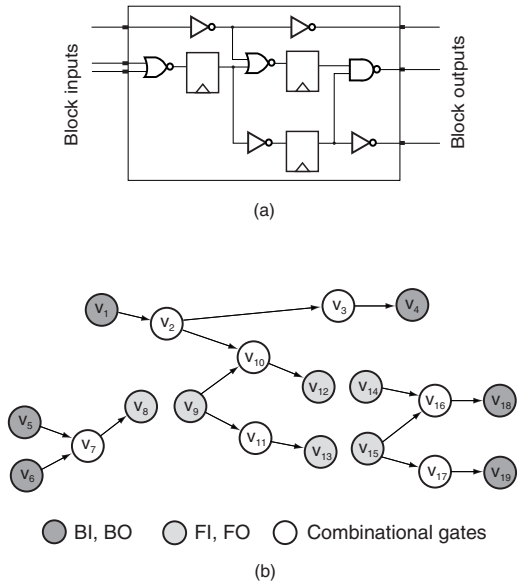

Fig. 6. (a) An example hierarchical block and (b) its graph representation.

also obtained; its correlation with the percentage of high-V_t gates turns out to be weaker (ρ = 0.82) as shown in Fig. 4.

IV. BPS-Based Time Budgeting

Fig. 5 shows the overall design based on BPS-based time budgeting. Initial logic synthesis is performed on RTL design. Hierarchical blocks are identified by designers through simply inheriting logical hierarchy present in RTL design or by manipulating (merging or partitioning) some logical blocks. After time budgeting, each block with its timing assertion is submitted, one by one, to dual-V_t allocation.

A. Problem Formulation

1) Block-Level Constraints: Each hierarchical block is modeled as a directed graph. Different from the graph that models a combinational circuit in Section III-A, however, this graph models a sequential circuit. Each flip-flop is represented by two unconnected vertices: one for master- and the other for slave-portion of the flip-flop as we discussed in Section II-A. Depending on how incremental delays are assigned to these two vertices, we effectively pick a choice of four different implementations. An example circuit is shown in Fig. 6(a) and its corresponding graph in Fig. 6(b), where we also model block inputs and outputs.

We now state block-level constraints for LP formulation:

$$a_i \geq x_i, \qquad \text{if } v_i \in \text{BI} \tag{1}$$

$$r_i \leq x_i, \qquad \text{if } v_i \in \text{BO} \tag{2}$$

$$r_i \leq P - T_{su} - \Delta d_i, \qquad \text{if } v_i \in \text{FI} \tag{3}$$

$$a_i \geq T_{cq} + \Delta d_i, \qquad \text{if } v_i \in \text{FO} \tag{4}$$

where BI (BO) are vertices representing block inputs (outputs), thus x_i is a variable (to be determined) corresponding to timing assertion at block boundary. FI (FO) are vertices representing flip-flop inputs (outputs); P denotes a clock period. For all other vertices corresponding to combinational gates, we require

$$a_i \geq a_j + d_i + \Delta d_i, \tag{5}$$

where v_j in a fanin of v_i. For all vertices where r_i is set, i.e. $\forall v_i \in \text{BO} \cup \text{FI}$, we require

$$r_i - a_i \geq 0. \tag{6}$$

Each incremental delay is bounded by B_i, i.e.

$$0 \leq \Delta d_i \leq B_i, \tag{7}$$

where B_i is the difference in delay when v_i is implemented in high- and low-V_t, respectively.

2) Top-Level LP Formulation: The linear constraints from (1) to (7) of all blocks are combined to constitute overall LP constraints, with objective of maximizing the sum of (bounded) incremental delays, i.e.

$$\text{Maximize} \sum_{\forall \text{blocks}} \sum_{\forall i} \Delta d_i. \tag{8}$$

B. Experimental Results

1) Experimental Setting: BPS-based time budgeting was implemented in the OpenAccess [16], open standard database, with standalone LP solver [17]. Dual-V_t allocation routine described in Section II-A was also implemented in the OpenAccess. As a reference of comparison, commercial time budgeting [6] was used, which is followed by the same dual-V_t allocation routine. All the experiments were performed in commercial 0.99 V, 45-nm bulk CMOS technology.

Six large circuits from the ISCAS and ITC benchmarks were taken, which are summarized in the first five columns of Table I. These designs are originally flat; they were arbitrarily partitioned with each partition being in comparable size. Six more circuits were taken from [7], which are originally hierarchical designs. Each design was initially synthesized [8] in low-V_t gates; the delay of longest timing path was found

TABLE I

COMPARISON OF LEAKAGE POWER AND RUN TIME BETWEEN CONVENTIONAL TIME BUDGETING AND BPS-BASED TIME BUDGETING, EACH ONE FOLLOWED BY THE SAME DUAL-V_t ALLOCATION

Benchmark					Conventional time budgeting			BPS-based time budgeting			BPS/Conv. (×)	
Name	# Gates	# FFs	# Hier. blocks	P (ns)	Leakage (nW)	Run time (s)		Leakage (nW)	Run time (s)		Leakage	Total run time
						Budgeting	Dual-V_t		Budgeting	Dual-V_t		
s13207	2260	490	4	1.3	24.6	4	42	19.3	11	26	0.79	0.80
s15850	2542	513	4	1.1	17.0	4	56	10.7	15	29	0.73	0.63
s35932	6097	1728	8	0.6	116.0	11	433	87.9	134	280	0.76	0.93
s38417	6899	1564	8	1.0	69.5	12	425	47.2	250	270	0.68	1.19
s38584	7638	1294	8	1.0	37.7	10	265	24.3	292	136	0.64	1.55
b17	23063	1414	8	2.8	296.0	98	2197	183.0	3477	1159	0.62	2.02
ac97	8756	2181	8	0.9	33.2	10	470	16.6	303	153	0.50	0.95
aeMB	14157	3330	8	2.2	92.8	34	1569	22.7	3061	282	0.25	2.09
oc54	13481	1426	5	2.3	84.6	24	787	71.8	934	662	0.85	1.97
ps2	2255	254	4	0.9	13.5	4	43	9.8	12	32	0.73	0.94
ucore	14149	1202	5	1.7	35.1	14	395	25.3	892	221	0.72	2.72
warp	21520	1670	7	2.0	124.9	38	1592	108.0	2311	1172	0.87	2.14
Average											0.68	1.49

and used as a clock period P reported in the fifth column. The arrival time at all (top-level) primary inputs were set to 0; the required arrival time at all primary outputs were set to P.

2) Leakage Power: Columns 6–8 show leakage power, run time for time budgeting, and run time for dual-V_t allocation when commercial time budgeting [6] is used. Corresponding figures with BPS-based time budgeting are shown in the next three columns. The ratios of leakage and total run time are shown in the last two columns.

Leakage power is reduced by 32% on average. This is substantial considering that, even though BPS and allocation are correlated very well (recall Fig. 4), time budgeting derives timing assertions while it maximizes sum of (bounded) incremental delays but those incremental delays are not exactly used in dual-V_t allocation because it simply uses sensitivity to determine the order of allocation. The saving of leakage in `aeMB`, which is 75%, is especially significant.

In Fig. 7(a), we compare leakage power of `s13207` between conventional and BPS-based time budgeting while we increase clock period from the minimum value of 1.3 ns used in Table I. With increasing clock period, more timing slacks are left, which are used to convert more gates to high-V_t, thereby decreasing leakage power. The difference between conventional and BPS-based time budgeting becomes smaller as clock period increases, because the majority of gates are now assigned to high-V_t; therefore, choice of time budgeting does not affect leakage much. On the other hand, this explains the importance of BPS-based time budgeting when timing constraint is very tight. The same experiment was performed for `ps2`, with its result shown in Fig. 7(b).

3) Run Time: Run time for budgeting (columns 7 and 10) increases significantly due to the use of LP formulation. The methods to reduce the number of variables and constraints, for example, by detecting and ignoring the blocks that are relatively insensitive to timing assertions, remain as a future work. There is, however, a decrease in run time for dual-V_t allocation (columns 8 and 11). This is because our implementation of dual-V_t allocation starts from all high-V_t gates, i.e. it initially converts all gates to high-V_t. It then gradually converts high-V_t gates to low-V_t ones until timing is satisfied. In BPS-based time budgeting, more gates remain in high-V_t, thus timing gets satisfied quicker with less iterations of allocation routine. Total run time including time for both budgeting and dual-V_t allocation increases by 49% on average as shown in the last column.

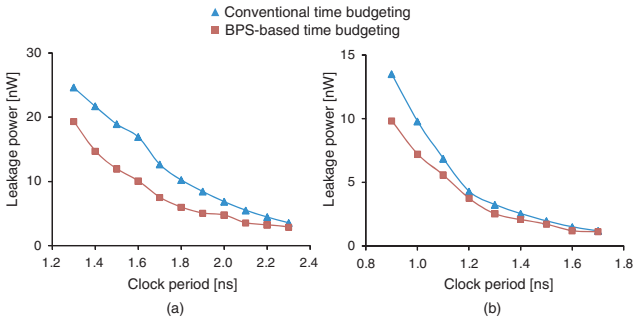

Fig. 7. Comparison of leakage power with varying clock period: (a) `s13207` and (b) `ps2`.

V. APPLICATION TO VOLTAGE ISLANDS WITH DUAL-VT

In this section, we consider voltage island design [18], i.e. each hierarchical block can have different supply voltage, where each island utilizes dual-V_t to manage leakage. Note that there is a conflict between voltage island design and dual-V_t allocation. Lowering V_{dd} can reduce both switching and leakage power; switching power is proportional to V_{dd}^2, subthreshold and gate leakage are roughly proportional to V_{dd}^3 and V_{dd}^4, respectively [19]. Less number of gates, however, can be assigned to high-V_t as V_{dd} decreases due to decreasing amount of timing slacks left in a circuit.

This provides a design space, which we explore in this section with example circuits. We first take `ps2`, which has four blocks, as an example. Low-V_{dd} of 0.99 V was initially assumed and the highest clock frequency was determined. It was followed by BPS-based time budgeting, block-by-block dual-V_t allocation, and estimation of total switching-

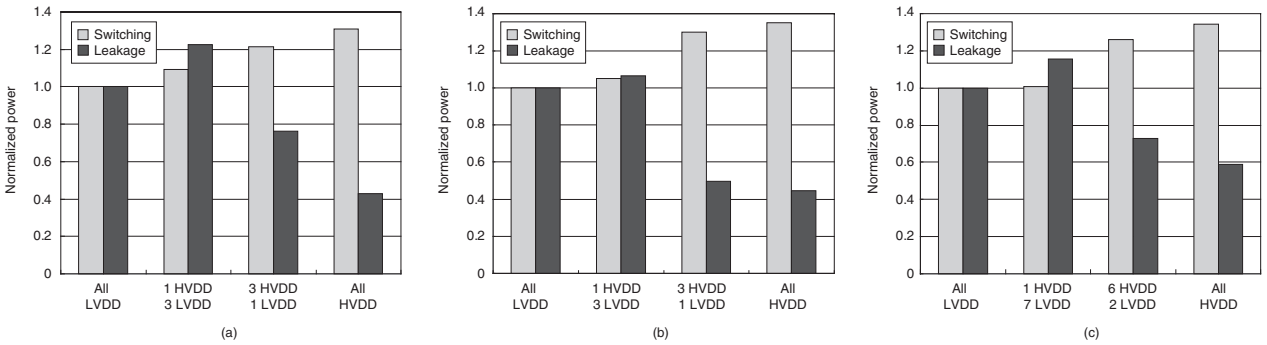

Fig. 8. Normalized switching and leakage power with different voltage island design followed by time budgeting and dual-V_t allocation: (a) `ps2`, (b) `s13207`, and (c) `s38417`.

and leakage-power. With clock frequency fixed, we arbitrarily generated three more configurations of voltage islands: one block set to high-V_{dd} of 1.14 V while three other blocks in low-V_{dd}, three blocks in high-V_{dd} and one block in low-V_{dd}, and all blocks in high-V_{dd}. Level converters were inserted wherever necessary. Each configuration of voltage islands were again followed by time budgeting, dual-V_t allocation, and power estimation. Fig. 8(a) shows the result where switching and leakage power are each normalized. With more number of blocks in high-V_{dd}, switching power increases but leakage power decreases due to more timing slacks being utilized by time budgeting and dual-V_t allocation. Leakage power increases rather than decreases when one block is put in high-V_{dd}; this is mainly because of extra level converters, which consume part of timing slacks. Similar experiments were repeated with `s13207` and `s38417`, and results are shown in Fig. 8(b) and (c), respectively.

VI. Conclusion

Traditional time budgeting for hierarchical design only considers timing closure, though it itself is a vital objective in most designs. We have addressed, for the first time, time budgeting that takes account of leakage power, in particular, when leakage is controlled by dual-V_t allocation. Bounded potential slack has been introduced as a measure of allocation, which has strong correlation with the proportion of high-V_t gates. Time budgeting was formulated as linear programming with objective of bounded potential slack.

Run time is a limitation of proposed time budgeting due to its use of LP formulation. We could work on a fast heuristic as a future work.

VII. Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST), F01-2007-000-10141-0.

References

[1] Y.-H. Chan, P. Kudva, L. Lacey, G. Northrop, and T. Rosser, "Physical synthesis methodology for high performance microprocessors," in *Proc. Design Automation Conf.*, June 2003, pp. 696–701.

[2] T. R. Bednar, P. H. Buffet, R. J. Darden, S. W. Gould, and P. S. Zuchowski, "Issues and strategies for the physical design of system-on-a-chip ASICs," *IBM Journal of Research and Development*, vol. 46, no. 6, pp. 661–674, Nov. 2002.

[3] S. Venkatesh, "Hierarchical timing-driven floorplanning and place and route using a timing budgeter," in *Proc. Custom Integrated Circuits Conf.*, May 1995, pp. 469–472.

[4] C. Kuo and A. Wu, "Delay budgeting for a timing-closure-driven design method," in *Proc. Int. Conf. on Computer Aided Design*, Nov. 2000, pp. 202–207.

[5] O. Omedes, M. Robert, and M. Ramdani, "A flexibility aware budgeting for hierarchical flow timing closure," in *Proc. Int. Conf. on Computer Aided Design*, Nov. 2004, pp. 261–266.

[6] Synopsys, "Budgeting for Synthesis User Guide," Sept. 2005.

[7] "Opencores," http://www.opencores.org/.

[8] Synopsys, "Design Compiler User Guide," Mar. 2007.

[9] M. Ketkar and S. Sapatnekar, "Standby power optimization via transistor sizing and dual threshold voltage assignment," in *Proc. Int. Conf. on Computer Aided Design*, Nov. 2002, pp. 375–378.

[10] Y. Ho and T. Hwang, "Low power design using dual threshold voltage," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 2004, pp. 205–208.

[11] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. Panda, and D. Blaauw, "Stand-by power minimization through simultaneous threshold voltage selelction and circuit sizing," in *Proc. Design Automation Conf.*, June 1999, pp. 436–441.

[12] T. Karnik, Y. Ye, J. Tschanz, L. Wei, S. Burns, V. Govindarajulu, V. De, and S. Borkar, "Total power optimization by simultaneous dual-Vt allocation and device sizing in high performance microprocessors," in *Proc. Design Automation Conf.*, June 2002, pp. 486–491.

[13] A. Srivastava, D. Sylvester, and D. Blaauw, "Statistical optimization of leakage power considering process variations using dual-vth and sizing," in *Proc. Design Automation Conf.*, June 2004, pp. 773–779.

[14] C. Chen, X. Yang, and M. Sarrafzadeh, "Potential slack: an effective metric of combinational circuit performance," in *Proc. Int. Conf. on Computer Aided Design*, Nov. 2000, pp. 198–201.

[15] K. Wang and M. Marek-Sadowska, "Potential slack budgeting with clock skew optimization," in *Proc. Int. Conf. on Computer Design*, Oct. 2004, pp. 265–271.

[16] "Openaccess," http://www.si2.org/.

[17] "GNU Linear Programming Kit," http://www.gnu.org/ software/glpk/.

[18] D. Lackey, P. Zuchowski, T. Bednar, D. Stout, S. Gould, and J. Cohn, "Managing power and performance for System-on-Chip designs using voltage islands," in *Proc. Int. Conf. on Computer Aided Design*, Nov. 2002, pp. 195–202.

[19] R. K. Krishnamurthy, A. Alvandpour, V. De, and S. Borkar, "High-performance and low-power challenges for sub-70nm microprocessor circuits," in *Proc. Custom Integrated Circuits Conf.*, May 2002, pp. 125–128.